
Mayan EDMS Documentation

Release 3.0.1

Roberto Rosario

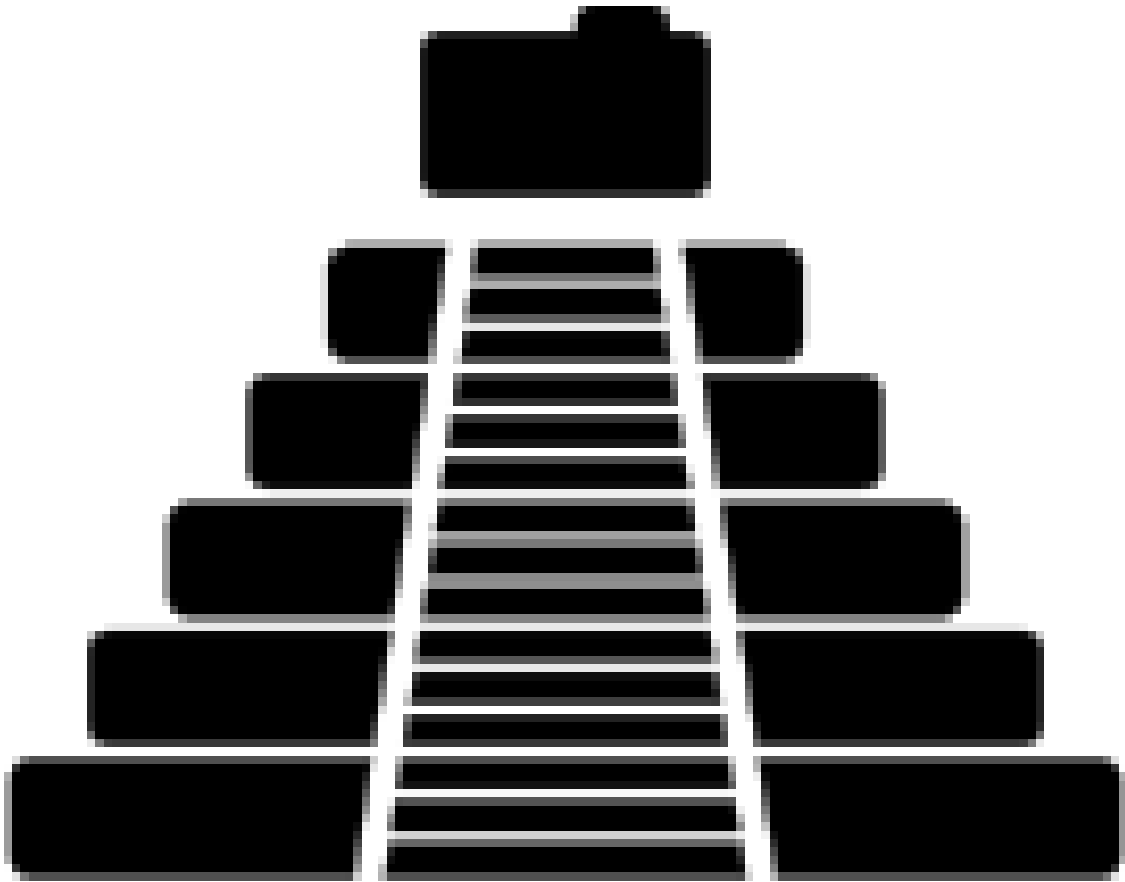
Jul 08, 2018

Contents

1	Installation	3
1.1	Hardware requirements	3
1.2	Docker procedure	3
1.3	Direct installation	3
2	Features	5
3	Release notes	7
3.1	Final releases	7
4	Access control lists	127
4.1	Inherited access control	127
5	Backups	129
6	Checkouts	131
7	Document types	133
8	File storage	135
9	Indexes	137
9.1	Mirroring	137
10	Languages	139
11	Document signatures	141
12	Mailing documents	143
13	Metadata	145
14	OCR backend	147
15	Permissions	149
16	Settings	151
16.1	Via environment variables	151
16.2	Via settings files	151

17 Sources	153
18 Smart links	155
19 Tags	157
20 Transformations	159
21 Upload wizard	161
22 Document versioning	163
23 Docker image	165
23.1 How to use this image	165
23.2 Accessing outside data	168
23.3 Performing backups	168
23.4 Restoring from a backup	168
23.5 Upgrading	168
23.6 Building the image	169
23.7 Customizing the image	169
23.8 Using Docker compose	170
23.9 Nightly images	170
24 Advanced deployment	171
24.1 Binary dependencies	171
25 Development	175
25.1 Project philosophies	175
25.2 Coding conventions	176
25.3 Source Control	178
25.4 Git branch structure	178
25.5 Steps to deploy a development version	179
25.6 Contributing changes	179
25.7 Debugging	179
25.8 Documentation	180
25.9 Installable package	180
26 App creation	183
26.1 App modules	183
26.2 Views	185
27 Pending work	187
27.1 Release blockers	187
27.2 Pending work	187
27.3 New features	189
28 Code statistics	195
29 Translations	205
30 License	207
30.1 License terms	207
31 FAQ	209

32 Contact	213
32.1 FAQ	213
32.2 Mailing list	213
32.3 Twitter	213
32.4 Bugs/ticket tracker	213
33 MECRs	215
33.1 Status	215
33.2 Type	224
34 Mayan EDMS Individual Contributor Assignment Agreement	225
34.1 1. Definitions	225
34.2 2. Grant of Rights	226
34.3 3. Agreement	227
34.4 4. Disclaimer	227
34.5 5. Consequential Damage Waiver	227
34.6 6. Miscellaneous	227
35 Mayan EDMS Entity Contributor Assignment Agreement	229
35.1 1. Definitions	229
35.2 2. Grant of Rights	230
35.3 3. Agreement	231
35.4 4. Disclaimer	231
35.5 5. Consequential Damage Waiver	231
35.6 6. Miscellaneous	231



MAYAN

Mayan EDMS is a [Free Open Source Electronic Document Management System](#), coded in the Python language using the [Django](#) web application framework and released under the [Apache 2.0 License](#). It provides an electronic vault or repository for electronic documents.

The easiest way to use Mayan EDMS is by using the official [Docker](#) image. Make sure Docker is properly installed and working before attempting to install Mayan EDMS.

1.1 Hardware requirements

- 2 Gigabytes of RAM (1 Gigabyte if OCR is turned off).
- Multiple core CPU (64 bit, faster than 1 GHz recommended).

1.2 Docker procedure

Docker is a computer program that performs operating-system-level virtualization also known as containerization. It allows independent “containers” to run within a single Linux instance, avoiding the overhead of starting and maintaining virtual machines (VMs).

Docker can be installed using their automated script:

```
wget -qO- https://get.docker.com/ | sh
```

This installs the latest versions of Docker. If you don't want run an automated script follow the instructions outlined in their documentation: <https://docs.docker.com/install/>

Once the Docker installtion is finished, proceed to the link below to install the Docker image for Mayan EDMS.

Docker image chapter: *Docker image*

1.3 Direct installation

For users with knowledge of Python, Django, Ubuntu, and databases.

Advanced deployment chapter: *Advanced deployment*

- *Document versioning.*
 - Store many versions of the same document, download or revert to a previous version.
- *Electronic signature verification.*
 - Check the authenticity of documents by verifying their embedded cryptographic signatures or upload detached signatures for document signed after they were stored.
- Collaboration tools.
 - Discuss documents, or comment on new versions of a document.
- Office document format support.
 - Mayan EDMS can detect the presence of Libre Office and use it to support word processing files, spreadsheets and presentations.
- User defined metadata fields.
 - Several metadata fields can be matched to a document type as per technical, legal or structural requirements such as the [Dublin core](#).
- Dynamic default values for metadata.
 - Metadata fields can have an initial value, which can be static or determined by a template code snippet provided by the user.
- Documents can be uploaded from different sources.
 - Local file or server side file uploads, multifunctional copier, or even via email.
- Batch upload many documents with the same metadata.
 - Clone a document's metadata for speedier uploads and eliminate repetitive data entry.
- Previews for many file formats.
 - Mayan EDMS provides image preview generation for many popular file formats.
- Full text searching.

- Documents can be searched by their text content, their metadata or any other file attribute such as name, extension, etc.
- Configurable document grouping.
 - Automatic linking of documents based on metadata values or document properties.
- *Roles support.*
 - It is possible to create an unlimited amount of different roles not being restricted to the traditional admin, operator, guest paradigm.
- *Fine grained permissions system.*
 - There is a permission for every atomic operation performed by users.
- Multi page document support.
 - Multiple page PDF and TIFF files are supported.
- Automatic OCR processing.
 - The task of transcribing text from documents via OCR can be distributed among several physical or virtual computers to decrease load and increase availability.
- Multilingual user interface.
 - Mayan EDMS being written using the [Django](#) framework, can be translated to practically any language spoken in the world. For a list of translated languages have a look at the [Transifex](#) project location.
- Multilingual OCR support.
 - The current language of the document is passed to the corresponding OCR engine to increase the text recognition rate.
- *Plugable storage backends.*
 - It is very easy to use 3rd party plugins such as the ones available for Amazon EC2.
- Color coded tagging.
 - Labeled and color coded tags can be assigned for intuitive recognition.
- Workflows.
 - Keep track of the state of documents, along with the log of the previous state changes.

Release notes for the official Mayan EDMS releases. Each release note will tell you what's new in each version, and will also describe any backwards-incompatible changes made in that version.

For those upgrading to a new version of Mayan EDMS, you will need to check all the backwards-incompatible changes and deprecated features for each 'final' release from the one after your current Mayan EDMS version, up to and including the latest version.

3.1 Final releases

Below are release notes through Mayan EDMS 3.0.1 and its minor releases. Newer versions of the documentation contain the release notes for any later releases.

3.1.1 3.0 series

Mayan EDMS v3.0.1 release notes

Released: July 08, 2018

What's new

- Pin javascript libraries to specific versions to avoid using potentially broken updates automatically. GitLab issue #486.
- French and Polish language translation updates.
- Merge request #25. Thanks to Daniel Albert @esclear for the patch.

Removals

- None

Upgrading from a previous version

Using PIP

Type in the console:

```
$ pip install mayan-edms==3.0.1
```

the requirements will also be updated automatically.

Using Git

If you installed Mayan EDMS by cloning the Git repository issue the commands:

```
$ git reset --hard HEAD
$ git pull
```

otherwise download the compressed archived and uncompress it overriding the existing installation.

Next upgrade/add the new requirements:

```
$ pip install --upgrade -r requirements.txt
```

Common steps

Migrate existing database schema with:

```
$ mayan-edms.py performupgrade
```

Add new static media:

```
$ mayan-edms.py collectstatic --noinput
```

The upgrade procedure is now complete.

Backward incompatible changes

- None

Bugs fixed or issues closed

- [GitLab issue #486](#) Docker Verison 3.0 not working

Mayan EDMS v3.0 release notes

Released: June 29, 2018

What's new

Turning Mayan EDMS into a single page app

Historically, Mayan EDMS has steered away from adding too much Javascript in its code. The goal was to be able to maintain a robust, backend-based page rendering method that will be as future-proof as possible. This approach comes at the cost of some page loading speed, and reduced user interface interactivity.

The whole system has been update to work as a modern Single Page App. Single Page Applications (SPAs) rewrite the current page dynamically rather than loading the entire page on each click of the mouse. This makes the web application feel and behave more like a desktop application. Because the majority of the styling and Javascript code is loaded only once, there is also the added benefit of less data down the wire. Thus the application becomes lighter and provides a faster response time to user events. Because the style is loaded and interpreted at the beginning, the browser is also able to apply it to the new content faster.

This redesign was achieved using only HTML5 and jQuery. Aside from two additional jQuery libraries, there are no extra framework dependencies. With the conversion to an SPA, many other petitions for user interface improvements are now possible.

Upgrading to Django 1.11

The move to Django 1.11 proved to be a real challenge. Even though Django 1.11 is a minor release, it breaks compatibility and interfaces in several key areas. Among these were templates and form widgets.

Mayan EDMS uses a complex template, form and widget system. The system mimics object-oriented concepts like inheritance at the rendering stage. This allows the more than 300 views to be serviced with just a handful of forms classes and base templates. Testing and auditing all the views and forms after the upgrade was a lot of work.

Along with the upgrade to Django 1.11, many deprecations warning were fixed in preparation for an eventual upgrade to Django 2.0.

Notification improvements

These work by allowing users to subscribe to a particular event like Document Uploads or to an event of a particular document like when an invoice is edited. If these events occurs, the user gets a reminder next to the bell icon in the main menu bar.

Dependencies upgrades

Most of the requirements, dependencies and libraries were upgraded to their latest release.

- Pillow: 5.0.0
- django-activity-stream: 0.6.5
- django-compressor: 2.2
- django-cors-headers: 2.2.0
- django-formtools: 2.1
- django-qsstats-magic: 1.0.0
- django-stronghold: 0.3.0
- django-suit: 0.2.26

- furl: 1.0.1
- graphviz: 0.8.2
- pyocr: 0.5.1
- python-dateutil: 2.6.1
- python-magic: 0.4.15
- pytz: 2018.3
- sh: 1.12.14
- rest_framework_swagger replaced with drf-yasg: 1.5.0
- FancyBox was upgraded to version 3, Font Awesome to version 5, jQuery to version 3.3.1. ajaxForm version 4.2.2, URL.js 1.19.1 and pace 0.7.8 were added as part of the conversion to single page app.

Search syntax

Searching without using a specialized search database is difficult. Mayan’s design calls avoiding a separate search engine at the cost of some missing search syntax. The OR and the negative term support are the first attempts at adding special syntax to Mayan’s search code.

By default now, search terms are routed to an “AND” query. That means that a search for:

```
Tag1 Tag2
```

will only return documents with both tags attached. To offer the opposite choice we added an “OR” syntax. Searching for:

```
Tag1 OR Tag2
```

will return documents with either tag attached.

Support for literals terms was also added.

Searching for:

```
blue car
```

will return documents with the words “blue” and “car”, even if they are not together. That means getting documents with the phrases “blue sky” and “slow car”. To search for exact terms enclose them in quotes:

```
“blue car”
```

This will return only documents with the exact phrase “blue car”.

Running multiple instances of Mayan EDMS

If you’ve ever tried running two instances of Mayan EDMS, you would have noticed that they both try to create a lock file in the `/tmp` directory with the same name. Only the first instance will be able to run.

The lock filename needs to be unique to each instance, yet predictable so that the workers of an instance can also access the same lock file.

This issues was solved by using a hexadecimal hash representation of the installation’s unique `SECRET_KEY` setting. The use of a hash makes reversing the value in order to obtain the `SECRET_KEY` impossible for all practical purposes.

Display resolution settings

Display sizes for document display, previews, and thumbnail were specified as a string that included the horizontal and the vertical resolution separated by the character “x”. Using an “x” character to separate resolution elements is not standard.

This version splits the settings for specifying resolutions into two settings for each size. One setting for horizontal resolution and another for vertical resolution.

The settings are now:

```
DOCUMENTS_DISPLAY_WIDTH,    DOCUMENTS_DISPLAY_HEIGHT,    DOCUMENTS_PREVIEW_WIDTH,
DOCUMENTS_PREVIEW_HEIGHT,  DOCUMENTS_PRINT_WIDTH,      DOCUMENTS_PRINT_HEIGHT,
DOCUMENTS_THUMBNAIL_WIDTH, DOCUMENTS_THUMBNAIL_HEIGHT
```

Dynamic upload wizard steps

The steps needed to upgrade a document using form-tools’ SessionWizard were hardcoded in the source app. This made it very difficult to add or remove wizard steps.

The steps of the wizard are now defined by a new class called `sources.wizard.WizardStep`. The existing steps to select a document type, enter metadata and tag the document, have been converted to function as `WizardSteps` subclasses. The converted steps now live in

```
sources.wizards.WizardStepDocumentType, tag.wizard_steps.WizardStepTags, and
metadata.wizard_steps.WizardStepMetadata.
```

The steps need to define the following methods:

- `done`: This method is executed when the wizard finished the last step and enters the step where the actual file is uploaded. This step is used to encode form data into the URL query string that will be passed to the document upload view for each file uploaded.
- `condition`: This method is used to display the step conditionally. If this method returns `True` it will be displayed during the upload wizard execution. To skip the step, return `False` or `None`.
- `get_form_initial`: This method is used to return the initial data for the step form. Use this method to set up initial values for the step’s form fields.
- `step_post_upload_process`: This method will be executed once the document finishes uploading. Use this method to process the information encoded in the URL query string by the step’s `done` method.

Once the `WizardStep` subclass is defined, it needs to be registered. This is done by calling the `.register` method of the `WizardStep` class with the subclass as the argument. Example:

```
WizardStep.register(WizardStepMetadata)
```

This statement must be located after the subclass definition. Finally, the module defining the wizard step must be imported so that it is loaded with the rest of the code and enabled. The best place to do this is in the `.ready` method of the apps’ `apps.py` module. Example:

```
class TagsApp(MayanAppConfig):
    has_rest_api = True
    has_tests = True
    name = 'tags'
    verbose_name = _('Tags')

    def ready(self):
```

(continues on next page)

(continued from previous page)

```
super(TagsApp, self).ready()
from actstream import registry

from .wizard_steps import WizardStepTags # NOQA
```

The WizardStep class also allows for unregistering existing steps. This is accomplished by calling the `.deregister` method of the WizardStep class and passing the subclass as the argument. This method should also be called inside the `.ready` method of an apps' `apps.py` module. Example:

```
class TagsApp(MayanAppConfig):
    has_rest_api = True
    has_tests = True
    name = 'tags'
    verbose_name = _('Tags')

    def ready(self):
        super(TagsApp, self).ready()
        from actstream import registry

        from metadata.wizard_steps import WizardStepMetadata # NOQA
        from sources.wizards import WizardStep # NOQA
        from .wizard_steps import WizardStepTags # NOQA

        WizardStep.deregister(WizardStepTags)
```

This will cause the tags assignment step to not be assigned to the upload wizard anymore.

New upload step

Using the new WizardStep class a new upload wizard step was added to assign documents being uploaded to any number of cabinets while being uploaded. This step was been assigned number 4 in the order of step for uploading a file.

Fix carousel item height issues

On some devices the height of the pages on the document preview view were squashed. The CSS for this view was updated to fix this issue.

Orientation detection

After reports that it is not working in 100% of the cases, the feature that detects and fixes the orientation of PDF has been marked experimental and now defaults to being disabled.

New proposal system

The Mayan EDMS Request for Comments or MERCs will be use to propose and or document the new features, the existing code, and the processes governing the project. MERCs 1 and 2 have been approved. MERC-1 outlines the MERC process itself and MERC-2 documents the way API tests are to be written for Mayan EDMS.

Duplicated documents

The duplicated documents system has been improved to also better detect when the duplicate of a primary document has been move to the trash. In this instance the duplicate count of the primary document would be zero and will cause the primary document to not show in the duplicated document list view.

If the duplicated document is deleted from the trash the system now will launch a background clean up task to permanently delete the empty primary document's duplicate document entry from the database.

Storage

It is now possible to pass arguments to the document, document cache and document signatures storage backends. To pass the arguments, use the new settings: `DOCUMENTS_STORAGE_BACKEND_ARGUMENTS`, `DOCUMENTS_CACHE_STORAGE_BACKEND_ARGUMENTS`, and `SIGNATURES_STORAGE_BACKEND_ARGUMENTS`.

The `FileBasedStorage` driver originally provided has been removed. With this change the setting `STORAGE_FILESTORAGE_LOCATION` has also been removed. The storage driver now default to Django's own `FileSystemStorage` driver. By using this driver each app is responsible of specifying their storage path. The path (or location) is configure via the `DOCUMENTS_STORAGE_BACKEND_ARGUMENTS`, `DOCUMENTS_CACHE_STORAGE_BACKEND_ARGUMENTS`, or `SIGNATURES_STORAGE_BACKEND_ARGUMENTS` for the documents, document cache and document signatures respectively.

For example, to change the document storage location use:

```
DOCUMENTS_STORAGE_BACKEND_ARGUMENTS = '{ location: <specific_path> }'
```

If no path is specified the backend will default to `mayan/media/document_storage`.

Finally, to standardize the way app use storage, the `storages.py` modules is now used instead of the `runtime.py` module.

User event filtering

When viewing the event list, the Actor (user) column is not displayed as a link. Clicking this link will filter the event list and display the events performed by that user. The view of event for each user can also be viewed using a new link added to the user list view in the setup menu.

Smart checkbox selection

A faster way to select multiple item has been added. Click the checkbox of the first, hold the Shift key, and then click the checkbox of the last item of the selection. This will select the first, the last and all items in between. To deselect multiple items the same procedure is used. This code was donated by the Paperattor project (www.paperattor.com).

Add JavaScript dependency manager

An internal utility to install and upgrade the JavaScript dependencies was added. This dependency manager allows for the easier maintenace of the JavaScript libraries used through the project.

Previously JavaScript libraries we downloaded and installed by manually. These libraries were them checked into the Git repository. Finally to enable them the correspondng imports were added to the base templates in the appearance app.

This new manager is the first step to start resolving these issues. The manager allows apps to specify their own dependencies. These dependencies are then downloaded when the project is installed or upgraded. As such they are not part of the repository and lower the file size of the project.

Workflow changes

Removing a document type from a workflow will now also remove all running instances of that workflow for documents of the document type just removed.

Adoption of Contributor Assignment Agreements

To facilitate the inclusion of submissions provided by third parties, the project has adopted the use of individual and entity contributor assignment agreements. These agreements make clear the process to transfer the rights to submissions. With these agreements in place we now have a documented and legally sound method to accept submissions that we couldn't before.

SQLite

Starting with version 3.0, a warning message will be shown in the console and in the user interface when using SQLite as the database engine. When it comes to Mayan EDMS, SQLite should only be used for development or testing, never for production. This is due to Mayan EDMS exceeding the concurrency capabilities of SQLite. The results are duplicated documents, frequency database locked errors, among other issues. Suggested database backends are PostgreSQL and MySQL (or MariaDB) using a transaction aware storage engine like InnoDB.

Received email processing

Parsing email messages is a complex task. To increase compatibility with the many interpretations of the standards that govern email messaging, Mayan EDMS now uses Mailgun's flanker library (<https://github.com/mailgun/flanker>). Thanks to flanker, Mayan EDMS now gains new capabilities when it comes to parsing incoming email. For example, in addition to mail attachments, it is now possible to process files included in emails as inline content.

Other changes worth mentioning

- Add Makefile target to check the format of the README.rst file.
- Fix permission filtering when performing document page searching
- base.js was splitted into mayan_app.js, mayan_image.js, and partial_navigation.js.
- Cabinet detail view pagination was fixed.
- Improve permission handling in the workflow app.
- The checkedout detail view permission is now required for the checked out document detail API view.
- Add missing services for the checkout API.
- Fix existing checkout APIs.
- Update API views and serializers for the latest Django REST framework version.
- Update to the latest version the packages for building, development, documentation and testing.
- Add statistics script to produce a report of the views, APIs and test for each app.

- Merge base64 filename patch from Cornelius Ludmann.
- SearchModel return interface changed. The class no longer returns the result_set value. Use the queryset returned instead.
- Remove the unused scrollable_content internal feature.
- Remove unused animate.css package.
- Add the MERC specifying javascript library usage.
- Documents without at least a version are not scanned for duplicates.
- Convert document thumbnails, preview, image preview and staging files to template base widgets.
- Unify all document widgets.
- Printed pages are now full width.
- Move the invalid document markup to a separate HTML template.
- Move transformations to their own module.
- Split documents.tests.test_views into:
 - base.py
 - test_deleted_document_views.py
 - test_document_page_views.py
 - test_document_type_views.py
 - test_document_version_views.py
 - test_document_views.py
 - test_duplicated_document_views.py
- Sort smart links by label.
- Rename the internal name of the document type permissions namespace. Existing permissions will need to be updated.
- Removed redundant permissions checks.
- Total test count increased to 753
- Fix documentation formatting.
- Add upload wizard step documentation chapter.
- Improve and add additional diagrams.
- Change documentation theme to rtd.
- Add the “to=” keyword argument to all ForeignKey, ManyToMany and OneToOne Fields.
- Rename the role groups link label from “Members” to “Groups”.
- Rename the group users link label from “Members” to “Users”.
- Don’t show full document version label in the heading of the document version list view.
- Show the number of pages of a document and of document versions in the document list view and document versions list views respectively.
- Display a document version’s thumbnail before other attributes.

- Use Django's provided form for setting an users password. This change allows displaying the current password policies and validation.
- Add method to modify a group's role membership from the group's view.
- Rename the group user count column label from "Members" to "Users".
- Backport support for global and object event notification. GitLab issue #262.
- Remove Vagrant section of the document. Anything related to Vagrant has been move into its own repository at: <https://gitlab.com/mayan-edms/mayan-edms-vagrant>
- Revise and improve permission requirements for the documents app API.
 - Downloading a document version now requires the document download permission instead of just the document view permission.
 - Creating a new document no longer works by having the document create permission in a global manner. It is now possible to create a document via the API by having the document permission for a specific document type.
 - Viewing the version list of a document now required the document version view permission instead of the document view permission. Not having the document version view permission for a document will not return a 403 error. Instead a blank response will be returned.
 - Reverting a document via API will new require the document version revert permission instead of the document edit permission.
 - Document view permissions is nor required to view the details of a trashed document.
- Revise and improve permission requirements for the document states app API.
 - Require the `Workflow view` permission for the workflow to be able to view a document type's workflow list.
 - Fix the permission check to create workflows. Previously it had not effect as it has using the `mayan_object_permissions` instead of the `mayan_view_permissions` dictionary.
 - Require the `Workflow view` permission to view the workflow instance list.
 - Require the `Workflow view` permission to view a workflow instance's details.
- Display a proper message in the document type metadata type relationship view when there are no metadata types exist.
- Improved styling and interaction of the multiple object action form.
- Add checkbox to allow selecting all item in the item list view.
- Update the role permission edit view require the permission grant or permission revoke permissions for the selected role.
- Add support for roles ACLs.
- Add support for users ACLs.
- Add support for groups ACLs.
- Sort permission namespaces and permissions in the role permission views.
- Invert the columns in the ACL detail view.
- Remove the data filters feature.
- Update Chart.js version.
- Improve line chart appearance. Fix issue with mouse over labels next other chart margin.

- Add support for passing arguments to the OCR backend.
- Fix issue when using workflows transitions with the new version upload event as trigger. Thanks to Sema @Miggaten for the find and the solution.
- Make error messages persistent and increase the timeout of warning to 10 seconds.
- Improve rendering of the details form.
- Update rendering of the readonly multiselect widget to conform to Django's updated field class interface.
- Add locking for interval sources. This reduces the chance of repeated documents from long running email downloads.
- Add the option to enable or disable parsing when uploading a document for each document type.
- Add a new setting option to enable automatic parsing for each new document type created.
- Add support for HTML bodies to the user mailers.
- Production ALLOWED_HOSTS settings now defaults to a safer ['127.0.0.1', 'localhost', ':::1']
- Capture menu resolution errors on invalid URLs. Closes GitLab issue #420.
- New environment variables: MAYAN_SECRET_KEY, MAYAN_CELERY_ALWAYS_EAGER, MAYAN_CELERY_RESULT_BACKEND, MAYAN_BROKER_URL, MAYAN_DATABASE_ENGINE, MAYAN_DATABASE_CONN_MAX_AGE, MAYAN_DATABASE_NAME, MAYAN_DATABASE_USER, MAYAN_DATABASE_PASSWORD, MAYAN_DATABASE_HOST, MAYAN_DATABASE_PORT, MAYAN_DEBUG.
- Stricter defaults. CELERY_ALWAYS_EAGER to False, ALLOWED_HOSTS to ['127.0.0.1', 'localhost', ':::1'].
- New initialization command. Creates media/system and populates the SECRET_KEY and VERSION files.
- Sane scanner source paper source now defaults to blank.
- Merge Docker image creation back into the main repository.
- Docker image now uses gunicorn and whitenoise instead of NGINX to server the app and the static media.
- All installation artifact are now created and read from the media folder.
- Debian is now the Linux distribution used for the Docker image.
- Most Docker Celery workers are now execute using a lower OS priority number.
- Add COMMON_PRODUCTION_ERROR_LOGGING setting to control the logging of errors in production. Defaults to False.
- Change the error log file handle class to RotatingFileHandle to avoid an indefinitely growing log file.
- Disable embedded signature verification during the perform upgrade command.
- Replace the DOCUMENTS_LANGUAGE_CHOICES setting option. Replaced with the new DOCUMENTS_LANGUAGE_CODES.
- Reduce default language code choice from 7800 to the top 100 spoken languages and related (https://en.wikipedia.org/wiki/List_of_languages_by_number_of_native_speakers)
- Fix error when trying to upload a document from and email account with 'from' and 'subject' metadata.
- Fix typo on message.header get from 'Subject' to 'Subject'.
- On multi part emails keep the original From and Subject properties for all subsequent parts if the sub parts don't specify them. Fixes issue #481. Thanks to Robert Schöftner @robert.schoeftner for the report and debug information.

- Don't provide a default for the scanner source `adf_mode`. Some scanners throw an error even when the selection is supported.
- Add a "Quick Download" action to reduce the number of steps to download a single document. GitLab issue #338.
- Recalculate a document's indexes when attaching or removing a tag from or to it.
- Recalculate all of a tag's documents when a tag is about to be deleted.

Removals

- Data filters app.
- `DOCUMENTS_LANGUAGE_CHOICES` setting option. Replaced with `DOCUMENTS_LANGUAGE_CODES`.

Known issues

The newly added 'flanker' dependency used to process email, produces a number of warnings on the console that are impossible to turn off. These are not critical and are related to coding practices in the library. All warning from flanker can be ignored.

Example: "WARNING:flanker.addresslib._parser.parser:Symbol 'domain' is unreachable"

Upgrading from a previous version

Using PIP

Type in the console:

```
$ pip install mayan-edms==3.0
```

the requirements will also be updated automatically.

Using Git

If you installed Mayan EDMS by cloning the Git repository issue the commands:

```
$ git reset --hard HEAD
$ git pull
```

otherwise download the compressed archived and uncompress it overriding the existing installation.

Next upgrade/add the new requirements:

```
$ pip install --upgrade -r requirements.txt
```

Common steps

Migrate existing database schema with:


```
$ mayan-edms.py performupgrade
```

Add new static media:

```
$ mayan-edms.py collectstatic --noinput
```

The upgrade procedure is now complete.

Backward incompatible changes

- None

Bugs fixed or issues closed

- [GitLab issue #211](#) Remove all workflows when removing them from a document type
- [GitLab issue #262](#) Event notifications
- [GitLab issue #278](#) Detect non migrated models
- [GitLab issue #302](#) ‘New Document’ button available to users who do not have permission
- [GitLab issue #332](#) Add option to pass configuration parameters to the OCR backend as mentioned in [#319](#)
- [GitLab issue #338](#) Quick download link in documents list view
- [GitLab issue #370](#) Advanced search appears to OR the fields instead of AND them
- [GitLab issue #380](#) Features removal for version 3.0
- [GitLab issue #405](#) Add wizard steps from external apps
- [GitLab issue #407](#) Improve search syntax to support search query types: AND, OR
- [GitLab issue #408](#) Improve document checkbox selection.
- [GitLab issue #420](#) Mayan raises a 500 instead of 404 in production mode
- [GitLab issue #427](#) /tmp/mayan_locks.tmp is always owned by root
- [GitLab issue #430](#) Can’t use STATICFILES_STORAGE that requires running collectstatic first
- [GitLab issue #437](#) Record users who upload or edit documents
- [GitLab issue #439](#) Toastr library missing after update
- [GitLab issue #444](#) Error in retrieving documents via POP3
- [GitLab issue #446](#) Document searcher
- [GitLab issue #447](#) API Security Bug Chinese wall breach
- [GitLab issue #449](#) OCR Error: ‘int’ object has no attribute ‘split’
- [GitLab issue #452](#) Workflow ACL Doesn’t works
- [GitLab issue #454](#) Invalid next month calculation in statistics app, causes failstop
- [GitLab issue #467](#) mail attachments without content-disposition are lost
- [GitLab issue #468](#) plain text e-mails without charset do not work
- [GitLab issue #470](#) Enable Django variable for HTML encoded emails

- [GitLab issue #474](#) Provide option to serve Mayan EDMS without a webserver (using Tornado o similar).
- [GitLab issue #480](#) Wrong Environment Variables names in documentation
- [GitLab issue #481](#) IMAP sources with metadata not working in 3.0rc1
- [GitLab issue #484](#) Document upload wizard only works as admin
- [GitHub issue #264](#) migrate fails on document_states 0004_workflow_internal_name
- [GitHub issue #269](#) Lack of authentication for document previews

3.1.2 2.0 series

Mayan EDMS v2.7.3 release notes

Released: September 11, 2017

What's new

- Fix task manager queue list view. Thanks to LeVon Smoker for the report.
- Fix resolved link class URL mangling when the `keep_query` argument is used. Fixes source navigation on the document upload wizard. Thanks to Nick Douma(LordGaav) for the report and diagnostic information. [GitLab issue #436](#).

Removals

- None

Upgrading from a previous version

Using PIP

Type in the console:

```
$ pip install -U mayan-edms
```

the requirements will also be updated automatically.

Using Git

If you installed Mayan EDMS by cloning the Git repository issue the commands:

```
$ git reset --hard HEAD
$ git pull
```

otherwise download the compressed archived and uncompress it overriding the existing installation.

Next upgrade/add the new requirements:

```
$ pip install --upgrade -r requirements.txt
```

Common steps

Migrate existing database schema with:

```
$ mayan-edms.py performupgrade
```

Add new static media:

```
$ mayan-edms.py collectstatic --noinput
```

The upgrade procedure is now complete.

Backward incompatible changes

- None

Bugs fixed or issues closed

- [GitLab issue #431](#) can't create new mailer
- [GitLab issue #436](#) New document source menu does not contain source_ids

Mayan EDMS v2.7.2 release notes

Released: September 06, 2017

What's new

Fixes to the new mailer creation view

Some last minute changes to the dynamic form creation code added to support workflow state actions broke the creation of new mailer profiles. This is fixed now and a test was added to avoid future regressions. [GitLab issue #431](#). Thanks to Robert Schöftner([@robert.schoeftner](#)) for the report and the solution.

Event consolidation

Several events were created to audit the creation of new documents. These events caused some confusion and were improved. When creating a new document the two 'document properties edited' events were removed as this is an internal process and not an user event. Another change is the preservation of the user ID that submits the file to create the document. Now instead of 'System' the actual username of the user that uploaded the document will appear in the events log. [GitLab issue #433](#). Thanks to Jesaja Everling([@jeverling](#)) for the report.

Cabinet list sorting

The root cabinet list is now displayed alphabetically sorted. The list of cabinets to which a document belongs to is now displayed sorted too. Thanks to Thomas Plotkowiak for the request.

Visual cue for the document cabinet list

In the document list item view, a simple visual cue in the form of italicized text was added to the document cabinet list. GitLab issue #435. Thanks to LeVon Smoker for the request.

Easier testing of production deployments

The testing library mock is now installed by default, making it easier to run the entire test suit on deployed instances.

Removals

- None

Upgrading from a previous version

Using PIP

Type in the console:

```
$ pip install -U mayan-edms
```

the requirements will also be updated automatically.

Using Git

If you installed Mayan EDMS by cloning the Git repository issue the commands:

```
$ git reset --hard HEAD
$ git pull
```

otherwise download the compressed archived and uncompress it overriding the existing installation.

Next upgrade/add the new requirements:

```
$ pip install --upgrade -r requirements.txt
```

Common steps

Migrate existing database schema with:

```
$ mayan-edms.py performupgrade
```

Add new static media:

```
$ mayan-edms.py collectstatic --noinput
```

The upgrade procedure is now complete.

Backward incompatible changes

- None

Bugs fixed or issues closed

- [GitLab issue #431](#) can't create new mailer
- [GitLab issue #433](#) Events are not created correctly for document upload
- [GitLab issue #435](#) Add visual cue to differentiate the Cabinet list on a document's preview card.

Mayan EDMS v2.7.1 release notes

Released: September 3, 2017

What's new

Unicode metadata when uploading a document

An issue with Django's `unquote_plus` caused documents being uploaded with an unicode, no English character in an initial metadata value field to fail. The `unquote_plus` usage was removed in favor of a dedicated URL parsing library called `furl`. [GitLab issue #423](#). Thanks to Gustavo Teixeira ([@gsteixei](#)) for the find.

Silent errors during initialization

When Python raises an `ImportError` exception, it is not possible to determine the cause of the error. The `local.py` import code was updated to interpret the text of the `ImportError` exception and ignore it only if the `local.py` was missing, which is the case during the initialization commands execute after the initial installation. Any error in the `local.py` file will now cause Mayan to exit. Thanks to MacRobb Simpson for the report and solution.

String usage in the `local.py` file

Python 2.7 uses byte strings by default. Byte strings cannot be used in conjunction with unicode strings. A missing import was causing strings in the `local.py` file to be interpreted as byte string while the rest of Mayan uses unicode strings. Using non-English special characters in a string located in the `local.py` file would have caused a Unicode error. For new installations from Mayan version 2.7.1 onwards, the line `from __future__ import absolute_import, unicode_literals` is included when generating the `local.py` for the first time. For existing installations, adding this line at the top is all that's needed. [GitLab issue #424](#). Thanks to Gustavo Teixeira ([@gsteixei](#)) for the find and researching the cause.

Removals

- None

Upgrading from a previous version

Using PIP

Type in the console:

```
$ pip install -U mayan-edms
```

the requirements will also be updated automatically.

Using Git

If you installed Mayan EDMS by cloning the Git repository issue the commands:

```
$ git reset --hard HEAD
$ git pull
```

otherwise download the compressed archived and uncompress it overriding the existing installation.

Next upgrade/add the new requirements:

```
$ pip install --upgrade -r requirements.txt
```

Common steps

Migrate existing database schema with:

```
$ mayan-edms.py performupgrade
```

Add new static media:

```
$ mayan-edms.py collectstatic --noinput
```

The upgrade procedure is now complete.

Backward incompatible changes

- None

Bugs fixed or issues closed

- [GitLab issue #423](#) Metadata can't handle non ascii chars on upload
- [GitLab issue #424](#) DjangoUnicodeDecodeError on document proprieties due document language field

Mayan EDMS v2.7 release notes

Released: August 30, 2017

What's new

Beta Python 3 support

Preliminary support for Python 3 has landed in this version. More testing is still needed but for the most part seems to be usable. This is just initial support and not meant for production. Please submit any issue with Python 3 to help improve the support for it.

PDF introspection improvements

Some PDF files encode their page rotation information using indirect values instead of actually storing the rotation value as an integer. Support these types of PDF files was added.

3rd party apps

Support was added to allow 3rd party app adding data columns to existing models to specify the order in which such new columns will appear. Support was also added to allow any app to remove existing main menus. App can now in addition to adding their own dashboard widget, remove existing widgets. As part of the dashboard updates support was also added to allow app developers to create multiple dashboards.

Converter customization improvements

For users wanting more control over the document image conversion process, support was added to change the internal format used for image conversion. By default JPG used but via the *pdftoppm_format* and *pillow_format* entries of the `CONVERTER_GRAPHICS_BACKEND_CONFIG` setting option any other format support by Python's Pillow can use used. Support was also added to change the DPI value used by the conversion process of PDF files to images. The default value for this conversion was set to 300 DPI. The entry used to specify this value is *pdftoppm_dpi*.

Workflow refactor

This version includes a preview release of the workflow refactor that includes three new features: transition triggers, state actions, and graphical previews. The transition triggers allow setting document events as triggers to perform a workflow transition automatically. State actions allow performing system actions when a workflow enters or leaves a specify state. For this release 5 actions were included: attaching and removing tags to a document, granting or revoking access via the ACL, and performing a HTTP POST request. As the feature matures more actions will be added. These two features make the workflow app the automation center for Mayan. This feature allow users to program behaviors to perform, even provoke changes in 3rd party software using the HTTP POST. This feature works very much like services like IFTTT [ifttt.com] (If This Then That) or conditionals in programming languages. The last improvement added to the workflow app is the ability to render a workflow in a graphical manner, useful for visually understanding, explaining and debugging workflows.

OCR refactor

As part of the plan to add OCR zone and barcode support the first set of changes was included in this version. These initial changes bring the OCR app up to standard with the rest of the system and splits the OCR app into two new apps: the OCR app and the Document parsing app. The document parsing app will read text content from documents that provide them and display the result under the "Content" document tab. The OCR app will also launch for each document even if they provide text content to recognize any text on images. This separation gives users the two choices

of text information one extracted from the document (not always available or of quality) and the other recognized by OCR.

Document parsing

Historically Mayan has had two methods to extract text from PDF files. First it will try the program called *pdftotext* and failing that will try the PDFMiner Python library. The official PDFMiner library is unmaintained and doesn't support Python 3 will be a requirement for Django 2.0, which will force Mayan to move to Python 3 exclusively in the near future. For this reason the PDFMiner parser has been removed. A new library called PyPDF2 was added in a past version to improve the PDF page count and rotation detection, initial experience with this library has been positive and since it supports text extraction might also replace PDFMiner as the secondary PDF text extraction strategy.

Document version UI

The list of versions of a document was updated to use the new item list view templated added in version 2.6 for document lists. Along with this update preview support was added for individual document version. It is also possible to explore and navigate different versions of a document much easier and with more information that previously available, being able to visually see for example the difference in a document's versions.

Events system

The events system has been updated to provide more information and improve navigation. The *Actor* field will now display *System* when an event was performed by the system instead of displaying the document name. The column *Action object* was added to help identify via which object the event was performed. This is significant when performing actions on objects which are children of another like document versions. The number and types of events that are monitored has been increased all of which can also be used to trigger a workflow transition. The current list:

- Document added to cabinet
- Document removed from cabinet
- Document automatically checked in
- Document checked in
- Document checked out
- Document forcefully checked in
- Document comment created
- Document comment deleted
- Document created
- Document downloaded
- Document properties edited
- New version uploaded
- Document type changed
- Document version reverted
- Document viewed
- Document version OCR finished

- Document version submitted for OCR
- Document version parsing finished
- Document version submitted for parsing
- Tag attached to document
- Tag removed from document

Metadata on document type change

Changing document types will no longer delete all metadata from the document. Any existing metadata whose type matches the metadata in the new type will be preserved.

Permission rebalance

In order to attach or remove a tag to a document, the tag view permissions was needed. This has been update to required the tag attach and remove permissions respectively.

Other Changes

- Add workaround for PDF with IndirectObject as the rotation value. GitHub #261.
- Add ACL list link with icon and use it for the document facet menu.
- Fix mailing app permissions labels.
- Add ACLs link and ACLs permissions to the mailer profile model.
- Improve mailer URL regex.
- Add ordering support to the SourceColumn class. GitLab issue #417.
- Shows the cabinets in the document list. GitLab #417 @corneliusludmann
- Update the index information columns to show the total number of documents and nodes contained in a level.
- Add workaround for pycountry versions without the bibliographical key. GitHub issue #250.
- Skip UUID migration on Oracle backends. GitHub issue #251.
- Allow changing the output format, DPI of the pdftoppm command, and the output format of the converter via the CONVERTER_GRAPHICS_BACKEND_CONFIG setting sub options: pdftoppm_dpi: 300, pdftoppm_format: jpeg, pillow_format: jpeg GitHub issues #256 #257 GitLab issue #416.
- Add support for workflow triggers.
- Add support for workflow actions. Includes actions to attach and remove tags, grant and remove access and perform an HTTP POST request.
- Add support for rendering workflows. Required graphviz binary.
- Add support for unbinding sub menus.
- Fix mailing profile test view.
- Disregard the last 3 dots that mark the end of the YAML document.
- Add support for multiple dashboards.
- Add support for removing dashboard widgets.

- Convert document version view to item list view.
- Add support for browsing individual document versions.
- Add support for dropdown menus to the item list view template.
- Add support for preserving the file extension when downloading a document version. GitLab #415.
- Split OCR app into OCR and parsing.
- Use the literal 'System' instead of the target name when the action user is unknown.
- When changing document types, don't delete the old metadata that is also found in the new document type. GitLab issue #421.
- Change the permission needed to attach and remove tags.
- Reduces debug verbosity during tests.
- Remove the NoMimetype match exception. Not needed now that this is a separate app from the OCR app.
- Make error messages persistent.
- Add 'Action object' column to the event list. Display the object or target type (document, tag, etc).
- Rebalance tag permissions. Change the required permission to attach and remove a tag from view to attach and remove respectively.
- Start of error log consolidation sub project.
- Implement field order for the action dynamic forms. Perform action class validation by importing the class and not relying on an instance of action model, which might not exist when still creating the action.
- Navigation improvements in the workflow app.
- Rename index nodes to index levels.
- Avoid Maximum recursion depth exceeded exception on index document list view.

Removals

- Folders app.
- The view to submit all document for OCR. The view to submit documents by type substitutes this one.
- The PDFMiner parser.

Upgrading from a previous version

Using PIP

Type in the console:

```
$ pip install -U mayan-edms
```

the requirements will also be updated automatically.

Using Git

If you installed Mayan EDMS by cloning the Git repository issue the commands:

```
$ git reset --hard HEAD
$ git pull
```

otherwise download the compressed archived and uncompress it overriding the existing installation.

Next upgrade/add the new requirements:

```
$ pip install --upgrade -r requirements.txt
```

Common steps

Migrate existing database schema with:

```
$ mayan-edms.py performupgrade
```

Add new static media:

```
$ mayan-edms.py collectstatic --noinput
```

The upgrade procedure is now complete.

Backward incompatible changes

- None

Bugs fixed or issues closed

- [GitHub issue #250](#) migrate fails on documents.0025_auto_20150718_0742
- [GitHub issue #251](#) migrate fails on documents.0032_auto_20160315_0537
- [GitHub issue #256](#) Make it possible to adjust values in appsconverterliterals.py from Settings
- [GitHub issue #257](#) Use the DEFAULT_FILE_FORMAT from literals.py in python.py
- [GitHub issue #261](#) fix_orientation method causes document add to crash
- [GitHub issue #263](#) Typo in mayan/apps/ocr/migrations/0004_documenttypesettings.py
- [GitLab issue #172](#) Metadata default value ignored when changing document type
- [GitLab issue #329](#) Move code to Python 3
- [GitLab issue #415](#) Wrong filename when downloading document version
- [GitLab issue #416](#) DPI value for OCR not taken from document metadata
- [GitLab issue #417](#) Display document cabinets in documents list
- [GitLab issue #421](#) Metadata lost when changing document type

Mayan EDMS v2.6.4 release notes

Released: July 25, 2017

What's new

This is a bug fix release and users are encouraged to upgrade.

Changes

- Add missing replacements of reverse to resolve_url.

Removals

- None

Upgrading from a previous version

Using PIP

Type in the console:

```
$ pip install -U mayan-edms
```

the requirements will also be updated automatically.

Using Git

If you installed Mayan EDMS by cloning the Git repository issue the commands:

```
$ git reset --hard HEAD
$ git pull
```

otherwise download the compressed archived and uncompress it overriding the existing installation.

Next upgrade/add the new requirements:

```
$ pip install --upgrade -r requirements.txt
```

Common steps

Migrate existing database schema with:

```
$ mayan-edms.py performupgrade
```

Add new static media:

```
$ mayan-edms.py collectstatic --noinput
```

The upgrade procedure is now complete.

Backward incompatible changes

- None

Bugs fixed or issues closed

- None

Mayan EDMS v2.6.3 release notes

Released: July 25, 2017

What's new

This is a bug fix release and users are encouraged to upgrade.

Changes

- Add makefile target to launch a PostgreSQL container.
- Use `resolve_url` instead of `redirect` to resolve the post login URL.
- Make the `intialsetup` and `performupgrade` management tasks work with signals to allow customization from 3rd party apps.
- PEP8 cleanups.
- Add `tag_ids` keyword argument to the `Source.handle_upload` model method. GitLab issue #413.
- Add overflow wrapping so wrap long titles in Firefox too.
- Makes Roles searchable. GitLab issue #402.
- Add line numbers to the debug and production loggers. Add date and time to the production logger.
- Add support for generating `setup.py` from a template. GitLab #149 #200.
- Add fade in animation to document images.

Removals

- None

Upgrading from a previous version

Using PIP

Type in the console:

```
$ pip install -U mayan-edms
```

the requirements will also be updated automatically.

Using Git

If you installed Mayan EDMS by cloning the Git repository issue the commands:

```
$ git reset --hard HEAD
$ git pull
```

otherwise download the compressed archived and uncompress it overriding the existing installation.

Next upgrade/add the new requirements:

```
$ pip install --upgrade -r requirements.txt
```

Common steps

Migrate existing database schema with:

```
$ mayan-edms.py performupgrade
```

Add new static media:

```
$ mayan-edms.py collectstatic --noinput
```

The upgrade procedure is now complete.

Backward incompatible changes

- None

Bugs fixed or issues closed

- [GitLab issue #149](#) Autogenerate setup.py
- [GitLab issue #200](#) dependencies in setup.py not up to date with requirements
- [GitLab issue #402](#) Make permissions and roles searchable.
- [GitLab issue #413](#) source.models.Source.handle_upload does not support tag_ids but upload_document does

Mayan EDMS v2.6.2 release notes

Released: July 22, 2017

What's new

This is a bug fix release and users are encouraged to upgrade.

Other Changes

- Fix deprecation warning to prepare upgrade to Django 1.11 and 2.0.
- Fix zoom feature in document page view.
- Add support to run tests against a MySQL or Postgres container.
- Improve tag widget customization by moving the markup to its own template.
- Fix document page widget appearance in the document page list view.
- Make document version order deterministic.
- Allow total page number introspection of encrypted PDF with non ASCII user properties. GitLab issue #411.

Removals

- None

Upgrading from a previous version

Using PIP

Type in the console:

```
$ pip install -U mayan-edms
```

the requirements will also be updated automatically.

Using Git

If you installed Mayan EDMS by cloning the Git repository issue the commands:

```
$ git reset --hard HEAD
$ git pull
```

otherwise download the compressed archived and uncompress it overriding the existing installation.

Next upgrade/add the new requirements:

```
$ pip install --upgrade -r requirements.txt
```

Common steps

Migrate existing database schema with:

```
$ mayan-edms.py performupgrade
```

Add new static media:

```
$ mayan-edms.py collectstatic --noinput
```

The upgrade procedure is now complete.

Backward incompatible changes

- None

Bugs fixed or issues closed

- None

Mayan EDMS v2.6 release notes

Released: July 18, 2017

What's new

Other Changes

- Fix issue when editing or removing metadata from multiple documents.

Removals

- None

Upgrading from a previous version

Using PIP

Type in the console:

```
$ pip install -U mayan-edms
```

the requirements will also be updated automatically.

Using Git

If you installed Mayan EDMS by cloning the Git repository issue the commands:

```
$ git reset --hard HEAD
$ git pull
```

otherwise download the compressed archived and uncompress it overriding the existing installation.

Next upgrade/add the new requirements:

```
$ pip install --upgrade -r requirements.txt
```


Common steps

Migrate existing database schema with:

```
$ mayan-edms.py performupgrade
```

Add new static media:

```
$ mayan-edms.py collectstatic --noinput
```

The upgrade procedure is now complete.

Backward incompatible changes

- None

Bugs fixed or issues closed

- None

Mayan EDMS v2.6 release notes

Released: July 18, 2017

What's new

Sending documents to multiple recipients

Support was added to send a document as an attachment, or a link to a document to multiple email recipients. To use this feature enter a comma separated list of email recipients in the "Email address" field.

Visual changes

Several patches to change and improve the user interface landed on this release. The first, by Macrobb Simpson @Macrobb, makes the content area width, match window area. This means that on almost all device screen sizes the content area will be almost fullscreen. Another patch from Macrobb Simpson, improves the visual appearance of the document metadata widget. The other big change is the new list item view template which lists documents in a column, row layout. With this layout document thumbnails are more clearly visible, much more information can be displayed for each document, and works much better on small screen devices like tablets and smartphone than a responsive table which requires two axis navigation on small screens. The height of the dashboard items is now adjusted via javascript to ensure correct layout regardless of screen size of message length when translated.

Search

This release adds users and groups to the list of objects that are searchable via the API. The current list of searchable objects is: metadata types, users, groups, tags, documents, document pages, and cabinets.

Logging

The logging configuration was improved to create a log for critical errors when running on production mode. The default location for this log file is: `/mayan/error.log`. This path can be changed with the `COMMON_PRODUCTION_ERROR_LOG_PATH` setting. This log file will capture application errors and request exceptions.

Cabinets

The access control for cabinets has been fixed in some regards and improved in others. The permission to add and remove documents can now be applied to individual root cabinets instead of globally for a role. Also, the permission to add or remove documents from cabinets must also now be granted to a document or document type. In other words, to add a document to a cabinet, the user's role must have the permission to add documents to cabinet, for the cabinet to receive the document and for the document about to be added.

New permission

The patch to add a permission to view a document's version list was backported from the development branch to make it accessible now. Like cabinets, the tag access control now works on two levels. Now to attach a tag to a document, the permission to attach tags must be granted to the tag to attach and to the document that will receive the tag.

ACL changes

The document type permissions namespace was renamed from "Document setup" to "Document types" for clarity. Along with that change, support was added for granting the document type edit, document type delete, and document type view permissions to individual document type instances instead of just globally.

Testing

The documents app view tests now test for view access and not just permission. Testing against access is more robust and also tests for permissions implicitly.

Other Changes

- Fix HTML mark up in window title. GitLab #397.
- Sort setting by namespace label and by global name second.
- Sort indexes by label.
- Switch the order of the `DEFAULT_AUTHENTICATION_CLASSES` of DRF. GitLab #400.
- Improve code to unbind menu entries.
- Increase the size of the mailing profile label field to 128 characters.

Removals

- None

Upgrading from a previous version

Using PIP

Type in the console:

```
$ pip install -U mayan-edms
```

the requirements will also be updated automatically.

Using Git

If you installed Mayan EDMS by cloning the Git repository issue the commands:

```
$ git reset --hard HEAD
$ git pull
```

otherwise download the compressed archived and uncompress it overriding the existing installation.

Next upgrade/add the new requirements:

```
$ pip install --upgrade -r requirements.txt
```

Common steps

Migrate existing database schema with:

```
$ mayan-edms.py performupgrade
```

Add new static media:

```
$ mayan-edms.py collectstatic --noinput
```

The upgrade procedure is now complete.

Backward incompatible changes

- None

Bugs fixed or issues closed

- [GitLab issue #378](#) Add metadata widget changes from @Macrobb
- [GitLab issue #396](#) Add support for emailing documents to a recipient list.
- [GitLab issue #397](#) Title on Index Page is Broken
- [GitLab issue #400](#) Django REST framework's BasicAuthentication doesn't work with Oauth2_proxy

Mayan EDMS v2.5.2 release notes

Released: July 08, 2017

What's new

Improve duplicate document scan

Previously the way document creation code was enclosed in a single database transactions. This cause the duplicate scan at upload code to received a document reference to uncommitted database data. The single database transaction was split into smaller units to make sure the duplicate scan receives saved and committed data.

Removals

- None

Upgrading from a previous version

Using PIP

Type in the console:

```
$ pip install -U mayan-edms
```

the requirements will also be updated automatically.

Using Git

If you installed Mayan EDMS by cloning the Git repository issue the commands:

```
$ git reset --hard HEAD  
$ git pull
```

otherwise download the compressed archived and uncompress it overriding the existing installation.

Next upgrade/add the new requirements:

```
$ pip install --upgrade -r requirements.txt
```

Common steps

Migrate existing database schema with:

```
$ mayan-edms.py performupgrade
```

Add new static media:

```
$ mayan-edms.py collectstatic --noinput
```

The upgrade procedure is now complete.

Backward incompatible changes

- None

Bugs fixed or issues closed

- None

Mayan EDMS v2.5.1 release notes

Released: July 07, 2017

What's new

This version is identical to version 2.5. It was released to workaround some issues with the recent migration of PyPI (<https://mail.python.org/pipermail/distutils-sig/2017-June/030766.html>)

Removals

- None

Upgrading from a previous version

Using PIP

Type in the console:

```
$ pip install -U mayan-edms
```

the requirements will also be updated automatically.

Using Git

If you installed Mayan EDMS by cloning the Git repository issue the commands:

```
$ git reset --hard HEAD
$ git pull
```

otherwise download the compressed archived and uncompress it overriding the existing installation.

Next upgrade/add the new requirements:

```
$ pip install --upgrade -r requirements.txt
```

Common steps

Migrate existing database schema with:

```
$ mayan-edms.py performupgrade
```

Add new static media:

```
$ mayan-edms.py collectstatic --noinput
```

The upgrade procedure is now complete.

Backward incompatible changes

- None

Bugs fixed or issues closed

- [GitLab issue #378](#) Add metadata widget changes from @Macrobb
- [GitLab issue #379](#) Add new document version list view permission.

Mayan EDMS v2.5 release notes

Released: July 07, 2017

What's new

OCR text download

A link and view were added to download the entire OCR text of a document as a separate file. The link can be found under the “Actions” dropdown when the “OCR” tab of a document is selected.

SANE document source

A validation error was being raised when the resolution field of the SANE document source was left blank. This issue has been fixed and works as expected now.

Mailing profiles

Previously, the way documents were emailed was controlled by configuration settings that only system administrator could change as the OS level. It is now possible to create mailing profiles from within the user interface. This allows for Mayan administrators to add mailing profiles without the intervention system administrators. It also provides the opportunity to create multiple mailing profiles. This is useful for sending documents via different email providers depending on things like priority of delivery, or email size limitations. For multi-tenant environments, this also means that each tenant can now send documents via email with their own respective email accounts. For system administrators, this means there is no longer a need to rely on a single email profile for the entirety of all the tenants in a deployment, which could be taxing email quota limits or triggering spam filters. For more information on the multi-tenant plugin visit the Mayan app store at: <http://www.mayan-edms.com/store/>

New transformation

A lineart transformation was added to reduce the amount of colors in a document's image to just 2. This is useful to increase the OCR accuracy on some kind of documents whose color or layout may confuse the OCR engine and lower the accuracy of the text recognition.

UI reorganization

The main menu was been reorganization for clarity of function. The “About” menu has been renamed to “System” to signify that the items in this menu relate to system configuration topics. The “Tools” and “Setup” sub-menus, were moved from the “Profile” menu to the new “System” menu. The “Profile” menu has been renamed to “User”. Additionally, the “User” menu is now part of the main menu instead of floating right on the layout. This change along with others improve the usability on small devices like tablets and smartphones.

PDF compatibility updates

Support for non-compliant, “broken”, and PDFs encrypted with no passwords has been added. Previously no effort was made to process the images for these files. The code for detecting the number of pages in a PDF has also been improved to retry several methods when failing on non-compliant PDF documents.

Office documents compatibility updates

Improvements to the Libre Office conversion code were added, including a workaround for Libre Office bug #37531 (https://bugs.documentfoundation.org/show_bug.cgi?id=37531) which sometimes manifested when uploading multiple office documents sequentially.

Metadata setup UI improvements

A new widget to define the document type to metadata type relationship has been added. The new widget provides a method to switch between required metadata and optional metadata for a document type. This new method is not only faster but does not force users to remove a metadata type before making the switch and thus avoid deletion of existing metadata entries. A new view was also added to change the document type to metadata type relation not only the document type view but also from the metadata type view eliminating travel between these two views when creating new metadata types and assigning them to document types.

Duplicated document scanning

Support to scan and list duplicated document scanning was added in the form of a new document list link under the “Documents” main menu. Every time a document is uploaded, a document scan will be triggered to determine if the new document is a duplicate of an existing document. Duplicate documents will be listed in a new “Duplicated documents” link in the main menu. A full document list scan can also be triggered by using the new “Duplicated document scan” button in the tools menu. Finally, a new tab in the document view has been added called “Duplicates” that will list all duplicates of the currently selected document when in the document’s view. Related to this feature is the addition of being able to search documents by their checksum. This was done by indexing the checksum database field and by adding the checksum as a search field in the advanced document search view and via the API.

Login session control

Support was added to control the length of time a log in session lasts. First from the user interface side of things a “Remember me” checkbox was added to the log in form that will cause the session to persist after the browser is closed. If this checkbox is left blank the session will be destroyed when the browser closes and the user will need to log in again when accessing any of the URLs. The second part of this feature is for administrators. The configuration setting `AUTHENTICATION_MAXIMUM_SESSION_LENGTH` was added to control the maximum time a logged in session will persist when users click the “Remember me” checkbox. The default of this setting is 30 days.

Document image cache disabling

It is now possible to disable the document page image caching. The document image cache works on two level and hence two setting options were added. The first is the `DOCUMENTS_DISABLE_BASE_IMAGE_CACHE` option which disables the first layer of caching, the generation of a master image file for each document page. This means that subsequent request for a page's image will trigger the conversion of the document from its original uploaded file. The second option, `DOCUMENTS_DISABLE_TRANSFORMED_IMAGE_CACHE`, disables just the caching of the transformed (rotated, resized, zoomed) images of document pages. The settings can be used together or separately depending on how much disk space saving is desired. These settings give control over the trade-off between disk space savings and higher CPU utilization. These settings are ideal for installations with a lot of documents, that want to conserve disk space, and have CPU capacity to spare. Multi-tenant installations can also benefit from these new settings.

Document filter by workflow state

A few versions over, a main menu item was added to list documents by their workflow and/or their current workflow state. Support for filtering by the initial workflow state has been added to this feature.

Support for restoring forgotten password

Views and templates were added to enable the typical "Forgotten password" workflow using a signed token via email.

Other Changes

- Add missing OCR migration.
- Improve error output of the `performupgrade` command to debug upgrade errors that could stop an upgrade (missing document files, etc).
- Enable the `django-mathfilters` app added in version 2.4.
- Do a complete pull and synchronization of the translations to fix missing translations for Polish. Thanks to Wojtek Warczakowski for the report.
- Allow null for the SANE source resolution field. Even though the field was marked as allowing blank values it was failing because it is a number field and number fields need to allow explicit null values when left blank.
- Rename the `mayan_task_manager` app to `task_manager`.
- Make the task manager translatable.
- Add Turkish to the list of processes languages.
- Use `Toastr` library for screen messages.
- Reduce verbosity of some debug messages in the `MayanAppConfig`, `settings` and, `mailing` discovery.
- Make sure lookup selection widgets also trigger the metadata update checkbox on change.
- Usability improvements on small displays.
- Removal of the `CONVERTER_LIBREOFFICE_PATH` and `CONVERTER_PDFTOPPM_PATH` settings. These setting have been consolidated into `CONVERTER_GRAPHICS_BACKEND_CONFIG`.
- Improve the documentation of the document creation API endpoint. GitHub issue #255. Thanks to @lcerliani opening the issue.

- Libre Office conversion improvements. Give every libreoffice instance its own separate \$HOME directory. Additionally give every libreoffice its own UserInstallation file in the \$HOME directory. Works around Libre Office issue: https://bugs.documentfoundation.org/show_bug.cgi?id=37531 Solves or affects GitLab issues #393 #258 #198 #175
- The trashed document deletion action is now a background task. This feature results in much faster trashed document deletion and trash can emptying.
- Remove animated spinners to lower browser memory usage and increase responsiveness.
- Render a document page placeholder while the real document page loads. This change avoids “jumping” effect when loading many thumbnails.
- Increase lazy load thresholds. More thumbnails and document pages will be loaded and visible by default when a view loads.
- Improve usability and appearance on medium and small devices like tablets and smartphones.
- Do hard word break on form titles to avoid horizontal scroll on small displays.
- Python3 compatibility improvements by removing all explicit conversion using the unicode() function.
- Unicode handling improvements.
- Update required versions of Pillow and django-suit.

Removals

- None

Upgrading from a previous version

Using PIP

Type in the console:

```
$ pip install -U mayan-edms
```

the requirements will also be updated automatically.

Using Git

If you installed Mayan EDMS by cloning the Git repository issue the commands:

```
$ git reset --hard HEAD
$ git pull
```

otherwise download the compressed archived and uncompress it overriding the existing installation.

Next upgrade/add the new requirements:

```
$ pip install --upgrade -r requirements.txt
```

Common steps

Migrate existing database schema with:

```
$ mayan-edms.py performupgrade
```

Add new static media:

```
$ mayan-edms.py collectstatic --noinput
```

The upgrade procedure is now complete.

Backward incompatible changes

- None

Bugs fixed or issues closed

- [GitHub issue #255](#) Uploading a local file via api
- [GitLab issue #215](#) Download text contents
- [GitLab issue #286](#) User configurable mailer
- [GitLab issue #337](#) Better way to switch Optional to Required Metadata
- [GitLab issue #373](#) (feature request) Allow selecting document types for metadata

Mayan EDMS v2.4 release notes

Released: June 23, 2017

What's new

SANE document source

A new document source has been added with the ability to retrieve documents from scanners directly. This new document source uses the SANE (Scanner Access Now Easy) (https://en.wikipedia.org/wiki/Scanner_Access_Now_Easy) API client to communicate with USB and network scanners. SANE must be properly installed for this document source to work. Your scanner must also be supported by the SANE API (<http://www.sane-project.org/sane-supported-devices.html>).

Automatic PDF orientation detection

The orientation of PDF documents is now detected at creation and a rotation transformation applied to each of the document's pages to correct the orientation.

Environment variables

Configuration options can now be updated from environment variables. To update a configuration option, prepend the string `MAYAN_` to the name of the configuration option. For example, to increase the number of documents displayed per search results page (from a default of 40) to 50 documents, set the environment variable `MAYAN_COMMON_PAGINATE_BY` to 50 with:

```
$ export MAYAN_COMMON_PAGINATE_BY=50
```

and restart Mayan EDMS. A list of the configuration options can be found in the *Setup* menu, under *Settings*.

Math filters

The `django-mathfilters` (<https://pypi.python.org/pypi/django-mathfilters>) package has been included to provide proper math filters in the indexes. An example of this is indexing documents by quarter:

```
{% load mathfilters %}{{ document.metadata_value_of.date|date:"Y" }}Q{{ document.  
↪metadata_value_of.date|date:"m"|sub:1|intdiv:3|add:1 }}
```

Expand searchable objects

Previously, only documents and later on document pages were searchable. This release add support for searching for tags, metadata types and cabinets. This search support is available via the dynamic search API.

Management command to reset locks

During testing or development error occur and locks can remain behind, blocking execution of a process or task until they expire. To help resolve this a management command has been added called `purgelocks` that will delete all locks in the system.

Index by workflow state

Support was added to update the a document indexes from workflow state changes. To make workflow referencing easier from the index template, a new fields was added to the workflow model called `internal_name`. For example, for a workflow called *Publishing Workflow* with an internal name of `publishing_workflow`, use the following string to reference the current state in an index:

```
{{ document.workflow.publishing_workflow.get_current_state }}
```

Task manager

A new app to monitor the distribution and consumption of background task has been added. This app is call *Task manager* and can be found in the *Tools* menu. Use this new tool to diagnose your background task workers or to determine when to scale up the number of workers.

Other Changes

- Improve render of documents with no pages.
- Fix repeated permission list API URL. GitLab issue #389.
- Fix role creation API endpoint not returning id. GitLab issue #390.
- Fix index rebuilding for multi value first levels. GitLab issue #391.
- Add hardware requirements. GitHub issue #247.
- Fix URL query string encoding for the document page navigation views. GitLab issue #383.
- Truncate views titles via the APPEARANCE_MAXIMUM_TITLE_LENGTH setting. GitLab issue #217.
- Add link to show a document's OCR errors. GitLab issue #291.

Removals

- None

Upgrading from a previous version

Using PIP

Type in the console:

```
$ pip install -U mayan-edms
```

the requirements will also be updated automatically.

Using Git

If you installed Mayan EDMS by cloning the Git repository issue the commands:

```
$ git reset --hard HEAD  
$ git pull
```

otherwise download the compressed archived and uncompress it overriding the existing installation.

Next upgrade/add the new requirements:

```
$ pip install --upgrade -r requirements.txt
```

Common steps

Migrate existing database schema with:

```
$ mayan-edms.py performupgrade
```

Add new static media:

```
$ mayan-edms.py collectstatic --noinput
```

The upgrade procedure is now complete.

Backward incompatible changes

- None

Bugs fixed or issues closed

- [GitHub issue #247](#) Hardware requirements
- [GitLab issue #132](#) Task queue viewer
- [GitLab issue #217](#) [Documents] Truncate document label for avoid double lines in mobile or reduce font
- [GitLab issue #221](#) Add flush lock management command
- [GitLab issue #291](#) Link to OCR error log from document OCR tab
- [GitLab issue #344](#) Add support for searching folders, tags, metadata.
- [GitLab issue #383](#) Page not found when deployed to sub-uri
- [GitLab issue #387](#) Document Scanning not Functioning on rotated landscape PDF's
- [GitLab issue #389](#) How can we map permissions with a particular role using REST API?
- [GitLab issue #390](#) Id is not getting in response
- [GitLab issue #391](#) "Rebuild indexes" does not work in Mayan EDMS 2.3

Mayan EDMS v2.3 release notes

Released: June 08, 2017

What's new

This is a bug-fix and minor feature release and all users are encouraged to upgrade.

Changes

- Index node expression template field changed from a 128 character field to an unlimited size text field to allow for complex indexing expressions.
- When updating the metadata of a document, any input in the value form field will select the adjacent checkbox.
- Support for passing the FUSE option *allow-other* and *allow-root* was added to the index mirroring management command.
- Added support for checking for the latest released version of Mayan from the About menu.
- Added support for rebuilding specific indexes instead of only being able to rebuild all index. [GitLab issue #372](#).
- Rewrite document indexing code to be faster and use less locking. Thanks to Macrobb Simpson (@Macrobb) for the initial implementation.

- Use a predefined file path for the file lock.
- Catch documents with not document version when displaying their thumbnails.
- Add custom `script_prefix` aware resolve function and use it for the document page navigation views. Fixes an issue when Mayan is installed as a sub URL app. Thanks to Gustavo Teixeira(@gsteixe) for the issue and investigation.
- Support was added to update document indexes after workflow state changes.
- An helper was added to access a documents workflow by name. To this end a new field was added to the Workflow class called *Internal name*. This new field makes it much easier to get a document's workflow instance. If for example a document has a workflow called *Publish* with the internal name *publish_workflow*, it will be accessible in the indexing template as `{{ document.workflow.publish_workflow }}`. The latest state of the workflow can be accessed using `{{ document.workflow.publish_workflow.get_current_state }}`.
- Added a new API endpoint to display a list of all the available search models.

Removals

- None

Upgrading from a previous version

Using PIP

Type in the console:

```
$ pip install -U mayan-edms
```

the requirements will also be updated automatically.

Using Git

If you installed Mayan EDMS by cloning the Git repository issue the commands:

```
$ git reset --hard HEAD
$ git pull
```

otherwise download the compressed archived and uncompress it overriding the existing installation.

Next upgrade/add the new requirements:

```
$ pip install --upgrade -r requirements.txt
```

Common steps

Migrate existing database schema with:

```
$ mayan-edms.py performupgrade
```

Add new static media:

```
$ mayan-edms.py collectstatic --noinput
```

The upgrade procedure is now complete.

Backward incompatible changes

- None

Bugs fixed or issues closed

- [GitLab issue #371](#) Auto select checkbox when updating metadata
- [GitLab issue #372](#) (Feature request) Allow ‘rebuild index’ to rebuild only a selected index
- [GitLab issue #383](#) Page not found when deployed to sub-uri
- [GitLab issue #385](#) mountindex: how to specify FUSE mount option allow_other?

Mayan EDMS v2.2 release notes

Released: April 26, 2017

What’s new

API changes

Refactor of the metadata API URLs to use the resource/sub resource paradigm.

Before:

```
/api/metadata/metadata_types/ /api/metadata/metadata_types/{pk}/ /api/metadata/document/metadata/{pk}/
/api/metadata/document/{pk}/metadata/ /api/metadata/document_type/{document_type_pk}/metadata_types/optional/
/api/metadata/document_type/{document_type_pk}/metadata_types/required/
```

After:

```
/api/metadata/metadata_types/ /api/metadata/metadata_types/{metadata_type_pk}/ /api/metadata/document_types/{document_type_pk}/
/api/metadata/document_types/{document_type_pk}/metadata_types/{metadata_type_pk}/
/api/metadata/documents/{document_pk}/metadata/ /api/metadata/documents/{document_pk}/metadata/{metadata_pk}/
```

Document API URLs updated to use the resource/sub resource paradigm. Before:

```
/api/documents/document_version<pk> /api/documents/document_pages<pk>
```

After: /api/documents/<pk>/version/<version_pk> /api/documents/<pk>/version/<version_pk>/pages/<page_pk>

Fields that reference a resource by URL now have the suffix ‘_url’ to differentiate them from fields include the resource.

Before:

```
‘document’: ‘/api/documents/10’
```

After:

```
‘document_url’: ‘/api/documents/10’
```

Removal of the document version revert API endpoint. To revert a document to a previous version using the API, use the DELETE verb to delete the most recent document version to be discarded.

Pages data is no longer included as part of the version data. Instead a link to the document version's pages has been added by the name 'pages_url'. This resolved to '/api/documents/<pk>/pages/<page_pk>/pages'.

- New API endpoints (initial work by @lokeshmanmode):
 - API endpoint to change an user's groups subscription.
 - API endpoint that list all available permissions types.
 - API endpoint to view or change a role's groups.
 - API endpoint to view or change a role's permissions.

Code cleanups

As with every release time was dedicated to improve the organization, size, and readability of code. To this end the licenses of each app were moved to their own module in every app, called licenses.py. As part of the code cleanup the seldom used app called 'installation' which tracked runtime Python packages installed alongside Mayan EDMS for debugging purposes has been removed. The dependency on django-filetransfer has been removed by using django-downloadviews which allows the creation of class based download views.

Performance

The document language list has been moved from the document model to the document form. This change speeds up loading time, document properties views and API documentation views. This version includes the new image caching pipeline which stores transformed (rotated, scaled, etc) versions of the document's images resulting in an overall display loading speed up. The fonts used are now loaded from Mayan EDMS itself and not from the web. This change also allow Mayan EDMS to work in a completely off-line manner.

Searching

Support for searching pages as well as documents has been added. This functionality has been exposed in the API too.

Security

This release enables the password validation for the user password validation support provided by Django. This change allows administrator to set password policies limiting the minimum amount of characters needed for example. For more information on how to configure the password validation feature refer to Django's documentation at: <https://docs.djangoproject.com/en/1.11/topics/auth/passwords/#enabling-password-validation>

Sources

To help test the interval sources (POP3 Email, IMAP Email, Watch folders) a "Check now" button was added that allows users to trigger the source's document fetching code instantly. Previously users had to wait until the next scheduled interval to verify if their source's settings were correct.

Testing

The testing process has been simplified by adding a new option ‘–mayan-apps’ to the test runner that automatically tests all Mayan EDMS apps that report to include tests. The app flag that indicates when an app has test was changed from ‘test’ to the more explicit ‘has_test’. The packaging manifest now includes test files, this means that tests can now be executed in production. The total number of tests was raised to 359 and the total coverage increased to 81%.

A custom test runner replacing the previous custom management command called *runtests*. Testing for orphaned temporary files and orphaned file handles is now optional and controlled by the `COMMON_TEST_FILE_HANDLES` and `COMMON_TEST_FILE_HANDLES` settings.

User interface

To avoid warping on long full names or usernames, the user’s full name or username is no longer displayed in the main menu. Instead the word “Profile” is displayed and the users’s full name or username is displayed when the “Profile” icon is clicked. Drop down menus support has been added and enabled for several apps like documents, folders, and tags. This change make navigation much faster and required less mouse travel.

Support was added for a dashboard widgets and several default widgets are included and enabled.

A view to clone a document page transformation to other pages has been added. A document page transformation navigation bug has been fixed. To aid visual lookup, tags are now alphabetically ordered by label.

A new workflow view that lists documents currently executing a workflow and documents by their specific current workflow state has been added to the main menu.

Other changes

- Cabinets app is now integrated as a core app.
- Now that the Cabinets app is included, the Folders app has been disabled by default. To enable the Folders apps add the following line to your settings/local.py file:

```
INSTALLED_APPS += ('folders',)
```

- Fix height calculation in resize transformation.
- Improve upgrade instructions.
- Update project to work with Django 1.10.
- Add support for attaching multiple tags to single or multiple documents.
- Refactor the workflow for removing tags from single and multiple documents.
- Move new version creation blocking from the documents app to the checkouts app.
- DEBUG now defaults to False.
- Production settings don’t override the DEBUG variable. DEBUG can be set to True on production install to debug errors live.
- Refactor add document to folder view to allow adding a documents to multiple folders at the same time.
- Refactor the remove document from folder view to allow removing documents from multiple folders at the same time.
- Refactor the document mailing views and add support for sending multiple documents via email at the same time.

- Refactor the document metadata views and add support for adding multiple metadata types to a document at the same time.
- Addition of a new OCR backend using PyOCR. This backend tries first to do OCR using libtesseract. If libtesseract is not available the backend defaults to calling the Tesseract executable.
- Make the `lock_manager.backends.file_lock.FileLock` the new default locking backend.
- New transformations added:
 - Rotate 90 degrees
 - Rotate 180 degrees
 - Rotate 270 degrees
 - Mirror (horizontal)
 - Flip (vertical)
 - Gaussian blur
 - Unsharp masking
- Add tool to launch all workflows. GitLab issue #355

Removals

- Removal of the `OCR_TESSERACT_PATH` configuration setting.
- Removal of the Tesseract OCR backend. Replaced with a PyOCR backend.
- Remove usage of `pytesseract` Python library.
- Installation app.
- Recent searches feature.

Upgrading from a previous version

If installed via PIP

Type in the console:

```
$ pip install -U mayan-edms
```

the requirements will also be updated automatically.

If installed using Git

If you installed Mayan EDMS by cloning the Git repository issue the commands:

```
$ git reset --hard HEAD
$ git pull
```

otherwise download the compressed archived and uncompress it overriding the existing installation.

Manually upgrade/add the new requirements:

```
$ pip install --upgrade -r requirements.txt
```

Remove deprecated requirements:

```
$ pip uninstall -y -r removals.txt
```

Common steps

Migrate existing database schema with:

```
$ mayan-edms.py performupgrade
```

Add new static media:

```
$ mayan-edms.py collectstatic --noinput
```

The upgrade procedure is now complete.

Backward incompatible changes

- None

Bugs fixed or issues closed

- [GitLab issue #185](#) Add support for nested menus
- [GitLab issue #285](#) Dashboard widgets
- [GitLab issue #294](#) Move new version creation blocking from the documents app to the checkouts app
- [GitLab issue #301](#) Remove the installation app
- [GitLab issue #303](#) Update urlpatterns in urls.py files to be a list of `django.conf.urls.url()` instances instead.
- [GitLab issue #304](#) Remove string view arguments of `url()` in urls.py files.
- [GitLab issue #307](#) Enter multiple Tags at once
- [GitLab issue #310](#) Metadata's lookup with chinese messages when new document
- [GitLab issue #311](#) acl page return `ContentType:Document`
- [GitLab issue #319](#) TransformationResize issue with very "long" image
- [GitLab issue #328](#) Upgrade Warning/Error during performupgrade (v2.1.3 to v2.1.4)
- [GitLab issue #342](#) Tags should be of `unordered / unsorted` data type
- [GitLab issue #343](#) Bootstrap's dependency on `fonts.googleapis.com` causes Mayan EDMS web interface load slowly if public internet is unreachable
- [GitLab issue #355](#) Workflow changes only on new added documents

Mayan EDMS v2.1.11 release notes

Released: March 14, 2017

What's new

This is a bug-fix release and all users are encouraged to upgrade. The focus of this micro release was REST API improvement.

Changes

- Added a quick rename serializer to the document type API serializer.
- Added per document type, workflow list API view. The URL for this endpoint is `GET /api/document_states/document_type/{pk}/workflows/`
- Added Developer Certificate of Origin. Mayan EDMS was adopted a version 1.1 of the Linux Foundation Developer Certificate of Origin. All commits must be signed (`git commit -s`) in order to be merged.
- Added the detail url of a permission in the permission serializer.
- Added endpoints for the ACL app API.
- Implemented document workflows transition ACLs. GitLab issue #321.
- Add document comments API endpoints. GitHub issue #249.
- Add support for overriding the Celery class. The setting is named `MAYAN_CELERY_CLASS` and expects a dotted python path to the class to use.
- Changed the document upload view in source app to not use the HTTP referer URL blindly, but instead re-compose the URL using known view name. Needed when integrating Mayan EDMS into other app via using iframes.
- Added size field to the document version serializer.
- Removed the serializer from the deleted document restore API endpoint it doesn't need a serializer being just an action POST endpoint.
- Added support for adding or editing document types to smart links via the API.

Removals

- None

Upgrading from a previous version

Using PIP

Type in the console:

```
$ pip install -U mayan-edms
```

the requirements will also be updated automatically.

Using Git

If you installed Mayan EDMS by cloning the Git repository issue the commands:

```
$ git reset --hard HEAD
$ git pull
```

otherwise download the compressed archived and uncompress it overriding the existing installation.

Next upgrade/add the new requirements:

```
$ pip install --upgrade -r requirements.txt
```

Common steps

Migrate existing database schema with:

```
$ mayan-edms.py performupgrade
```

Add new static media:

```
$ mayan-edms.py collectstatic --noinput
```

The upgrade procedure is now complete.

Backward incompatible changes

- None

Bugs fixed or issues closed

- [Github issue #249](#) Add document comments API [\$50 US]
- [GitLab issue #321](#) Transition ACLS
- [GitLab issue #357](#) It should be possible to retrieve all workflows for a given DocumentType from the API

Mayan EDMS v2.1.10 release notes

Released: February 13, 2017

What's new

This is a micro release equal to the previews version from the user's point of view. The version number was increase to workaround some issues with the Python Package Index not allowing re-uploads.

Changes

- Update Makefile to use twine for releases.
- Add Makefile target to make test releases.

Removals

- None

Upgrading from a previous version

Using PIP

Type in the console:

```
$ pip install -U mayan-edms
```

the requirements will also be updated automatically.

Using Git

If you installed Mayan EDMS by cloning the Git repository issue the commands:

```
$ git reset --hard HEAD
$ git pull
```

otherwise download the compressed archived and uncompress it overriding the existing installation.

Next upgrade/add the new requirements:

```
$ pip install --upgrade -r requirements.txt
```

Common steps

Migrate existing database schema with:

```
$ mayan-edms.py performupgrade
```

Add new static media:

```
$ mayan-edms.py collectstatic --noinput
```

The upgrade procedure is now complete.

Backward incompatible changes

- None

Bugs fixed or issues closed

- None

Mayan EDMS v2.1.9 release notes

Released: February 13, 2017

What's new

This is a micro release equal to the previews version from the user's point of view. The version number was increase to workaround some issues with the Python Package Index not allowing re-uploads.

Changes

- Update make file to Workaround long standing pypa wheel bug #99

Removals

- None

Upgrading from a previous version

Using PIP

Type in the console:

```
$ pip install -U mayan-edms
```

the requirements will also be updated automatically.

Using Git

If you installed Mayan EDMS by cloning the Git repository issue the commands:

```
$ git reset --hard HEAD
$ git pull
```

otherwise download the compressed archived and uncompress it overriding the existing installation.

Next upgrade/add the new requirements:

```
$ pip install --upgrade -r requirements.txt
```

Common steps

Migrate existing database schema with:

```
$ mayan-edms.py performupgrade
```

Add new static media:

```
$ mayan-edms.py collectstatic --noinput
```

The upgrade procedure is now complete.

Backward incompatible changes

- None

Bugs fixed or issues closed

- None

Mayan EDMS v2.1.8 release notes

Released: February 12, 2017

What's new

This is a bug-fix release and all users are encouraged to upgrade. The focus of this micro release was REST API improvement.

Changes

- Fixes in the trashed document API endpoints.
- Improved tags API PUT and PATCH endpoints.
- Bulk document adding when creating and editing tags.
- The version of django-mptt is preserved in case mayan-cabinets is installed.
- Add Django GPG API endpoints for signing keys.
- Add API endpoints for the document states app.
- Add API endpoints for the message of the day (MOTD) app.
- Add Smart link API endpoints.
- Add writable versions of the Document and Document Type serializers (GitLab issues #348 and #349).

Removals

- None

Upgrading from a previous version

Using PIP

Type in the console:

```
$ pip install -U mayan-edms
```

the requirements will also be updated automatically.

Using Git

If you installed Mayan EDMS by cloning the Git repository issue the commands:

```
$ git reset --hard HEAD
$ git pull
```

otherwise download the compressed archived and uncompress it overriding the existing installation.

Next upgrade/add the new requirements:

```
$ pip install --upgrade -r requirements.txt
```

Common steps

Migrate existing database schema with:

```
$ mayan-edms.py performupgrade
```

Add new static media:

```
$ mayan-edms.py collectstatic --noinput
```

The upgrade procedure is now complete.

Backward incompatible changes

- None

Bugs fixed or issues closed

- [GitLab issue #310](#) Metadata's lookup with chinese messages when new document
- [GitLab issue #348](#) REST API: Document version comments are not getting updated
- [GitLab issue #349](#) REST API: Document Label, Description are not able to update

Mayan EDMS v2.1.7 release notes

Released: February 2, 2017

What's new

This is a bug-fix release and all users are encouraged to upgrade. The focus of this micro release was REST API improvement.

Changes

- Improved user management API endpoints (initial work by @lokeshmanmode):
 - Improved user creation API endpoint to allow specifying the group membership.
 - Improved user editing API endpoint to allow specifying the group membership.
- Improved permissions API endpoints (initial work by @lokeshmanmode):
 - Add permission list API endpoint. This API endpoint lists all possible permissions in the system.
 - Improved role creation API endpoint to allow specifying the role's group membership and role's permissions.
 - Improved role editing API endpoint to allow specifying the role's group membership and role's permissions.
- Improvements in the API tests of a few apps.
- Add content type list API view to the common app. Content type is required when querying the events of an object, this view show list of content types available.
- Add event type list api view. This API view shows all the possible events that are registered in the system.
- Add event list API view. This view shows all the events that have taken place in the system.
- Add object event list API view. This view show all the events for a specific object (document, etc). The content type of the object whose events are being requested must be specified. The list of available content types is provided now by the common app API.
- The parser and validation fields of the metadata type model have been enable in the metadata type API serializer.

Removals

- None

Upgrading from a previous version

Using PIP

Type in the console:

```
$ pip install -U mayan-edms
```

the requirements will also be updated automatically.

Using Git

If you installed Mayan EDMS by cloning the Git repository issue the commands:

```
$ git reset --hard HEAD
$ git pull
```

otherwise download the compressed archived and uncompress it overriding the existing installation.

Next upgrade/add the new requirements:

```
$ pip install --upgrade -r requirements.txt
```

Common steps

Migrate existing database schema with:

```
$ mayan-edms.py performupgrade
```

Add new static media:

```
$ mayan-edms.py collectstatic --noinput
```

The upgrade procedure is now complete.

Backward incompatible changes

- None

Bugs fixed or issues closed

- None

Mayan EDMS v2.1.6 release notes

Released: November 23, 2016

What's new

This is a bug-fix release and all users are encouraged to upgrade.

Changes

- Fix variable name typo in the rotation transformation class.
- Update translations

Removals

- None

Upgrading from a previous version

Using PIP

Type in the console:

```
$ pip install -U mayan-edms
```

the requirements will also be updated automatically.

Using Git

If you installed Mayan EDMS by cloning the Git repository issue the commands:

```
$ git reset --hard HEAD
$ git pull
```

otherwise download the compressed archived and uncompress it overriding the existing installation.

Next upgrade/add the new requirements:

```
$ pip install --upgrade -r requirements.txt
```

Common steps

Migrate existing database schema with:

```
$ mayan-edms.py performupgrade
```

Add new static media:

```
$ mayan-edms.py collectstatic --noinput
```

The upgrade procedure is now complete.

Backward incompatible changes

- None

Bugs fixed or issues closed

- None

Mayan EDMS v2.1.5 release notes

Released: November 8, 2016

What's new

This is a bug-fix release and all users are encouraged to upgrade.

Other changes

- Backport resize transformation math operation fix (GitLab #319).
- Update Pillow to 3.1.2
 - <https://pillow.readthedocs.io/en/3.4.x/releasenotes/3.1.1.html>
 - <https://pillow.readthedocs.io/en/3.4.x/releasenotes/3.1.2.html>
- Backport zoom performance improvement (GitLab #334).
- Backport trash can navigation link resolution fix (GitLab #331).
- Improve documentation regarding the use of GPG version 1 (GitLab #333).
- Fix ACL create view HTML response type. (GitLab #335).
- Expand staging folder and watch folder explanation.

Removals

- None

Upgrading from a previous version

Using PIP

Type in the console:

```
$ pip install -U mayan-edms
```

the requirements will also be updated automatically.

Using Git

If you installed Mayan EDMS by cloning the Git repository issue the commands:

```
$ git reset --hard HEAD
$ git pull
```

otherwise download the compressed archived and uncompress it overriding the existing installation.

Next upgrade/add the new requirements:

```
$ pip install --upgrade -r requirements.txt
```

Common steps

Migrate existing database schema with:

```
$ mayan-edms.py performupgrade
```

Add new static media:

```
$ mayan-edms.py collectstatic --noinput
```

The upgrade procedure is now complete.

Backward incompatible changes

- None

Bugs fixed or issues closed

- [GitLab issue #319](#) TransformationResize issue with very “long” image
- [GitLab issue #331](#) Trash List View: Items actions should be limited
- [GitLab issue #333](#) “Unable to run gpg - it may not be available.”
- [GitLab issue #334](#) Performance improvment: prevent unnecessary image.resize in TransformationZoom
- [GitLab issue #335](#) Wrong HTML Content-Type in ACL->NEW

Mayan EDMS v2.1.4 release notes

Released: October 28, 2016

What’s new

This is a bug-fix release and all users are encouraged to upgrade.

Other changes

- Fix statistics namespace list display view
- Fix events list display view
- Update required Django version to 1.8.15
- Update required python-gnupg version to 0.3.9
- Improved orphaned temporary files test mixin
- Re-enable and improve GitLab CI MySQL testing
- Improved GPG handling
- New GPG backend system
- Minor documentation updates

Removals

- None

Upgrading from a previous version

Using PIP

Type in the console:

```
$ pip install -U mayan-edms
```

the requirements will also be updated automatically.

Using Git

If you installed Mayan EDMS by cloning the Git repository issue the commands:

```
$ git reset --hard HEAD
$ git pull
```

otherwise download the compressed archived and uncompress it overriding the existing installation.

Next upgrade/add the new requirements:

```
$ pip install --upgrade -r requirements.txt
```

Common steps

Migrate existing database schema with:

```
$ mayan-edms.py performupgrade
```

Add new static media:

```
$ mayan-edms.py collectstatic --noinput
```

The upgrade procedure is now complete.

Backward incompatible changes

- None

Bugs fixed or issues closed

- [GitLab issue #311](#) acl page return ContentType:Document
- [GitLab issue #316](#) Error when trying to access the statistics
- [GitLab issue #324](#) Document signature tests fail in Ubuntu 16.10

Mayan EDMS v2.1.3 release notes

Released: June 29, 2016

What's new

This is a bug-fix release and all users are encouraged to upgrade.

Temporary files cleanup

When uploading PDF files that had been OCR'd by previous software, the text parser backend that uses Poppler, would leave behind some temporary files in the /tmp folder. The issue has been resolved and from the fix a test mixin system check has been devised that will identify places in the codebase with similar behaviors, reducing the recurrence of similar issues in the future.

Other changes

- Add help message when initialsetup migration phase fails. Relates to GitLab issue #296
- Start using self.setdout instead of print as per documentation.
- Fix GitLab issue #295, "When editing a user the top bar jumps to the name of the user".
- Normalize handling of temporary file and directory creation.
- Explicitly check for residual temporary files in tests.
- Add missing temporary file cleanup for office documents.
- Fix file descriptor leak in the document signature download test.

Removals

- None

Upgrading from a previous version

Using PIP

Type in the console:

```
$ pip install -U mayan-edms
```

the requirements will also be updated automatically.

Using Git

If you installed Mayan EDMS by cloning the Git repository issue the commands:

```
$ git reset --hard HEAD
$ git pull
```

otherwise download the compressed archived and uncompress it overriding the existing installation.

Next upgrade/add the new requirements:


```
$ pip install --upgrade -r requirements.txt
```

Common steps

Migrate existing database schema with:

```
$ mayan-edms.py performupgrade
```

Add new static media:

```
$ mayan-edms.py collectstatic --noinput
```

The upgrade procedure is now complete.

Backward incompatible changes

- None

Bugs fixed or issues closed

- [GitLab issue #295](#) When editing a user the top bar jumps to the name of the user
- [GitLab issue #309](#) Temp files quickly filling-up my /tmp (1GB tmpfs)

Mayan EDMS v2.1.2 release notes

Released: May 20, 2016

What's new

This is a bugfix release and all users are encouraged to upgrade.

Language list sorting

The document language list and the user locale profile language list are now sorted to make it easier to find the desired language.

Fixed the metadata lookup options: `{{ users }}` and `{{ groups }}`

When configuring metadata type lookup options the `{{ users }}` and `{{ groups }}` special options can be used to display a list of users or a list of groups. These options were producing a list in the wrong format and were updated.

Other changes

- Add Makefile for common development tasks

Removals

- None

Upgrading from a previous version

Using PIP

Type in the console:

```
$ pip install -U mayan-edms
```

the requirements will also be updated automatically.

Using Git

If you installed Mayan EDMS by cloning the Git repository issue the commands:

```
$ git reset --hard HEAD
$ git pull
```

otherwise download the compressed archived and uncompress it overriding the existing installation.

Next upgrade/add the new requirements:

```
$ pip install --upgrade -r requirements.txt
```

Common steps

Migrate existing database schema with:

```
$ mayan-edms.py performupgrade
```

Add new static media:

```
$ mayan-edms.py collectstatic --noinput
```

The upgrade procedure is now complete.

Backward incompatible changes

- None

Bugs fixed or issues closed

- [GitLab issue #290](#) Unicode characters not supported as metadata values
- [GitLab issue #292](#) Sort languages by name not by abbreviation

Mayan EDMS v2.1.1 release notes

Released: May 17, 2016

What's new

This is a bugfix release and all users are encouraged to upgrade.

Fix object column resolution issue in navigation app

Version 2.1 includes a navigation feature that allows model instances from a queryset generated using the `.defer()` or `.only()` Django filter optimization features to resolve to their parent class transparently. This optimization caused problems with the sources app which uses a

Missing Tesseract language files

The Tesseract OCR backend now reports if the tesseract language file is missing for the requested document's language.

Other changes

- Ensure the automatic default index is created after the default document type.

Removals

- None

Upgrading from a previous version

Using PIP

Type in the console:

```
$ pip install -U mayan-edms
```

the requirements will also be updated automatically.

Using Git

If you installed Mayan EDMS by cloning the Git repository issue the commands:

```
$ git reset --hard HEAD
$ git pull
```

otherwise download the compressed archived and uncompress it overriding the existing installation.

Next upgrade/add the new requirements:

```
$ pip install --upgrade -r requirements.txt
```

Common steps

Migrate existing database schema with:

```
$ mayan-edms.py performupgrade
```

Add new static media:

```
$ mayan-edms.py collectstatic --noinput
```

The upgrade procedure is now complete.

Backward incompatible changes

- None

Bugs fixed or issues closed

- [GitLab issue #288](#) Can't add sources in mayan-edms 2.1.
- [GitLab issue #289](#) OCR fails with Exception.

Mayan EDMS v2.1 release notes

Released: May 14, 2016

What's new

Upgrade to use Django 1.8.13

With the end of life support for Django 1.7, moving to the next Mayan EDMS minor version was a target for this release. The Django minor release chosen was 1.8 as it is very compatible with 1.7 and required minimal changes. Django 1.8 is an LTS release (Long Term Support) meaning that is no new big feature of a new Django version is required, the project can stay in Django 1.8 for a good amount of time with no downsides.

Remove remaining references to Django's User model

The few remaining hard code references to Django's User model that were missed in a previous release have been removed. Using a custom User model with Mayan should present very little if any obstacles.

Remove included login required middleware

The custom middleware include with Mayan EDMS that forces user to be authenticated before being able to access any view has been removed in favor of a dedicated 3rd party Django app for that purpose. The app chosen was `django-stronghold` (<http://mikegrouchy.com/django-stronghold/>).

Improve generation of success and error messages for class based views

In the past success messages for actions would show a generic mention to the object being manipulated (document, folder, tag). Now the errors and success messages will be more explicit in describing what the view has or was trying to manipulate.

Remove ownership concept from folders

Currently Folders in Mayan EDMS have a field that stores a reference to the user that has created that folders. One of the design decisions of Mayan EDMS is that there should never be any explicit ownership of any object. Ownership is relative and is defined by the Access Control List of an object. The removal of the user field from the Folders model brings this app in line with the defined behavior.

Replacement of strip_spaces middleware with the spaceless template tag

As a size optimization technique HTML content was dynamically stripped of spaces as it was being served. The technique used involved detecting the MIME type of the content being served and if found to be of text/HTML type spaces between tags were stripped. An edge case was found where this did not work always. The approach has been changed to use Django's official tag to strip spaces. In addition to using an official approach, the removal of spaces only happens when the template is compiled and not at each HTTP response. The optimization is minimal but since it happened at every response a small increase in speed is expected for all deployment scenarios.

Deselect the update checkbox for optional metadata by default

During the last releases the behavior of the of metadata edit checkbox has seen several tune ups. Thanks to community feedback one small change has been introduced. The edit checkbox will be deselected by default for all optional document type metadata entries.

Implement per document type document creation permission

It is now possible to grant the document creation permission to a role for a document type. Previously document creation was a "blanket" permission. Having the permission meant that user could create any type of document. With this change it is now possible to restrict which types of document users of a specific role can create.

Make document type delete time period optional

The entries that defined after how long a document in the trash would be permanently deleted have been made optional. This means that if a document type has this option blank, the corresponding document of this type would never be deleted from the trash can.

Fixed date locale handling in document properties, checkout and user detail views

A few releases back the ability to for users to set their timezone was added. This change also included a smart date rendering update to adjust the dates and times fields to the user's timezone. Some users reported a few views where this timezone adjustment was not happening, this has been fully fixed.

Default index

During new installations a default index that organizes document by year/month when they were uploaded will be created to help users better understand the concept of indexes in Mayan EDMS.

HTML5 upload widget

A common request is the ability to just drap and drop documents from other windows into Mayan EDMS's document upload wizard. This release includes that capability and will also show a completion bar for the upload. Document uploading is sped up dramatically with this change.

Message of the Day app

Administrators wanting to display announcements has no other way to do so than to customize the login template. To avoid this a new app has been added that allows for the creation of messages to be shown at the user login screen. These messages can have an activation and an expiration date and time. These messages are useful for display company access policies, maintenance announcement, etc.

Document signing

The biggest change for this release is the addition of document signing from within the UI. Enterprise users request this feature very often as in those environments cryptographic signatures are a basic requirement. Previously Mayan EDMS had the ability to automatically check if a document was signed and if signed, verify the validity of the signature. However, to sign documents user had to download the document, sign the document offline, and either re-upload the signed document as a new version or upload a detached signature for the existing document version. Aside from being now able to sign documents from the web user interface, the way keys are handled has been rewritten from scratch to support distributed key storage. This means that a key uploaded in one computer by one user can be used transparently by other users in other computers to sign documents. The relevant access control updates were added to the new document signing system. Users wanting to sign a document need the signing permission for the document (or document type), for the private key they intend to use, and the passphrase (if the key has one). Finally documents are now checked just once for signatures and not every time they are accessed, this provides a very sizable speed improvement in document access and availability.

Other changes

- Upgrade Python requirements to recent versions.
- Rename 'Content' search box to 'OCR'.
- Silence all Django 1.8 model import warnings.
- Add icons to the document face menu links.
- Increase icon to text spacing to 3px.
- Add new permission: checkout details view.
- Add HTML tags stripping to the browser title generation template.
- Folder and Tag creation API calls now return the id of the created instances.
- Update Document model's uuid field to use Django's native UUIDField class.
- Add new split view index navigation

- Newly uploaded documents appear in the Recent document list of the user.
- Start migration from django-sendfile to django-downloadview.
- Index more model fields.
- Navigation system support querysets using `.defer()` or `.only()` optimizations.
- API fixes and improvements.
- Increase total test count to 311.
- Increase test coverage to 77%.
- Documentation improvements.
- Handle unicode filenames in staging folders.
- Add staging file deletion permission.
- New `document_signature_view` permission.
- Instead of multiple keyserver only one keyserver is now supported.
- Replace document type selection widget with an opened selection list.
- Add roadmap documentation chapter.

Removals

- None

Upgrading from a previous version

Using PIP

Type in the console:

```
$ pip install -U mayan-edms
```

the requirements will also be updated automatically.

Using Git

If you installed Mayan EDMS by cloning the Git repository issue the commands:

```
$ git reset --hard HEAD
$ git pull
```

otherwise download the compressed archived and uncompress it overriding the existing installation.

Next upgrade/add the new requirements:

```
$ pip install --upgrade -r requirements.txt
```

Common steps

Migrate existing database schema with:

```
$ mayan-edms.py performupgrade
```

Add new static media:

```
$ mayan-edms.py collectstatic --noinput
```

The upgrade procedure is now complete.

Backward incompatible changes

- None

Bugs fixed or issues closed

- [GitLab issue #137](#) Add app creation chapter to documentation.
- [GitLab issue #147](#) Add in app document signing.
- [GitLab issue #161](#) Email backend setup documentation.
- [GitLab issue #162](#) Add HTML5 file uploader.
- [GitLab issue #191](#) Split index contents title into title and path/breadcrumb widget.
- [GitLab issue #206](#) Support for dynamic LOGIN_EXEMPT_URLS.
- [GitLab issue #208](#) Add tagging step to upload wizard.
- [GitLab issue #218](#) Cookie cutter template for Mayan apps.
- [GitLab issue #222](#) Add notice board or Message of the Day.
- [GitLab issue #225](#) Remove hard coded User model.
- [GitLab issue #232](#) “Create documents” is a blanket permission for a user to create a document of any document type.
- [GitLab issue #246](#) Upgrade to Django version 1.8 as Django 1.7 is end-of-life.
- [GitLab issue #251](#) Add method to disable metadata edit form “update” checkbox when not needed.
- [GitLab issue #255](#) UnicodeDecodeError in apps/common/middleware/strip_spaces_widdleware.py.
- [GitLab issue #256](#) typo in locale settings (Dutch).
- [GitLab issue #261](#) Feature: Document Access Audit Logging.
- [GitLab issue #265](#) Indexes show list (show index only if the user has ACLs on document type).
- [GitLab issue #266](#) Smart links : Dynamic label with Postgresql.
- [GitLab issue #267](#) Release 2.1 RC1 : Notes and ideas.
- [GitLab issue #268](#) Release 2.1 RC1 : Bug to access inside an indexes.
- [GitLab issue #270](#) Release 2.1 RC1 : Bug statistics.
- [GitLab issue #274](#) [Release 2.1 RC2] Web Tests.

- [GitLab issue #275](#) [Release 2.1 RC2] Notes.
- [GitLab issue #276](#) [Release 2.1 RC2] API Tests.

Mayan EDMS v2.0.2 release notes

Released: February 15, 2016

Welcome to Mayan EDMS v2.0.2

What's new

Fine tune "Update" checkbox from the metadata entry form

Previously the update checkbox was ignored during the metadata step of the document upload wizard with the wizard always creating a metadata entry for the new document even if the entry was left blank. The checkbox now controls whether or not the wizard will store try to create the metadata entry.

Fix empty optional lookup metadata fields behavior

An edge case was fixed that caused validation to be executed for empty metadata fields that had a value lookup list.

Remove Docker files

Included Docker and Docker Compose files were removed since the Mayan EDMS Docker (<https://gitlab.com/mayan-edms/mayan-edms-docker>) repository is stable.

Other changes

- Only extract validation error messages from `ValidationError` exception instances.
- Add 2 second delay to document version tests to workaround MySQL not storing the millisecond part of the document version's timestamp.
- Install testing dependencies when installing development dependencies.
- Fix OCR API test for document version page OCR content.
- Move metadata form value validation to `.clean()` method.
- Add HTML tags stripping to the browser title generation template.

Removals

- None

Upgrading from a previous version

Using PIP

Type in the console:

```
$ pip install -U mayan-edms
```

the requirements will also be updated automatically.

Using Git

If you installed Mayan EDMS by cloning the Git repository issue the commands:

```
$ git reset --hard HEAD
$ git pull
```

otherwise download the compressed archived and uncompress it overriding the existing installation.

Next upgrade/add the new requirements:

```
$ pip install --upgrade -r requirements.txt
```

Common steps

Migrate existing database schema with:

```
$ mayan-edms.py performupgrade
```

Add new static media:

```
$ mayan-edms.py collectstatic --noinput
```

The upgrade procedure is now complete.

Backward incompatible changes

- None

Bugs fixed or issues closed

- [GitLab issue #250](#) Empty optional lookup metadata trigger validation error.

Mayan EDMS v2.0.1 release notes

Released: January 22, 2016

Welcome to Mayan EDMS v2.0.1

What's new in Mayan EDMS v2.0.1

Required metadata was not enforce correctly

Fixed a situation where documents having required metadata could still be uploaded without entering a value for the required metadata.

Fix multiple document metadata adding

Fixed a bug when adding metadata to multiple documents.

Fix multiple document metadata editing

Fixed a bug that made it impossible to edit multiple documents' metadata values if one of the documents had no previous value for it's metadata.

Updated Vagrant file

The included Vagrant file now provide 2 boxes: development and production. Selection which kind of box to provision is as easy as executing:

```
vagrant up development
```

or

```
vagrant up production
```

Other changes

- None

Removals

- None

Upgrading from a previous version

Using PIP

Type in the console:

```
$ pip install -U mayan-edms
```

the requirements will also be updated automatically.

Using Git

If you installed Mayan EDMS by cloning the Git repository issue the commands:

```
$ git reset --hard HEAD
$ git pull
```

otherwise download the compressed archived and uncompress it overriding the existing installation.

Next upgrade/add the new requirements:

```
$ pip install --upgrade -r requirements.txt
```

Common steps

Migrate existing database schema with:

```
$ mayan-edms.py performupgrade
```

Add new static media:

```
$ mayan-edms.py collectstatic --noinput
```

The upgrade procedure is now complete.

Backward incompatible changes

- None

Bugs fixed or issues closed

- [GitLab issue #243](#) System allows a user to skip entering values for a required metadata field while uploading a new document
- [GitLab issue #245](#) Add multiple metadata not possible

Mayan EDMS v2.0 release notes

Released: December 2015

Welcome to Mayan EDMS v2.0

What's new in Mayan EDMS v2.0

Update to Django 1.7

The biggest change of this release comes in the form of support for Django 1.7. Mayan EDMS makes use of several new features of Django 1.7 like: migrations, app config and transaction handling. The version of Django supported in this version is 1.7.10. With the move to Django 1.7, support for South migrations and Python 2.6 is removed. The switch to Django 1.7's app config means that the startup order of app should not longer have any relevance, cause any import or startup problems.

Frontend UI

The frontend UI HTML has been re-factored to use Bootstrap. Along with this update a lot of legacy HTML and CSS was removed, greatly simplifying the existing template and allowing the removal of some.

Theming and re-branding

All the presentation logic and markup has been moved into it's own app, the 'appearance' app. All modifications required to customize the entire look of the Mayan EDMS can now be done in a single app. Very little markup remains in the other apps, and it's usually because of necessity, namely the widgets.py modules.

Improved page navigation interface

Previously the document page interface used a fancybox windows leaving the current document in the background. This UI workflow as been improved and the document page navigation behaves like the rest of the document views.

Menu reorganization

To improve user experience, the main menu has been restructured based on function usage, moving seldom used buttons inside other views.

Removal of famfam icon set

The previously used icon set and icon display code was removed and a new system that favor font icon was added.

Document preview generation

The image conversion system was re-factored from the ground up and uses a much smarted caching system. The document image cache has it's own Django file storage driver and no longer default to the system /tmp directory. By moving the document image cache to a Django file storage, the cache doesn't need to reside in the same filesystem or even computer serving the document images. This change also allows nodes in a clustered install to share the document image cache.

Document submission for OCR changed to POST

Previously submitting a document for OCR could be done with a GET request to the corresponding URL. This design decision allowed for fast user experience but caused massive document submissions when sites were scanned by web spiders. The new workflow is to submit documents to the OCR queue only on POST request.

New YAML based settings system

The first phase of the new distributed settings system has landed in this version. This first change causes settings to be serialized to YAML. This also means that it is not possible to pass functions or custom classes as values to settings. Setting that related to a class or function, now specify the path to those classes or functions and they are imported dynamically at runtime. Example:

```
DOCUMENTS_STORAGE_BACKEND = 'storage.backends.filebasedstorage.FileBasedStorage'
```

Removal of auto admin creation

The auto admin user creation code used during new installs has been removed and it is its own reusable Django app. The app is available at <https://pypi.python.org/pypi/django-autoadmin>

Removal of dependencies

Through optimizations and code reduction several Python libraries and Django app are no longer required. These are:

- South
- GitPython
- django-pagination
- psutil
- python-hkp
- sendfile
- slate

ACL system re-factor

The Access Control System has been greatly simplified and optimized. The logistics to grant and revoke permissions are now as follows: Only Roles can hold permissions, groups and user can no longer on their own be granted a permission. Groups are now only organizational units that hold users and Roles are collections of groups. User are just a profile and authentication information object. So to grant a permission or access to a document to a user, grant those permissions to a new or existing role, add the desired user to a group and add that group to the role to which you granted the permission. When thinking about granting permissions think of it this way:

Permissions -> Roles -> Groups -> User

Permissions for a document -> Roles -> Groups -> User

Permissions for a type of document -> Roles -> Groups -> User

Object access control inheritance

A frequently asked feature is the ability to change the access control of a group of documents. This feature has been implemented in the form of object access control inheritance. This means that if you grant a permission to a role for a document type, that role will inherit that permission for all document that are later created of that type. If you revoke a permission from a role for a document type, that role loses that permission for all documents of that type. With this new system changing the access control of individual documents should be an edge case. This new ability of modifying the access control of document types is the new recommended method.

Removal of anonymous user support

Allowing anonymous users access to your document repository is no longer support. Administrators wanting to make a group of documents public are encouraged to create an user, group and role for that purpose.

Metadata validators re-factor

The metadata validators have been split into: Validators and Parsers. Validators will just check that the input value conforms to certain specification, raising a validation error is not and blocking the user from submitting data. The Parsers will transform user input and store the result as the metadata value.

Trash can support

To avoid accidental data loss, documents are not deleted but moved to a virtual trash can. From that trash can documents can then be deleted permanently. The deletion document documents and the moving of documents to the trash can are governed by two different permissions.

Retention policies

Support for retention policies was added and is control on a document type basis. Two aspects can be controlled: the time at which documents will be automatically moved to the trash can and the time after which documents in the trash can will be automatically deleted. By default all new document types created will have a retention policy that doesn't move documents to the trash can and that permanently deletes documents in the trash can after 30 days.

Support to share an index as a FUSE filesystem

Index mirror has been added after being removed several version ago. This time mirroring works by creating a FUSE filesystem that is then mounted anywhere in the filesystem. The previous implementation used symbolic links that while fast, required constant modification to keep in sync with the indexes structure and only worked when the document storage and the index mirror resided in the same physical computer or node. This new implementation allowing mirroring of indexes even across a network or if the document storage is not a traditional filesystem but a remote object store. Since this new FUSE mirroring uses direct read access to the database caching is provided and is controlled by the `MIRRORING_DOCUMENT_CACHE_LOOKUP_TIMEOUT` and `MIRRORING_NODE_CACHE_LOOKUP_TIMEOUT` setting options. Both setting have a default of 10 seconds.

Clickable preview images titles

To reduce the amount of clicks required to access a document, document previews titles are now clickable and will take the user straight to the document view.

Removal of eval

Use of Python's eval statement has been completely removed. Metadata type defaults, lookup fields, smart links and indexes templates now use Django's own template language.

Smarter OCR

Document OCR workflow has been improved to try to parse text for each document page and in failing to parse text will only perform OCR on that specific page, returning to the parsing behavior for the next page. This allowing proper text extraction of documents containing both, embedded text and images.

Failure tolerance

Previous versions made use of transactions to prevent data loss in the event of an unexpected error. This release improves on that approach by also reacting to infrastructure failures. Mayan EDMS can now recover without any or minimal data loss from critical events such as loss of connectivity to the database manager. This changes allow installation of using database managers that do not provide guaranteed concurrency such as SQLite, to scale to thousand of documents. While this configuration is still not recommended, Mayan EDMS will now work and scale much better in environments where parts of the infrastructure cannot be changed (such as the database manager).

For more information about this change read the blog post: <http://blog.robertorosario.com/testing-django-project-infrastructure-failure-tolerance/>

As a result of this work a new Django app called Django-sabot was created that gives Django projects the ability to create unit tests for infrastructure failure tolerance: <https://pypi.python.org/pypi/django-sabot>

RGB tags

Previously tags could only choose from a predetermined number of color. This release changes that and tags be of any color. Tags now store the color selected in HTML RGB format. Existing tags are automatically converted to this new scheme.

Default document type and default document source

After installation a default document type and document source are created, this means that users can start uploading documents as soon as Mayan EDMS is installed without having to do any configuration setting changes. The default document type and default document source are both called 'Default'.

Link unbinding

Support for allowing 3rd party apps to unbind links binded by the core apps was added to further improve re-branding and customization.

Statistics re-factor

Statistics gathering and generation has been overhauled to allow for the creation of scheduled statistics. This allows statistics computation to be scheduled during low load times. A new management command was added to purge stale or orphan schedules left behind by the editing of statistics scheduled. The command is *purgestatistics* and has no parameters.

Apps merge

Several app were merge to reduce complexity of the code based on function. These are: the *home*, *common*, *project_tools* and *project_setup* apps, as well as the *documents* and *document_acls* apps.

New signals

Two new signals are provided to better trigger processing documents at the correct moment, these are:

- `common/perform_upgrade` - Launched on the *performupgrade* management command to allow 3rd party apps to execute custom upgrade procedures in an unified manner.
- `common/post_initial_setup` - Launched on the *initialsetup* management command to allow for post install initialization or setup.
- `common/post_upgrade` - Launched after the *performupgrade* management command finishes.
- `documents/post_version_upload` = Launched after a new document version is uploaded.
- `document/post_document_type_change` = Launched after the document type of a document is changed.
- `documents/post_document_created` = Launched after a document is finally ready to be accessed, not when it is created.
- `ocr/post_document_version_ocr` - Launched when the OCR of a document version has finished.

Test improvements

Instead of a flat `tests.py` file, each app now has a `tests/` directory containing tests modules for each particular aspect of an apps, ie: `test_models.py`, `test_views.py`, `test_classes.py`. The total number and coverage of tests has been greatly increased.

Indexes recalculation

Indexes are now recalculated on when a new document is ready as well as the when the metadata of a document changes. This allows indexing documents not only based on their metadata but also based on their properties.

Upgrade command

To reduce the steps and complexity of upgrades, the new *performupgrade* management command was been added. All the upgrade steps will be performed by this command.

Admin changes

Installation admins are no longer required to have the *superusers* or *staff* Django account flags. All setup tasks are now governed by a permission which can be assigned to a role.

OCR functions split

The textual content of a document as interpreted by the OCR now resides as data in the *ocr* app and not in the *documents* app as before. OCR content might not be available for all documents after the upgrade and might need to be queued again. To help with this situation there is new tool called *OCR all documents* for this exact situation.

New internal document creation workflow

The new document upload code now returns a document stub while content is processing. This allows API users to have the document id of the document just uploaded and perform other actions on it while it becomes ready for access.

Auto logging

App logging to the console is now automatically enabled. If Django's `DEBUG` flag is `True` the default level for auto logging is `DEBUG`. If Django's `DEBUG` flag is `False` (as in production), the default level changes to `INFO`. This should make it easier to add relevant messages to issue tickets as well as a adequate logging during production.

Other changes

- Merge of `document_print` and `document_hard_copy` views.
- New class based and menu based navigation system.
- Re-purpose the installation app.
- New class based transformations.
- Usage of Font Awesome icons set.
- Move document text content display code to the OCR app.
- Add new permissions `PERMISSION_OCR_CONTENT_VIEW`.
- Document type OCR settings move to the OCR app.
- New dependencies:
 - PyYAML
 - django-autoadmin
 - django-pure-pagination
 - djangorestframework-recursive
- Management command to remove obsolete permissions: *purgepermissions*.
- Normalization of 'title' and 'name' fields to 'label'.
- Improved API, now at version 1.
- Invert page title/project name order in browser title.
- Use Django's class based views pagination.
- Reduction of text strings.
- OCR all documents.
- Add tool to OCR all documents of a type.
- Fix rendering of text files with Unicode characters.
- Capture body of emails as a text document.
- All app APIs are top level URLs.
- CI using gitlab-ci.
- Coverage report with codecov.io.
- Thumbnails for documents in trash.
- Production deployment documentation chapter.
- Command line to create an initial settings file: *createsettings*.
- Initialsetup now continues even is a `settings/local.py` exists.

- `default_app_config` for each app.
- Natural key support for many models allowing database migrations using dumped data.
- Separate documentation requirements file to allow for contributor who only want to work on documentation.
- Centralized testing with a new management command, *runtests*.
- Addition of a tox testing configuration.
- Email test body capture.
- Email subject and from values storage.
- Gitlab CI support.
- Codecov support.
- Improve text file rendering.
- Show other packages licenses.
- Task delay to allow DB replication.
- Automatic debug logging and info logging during production.

Removals

- Removal of the `CombinedSource` class.
- Removal of default class ACLs.
- Removal of the `ImageMagick` and `GraphicsMagick` converter backends.
- Remove support for applying roles to new users automatically.
- Removal of the `DOCUMENT_RESTRICTIONS_OVERRIDE` permission.
- Removed the `page_label` field.
- Removal of custom HTTP 505 error view.

Upgrading from a previous version

Using PIP

Type in the console:

```
$ pip install -U mayan-edms
```

the requirements will also be updated automatically.

Using Git

If you installed Mayan EDMS by cloning the Git repository issue the commands:

```
$ git reset --hard HEAD
$ git pull
```

otherwise download the compressed archived and uncompress it overriding the existing installation.

Next upgrade/add the new requirements:

```
$ pip install --upgrade -r requirements.txt
```

Common steps

Migrate existing database schema with:

```
$ mayan-edms.py performupgrade
```

During the migration several messages of stale content types can occur:

```
The following content types are stale and need to be deleted:
```

```
XX | XX
```

```
Any objects related to these content types by a foreign key will also  
be deleted. Are you sure you want to delete these content types?
```

```
If you're unsure, answer 'no'.
```

```
Type 'yes' to continue, or 'no' to cancel:
```

You can safely answer “yes” to all.

Add new static media:

```
$ mayan-edms.py collectstatic --noinput
```

Remove unused dependencies:

```
$ pip uninstall South  
$ pip uninstall GitPython  
$ pip uninstall psutil  
$ pip uninstall python-hkp  
$ pip uninstall django-sendfile  
$ pip uninstall django-pagination  
$ pip uninstall slate
```

The upgrade procedure is now complete.

Backward incompatible changes

- Current document and document sources transformations will be lost during upgrade.
- Permissions and Access Controls granted to users and/or groups will be lost during upgrade.

Bugs fixed or issues closed

- [GitHub issue #33](#) Update to Django 1.7
- [GitHub issue #59](#) New bootstrap based UI
- [GitHub issue #60](#) Backport class based navigation code from the unstable branch

- [GitHub issue #62](#) Simplify and reduce code in templates
- [GitHub issue #67](#) Python 3 compatibility: Update models `__unicode__` methdo to `__str__` methods (using Django's six library)
- [GitHub issue #121](#) Twitter Bootstrap theme for Mayan EDMS
- [GitHub issue #155](#) Header does not fit list on documents/list on small screens (laptop)
- [GitHub issue #170](#) Remove use of python-hkp
- [GitHub issue #182](#) Reorganize signal processors
- [GitLab issue #131](#) error on initialsetup: GPG initialization error
- [GitLab issue #135](#) Add document indexing filesystem mirroring
- [GitLab issue #141](#) Merge common and main app
- [GitLab issue #142](#) New authentication app
- [GitLab issue #145](#) Convert document tags to user RGB value for code instead of predetermined choices
- [GitLab issue #150](#) Add 'trash can' support
- [GitLab issue #151](#) Add support for data retention policies
- [GitLab issue #152](#) JSON API 500 error
- [GitLab issue #154](#) /documents API endpoint should return document pk
- [GitLab issue #155](#) Remove unused document page label field
- [GitLab issue #156](#) Remove post OCR language cleanup
- [GitLab issue #158](#) Django REST Swagger not working
- [GitLab issue #159](#) Error during template rendering on /document/folder/add with non-admin user
- [GitLab issue #160](#) Add audit logging
- [GitLab issue #163](#) Removal of the compressed file support
- [GitLab issue #164](#) Keep fancybox prev & next buttons enabled all the time
- [GitLab issue #167](#) Add workflow completion number to states
- [GitLab issue #168](#) Add field to store last error of source during execution
- [GitLab issue #171](#) tesseract fails with german language (wrong abbreviation)
- [GitLab issue #173](#) Add `post_document_upload` signal
- [GitLab issue #174](#) Bootstrap UI with master branch
- [GitLab issue #176](#) Replace default email domain
- [GitLab issue #177](#) Multi page tiff preview is not working
- [GitLab issue #178](#) Add separate missing optional metadata and missing required metadata tools
- [GitLab issue #181](#) Move task `<->` queue assignment to `apps.py`
- [GitLab issue #182](#) Document tags widget is not permissions aware
- [GitLab issue #183](#) Separate metadata validators into: `validators` and `parsers`
- [GitLab issue #184](#) Move literals in `checkouts apps.py` and `tasks.py` to `literals.py`
- [GitLab issue #186](#) Scheduled task to delete all document stubs of more than X age.

- [GitLab issue #187](#) Add tests for multi page tiff files
- [GitLab issue #189](#) Use transient queues
- [GitLab issue #190](#) Bump API version number
- [GitLab issue #192](#) Use local model for document comments
- [GitLab issue #197](#) Add continuous integration that is compatible with Gitlab
- [GitLab issue #201](#) Untranslated items
- [GitLab issue #202](#) AutoAdminSingleton matching query does not exist.
- [GitLab issue #203](#) KeyError at /sources/upload/document/new/interactive/
- [GitLab issue #204](#) Problems to add required metadata after changin the document type
- [GitLab issue #216](#) Add default_app_config value to each app
- [GitLab issue #223](#) [Documents] Trigger event_document_type_change on the model not on the view
- [GitLab issue #227](#) decoder zip not available
- [GitLab issue #228](#) Attribute error when trying to attach a tag for a user with inadequate permissions
- [GitLab issue #229](#) Attribute error when a user tries to download a document - version 2.0.0b2
- [GitLab issue #230](#) No option to create new document version even though user given permission in document ACL
- [GitLab issue #231](#) User shown option to upload new version of a document even though it is blocked by checkout - v2.0.0b2
- [GitLab issue #233](#) Available users instead of available groups
- [GitLab issue #237](#) Forcefully checking in a document by a user without adequate permissions throws out an error

3.1.3 1.0 series

Mayan EDMS v1.1.1 release notes

Released: May 21, 2015

What's new

Update to Django 1.6.11

The required version of Django has been updated to version 1.6.11.

Minor changes

- Fix make_dist.sh script
- Add test for issue #163
- Activate tests for the sources app
- Removal of the registration app

- New simplified official project description
- Improvements to the index admin interface
- Removal of installation statistics gathering
- Remove unused folder tag
- Fix usage of ugettext to ugettext_lazy
- Increase size of the lock name field
- New style documentation

Upgrading from a previous version

Using PIP

Type in the console:

```
$ pip install mayan-edms==1.1.1
```

the requirements will also be updated.

Using Git

If you installed Mayan EDMS by cloning the Git repository issue the commands:

```
$ git reset --hard HEAD  
$ git pull
```

otherwise download the compressed archived and uncompress it overriding the existing installation.

Next upgrade/add the new requirements:

```
$ pip install --upgrade -r requirements.txt
```

Common steps

Add new static media:

```
$ mayan-edms.py collectstatic --noinput
```

The upgrade procedure is now complete.

Backward incompatible changes

None

Bugs fixed or issues closed

None

Mayan EDMS v1.1 release notes

Released: February 10, 2015

Welcome to Mayan EDMS v1.1

What's new in Mayan EDMS v1.1

Celery

All background tasks processing has been converted to use [Celery](#). By default Mayan EDMS runs in “Eager” until a broker and result backend are configured as per [Celery's documentation](#). This change made the built-in scheduler and `job_processing` apps obsolete, both were removed.

Views namespaces

All views are [namespaced](#) with the name of the app which defines them. If you have developed 3rd party apps for Mayan EDMS be sure to update any reference to a view by prepending the app name to the view name.

Removal of the splash screen

The static image home screen has been replaced with a quick links view, showing the most used actions: Uploading documents, viewing recent documents, viewing all documents and searching documents.

Sending and receiving documents via email

A link or entire documents can be sent as attachments via email. Documents can also be received via email with the addition of two document sources named IMAP and POP3 which correspond to the mail protocol used to fetch the documents. Read Django's [email configuration settings documentation](#) for more details on how to set up mail serving.

Update to Django 1.6.8

Mayan EDMS has been updated to use Django 1.6.8.

Events app

The built-in history app has been removed in favor of a new events wrapper app for [Django activity stream](#)

Watch folders

Filesystem folders can be monitored for change and their files automatically uploaded as documents in Mayan EDMS.

Vagrant file included

A vagrant file is now included to allow developers to provision a virtual machine with the latest development version of Mayan EDMS.

User locale profile (language and timezone)

Interface language and locale setting can now be setup for each user and are not installation wide as before. Date and times offsets are automatically adjusted to each user's timezone settings.

Document states

A new simple workflow app that can represent document states has been included.

Explicit document types needed per index

Indexes can now be tied to document types, eliminating the need to update indexes for every document update. Indexes will only update when a document of the type to which they are associated is updated.

Optional and required metadata types

Metadata types can now be assigned in two ways to documents types, as optional or required. Values for required metadata types as the name implies, must be entered for documents to be able to be uploaded. Optional metadata types on the other hand can be left blank by the user.

Bulk document type change

It is now possible to change the document type of previously uploaded documents. When the document type of a document is changed the metadata values are reset and the metadata types of the new document type are automatically assigned.

New release cycle

Starting with this version a new release cycle methodology will come into effect. The goal of this release cycle is to allow two series of versions of Mayan EDMS to be active at a given time: A new major version with new functionality and a minor version providing upgrades and fixes. This release (1.1) will be active and supported during releases of versions 2.x, but will go into end-of-life as soon as version 3.0 is released, at which time version series 2.x will go into maintenance mode.

Deprecation of Python 2.6

Series 1.0 of Mayan EDMS will be the last series supporting Python 2.6. Series 2.0 will be using Django 1.7.x which itself requires Python 2.7 or later.

Improved testings

Mayan EDMS is now automatically tested against SQLite, MySQL and PostgreSQL.

API updates

Many new API endpoints have been added exposing the majority of Mayan EDMS functionality.

Messages contextual improvements

Many updates and simplifications were made to the source text messages to reduce the difficulty of translating Mayan EDMS and maintaining the contextual meaning of the text messages.

Improved method for custom settings

Custom settings now use a string based value, it is no longer needed to import classes when customizing a setting:

```
from custom_app.backends import CustomStorageBackend
DOCUMENTS_STORAGE_BACKEND = CustomStorageBackend
```

Instead the fully qualified name of the class must be passed as the setting value:

```
DOCUMENTS_STORAGE_BACKEND = 'custom_app.backends.CustomStorageBackend'
```

Removal of the OCR config setting

OCR behavior is now a document type property meaning that it can be turned on or off for specific document types.

Per document language setting

Previously the document language used for OCR was specified for the entire installation. If documents in multiple languages were uploaded some suffered lower success rates. Now the language of each document can be specified.

Metadata validation and parsing support

It is now possible to create functions to validate metadata value input or parse and store corrected values. Three sample metadata validation functions are included: `Parse date and time`, `Parse date and time` and `Parse time`.

Removal of 960 grid system in favor of Pure CSS's grid system

By using Pure CSS's column based grid system, the move towards a Bootstrap UI migration has advanced greatly.

Simplified UI

All user actions as well as the logout button are now under the user functions section.

Stable PDF previews generation

The way PDFs were being generated has been improved greatly eliminating spurious segmentation faults at the expense of a small speed penalty.

More technical documentation

Many new sub topics were added to the development section of the documentation to allow developers to better understand the inner workings and philosophies of Mayan EDMS.

Other changes

- Removal of the MAIN_SIDE_BAR_SEARCH setting
- Removal of THEMES and themes support
- Removal of VERBOSE_LOGIN setting
- Removal of graphics backend supported file format list view
- Removal of the MAIN_DISABLE_ICONS setting
- Removal of specialized Sentry support
- Removal of the MAIN_ENABLE_SCROLL_JS setting
- Remove hardcoded root (/) redirections
- Removal of APScheduler as a requirement
- Removal of the scheduled jobs view
- Removal of the web_theme app
- Removal of the sources icon selection support
- Removal of the in-app help panels
- Removal of the duplicate document search feature
- Removal of filesystem document indexes mirroring feature
- Improve sources app model sub classes and inheritance handling
- Addition of CORS support to the API

Upgrading from a previous version

IMPORTANT! Before running the upgrade make sure none of your documents have duplicated metadata types, meaning that the same metadata type must not appear twice for any given document.

If you installed Mayan EDMS by cloning the Git repository issue the commands:

```
$ git reset --hard HEAD
$ git pull
```

otherwise download the compressed archived and uncompress it overriding the existing installation.

Next upgrade/add the new requirements:

```
$ pip install --upgrade -r requirements.txt
```

Migrate existing database schema with:

```
$ mayan-edms.py migrate dynamic_search 0001 --fake
$ mayan-edms.py migrate history 0001 --fake
$ mayan-edms.py migrate linking 0001 --fake
$ mayan-edms.py migrate lock_manager 0001 --fake
$ mayan-edms.py migrate tags 0001 --fake
$ mayan-edms.py migrate
```

During the migration several messages of stale content types can occur:

The following content types are stale and need to be deleted:

```
metadata | documenttypedefaults
metadata | metadataset
metadata | metadatasetitem
ocr | documentqueue
ocr | queuedocument
sources | watchfolder
sources | outofprocess
sources | webform
sources | stagingfolder
tags | tagproperties
```

Any objects related to these content types by a foreign key will also be deleted. Are you sure you want to delete these content types? If you're unsure, answer 'no'.

Type 'yes' to continue, or 'no' to cancel:

You can safely answer “yes”.

Add new static media:

```
$ mayan-edms.py collectstatic --noinput
```

Remove unused dependencies:

```
$ pip uninstall APScheduler
$ pip uninstall django-taggit
```

The upgrade procedure is now complete.

Backward incompatible changes

- You will have to redefine your document sources due to the new extended models for this app.
- Check your configurations of smart links and indexes to use the newly provided arguments.

Bugs fixed or issues closed

- [issue #30](#) Document workflows
- [issue #32](#) Watched folders
- [issue #34](#) Postprocessing document queue
- [issue #35](#) Metadata validation

- [issue #37](#) Add from `__future__` import `unicode_literals`
- [issue #39](#) Capitalization of messages
- [issue #40](#) Update references to root ('/') path
- [issue #46](#) Advanced search past 1st page
- [issue #49](#) Problems with large pdf files
- [issue #50](#) raise `CommandNotFound(path)`
- [issue #51](#) Search with ANONYMOUS error
- [issue #55](#) Document approval cycle?
- [issue #56](#) Removal of non essential features, views, models
- [issue #57](#) Migrate to Celery for task query and periodic tasks
- [issue #64](#) Pluralize messages properly
- [issue #65](#) Backport the ability to receive documents via email
- [issue #66](#) Python 3 compatibility: Add from `__future__` import `unicode_literals` and remove all `u'`
- [issue #68](#) Revise and update the use `gettext` vs. `gettext_lazy`
- [issue #69](#) Feature removal: remove "Unregistered" message from the title bar
- [issue #71](#) Add retry support to the converter task
- [issue #72](#) Delete unused static icons
- [issue #74](#) Cache a document's first document version
- [issue #75](#) Move automatic OCR queuing from a configuration settings to a property of Document Type model
- [issue #77](#) Add document view permission support to the search app
- [issue #78](#) `COMMON_TEMPORARY_DIRECTORY` seems not to be used everywhere
- [issue #79](#) Error installing
- [issue #82](#) Make document type a required field
- [issue #83](#) Simplify source app views and navigation
- [issue #84](#) Remove template context variable `'object_name'` to improve translations
- [issue #85](#) Reset page count for a single document
- [issue #86](#) Move migrations to new `'south_migrations'` folders
- [issue #87](#) Per document language selection
- [issue #88](#) Remove metadata type selection from the upload wizard
- [issue #89](#) Allow metadata types to be required for specific document types
- [issue #90](#) Remove the `app_registry` app
- [issue #91](#) Don't preserve the `?page=` URL query string value when switching sources during document upload
- [issue #92](#) Make `register_multi_item_links` class aware
- [issue #95](#) Installation error on Mac OSX; `OSError: [Errno 2] No such file or directory`
- [issue #96](#) Remove hard code User model references
- [issue #97](#) Make multi item links a drop down list

- [issue #104](#) Finish polishing metadata validation patch
- [issue #105](#) Tie smart links setups to document types
- [issue #106](#) Convert document indexing app actions to Celery
- [issue #107](#) Restrict document metadata addition and removal
- [issue #108](#) New home screen
- [issue #109](#) Add Roles API endpoints
- [issue #111](#) Add Checkouts API endpoints
- [issue #112](#) Add OCR API endpoints
- [issue #114](#) Implement UI language as user preference
- [issue #116](#) Add documentation topic explicitly noting the binary requirements
- [issue #118](#) When a metadata type is removed from a document type, remove it from all the documents of that type
- [issue #119](#) When a required metadata type is added to a document type, add it to all documents of that type
- [issue #126](#) Failing migration with SQLite
- [issue #127](#) Failing migration with Postgres
- [issue #128](#) Add Indexes API endpoints
- [issue #129](#) Search api shouldn't memorize requested page as part of the query
- [issue #130](#) Users API is not working correctly
- [issue #131](#) Is there an API to update a user's password?
- [issue #137](#) Enhancement of language selection
- [issue #138](#) Possibility to keep zoom factor
- [issue #139](#) Translatability of language selection
- [issue #140](#) Thumbnail creation for ods crashing
- [issue #143](#) Exception Value: 'exceptions.ValueError' object has no attribute 'messages'
- [issue #144](#) Behavior of 'Edit metadata' (Recent Documents)
- [issue #146](#) Periodic task not initiated for mail boxes and watch folders
- [issue #149](#) Attribute error in document download
- [issue #150](#) Double second menu entry
- [issue #152](#) Document content empty
- [issue #153](#) south migration with postgres: documents: 031_remove_orphan_documents
- [issue #154](#) south migration with postgres: tags: 0002_auto__add_tag__chg_field_tagproperties_tag
- [issue #157](#) upload new version of a document not working
- [issue #158](#) Plural form not matching singular form in ocr app bug i18n

Mayan EDMS v1.0 release notes

August 26, 2014

Welcome to Mayan EDMS v1.0!

Overview

After a year of low activity the focus of this release was to get the code updated to work with the most recent version of Django and the most recent version of the dependencies Mayan EDMS uses. The goal was to have a strong and stable 1.0 release so reduction, trimming, warning reductions and bug fixes were the primary focus of this cycle. Because of this not much emphasis was placed on adding new features, or at least new features that could have the potential to break things.

What's new in Mayan EDMS v1.0

New home

The growth and reach of the project has necessitated for a while the move of the project to its own organization in Github. This move has finally been done, the new URL of the project is: <https://github.com/mayan-edms/mayan-edms>

Django 1.6

This release updates the required version of Django to 1.6, bringing with it not only new features, but also a lot of security updates, a new project directory structure and new deployment methods.

Translation updates

The translation for all languages were synchronized to the latest transifex project sources. Translation completion as reported by Transifex:

- English - 100%
- Spanish - 100%
- Arabic - 96%
- Bosnian - 96%
- French - 96%
- German - 96%
- Romanian - 96%
- Russian - 96%
- Italian - 77%
- Portuguese - 76%
- Dutch - 68%
- Portugese (Brazil) - 67%
- Bulgarian - 62%

- Danish - 42%
- Vietnamese - 40%
- Polish - 39%
- Hungarian - 27%
- Indonesian - 18%
- Slovenian - 17%
- Persian - 6%
- Croatian - 3%
- Turkish - 3%

Model updates

There were some convenience properties created to allow quick access to a document's version and pages. These custom properties were removed and an official method to access these properties as provided by Django is now used.

Source code improvements

A circular import of metadata and document_index apps code from the documents app was removed. Document index updates are now handled via signals, not called directly as before. Hundreds of PEP8 style fixes, unused import removals, unused variables removals and removal of remarked or unused code. Removal of the DEVELOPMENT flag (was used to trigger static media serving during development), this is now handled by the DEBUG flag. The DEBUG flag is now set to *True* by default as per Django 1.6 defaults. Removed usage of Django's JSON libraries using Python's JSON library instead. Update of time and date use to use Django's new timezone aware data and time handling. Removal of custom code in favor of using modules provided by Django or by existing 3rd party libraries. Unification of code used for equal or similar purpose in various modules.

Removal of included 3rd party modules

One last 3rd party module was included with the source code of Mayan EDMS. This module is now available on PyPI and fetched during the installation instead of being included.

Automatic testing and code coverage check

Some initial tests were added, which will help with the detection of regressions or bugs when adding new features. More tests are needed, but the initial work has being started.

Update of required modules and libraries versions

Many of the required modules and libraries have been updated to a more recent version if not to their most recent released version.

Database connection leaks

Stale database connection being left open by scheduler tasks are now explicitly closed. This avoids consumption of the pool of database connections, increases stability and reduces memory usage.

Deletion of detached signatures

Detached signatures can now be deleted, if accidentally added to the wrong document.

Removal of fabric based installations

These files are now part of their own project and located at <https://github.com/mayan-edms/mayan-fabric>

Pluggable OCR backends

A commonly requested feature, it is now possible to write backends drivers to do document OCR using software or services other than Tesseract.

OCR improvements

OCR queue state is now reset when reloading Mayan EMDS, avoiding the OCR queue to remain locked. *unpaper* binary is now an optional pre OCR requirement, the OCR queue will now continue working if *unpaper* is not installed. Addition of post OCR processing support for French and German.

License change

Mayan EDMS is now licensed under the Apache 2.0 license. This means many things but the main change is that inclusion of Mayan EDMS into commercial products is now explicitly allowed.

PyPI package

Mayan EDMS has been packaged and submitted to the [PyPI](https://pypi.org/) Python Package Index making it even easier to install and use.

New REST API

This release feature a completely new REST API and automatic API documentation. This new API is also used internally by *Mayan EDMS* itself.

Other changes

More office document types are now recognized and supported. More file types are now supported as text files and properly previewed and parsed. Removal of the legacy `runserver.sh` and `runserver_plus.sh` scripts. New document preview generation and display pipeline, faster, simpler. Inclusion of a proof of concept compressed storage backend.

Upgrading from a previous version

If you installed Mayan EDMS by cloning the Git repository issue the commands:

```
$ git reset --hard HEAD
$ git pull
```

otherwise download the compressed archived and uncompress it overriding the existing installation.

Next upgrade/add the new requirements:

```
$ pip install --upgrade -r requirements.txt
```

Migrate existing database schema with:

```
$ mayan-edms.py migrate acls 0001 --fake
$ mayan-edms.py migrate taggit 0001 --fake
$ mayan-edms.py migrate ocr 0001 --fake
$ mayan-edms.py migrate
```

Add new static media:

```
$ mayan-edms.py collectstatic --noinput
```

The upgrade procedure is now complete.

Backward incompatible changes

- If using the SQLite3 database option, rename the file named *mayan.sqlite* file to *db.sqlite3* and move it to the new *media* provided folder.
- Also move to the *media* folder any *gpg_home*, *document_storage* and *image_cache* folders from your previous installation.

Bugs fixed or issues closed

- None

3.1.4 0.13 release

Mayan EDMS v0.13 release notes

December 2012

Welcome to Mayan EDMS v0.13!

Overview

Initially this version was meant as a the third maintenance release of the 0.12 series, but with the amount of changes and updates that were included it was obvious this was not just a bug fix version anymore hence the version jump to 0.13 instead of 0.12.3

What's new in Mayan EDMS v0.13

Django 1.3.5

This release updates the required version of Django to 1.3.5 to take advantage of the security fixes added to that version of the framework.

Bulk document tagging and untagging

As requested by issue [issue #31](#) this feature has been implemented and enabled in this version. Attaching or removing tags from a large number of documents is now much easier.

Registration

Based on requests made by the community for greater commercial support and services for Mayan EDMS, a new feature has been added that allows users to register their copies of Mayan EDMS and better help users with commercial support packages. Registration for non commercial users is voluntary and optional, and in no way affects the functionality of Mayan EDMS. However even for non commercial users registration offers the advantage of automatically branding the user's copy of Mayan EDMS with their name or the company name in the title area.

Per document type indexing

Index can now be restricted to update only on specific document types, this greatly increases the usefulness of indexes, and prevents unwanted index updates.

Bootstrap

Setting up Mayan EDMS after installation has been identified by users as the main difficulty when knowledge about Mayan EDMS is relatively low. To address this situation a new feature that provides preconfigured setups has been added. These preconfigured setups are published in the Mayan EDMS website and upon synchronization are available to users, this gives users access and integrators access to new setups without having to wait for new versions of Mayan EDMS. Aside from including preconfigured setups, the new bootstrap app has the ability of dumping an user's current setup into a serialized text file which can be tweaked by hand and sent via email to other users. The possibilities enabled by this range from company wide defaults setups to consultants providing their clients with customized setups without having to access their clients' Mayan EDMS instances. JSON, YAML and a custom YAML format (<http://djangosnippets.org/snippets/2461/>) are supported by this new app.

Add documents to folders in bulk

As requested, the ability to add more than one document at a time to a selected folder has been added.

Translation updates

The translation for all the current languages were synchronized to the latest transifex project sources.

Model updates

Several small fixes to the behavior of some model were added, namely that the names of some models should be unique. The document type name as well as the metadata set name were two models that were updated to behave this way.

Navigation updates

There were some additions and changes to the navigation to make it more intuitive or to add an alternate way to access the same information better. The bulk attachment of tags was one of these. Previously users were added or removed from groups, now users can be assigned to groups without leaving the user view. The name of an existing metadata set can now be edited and what was previously called metadata set edit is now more aptly named metadata members which adds or removes metadata types into a single organizational unit. An error that caused a duplicate menu link in the document type filename view was also fixed.

Support for UNOCONV dropped

Support for converting office documents by calling LibreOffice via UNOCONV has been disabled for a while ever since LibreOffice start including document conversion support from the command line. This version completely removes any traces of code and configuration options related to UNOCONV.

Optimizations

Inspired by the idea of getting Mayan EDMS running effectively on low power hardware such as the Raspberry Pi, several rounds of profiling and optimization were done.

Some caching optimization were introduced to the permission model, resulting in a speed increase of 33% in rendering time on views with heavy permission checking and a 18% query reduction on cache hits.

Upgrading from a previous version

If you installed Mayan EDMS by cloning the Git repository issue the commands:

```
$ git reset --hard HEAD
$ git pull
```

otherwise download the compressed archived and uncompress it overriding the existing installation.

Next add the new requirements:

```
$ pip install -r requirements/production.txt
```

Migrate existing database schema with:

```
$ ./manage.py migrate document_indexing
$ ./manage.py migrate metadata 0001 --fake
$ ./manage.py migrate metadata
$ ./manage.py migrate bootstrap
$ ./manage.py migrate documents
$ ./manage.py migrate registration
```

Add new static media:

```
$ ./manage.py collectstatic --noinput
```

The upgrade procedure is now complete.

Backward incompatible changes

- None

Bugs fixed or issues closed

- [issue #28](#) “Document indexing based on filesystem problem”
- [issue #29](#) Default ACL documentation
- [issue #30](#) “Ubuntu 12.0.4 LTS complaining about missing ‘fabfile/templates/settings_local.py’ (master branch)”
- [issue #31](#) “Bulk add tags Feature request”
- [issue #32](#) “Error run \$python manage.py syncdb”
- [issue #33](#) “Fetches translations from Transifex. “
- [issue #34](#) “Initial syncdb fails because of import time database access”
- [issue #35](#) “Fix first install syncdb on Postgres”
- [issue #37](#) “When 19 >= &document_type_id >= 10 in url, error page not found”
- [issue #39](#) Feature request “Bulk move folders”

3.1.5 0.12 release

Mayan EDMS v0.12.2 release notes

June 2012

This is the second maintenance release of the 0.12 series.

Overview

As with the previous release bug fixes and minor feature were the focus for this release too. [issue #24](#) has been fixed and document check outs have been added too as per the feature request posted as [issue #26](#). The way the history events for a document are presented has been improved and it is now more useful as it provides filtering by event type. To improve the diagnosis of installation of runtime error a simple view showing the number of internal interval jobs being used by Mayan EDMS as well as a new app which shows a detail of the current installation environment were added.

What’s new in Mayan EDMS v0.12.2

Smarter auto admin creation

Mayan EDMS creates a administrator user during the database creation phase to reduce the amount of steps required for a functional install. The creation of this account is controlled by the configuration

option `COMMON_AUTO_CREATE_ADMIN`, the username of the account is specified with the configuration option `COMMON_AUTO_ADMIN_USERNAME` and the password of this account by the option `COMMON_AUTO_ADMIN_PASSWORD`. Previously the `COMMON_AUTO_ADMIN_PASSWORD` defaulted to 'admin' which created an administrator account of username 'admin' with a password of 'admin'. The new default is to randomize an initial password and show this password at the login screen until the administrator password is changed.

Document check outs

As per the feature request filed under [issue #26](#), a new document check out and check in functionality has been added. Users can now check out a document and lock new version of it from being uploaded to avoid editing conflicts. Document check outs have an expiration period after which Mayan EDMS will automatically check them back in to avoid a permanent document lockout. Only the user who has checked out a document can upload new versions of it or check the document back in before the expiration period, unless being granted the `Allow overriding check out restrictions` or `Forcefully check in documents` permission respectively.

Installation environment app

Diagnosing remote installations of web based applications without access to the command line can be a bit hard, to alleviate this situation a new installation environment details app has been added. The purpose of this app is to provide support staff information about the physical environment where Mayan EDMS has been installed. To avoid possible security compromises only administrators can access this app.

Editable compressed documents filename

Previously when downloading more than one document in a compressed manner, Mayan EDMS would produce a file with the name `document_bundle.zip` for download. A different filename can now be specified at the same download dialog.

German translation

A German language translation has been added thanks to Tetja Rediske and Tilmann Sittig.

Statistics gathering

Previous attempts at gathering usage statistics have been met with deficient results. User participation in surveys as well as the quality of the data entered by users was disappointing. That is why this version of Mayan EDMS features an anonymous statistics gathering functionality.

Upgrading from a previous version

Start off by adding the new requirements:

```
$ pip install -r requirements/production.txt
```

Migrate existing database schema with:

```
$ ./manage.py migrate common 0001 --fake
$ ./manage.py migrate common
$ ./manage.py migrate checkouts
$ ./manage.py migrate installation
```

Add new static media:

```
$ ./manage.py collectstatic --noinput
```

The upgrade procedure is now complete.

Backward incompatible changes

- None

Bugs fixed

- [issue #24](#) “Duplicated filename extension when uploading a new version of a document”
- [issue #26](#) “checkout feature request”

Stuff removed

- Feedback app

Mayan EDMS v0.12.1 release notes

May 2012

This is the first maintenance release of the 0.12 series.

Overview

While bug fixes and minor feature were the focus for this release, some bigger changes were included because of their importance. The parsing of documents saw a complete rewrite being now class based and allows for more than one parser per mimetype with sequential fallback. This provides the best text extraction on deployments where users have control over the installation and basic extraction when deploying on the cloud or other environments where users don't have the ability to install OS level binaries.

What's new in Mayan EDMS v0.12.1

Fabric file (fabfile)

A Fabric file is included to help users not very familiar with Ubuntu, Python and Django install Mayan EDMS, or for system administrators looking to automate the install whether in local or remote systems. At the moment the fabfile will install Mayan EDMS in the same configurations listed in this documentation, that is: (Ubuntu/Debian/Fedora) + virtualenv + Apache + MySQL. Feel free to submit your configuration settings and files for different databases, webserver or Linux distribution. More configurations will be added to the fabfile as more are tested.

Documentation update

The installation instructions were updated to include the installation of the `libpng-dev` and `libjpeg-dev` libraries as well as the installation of the `poppler-utils` package. An additional step to help users test their new installation of Mayan EDMS was also added.

Translations

The Italian translation has been synchronized with the source files at Transifex and finished to %100 completion.

Usability improvements

The index instance view now feature the same multi document action buttons (Submit to OCR, delete, download, etc) as the mail and recent document views.

Better office document conversion

A new method of converting office documents has been implemented, this new method doesn't require the use of the command line utility `UNOCONV`. If this new method proves to work better than previous solutions the use of `UNOCONV` may be deprecated in the future. The conversion method adds just one new configuration option: `CONVERTER_LIBREOFFICE_PATH` which defaults to `'/usr/bin/libreoffice'`.

Better PDF text parsing

Brian E. submitted a patch to use the Poppler package `pdftotext` utility to extract text from PDF files. This is now the default method Mayan EDMS will execute to try to extract text from a PDF and failing that will fallback to the previous method. This change add a new configuration option: `OCR_PDFTOTEXT_PATH` to specify the location of the `pdftotext` executable, it defaults to `'/usr/bin/pdftotext'`. Be sure to install the `poppler-utils` os package to take advantage of this new parser.

Changed defaults

The OCR queue is now active by default when first created during the `syncdb` phase and the `OCR_AUTOMATIC_OCR` option now defaults to `True`. These two changes are made to reduce the steps required for new users to start enjoying the benefits of automatic text extraction from uploaded documents without having to read the documentation and have a more functional default install.

Upgrading from a previous version

Start off by adding the new requirements:

```
$ pip install -r requirements/production.txt
```

Migrate existing database schema with:

```
$ ./manage.py migrate documents
```

Install the `poppler-utils` package:

- Ubuntu, Debian:

```
$ apt-get install -y poppler-utils
```

- Fedora:

```
$ yum install -y poppler-utils
```

The upgrade procedure is now complete.

Backward incompatible changes

- None

Bugs fixed

- Issue #25 “Office document conversion error”

Stuff removed

- None

Mayan EDMS v0.12 release notes

February 2012

Welcome to Mayan EDMS v0.12!

This release commemorates Mayan EDMS first anniversary!

Overview

Aside from new features, the focus of this release of Mayan EDMS also been about improving the code and documentation quality standard even further. The permission system has been completely overhauled to make it entire class based. The other big change is the addition of object level permissions, with this new system being applied to documents, folder, tags and smart links. There is also a small batch of navigation improvements. Big code cleanup and lots of changes ‘under the hood’, most of these are not visible to the end user, but make the code cleaner and more manageable so that more and better features can be added in future releases:

- Absolute imports used throught the code
- All app permissions have been move to a separate permissions.py file per app
- Complete permission system refactor.
- Document signining code move to it’s own app
- Initial unit tests
- A lot of logging used throught the entire project.
- Much functionality moved to model managers.
- A lot of code converted into classes.

- Coding style improvements.
- Template user authentication state logic improvements, for stonger prevention against intrusion or unintentional display or access of restricted data.
- Removal of remarked code.

What's new in Mayan EDMS v0.12

ACL support

- Object level access control is now in place for documents, folders, tags and smart links. What this means is that administrators can now grant permissions to users, groups or roles on for specific objects. A more in-depth explanation of how this new ACL system can be found in the *3 tier access control* section of the permissions chapter.
- Default class ACL support. Administrators can setup the access control lists that new documents, folders and tags will automatically inheric when created. Aside from asigning permission to users, groups and roles to specific objects, there is a special user called *Creator*, use to allow the access control list that the actual creator of an object will inherit.

Anonymous user support

Anonymous user support is a two tier function, first is the addition of the `COMMON_ALLOW_ANONYMOUS_ACCESS` configuration option that allows non authenticated user to browse all the pages of a Mayan EDMS installation. The second part of this support is the ability to assign permissions or individual access to objects to anonymous users.

Translations

A new Italian translation is available, provided by SeeOpen.IT (www.seeopen.it, info@seeopen.it) as well as complete Russian translation update by Sergei Glita. Included in this release also the initial translation to Polish by mic.

Usability improvements

- Detached signature behavior improved, uploading a new detached signature erases the previous one.
- Usability improvement in the role member's add/removal form, by using HTML's option groups tag property

2 Step download process

The code for downloading single and multiple document and document versions has been merged with compression support also added. This allows for the download of documents in their original format or compressed and well as the download of several documents in a single compressed file.

Customizable GPG home directory

Addition of the `SIGNATURES_GPG_HOME` configuration option to let administrators set Mayan EDMS's GPG instance home directory, used to store keyrings and other GPG configuration files.

Out of process bulk uploading

A management command has been added to help upload a large number of documents from a compressed file. For information about this new feature check the Initial data loading chapter.

Out of process user import

A management command has been added to import a large number users from a CSV file. More information about this new feature can also be found in the Initial data loading chapter.

Refactored document indexing

The document indexing functionality has been improved and moved from experimental stage to beta stage. Index configuration menus are now available on the `Setup` menu and allows administrators to create skeleton trees that will be populated with document links depending on their metadata and properties. These populated trees can also be mirrored on the physical filesystem and shared using Samba or another filesharing server giving users a structured view of the documents contained within Mayan EDMS from the `Indexes` tab or from a mirrored index shared via the network. A new configuration option has been added, `DOCUMENT_INDEXING_FILESYSTEM_SERVING`, which maps the index internal name with the physical directory where such index will be mirrored on disk.

Help shape Mayan EDMS's future

Included in this version is a small feedback application, found under the `About` main menu, where users by just answering some questions can help determine the priority of the next planned features on the pipeline, or even help add new features if enough requests are received. All questions are optional but answering as many as possible will help greatly understand the need of the Mayan EDMS user base.

Staging file previews

The staging file previews now show the filename for easier identification and speedier upload selection. The staging files previews are now treated as a gallery which means that users can preview an entire page of staging files without having to click and close each one individually.

Upgrading from a previous version

Start off by adding the new requirements:

```
$ pip install -r requirements/production.txt
```

Then create the new database structures with:

```
$ ./manage.py syncdb
```

Afterwards migrate existing database schema with:

```
$ ./manage.py migrate permissions 0001 --fake
$ ./manage.py migrate permissions
```

When the following message appears

The following content types are stale and need to be deleted:

```
permissions | permission
```

Any objects related to these content types by a foreign key will also be deleted. Are you sure you want to delete these content types?

If you're unsure, answer 'no'.

Type 'yes' to continue, or 'no' to cancel:

Type **yes** and press **Enter**

And continue migrating database schema with:

```
$ ./manage.py migrate documents
$ ./manage.py migrate document_signatures
$ ./manage.py migrate folders 0001 --fake
$ ./manage.py migrate folders
$ ./manage.py migrate document_indexing 0001 --fake
$ ./manage.py migrate document_indexing
$ ./manage.py migrate sources 0001 --fake
$ ./manage.py migrate sources
```

Again when a similar messages appears

The following content types are stale and need to be deleted:

```
document_indexing | indexinstance
```

Any objects related to these content types by a foreign key will also be deleted. Are you sure you want to delete these content types?

If you're unsure, answer 'no'.

Type 'yes' to continue, or 'no' to cancel:

Type **yes** and press **Enter**

The upgrade procedure is now complete.

Backward incompatible changes

The permission system has been completely reworked so sadly this is a place where even data migration can't help and the permissions assigned to roles will be lost during the upgrade to version 0.12. Users, groups and roles will be preserved only permissions need to be assigned again, so write down your role permission setup before upgrading.

Bugs fixed

- Issue #17, special thanks to Dave Herring for all the help including access to a machine suffering with the issue, and to Sergei Glita for his research and eventual find of the core cause.
- Statistics fixes.
- Fixed `get_image_cache_name` regression in the OCR app.

Stuff removed

- Support for Celery and Sentry has been drop for now.
- Removed the ‘db_index’ argument from Text fields definition and migrations as it was causing error messages for MySQL users, thanks to Sergei Glita for reporting this one.
- Configuration options removed:
 - `OCR_CACHE_URI`
 - `DOCUMENT_INDEXING_FILESYSTEM_FILESERVING_PATH` - Use the newest `DOCUMENT_INDEXING_FILESYSTEM_SERVING`
 - `DOCUMENT_INDEXING_FILESYSTEM_FILESERVING_ENABLE` - Use the newest `DOCUMENT_INDEXING_FILESYSTEM_SERVING`

3.1.6 0.11 release

Version 0.11.1

- Fixed a document deletion regression
- Improves error detection when importing keys from a keyserver, catching the exception `KeyImportError` and not `KeyFetchingError`
- Fixes a wrong method call when verifying signatures for the first time upon document uploading
- `django-compress` is now disabled by default to avoid problems when deploying with `DjangoZoom`
- Improve post metadata set delete redirection

Version 0.11

- Support for signed documents verification added, embedded and detached signatures are supported. When verifying a document Mayan EDMS will try to fetch the public key from the list of keyservers provided in the configuration option `SIGNATURES_KEYSERVERS` (which defaults to ‘pool.sks-keyservers.net’). A public key management view has been added to the setup menu as well as a key query and fetching view to manually import keys from a keyserver.
- Added support for document versions. Users can upload unlimited amount of versions for a document using a very flexible document version numbering system, users can also revert to a previous document version.
- OCR queue processing improvements.
- Office documents handling improvements.
- Text extraction support for office documents.
- RTF text documents are now handled as office documents.
- Added a view to delete the document image cache, useful when switching converter backends or doing diagnostics.
- Added South to the requirements.
- Merged documents’ filename and extension database fiels into a single filename field, filename are store as uploaded not manipulation is done Users with existing data must install South and run the appropriate migrate commands:

```
$ pip install -r requirements/production.txt
$ ./manager syncdb
$ ./manage.py migrate documents 0001 --fake
$ ./manage.py migrate documents
```

- Added new office document mimetype * application/vnd.ms-office
- Fixed documents not saving the file encoding
- Removed extra slash in ajax-loader.gif URL fixes #15, thanks to IHLeanne for finding this one

3.1.7 0.10 release

Version 0.10.1

- Upgraded django-compressor to version 1.1.1, run:

```
$ pip install --upgrade -r requirements/production.txt
```

to upgrade

- django-compressor is now disabled by default, users must explicitly enable it adding COMPRESS_ENABLED=True to their settings_local.py file

Version 0.10

- Added a proper setup views for the document grouping functionality.
- Document grouping is now called smart linking as it relates better to how it actually works. The data base schema was changed and users must do the required:

```
$ ./manager syncdb
```

for the new tables to be created.

- Grappelli is no longer required as can be uninstalled.
- New smarter document preview widget that doesn't allow zooming or viewing unknown or invalid documents.
- New office document converter, requires:
 - LibreOffice (<https://www.libreoffice.org/>)
 - unoconv [version 0.5] (<https://github.com/dagwieers/unoconv>)
- The new office documents converter won't convert files with the extension .docx because these files are recognized as zip files instead. This is an issue of the libmagic library.
- New configuration option added CONVERTER_UNOCONV_USE_PIPE that controls how unoconv handles the communication with LibreOffice. The default of True causes unoconv to use **pipes**, this approach is slower than using **TCP/IP** ports but it is more stable.
- Initial *REST API* that exposes documents properties and one method, this new *API* is used by the new smart document widget and requires the package `djangoRESTframework`, users must issue a:

```
$ pip install -r requirements/production.txt
```

to install this new requirement.

- MIME type detection and caching performance updates.
- Updated the included version of jQuery to 1.7
- Updated the included version of JQueryAsynchImageLoader to 0.9.7
- Document image serving response now specifies a MIME type for increased browser compatibility.
- Small change in the scheduler that increases stability.
- Russian translation updates Sergey Glita
- Improved and generalized the OCR queue locking mechanism, this should eliminate any possibility of race conditions between Mayan EDMS OCR nodes.
- Added support for signals to the OCR queue, this results in instant OCR processing upon submittal of a document to the OCR queue, this works in addition to the current polling processing which eliminates the possibility of stale documents in the OCR queue.
- Added multiple document OCR submit link
- Re enabled tesseract language specific OCR processing and added a one (1) time language neutral retry for failed language specific OCR

3.1.8 0.9 release

Version 0.9.1

- Added handling percent encoded unicode query strings in search URL, thanks to Sergey Glita for reporting.
- Added a FAQ explaining how to fix MySQL collation related error when doing searches also thanks to Sergey Glita for reporting this one.

Version 0.9

- Simplified getting mimetypes from files by merging 2 implementations (document based and file based)
- Updated python converter backend, document model and staging module to use the new get_mimetype API
- Only allow clickable thumbnails for document and staging files with a valid image
- Removed tag count from the group document list widget to conserve vertical space
- Updated required Django version to 1.3.1
- Removed the included 3rd party module django-sendfile, now added to the requirement files.
 - User should do a pip install -r requirements/production.txt to update
- Changed to Semantic Versioning (<http://semver.org/>), with recommendations 7, 8 and 9 causing the most effect in the versioning number.
- Added Russian locale post OCR cleanup backend Sergey Glita
- Reduced severity of the messages displayed when no OCR cleanup backend is found for a language
- Complete Portuguese translation (Emerson Soares and Renata Oliveira)
- Complete Russian translation Sergey Glita)
- Added animate.css to use CSS to animate flash messages with better fallback on non JS browsers
- The admin and sentry links are no longer hard-coded (Meurig Freeman)

- Improved appearance of the document tag widget (<https://p.twimg.com/Ac0Q0b-CAAE1lfA.png:large>)
- Added `django_compress` and `cssmin` to the requirements files and enabled `django_compress` for CSS and JS files
- Added granting and revoking permission methods to the permission model
- Correctly calculate the mimetype icons paths when on development mode
- Added a new more comprehensive method of passing multiple variables per item in multi item selection views
- Used new multi parameter passing method to improve the usability of the grant/revoke permission view, thanks to Cezar Jenkins (<https://twitter.com/#!/emperorcezar>) for the suggestion
- Added step to the documentation explaining how to install Mayan EDMS on Webfaction
- Added an entry in the documentation to the screencast explaining how to install Mayan EDMS on DjangoZoom
- Added required changes to add Mayan EDMS to Transifex.com
- Fixed the apache contrib file static file directory name
- Added improved documentation

3.1.9 0.8 release

Version 0.8.3

- Added a Contributors file under the docs directory
- Moved the document grouping subtemplate windows into a document information tab
- Change the mode the setup options are shown, opting to use a more of a dashboard style now
- Changed the tool menu to use the same button layout of the setup menu
- Moved OCR related handling to the tools main menu
- Improved the metadata type and metadata set selection widget during the document upload wizard
- Added a view to the about menu to read the LICENSE file included with Mayan
- Added converter backend agnostic image file format descriptions
- Disable whitelist and blacklist temporarily, removed `document_type` field from interactive sources
- Fully disabled watch folders until they are working correctly
- Updated the project title to ‘Mayan EDMS’
- If ghostscript is installed add PDF and PS to the list of file formats by the python converter backend
- Use Pillow (<http://pypi.python.org/pypi/Pillow>) instead of PIL
 - Pillow is a fork of PIL with several updates including better jpeg and png library detection
 - Users must uninstall PIL before installing Pillow
- Updated the static media url in the login exempt url list
- Added remediation code to sidestep issue #10 caused by DjangoZoom’s deployment script executing the `collectstatic` command before creating the database structure with `syncdb`. Thanks to Joost Cassee (<https://github.com/jcassee>) for reporting this one.
- Perform extra validation of the image cache directory and fallback to creating a temporary directory on validation failure

- Fixed a source creation bug, that caused invalid links to a non existing source transformation to appear on the sidebar

Version 0.8.2

- Moved code to Django 1.3
 - Users have to use the `collectstatic` management command:

```
$ ./manage.py collectstatic
```

- The `site_media` directory is no more, users must update the media serving directives in current deployments and point them to the `static` directory instead
- The changelog is now available under the `about` main menu
- Grappelli no longer bundled with Mayan
 - Users must install Grappelli or execute:

```
$ pip install --upgrade -r requirements/production.txt
```

- Even easier UI language switching
- Added email login method, to enable it, set:

```
AUTHENTICATION_BACKENDS = ('common.auth.email_auth_backend.EmailAuthBackend',)
COMMON_LOGIN_METHOD = 'email'
```

Version 0.8.1

- Tags can now also be created from the main menu
- Added item count column to index instance list view
- Updated document indexing widget to show different icon for indexes or indexes that contain documents
- Replaced the Textarea widget with the TextAreaDiv widget on document and document page detail views
 - This change will allow highlighting search terms in the future
- Unknown document file format page count now defaults to 1
 - When uploading documents which the selected converted backend doesn't understand, the total page count will fallback to 1 page to at least show some data, and a comment will be automatically added to the document description
- Added new `MAIN_DISABLE_ICONS` to turn off all icons
 - This options works very well when using the `default` theme
- The default theme is now `activo`
- Improved document page views and document page transformation views navigation
- Added OCR queue document transformations
 - Use this for doing resizing or rotation fixes to improve OCR results
- Added reset view link to the document page view to reset the zoom level and page rotation
- Staging files now show a thumbnail preview instead of preview link

Version 0.8

- Distributed OCR queue processing via celery is disabled for the time being
- Added support for local scheduling of jobs
 - This addition removes celery beat requirement, and make it optional
- Improve link highlighting
- Navigation improvements
- Documents with an unknown file format now display a mime type placeholder icon instead of an error icon
- Mayan now does pre-caching of document visual representation improving overall thumbnail, preview and display speed
 - Page image rotation and zooming is faster too with this update
- Removed all QUALITY related settings
- `COMMON_TEMPORARY_DIRECTORY` is now validated when Mayan starts and if not valid falls back to creating its own temporary folder
- Added PDF file support to the python converter backend via ghostscript
 - This requires the installation of:
 - * ghostscript python package
 - * ghostscript system binaries and libraries
- Added PDF text parsing support to the python converter backend
 - This requires the installation of:
 - * pdfminer python package
- Added PDF page count support to the python converter backend
- Added python only converter backend supporting resizing, zooming and rotation
 - This backend required the installation of the python image library (PIL)
 - This backend is useful when Graphicsmagick or Imagemagick can not be installed for some reason
 - If understand fewer file formats than the other 2 backends
- Added default transformation support to document sources
- Removed `DOCUMENT_DEFAULT_TRANSFORMATIONS` setup options
- Document sources are now defined via a series of views under the setup main menu
- This removes all the `DOCUMENT_STAGING` related setup options
 - Two document source types are supported local (via a web form), and staging
 - However multiple document sources can be defined each with their own set of transformations and default metadata selection
- Use `python-magic` to determine a document's mimetype otherwise fallback to use python's `mimetypes` library
- Remove the included sources for `python-magic` instead it is now fetched from github by pip
- Removed the document subtemplates and changed to a tabbed style
- Added link to document index content view to navigate the tree upwards

- Added new option `MAIN_DISABLE_HOME_VIEW` to disable the home main menu tab and save some space
- Added new option to the web theme app, `WEB_THEME_VERBOSE_LOGIN` that display a more information on the login screen (version, copyright, logos)
- Added a confirmation dialog to the document tag removal view

3.1.10 0.7 release

Version 0.7.6

- Added recent searches per user support
 - The ammount of searches stored is controlled by the setup option `SEARCH_RECENT_COUNT`
- The document page zoom button are now disabled when reaching the minimum or maximum zoom level
- The document page navigation links are now disabled when view the first and last page of a document
- Document page title now displays the current page vs the total page count
- Document page title now displays the current zoom level and rotation degrees
- Added means set the expansion compressed files during document creation, via web interface removing the need for the configuration options: `UNCOMPRESS_COMPRESSED_LOCAL_FILES` and `UNCOMPRESS_COMPRESSED_STAGING_FILES`
- Added ‘search again’ button to the advances search results view
- Implementes an advanced search feature, which allows for individual field terms
 - Search fields supported: document type, MIME type, filename, extension, metadata values, content, description, tags, comments

Version 0.7.5

- Added a help messages to the sidebar of some views
- Renamed some forms submit button to more intuitive one
 - ‘Search’ on the submit button of the search form
 - ‘Next step’ on the document creation wizard
- Added view to list supported file formats and reported by the converter backend
- Added redirection support to multi object action views
- Renamed ‘document list’ link to ‘all documents’ and ‘recent document list’ to ‘recent documents’
- Removed ‘change password’ link next to the current user’s name and added a few views to handle the current user’s password, details and details editing

Version 0.7.4

- Renamed ‘secondary actions’ to ‘secondary menu’
- Added document type setup views to the setup menu
- Added document type file name editing views to the setup menu
- Fixed document queue properties sidebar template not showing

Version 0.7.3

- Refactored main menu navigation and converted all apps to this new system
- Multi item links are now displayed on top of generic lists as well as on the bottom
- Spanish translation updates
- Updated requirements to use the latest development version of django-mptt
- Improved user folder document removal views
- Added ability to specify default metadata or metadataset per document type
- Converted filename handling to use os.path library for improved portability
- Added edit source object attribute difference detection and logging to history app
- Missing metadata type in a document during a multi document editing doesn't raise errors anymore.
 - This allows for multi document heterogeneous metadata editing in a single step.
- Added document multi item links in search results
 - Direct editing can be done from the search result list
- Permissions are now grouped and assigned a group name
- Improved role management views
- Document type is now an optional document property
 - Documents can be created without an explicit document type
- Added support for per user staging directories
- Updated logos

Version 0.7

- Added confirmation dialogs icons
- Added comment app with support for adding and deleting comments to and from documents
- Updated requirements files as per issue #9
- Show tagged item count in the tag list view
- Show tagget document link in the tags subtemplate of documents
- Made comment sorted by oldest first, made comment subtemplate scrollable
- Rename comments app to document_comment to avoid conflict with Django's comment app
- Made document comments searchable

3.1.11 0.5 release

Version 0.5.1

- Applied initial merge of the new subtemplate renderer
- Fixed tag removal logic
- Initial commit to support document comments

- Updated so that loading spinner is displayed always
- Exclude tags from the local document upload form
- Added document tagging support
 - Requires installing `django-taggit` and doing a `sync-db`

Version 0.5

- Added tag list view and global tag delete support
- Added tag editing view and listing documents with an specific tag
- Changed the previewing and deleting staging files views to required `DOCUMENT_CREATE` permission
- Added no-parent-history class to document page links so that iframe clicking doesn't affect the parent window history
 - Fixes back button issue on Chrome 9 & 10
- Added per app version display tag
- Added loading spinner animation
- Messages tweaks and translation updates
- Converter app cleanups, document pre-cache, magic number removal
- Added OCR view displaying all active OCR tasks from all cluster nodes
- Disabled `CELERY_DISABLE_RATE_LIMITS` by default
- Implement local task locking using Django locmem cache backend
- Added doc extension to office document format list
- Removed redundant transformation calculation
- Make sure OCR in processing documents cannot be deleted
- PEP8, pylint cleanups and removal of relative imports
- Removed the obsolete `DOCUMENTS_GROUP_MAX_RESULTS` setting option
- Improved visual appearance of messages by displaying them outside the main form
- Added link to close all notifications with one click
- Made the queue processing interval configurable by means of a new setting: `OCR_QUEUE_PROCESSING_INTERVAL`
- Added detection and reset of orphaned ocr documents being left as 'processing' when celery dies
- Improved unknown format detection in the graphicsmagick backend
- Improved document conversion API
- Added initial support for converting office documents (only ods and docx tested)
- Added sample configuration files for supervisor and apache under contrib/
- Avoid duplicates in recent document list
- Added the configuration option `CONVERTER_GM_SETTINGS` to pass GraphicsMagicks specific commands the the GM backend
- Lower image conversion quality if the format is jpg

- Inverted the rotation button, more intuitive this way
- Merged and reduced the document page zoom and rotation views
- Increased permissions app permission's label field size
 - DB Update required
- Added support for metadata group actions
- Reduced the document pages widget size
- Display the metadata group numeric total in the metadata group form title
- Reorganized page detail icons
- Added first & last page navigation links to document page view
- Added interactive zoom support to document page detail view
- Spanish translation updates
- Added `DOCUMENTS_ZOOM_PERCENT_STEP`, `DOCUMENTS_ZOOM_MAX_LEVEL`, `DOCUMENTS_ZOOM_MIN_LEVEL` configuration options to allow detailed zoom control
- Added interactive document page view rotation support
- Changed the side bar document grouping with carousel style document grouping form widget
- Removed the obsolete `DOCUMENTS_TRANFORMATION_PREVIEW_SIZE` and `DOCUMENTS_GROUP_SHOW_THUMBNAIL` setting options
- Improved double submit prevention
- Added a direct rename field to the local update and staging upload forms
- Separated document page detail view into document text and document image views
- Added grab-scroll to document page view
- Disabled submit buttons and any buttons when during a form submit
- Updated the page preview widget to display a infinite-style horizontal carousel of page previews
- Added support user document folders
 - Must do a `syncdb` to add the new tables
- Added support for listing the most recent accessed documents per user
- Added document page navigation
- Fixed diagnostics url resolution
- Added confirmation dialog to document's find missing document file diagnostic
- Added a document page edit view
- Added support for the command line program `pdftotext` from the `poppler-utils` packages to extract text from PDF documents without doing OCR
- Fixed document description editing
- Replaced page break text with page number when displaying document content
- Implemented detail form readonly fields the correct way, this fixes copy & paste issues with Firefox
- New document page view
- Added view to add or remove user to a specific role

- Updated the jQuery packages with the web_theme app to version 1.5.2
- Made `AVAILABLE_INDEXING_FUNCTION` setting a setting of the documents app instead of the filesystem_serving app
- Fixed document download in FireFox for documents containing spaces in the filename
- If mime detection fails set mime type to `''` instead of `'unknown'`
- Use document MIME type when downloading otherwise use `'application/octet-stream'` if none
- Changed the way document page count is parsed from the graphics backend, fixing issue #7
- Optimized document metadata query and display
- Implemented OCR output cleanups for English and Spanish
- Redirect user to the website entry point if already logged and lands in the login template
- Changed from using SimpleUploadedFile class to stream file to the simpler File class wrapper
- Updated staging files previews to use sendfile instead of serve_file
- Moved staging file preview creation logic from documents.views to staging.py
- When deleting staging file, it's cached preview is also deleted
- Added a new setup option:
 - `FILESYSTEM_INDEXING_AVAILABLE_FUNCTIONS` - a dictionary to allow users to add custom functions
- Made automatic OCR a function of the OCR app and not of Documents app (via signals)
 - Renamed setup option `DOCUMENT_AUTOMATIC_OCR` to `OCR_AUTOMATIC_OCR`
- Clear node name when requeueing a document for OCR
- Added support for editing the metadata of multiple documents at the same time
- Added Graphics magick support by means of user selectable graphic conversion backends
 - Some settings renamed to support this change:
 - * `CONVERTER_CONVERT_PATH` is now `CONVERTER_IM_CONVERT_PATH`
 - * `CONVERTER_IDENTIFY_PATH` is now `CONVERTER_IM_IDENTIFY_PATH`
 - Added options:
 - * `CONVERTER_GM_PATH` - File path to graphicsmagick's program.
 - * `CONVERTER_GRAPHICS_BACKEND` - Backend to use: ImageMagick or GraphicMagick
- Raise ImportError and notify user when specifying a non existant converter graphics backend
- Fixed issue #4, avoid circular import in permissions/__init__.py
- Add a user to a default role only when the user is created
- Added total page count to statistics view
- Added support to disable the default scrolling JS code included in web_theme app, saving some KBs in transfer
- Clear last ocr results when requeueing a document
- Removed the 'exists' column in document list view, diagnostics superceded this
- Added 3rd party sendfile app (support apache's X-sendfile)
- Updated the get_document_image view to use the new sendfile app

- Fixed the issue of the strip spaces middleware conflicting with downloads
- Removed custom IE9 tags
- Closed Issue #6
- Allow deletion of non existing documents from OCR queue
- Allow OCR requeue of pending documents
- Invalid page numbers now raise Http404, not found instead of error
- Added an additional check to lower the chance of OCR race conditions between nodes
- Introduce a random delay to each node to further reduce the chance of a race condition, until row locking can be implemented or is implemented by Django
- Moved navigation code to its own app
- Reimplemented OCR delay code, only delay new document Added a new field: delay, update your database schema accordingly
- Made the concurrent ocr code more granular, per node, every node can handle different amounts of concurrent ocr tasks Added a new field: node_name, update your database schema accordingly
- Reduced default ocr delay time
- Added a new diagnostics tab under the tools menu
- Added a new option `OCR_REPLICATION_DELAY` to allow the storage some time for replication before attempting to do OCR to a document
- Added OCR multi document re-queue and delete support
- Added simple statistics page (total used storage, total docs, etc)
- Implemented form based and button based multi item actions (button based by default)
- Added multi document delete
- Fixed a few HTML validation errors
- Issues are now tracked using github
- Added indexing flags to ocr model
- Small optimization in document list view
- Small search optimization
- Display “DEBUG mode” string in title if `DEBUG` variable is set to True
- Added the fix-permissions bash script under misc/ folder
- Plugged another file descriptor leak
- Show class name in config settings view
- Added missing config option from the setup menu
- Close file descriptor to avoid leaks
- Don't allow duplicate documents in queues
- Don't raise `PermissionDenied` exception in `PermissionDenied` middleware, even while debugging
- Fixed page number detection
- Created 'simple document' for non technical users with all of a document pages content

- Use document preview code for staging file also
- Error picture literal name removal
- Spanish translation updates
- Show document file path in regards of its storage
- Added new setting: side bar search box
- Implemented new `PermissionDenied` exception middleware handler
- Permissions app api now returns a `PermissionDenied` exception instead of a custom one
- Added new 403 error template
- Updated the 404 template to display only a not found message
- Moved the login required middleware to the common app
- Fixed search app's `model.objects.filter` indentation, improved result count calculation
- Added dynamic comparison types to search app
- Separated search code from view code
- Correctly calculate show result count for multi model searches
- Fixed OCR queue list showing wrong thumbnail
- Fixed staging file preview
- Show current metadata in document upload view sidebar
- Show sentry login for admin users
- Do not reinitialize document queue and/or queued document on reentry
- Try extra hard not to assign same uuid to two documents
- Added new transformation preview size setting
- Renamed document queue state links
- Changed ocr status display sidebar from form based to text based
- Added document action to clear all the document's page transformations
- Allow search across related fields
- Optimized search for speed and memory footprint
- Added `LIMIT` setting to search
- Show search elapsed time on result page
- Converter now differentiates between unknown file format and convert errors
- Close file descriptors when executing external programs to prevent/reduce file descriptor leaks
- Improved exception handling of external programs
- Show document thumbnail in document ocr queue list
- Make ocr document date submitted column non breakable
- Fix permissions, directories set to mode 755 and files to mode 644
- Try to fix issue #2, "random ORM field error on search while doing OCR"
- Added configurable location setting for file based storage

- Prepend storage name to differentiate config options
- Fixed duplicated document search
- Optimized document duplicate search
- Added locale middleware, menu bar language switching works now
- Only show language selection list if localemiddleware is active
- Spanish translation updates
- Added links, views and permissions to disable or enable an OCR queue
- Enabled Django's template caching
- Added document queue property side bar window to the document queue list view
- Added HTML spaceless middleware to remove whitespace in HTML code
- If current user is superuser or staff show thumbnail & preview generation error messages
- Added a setting to show document thumbnail in metadata group list
- Started adding configurations setting descriptions
- Initial GridFS storage support
- Implemented size and delete methods for GridFS
- Implement GridFS storage user settings
- Added document link in the OCR document queue list
- Link to manually re queue failed OCR
- Don't separate links (enclose object list links with white-space: nowrap;)
- Added document description to the field search list
- Sort OCR queued documents according to submitted date & time
- Document filesystem serving is now a separate app
 - Steps to update (Some warnings may be returned, but these are not fatal as they might be related to missing metadata in some documents):
 - * rename the following settings:
 - DOCUMENTS_FILESYSTEM_FILESERVING_ENABLE to FILESYSTEM_FILESERVING_ENABLE
 - DOCUMENTS_FILESYSTEM_FILESERVING_PATH to FILESYSTEM_FILESERVING_PATH
 - DOCUMENTS_FILESYSTEM_SLUGIFY_PATHS to FILESYSTEM_SLUGIFY_PATHS
 - DOCUMENTS_FILESYSTEM_MAX_RENAME_COUNT to FILESYSTEM_MAX_RENAME_COUNT
 - * Do a ./manage.py syncdb
 - * Execute 'Recreate index links' locate in the tools menu
 - * Wait a few minutes
- Added per document duplicate search and a tools menu option to search all duplicated documents
- Added document tool that deletes and re-creates all documents filesystem links
- Increased document's and document metadata index filename field's size to 255 characters
- Added sentry to monitor and store error for later debugging

- Zip files can now be uncompressed in memory and their content uploaded individually in one step
- Added support for concurrent, queued OCR processing using celery
- Apply default transformations to document before OCR
- Added unpaper to the OCR conversion pipe
- Added views to create, edit and grant/revoke permissions to roles
- Added multipage documents support (only tested on pdfs)
 - To update a previous database do: [d.update_page_count() for d in Document.objects.all()]
- Added support for document page transformation (no GUI yet)
- Added permissions and roles support
- Added python-magic for smarter MIME type detection (<https://github.com/ahupp/python-magic>).
- Added a new Document model field: file_mime_encoding.
- Show only document metadata in document list view.
- If one document type exists, the create document wizard skips the first step.
- Changed to a liquid css grid
- Added the ability to group documents by their metadata
- New abstracted options to adjust document conversion quality (default, low, high)

Access control lists

Besides the permissions system explained in *Permissions*, Mayan EDMS provides per object permission granting. This feature is used to grant a permission to a role, but this permission can only be executed for a limited number of objects (documents, folders, tags) instead of being effective system-wide.

Example:

In this scenario only users in groups belonging to the `Accountants` role would be able to view the `2015 Payroll report.txt` document.

4.1 Inherited access control

It is also possible to grant a permission to a role for a specific document type (*Document types*). Under this scheme all users in groups belonging to that role will inherit that permission for all documents of that type.

Example:

The role `Accountants` is given the permission `document view` for the document type `Payroll reports`. Now all users in groups belonging to the `Accountants` role can view all documents of the type `Payroll reports` without needing to have that permissions granted for each particular `Payroll reports` type document.

If access control for the `Payroll reports` documents needs to be updated it only needs to be done for the document type and not for each document of the type `Payroll reports`.

Backups

To backup your install of Mayan EDMS just copy the actual document files and the database content. If you are using the default storage backend, the document files should be found in the `media` folder of your installation.

To dump the content of your database manager refer to the documentation chapter regarding database data “dumping”.

Here is an example of how to perform a backup and a restore of a PostgreSQL database.

To dump the database into an SQL text file:

```
pg_dump -h <host> -U <database user> -c <database name> -W > `date +%Y-%m-%d"%H-%M-%S`.sql
```

Example:

```
pg_dump -h 127.0.0.1 -U mayan -c mayan -W > `date +%Y-%m-%d"%H-%M-%S`.sql
```

To restore the database from the SQL text file:

```
psql -h <host> -U <database user> -d <database name> -W -f <sql dump file>
```

Example:

```
psql -h 127.0.0.1 -U mayan -d mayan -W -f 2018-06-07_18-10-56.sql
```

Here is an example of how to perform a backup and a restore of a PostgreSQL Docker container using a compressed dump file. A dump file is not compatible or can be used interchangeable with an SQL text file.

To backup a PostgreSQL Docker container:

```
docker exec <container name> pg_dump -U <database user> -Fc -c <database name> > `date +%Y-%m-%d"%H-%M-%S`.dump
```

Example:

```
docker exec mayan-edms-db pg_dump -U mayan -Fc -c mayan > `date +%Y-%m-%d"%H-%M-%S`.dump
```

This will produce a compressed dump file with the current date and time as the filename.

To restore a PostgreSQL Docker container:

```
docker exec -i <container name> pg_restore -U <database user> -d <database name> <
↪<dump file>
```

Since it is not possible to drop a currently open PostgreSQL database, this command must be used on a new and empty PostgreSQL container.

Example:

```
docker run -d \
--name mayan-edms-pg-new \
--restart=always \
-p 5432:5432 \
-e POSTGRES_USER=mayan \
-e POSTGRES_DB=mayan \
-e POSTGRES_PASSWORD=mayanuserpass \
-v /docker-volumes/mayan-edms/postgres-new:/var/lib/postgresql/data \
-d postgres:9.5

docker exec -i mayan-edms-pg-new pg_restore -U mayan -d mayan < 2018-06-07_17-09-34.
↪dump
```

More information at:

- PostgreSQL: <http://www.postgresql.org/docs/current/static/backup.html>
- MySQL: <https://dev.mysql.com/doc/refman/5.7/en/mysqldump.html>
- SQLite: Just copy the file `mayan/media/db.sqlite3`

CHAPTER 6

Checkouts

Checkouts are a way to block certain accesses or actions of a document for a period of time.

An user can choose to checkout a document to work on an update and block new versions of that document to be uploaded by other users. Document are checked out for a certain amount of time and if not manually checked in by the original user, will be checked in automatically by the system.

To be able to check in documents that were checked out by other users, the permission 'Forcefully check in documents' is required.

Document types

The basic unit of data in Mayan EDMS is the `document type`. A document type can be interpreted also as a document category, a document class, or a document template. Document types need to be created before documents can be uploaded. It is not possible to upload documents without assigning them a document type. Examples of document type: **invoices, blueprints, receipts**.

Examples:

Settings and attributes are applied to document types and documents will inherit those settings and attributes based on the document type they were assigned when uploaded into Mayan EDMS. A document can only be of one type at a given moment, but if needed, the type of a document can be changed. Upon changing its type, the document will lose its previous settings and attributes, and will inherit the settings and attributes of its new type.

The files are stored and placed under Mayan EDMS “control” to avoid filename clashes each file gets renamed to its UUID (Universally Unique ID), without extension, and stored in a simple flat arrangement in a directory.

This doesn’t stop access to the files but renaming, moving or updating directly them is not recommended because it would throw the database out of sync.

Because Mayan EDMS components are as decoupled from each other as possible, storage in this case is decoupled and its behavior is controlled not by the project but by the `Storage` module class. All the other modules don’t make any assumptions about how the actual document files are stored. This way files can be saved locally, over the network or even across the Internet and everything will still operate exactly the same.

The default file storage backend: `storage.backends.filebasedstorage.FileBasedStorage` is a simple backend that only supports paths and not IP addresses. In case you are interested in using remote volumes to store documents (NFS, SAMBA), first mount these volumes so that they appear as a directories to Mayan EDMS. For direct support for remote volumes a custom backend would be needed such as those provided by the Django Storages project (<https://django-storages.readthedocs.org/en/latest/>).

Indexes are an automatic method to hierarchically organize documents in relation to their properties (*Metadata*, label, MIME type, etc). To use indexes you need to first create an index template. Once created, associate the index to one or more *Document types*.

Index are hierarchical models so a tree template needs to be specified for them. This tree template will contain references to document metadata or properties that will be replaced with the actual value for those metadata or properties.

Example:

- Document type: `Product sheet`
- Metadata type: `Product year`, associated as a required metadata for the document type `Product sheet`.
- Index: `Product sheets per year`, and associated to the document type `Product sheet`.
- Index slug: `product-sheets-per-year`. Slugs are internal unique identifiers that can be used by other Mayan EDMS modules to reference each index.
- Index tree template as follows:

Now every time a new `Product sheet` is uploaded a hierarchical unit with the value of the metadata type `Product year` is created and a link to the uploaded `Product sheet` added to it.

Example:

Suppose three `Product sheets` are uploaded with the following values as their `Product year` metadata: 2001, 2002, 2001 respectively. The result index that will be generate based on the tree template would be as follows:

9.1 Mirroring

Indexes can be exported as **FUSE** filesystems. Using the management command `mountindex` we could export the previous example index as follows:

```
mkdir -p ~/indexes/products
mayan-edms.py mountindex product-sheets-per-year ~/indexes/products
```

The `~/indexes/products` directory will now have a directory and files structure identical to that of the index. Once indexes are mounted with this command, they behave like any other filesystem directory and can even be further shared via the network with network file system software like [Samba](#) or [NFS](#).

Indexes and mirrored indexes are Read Only as they are generated as a result of prior activities like document uploads, metadata changes.

CHAPTER 10

Languages

The list of languages choices in the language dropdown used for documents is based on the current ISO 639 list. This list can be quite extensive. To reduce the number of languages available use the settings DOCUMENTS_LANGUAGE_CODES, and set it to a nested list of abbreviations + languages names like:

```
DOCUMENTS_LANGUAGE_CODES = ('eng', 'spa')
```

The default language to appear on the dropdown can also be configured using:

```
DOCUMENTS_LANGUAGE = 'spa'
```

Use the correct ISO 639-3 language abbreviation (https://en.wikipedia.org/wiki/ISO_639) as this code is used in several subsystems in Mayan EDMS such as the OCR app to determine how to interpret the document.

Document signatures

Mayan EDMS supports two types of document signatures: embedded and detached signatures. When a document with an embedded signature is uploaded, this signature is readily detected as part of the document inspection step. The status of the signature can be verified by accessing the signatures sections of a document.

Signed documents are tamper-proof. That means that if any part of the document's file is edited (even just one character or one pixel), the document's signature becomes invalid alerting that an unauthorized change has occurred.

Existing non signed documents can be signed in one of two ways: by downloading the document, signing it, and uploading the signed document as a new version of the existing one or by creating a detached signature for the non signed document and uploading such detached signature file.

Maintenance of the public keyring can be done using the `Key` management functionality in the `Setup` menu.

From this menu, key servers can be queried and the results imported. Public keys no longer needed can also be deleted from this menu.

Only `GNU Privacy Guard` signatures are support at the moment.

Only version 1 of `GNU Privacy Guard` is supported at the moment.

CHAPTER 12

Mailing documents

To be able to send documents via email from inside Mayan EDMS you need to add and configure the following configuration variables in your `mayan/settings/local.py` file:

```
EMAIL_HOST = 'smtp.gmail.com' # Or similar
EMAIL_PORT = 587
EMAIL_HOST_USER = '<your smtp username>'
EMAIL_HOST_PASSWORD = '<your smtp password>'
EMAIL_USE_TLS = True
```

“Mail is sent using the SMTP host and port specified in the `EMAIL_HOST` and `EMAIL_PORT` settings. The `EMAIL_HOST_USER` and `EMAIL_HOST_PASSWORD` settings, if set, are used to authenticate to the SMTP server, and the `EMAIL_USE_TLS` and `EMAIL_USE_SSL` settings control whether a secure connection is used.”

For more details consult Django’s documentation on the topic: <https://docs.djangoproject.com/en/1.8/ref/settings/#email-host>

Metadata

Metadata is the name of the attribute of a document. The concept of metadata is divided in two: **metadata types** (size, color, distance) and **metadata values** for those types. Metadata types are defined in the setup menu and associated with document types. Then when a document is uploaded, a value for that metadata can be entered. There are two kinds of metadata type to document type relations: optional and required. When a metadata type is optional for a document type, it can be left blank for a document being uploaded and the upload will still be successful. On the other hand required metadata type must be given a value or it will not be possible to upload the document at hand.

Examples of metadata type: Invoice number, color, employee id.

The data entry of metadata types can be set to allow any value to be provided (the default) or a list of possible values can be entered in the `Lookup` configuration option and users will be presented with a drop down list of options instead of the default text entry box.

If metadata types are setup to allow any value to be entered a `validation` option can be chosen to block the entry of invalid data. Metadata types also provide `parsers` which will not block the entry of data but are able to interpret and modify the value provided by the user to conform to a specific format. An example of a provided parser is the date parser which will interpret and correct dates provided by users regardless of the format in which they are entered.

CHAPTER 14

OCR backend

Mayan EDMS ships an OCR backend that uses the FLOSS engine Tesseract (<https://github.com/tesseract-ocr/tesseract/>), but it can use other engines. To support other engines create a wrapper that subclasses the `OCRBackendBase` class defined in `mayan/apps/ocr/classes`. This subclass should expose the `execute` method. For an example of how the Tesseract backend is implemented take a look at the file `mayan/apps/ocr/backends/tesseract.py`

Once you create your own backend, in your `local.py` settings add the option `OCR_BACKEND` and point it to your new OCR backend class path.

The default value of `OCR_BACKEND` is `"ocr.backends.tesseract.Tesseract"`

To add support to OCR more languages when using Tesseract, install the corresponding language file. If using a Debian based OS, this command will display the available language files:

```
apt-cache search tesseract-ocr
```


CHAPTER 15

Permissions

Mayan EDMS provides very fine control over which actions users can perform. Action control works by allowing roles, that are composed of groups of users to be granted a permission such that the holder of that permission can exercise it throughout the entire system.

In other words, users themselves can't hold a permission, permissions are granted only to roles. Users can't directly belong to a role, they can only belong to a group. Groups can be members of roles. Roles are system permission units and groups are business organizational units.

Mayan EDMS can be configured via environment variables or by setting files.

16.1 Via environment variables

To use environment variables, look up the name of the setting you want to override in the “Settings” menu. The “Settings” menu is located inside the “Setup” main menu. To pass a value via an environment variable append "MAYAN_" to the name of the settings option. For example, to change the number of documents displayed per page (COMMON_PAGINATE_BY, by default 40), use:

```
export MAYAN_COMMON_PAGINATE_BY=10
```

Restart Mayan EDMS and the new value will take effect. The “Settings” menu can be used to verify if the overridden setting value is being interpreted correctly.

16.2 Via settings files

Another way to configure Mayan EDMS is via Python-style, settings files. If Mayan EDMS was installed using the Python package a `mayan_settings` folder will be created for this purpose. If you installed Mayan EDMS according to the *Advanced deployment* instructions provided in this documentation your `mayan_settings` folder should be located in the directory: `/usr/share/mayan-edms/mayan/media/mayan_settings`.

If Mayan EDMS was installed using Docker, the `mayan_settings` folder will be found inside the install Docker volume. If you installed Mayan EDMS according to the *Docker image* instructions provided in this documentation your `mayan_settings` folder should be located in the directory: `/docker-volumes/mayan/mayan_settings`.

Create a file with any valid name and a `.py` extension in the `mayan_settings` folder. The file must start with a global import of `mayan.settings.production`. In the form:

```
from mayan.settings.production import *
```

Now add the corresponding lines to override the default settings. In the settings file, it is not necessary to prepend the string `MAYAN_` to the setting option. For example, to change the number of documents displayed per page (`COMMON_PAGINATE_BY`, by default 40), use:

```
COMMON_PAGINATE_BY=10
```

versus:

```
export MAYAN_COMMON_PAGINATE_BY=10
```

when using the environment variable method.

For this example let's assume the file was saved with the name `mysettings.py`.

The way used to tell Mayan EDMS to import this file will vary based on the installation method.

For the *Advanced deployment* method, the full import path will be `mayan.media.mayan_settings.mysettings` and can be passed via the `--settings` command line argument like this:

```
python manage.py runserver --settings=mayan.media.mayan_settings.mysettings
```

or via the `DJANGO_SETTINGS_MODULE` environment variable like this:

```
export DJANGO_SETTINGS_MODULE=mayan.media.mayan_settings.mysettings
```

For the *Docker image* installation method, the full import path will be `mayan_settings.mysettings` and can only be passed via the `MAYAN_SETTINGS_MODULE` environment variable like this:

```
docker run <...> -e MAYAN_SETTINGS_MODULE=mayan_settings.mysettings
```

Document sources define places from which documents can be uploaded or gathered.

The current document sources supported are:

- Web - HTML forms with a `Browse` button that will open the file dialog when clicked to allow selection of files in the user's computer to be uploaded as documents.
- POP3 email - Provide the email, server and credential of a POP3 based email to be scanned periodically for email. The body of the email is uploaded as a document and the attachments of the email are uploaded as separate documents.
- IMAP email - Same as the POP3 email source but for email accounts using the IMAP protocol.
- Watch folder - A filesystem folder that is scanned periodically for files. Any file in the watch folder is automatically uploaded.
- Staging folder - Folder where networked attached scanned can save image files. The files in these staging folders are scanned and a preview is generated to help the process of upload. Staging folders and Watch folders work in a similar way with the main difference being that Staging folders are interactive while Watch folders are automatic; documents in a Watch folder are uploaded periodically and documents in a Staging folder remain indefinitely there until an user uploads them. A preview for files in a Staging folder is also provided. An example of Staging folder use is when multiple people are scanning documents but only one person must be allowed to upload those documents. This one person examines the scans quality and decides what to upload and what to reject and have re-scanned. Watch folders can be used when the quality of the scans is irrelevant or when they will be known to be of good quality, such as when receiving e-faxes as PDFs.

Document source can be configure to allow document bundles to uploaded as compressed files which are decompressed and their content uploaded as separate documents. This feature is useful when migrating from another document manager system.

Smart links

Smart links are a way to link documents without changing how they are organized in their respective indexes. Smart links are useful when two documents are related somehow but are of different type or different hierarchical units.

Example: A patient record can be related to a prescription drug information document, but they each belong to their own *Indexes*.

Smart links are rule based, but don't create any organizational structure. Smart links just show the documents that match the rules as evaluated against the metadata or properties of the currently displayed document.

Indexes are automatic hierarchical units used to group documents, smart links are automatic references between documents.

Example:

- Document type: `Patient records`
- Metadata type: `Prescription`, associated as an optional metadata for the document type `Patient records`.
- Document type: `Prescription information sheets`

A smart link with the following condition, will automatically links patient records to the prescription information sheets based on the value of the metadata type of the patient record.

```
foreign label is equal to {{ document.metadata_value_of.prescription }}
```


CHAPTER 19

Tags

Tags allow giving documents a binary property. Documents can also be tagged with more than one tag. Once tagged, documents can be searched also by their tags and from the tags main menu a list of all the documents with a particular tag can be obtained easily. Aside from their texts, tags can be assigned a particular color.

CHAPTER 20

Transformations

Transformations are persistent manipulations to the previews of the stored documents. For example: a scanning equipment may only produce landscape PDFs. In this case a useful transformation for that document source would be to rotate all scanned documents by 270 degrees after being uploaded. By adding this transformation to the Mayan EDMS source that is connected to the scanner, all pages scanned via that source will inherit the transformation as they are created. The result is that whenever a document is uploaded from that scanner, it will appear in portrait orientation, instead of landscape orientation.

Transformations can also be added to existing documents by clicking on a document's page and then clicking on "transformations". In this view the Actions menu will have a new option that reads "Create new transformation". Currently, the available transformations are: rotation, zoom, crop, and resize. Once the document image has been corrected, resubmit it for OCR for improved results.

Transformations are not destructive and do not physically modify the document file, they just modify the document's graphical representation.

CHAPTER 21

Upload wizard

The steps needed to upgrade a document using form-tools' `SessionWizard` were hardcoded in the source app. This made it very difficult to add or remove wizard steps.

The steps of the wizard are now defined by a new class called `sources.wizard.WizardStep`. The existing steps to select a document type, enter metadata and tag the document, have been converted to function as `WizardSteps` subclasses. The converted steps now live in

```
sources.wizards.WizardStepDocumentType, tag.wizard_steps.WizardStepTags, and
metadata.wizard_steps.WizardStepMetadata.
```

The steps need to define the following methods:

- `done`: This method is executed when the wizard finished the last step and enters the step where the actual file are uploaded. This step is used to encode form data into the URL query string that will be passed to the document upload view for each file uploaded.
- `condition`: This method is used to display the step conditionally. If this method returns `True` it will be displayed during the upload wizard execution. To skip the step, return `False` or `None`.
- `get_form_initial`: This method is used to return the initial data for the step form. Use this method to set up initial values for the step's form fields.
- `step_post_upload_process`: This method will be executed once the document finishes uploading. Use this method to process the information encoded in the URL query string by the step's `done` method.

Once the `WizardStep` subclass is defined, it needs to be registered. This is done by calling the `.register` method of the `WizardStep` class with the subclass as the argument. Example:

```
WizardStep.register(WizardStepMetadata)
```

This statement must be located after the subclass definition. Finally, the module defining the wizard step must be imported so that it is loaded with the rest of the code and enabled. The best place to do this is in the `.ready` method of the app's `apps.py` module. Example:

```
class TagsApp(MayanAppConfig):
    has_rest_api = True
```

(continues on next page)

(continued from previous page)

```
has_tests = True
name = 'tags'
verbose_name = _('Tags')

def ready(self):
    super(TagsApp, self).ready()
    from actstream import registry

    from .wizard_steps import WizardStepTags # NOQA
```

The WizardStep class also allows for unregistering existing steps. This is accomplished by calling the `.deregister` method of the WizardStep class and passing the subclass as the argument. This method should also be called inside the `.ready` method of an apps' `apps.py` module. Example:

```
class TagsApp(MayanAppConfig):
    has_rest_api = True
    has_tests = True
    name = 'tags'
    verbose_name = _('Tags')

    def ready(self):
        super(TagsApp, self).ready()
        from actstream import registry

        from metadata.wizard_steps import WizardStepMetadata # NOQA
        from sources.wizards import WizardStep # NOQA
        from .wizard_steps import WizardStepTags # NOQA

        WizardStep.deregister(WizardStepTags)
```

This will cause the tags assignment step to not be assigned to the upload wizard anymore.

Document versioning

Mayan EDMS has the ability to store different versions of the same document. A comment field is provided to allow users to summarize the new version changes in comparison with the previous one. If a new version was uploaded by mistake or such new version is no longer necessary the option to revert to a previous version of the document is provided.

Only the interactive document sources (*Sources*) (*Web and Staging folders*) are available to upload new document versions.

There is no limit to the number of versions a document can have.

By default, the last version will be showed when working with the document but any of the versions can be inspected and viewed.

23.1 How to use this image

23.1.1 Start a Mayan EDMS instance

With Docker properly installed, proceed to download the Mayan EDMS image using the command:

```
docker pull mayanedms/mayanedms:<version>
```

Then download version 9.5 of the Docker PostgreSQL image:

```
docker pull postgres:9.5
```

Create and run a PostgreSQL container:

```
docker run -d \  
--name mayan-edms-postgres \  
--restart=always \  
-p 5432:5432 \  
-e POSTGRES_USER=mayan \  
-e POSTGRES_DB=mayan \  
-e POSTGRES_PASSWORD=mayanuserpass \  
-v /docker-volumes/mayan-edms/postgres:/var/lib/postgresql/data \  
-d postgres:9.5
```

The PostgreSQL container will have one database named `mayan`, with an user named `mayan` too, with a password of `mayanuserpass`. The container will expose its internal 5432 port (PostgreSQL's default port) via the host's 5432 port. The data of this container will reside on the host's `/docker-volumes/mayan-edms/postgres` folder.

Finally create and run a Mayan EDMS container. Change `<version>` with the latest version in numeric form (example: 2.7.3) or use the `latest` identifier:

```
docker run -d \  
--name mayan-edms \  
--restart=always \  
-p 80:8000 \  
-e MAYAN_DATABASE_ENGINE=django.db.backends.postgresql \  
-e MAYAN_DATABASE_HOST=172.17.0.1 \  
-e MAYAN_DATABASE_NAME=mayan \  
-e MAYAN_DATABASE_PASSWORD=mayanuserpass \  
-e MAYAN_DATABASE_USER=mayan \  
-e MAYAN_DATABASE_CONN_MAX_AGE=60 \  
-v /docker-volumes/mayan-edms/media:/var/lib/mayan \  
mayanedms/mayanedms:<version>
```

The Mayan EDMS container will connect to the PostgreSQL container via the 172.17.0.1 IP address (the Docker host's default IP address). It will connect using the `django.db.backends.postgresql` database driver and connect to the `mayan` database using the `mayan` user with the password `mayanuserpass`. The container will keep connections to the database for up to 60 seconds in an attempt to reuse them increasing response time and reducing memory usage. The files of the container will be store in the host's `/docker-volumes/mayan-edms/media` folder. The container will expose its web service running on port 8000 on the host's port 80.

The container will be available by browsing to `http://localhost` or to the IP address of the computer running the container.

If another web server is running on port 80 use a different port in the `-p` option. For example: `-p 81:8000`.

23.1.2 Stopping and starting the container

To stop the container use:

```
docker stop mayan-edms
```

To start the container again:

```
docker start mayan-edms
```

23.1.3 Environment Variables

The Mayan EDMS image can be configure via environment variables.

`MAYAN_DATABASE_ENGINE`

Defaults to `None`. This environment variable configures the database backend to use. If left unset, SQLite will be used. The database backends supported by this Docker image are:

- `'django.db.backends.postgresql'`
- `'django.db.backends.mysql'`
- `'django.db.backends.sqlite3'`

When using the SQLite backend, the database file will be saved in the Docker volume. The SQLite database as used by Mayan EDMS is meant only for development or testing, never use it in production.

`MAYAN_DATABASE_NAME`

Defaults to `'mayan'`. This optional environment variable can be used to define the database name that Mayan EDMS will connect to. For more information read the pertinent Django documentation page: [Connecting to the database](#)

`MAYAN_DATABASE_USER`

Defaults to 'mayan'. This optional environment variable is used to set the username that will be used to connect to the database. For more information read the pertinent Django documentation page: [Settings, USER](#)

`MAYAN_DATABASE_PASSWORD`

Defaults to ''. This optional environment variable is used to set the password that will be used to connect to the database. For more information read the pertinent Django documentation page: [Settings, PASSWORD](#)

`MAYAN_DATABASE_HOST`

Defaults to *None*. This optional environment variable is used to set the hostname that will be used to connect to the database. This can be the hostname of another container or an IP address. For more information read the pertinent Django documentation page: [Settings, HOST](#)

`MAYAN_DATABASE_PORT`

Defaults to *None*. This optional environment variable is used to set the port number to use when connecting to the database. An empty string means the default port. Not used with SQLite. For more information read the pertinent Django documentation page: [Settings, PORT](#)

`MAYAN_BROKER_URL`

This optional environment variable determines the broker that Celery will use to relay task messages between the frontend code and the background workers. For more information read the pertinent Celery Kombu documentation page: [Broker URL](#)

This Docker image supports using Redis and RabbitMQ as brokers.

Caveat: If the `MAYAN_BROKER_URL` and `MAYAN_CELERY_RESULT_BACKEND` environment variables are specified, the built-in Redis server inside the container will be disabled.

`MAYAN_CELERY_RESULT_BACKEND`

This optional environment variable determines the results backend that Celery will use to relay result messages from the background workers to the frontend code. For more information read the pertinent Celery Kombu documentation page: [Task result backend settings](#)

This Docker image supports using Redis and RabbitMQ as result backends.

Caveat: If the `MAYAN_BROKER_URL` and `MAYAN_CELERY_RESULT_BACKEND` environment variables are specified, the built-in Redis server inside the container will be disabled.

`MAYAN_SETTINGS_MODULE`

Optional. Allows loading an alternate settings file.

`MAYAN_DATABASE_CONN_MAX_AGE`

Amount in seconds to keep a database connection alive. Allow reuse of database connections. For more information read the pertinent Django documentation page: [Settings, CONN_MAX_AGE](#)

`MAYAN_SETTINGS_FILE`

Optional. Previously only the `local.py` file was the only settings file available to allow users to make configuration changes to their installations. Now with this environment variable, users are free to create multiple settings files and tell the Mayan EDMS container which setting file to import. The only requirement is that the setting file starts with a global import of `mayan.settings.production`. In the form:

```
from mayan.settings.production import *
```

`MAYAN_GUNICORN_WORKERS`

Optional. This environment variable controls the number of frontend workers that will be executed. If not specified the default is 2. For heavier loads, use a higher number. A formula recommended for this setting is the number of CPU cores + 1.

23.2 Accessing outside data

To use Mayan EDMS's staging folders or watch folders from Docker, the data for these source must be made accessible to the container. This is done by mounting the folders in the host computer to folders inside the container. This is necessary because Docker containers do not have access to host data on purpose. For example, to make a folder in the host accessible as a watch folder, add the following to the Docker command line when starting the container:

```
-v /opt/scanned_files:/srv/watch_folder
```

The command line would look like this:

```
docker run ... -v /opt/scanned_files:/srv/watch_folder mayanedms/mayanedms:latest
```

Now create a watch folder in Mayan EDMS using the path `/srv/watch_folder` and the documents from the host folder `/opt/scanned_files` will be automatically available. Use the same procedure to mount host folders to be used as staging folders. In this example `/srv/watch_folder` was as the container directory, but any path can be used as long as it is not an already existing path or a path used by any other program.

23.3 Performing backups

To backup the existing data, stop the image and copy the content of the volume. For the example:

```
docker run -d --name mayan-edms --restart=always -p 80:8000 \  
-v /docker-volumes/mayan:/var/lib/mayan \  
-v /opt/scanned_files:/srv/watch_folder mayanedms/mayanedms:latest
```

That would be the `/docker-volumes/mayan` folder:

```
sudo tar -zcvf backup.tar.gz /docker-volumes/mayan  
sudo chown `whoami` backup.tar.gz
```

If using an external PostgreSQL or MySQL database or database containers, these too need to be backed up using their respective procedures. A simple solution is to copy the entire database container volume after the container has been stopped.

23.4 Restoring from a backup

Uncompress the backup archive in the original docker volume using:

```
sudo tar -xvzf backup.tar.gz -C /
```

23.5 Upgrading

Upgrading a Mayan EDMS Docker container is actually a matter of stopping and deleting the container, downloading the most recent version of the image and starting a container again. The container will take care of updating the

database structure to the newest version if necessary.

IMPORTANT! Do not delete the volume storing the data, only the container.

Stop the container to be upgraded:

```
docker stop mayan-edms
```

Remove the container:

```
docker rm mayan-edms
```

Pull the new image version:

```
docker pull mayanedms/mayanedms:latest
```

Start the container again with the new image version:

```
docker run -d --name mayan-edms --restart=always -p 80:8000 -v /docker-volumes/mayan:/  
→var/lib/mayan mayanedms/mayanedms:latest
```

23.6 Building the image

Clone the repository with:

```
git clone https://gitlab.com/mayan-edms/mayan-edms.git
```

Change to the directory of the cloned repository:

```
cd mayan-edms
```

Execute Docker's build command using the provided makefile:

```
make docker-build
```

Or using an apt cacher to speed up the build:

```
make docker-build-with-proxy APT_PROXY=172.17.0.1:3142
```

Replace the IP address *172.17.0.1* with the IP address of the computer running the APT proxy and caching service.

23.7 Customizing the image

23.7.1 Simple method

If you just need to add a few Ubuntu or Python packages to your installation, you can use the following environment variables:

MAYAN_APT_INSTALLS

Specifies a list of Ubuntu *.deb* packages to be installed via APT when the container is first created. The installed packages are not lost when the image is stopped. Example: To install the Tesseract OCR language packs for German and Spanish add the following in your `docker start` command line:

```
-e MAYAN_APT_INSTALLS="tesseract-ocr-deu tesseract-ocr-spa"
```

MAYAN_PIP_INSTALLS

Specifies a list of Python packages to be installed via `pip`. Packages will be downloaded from the Python Package Index (<https://pypi.python.org>) by default.

23.8 Using Docker compose

To deploy a complete production stack using the included Docker compose file execute:

```
docker-compose -f docker-compose.yml up -d
```

This Docker compose file will provision four containers:

- Postgres as the database
- Redis as the Celery result storage
- RabbitMQ as the Celery broker
- Mayan EDMS using the above service containers

To stop the stack use:

```
docker-compose -f docker-compose.yml stop
```

The stack will also create four volumes to store the data of each container. These are:

- `mayan_app` - The Mayan EDMS data container, normally called `mayan_data` when not using Docker compose.
- `mayan_broker` - The broker volume, in this case RabbitMQ.
- `mayan_db` - The database volume, in this case Postgres.
- `mayan_results` - The celery result backend volume, in this case Redis.

23.9 Nightly images

The continuous integration pipeline used for testing development builds also produces a resulting Docker image. These are build automatically and their stability is not guaranteed. They should never be used in production. If you want to try out the Docker images the development uses or want a sneak peek at the new features being worked on checkout the container registry at: https://gitlab.com/mayan-edms/mayan-edms/container_registry

Advanced deployment

Mayan EDMS should be deployed like any other [Django](#) project and preferably using [virtualenv](#). Below are some ways to deploy and use Mayan EDMS. Do not use more than one method.

Being a [Django](#) and a [Python](#) project, familiarity with these technologies is recommended to better understand why Mayan EDMS does some of the things it does.

24.1 Binary dependencies

24.1.1 Ubuntu

If using a [Debian](#) or [Ubuntu](#) based Linux distribution, get the executable requirements using:

```
sudo apt-get install g++ gcc ghostscript gnupg1 graphviz libjpeg-dev libmagic1 \  
libpq-dev libpng-dev libreoffice libtiff-dev poppler-utils postgresql \  
python-dev python-pip python-virtualenv redis-server sane-utils supervisor \  
tesseract-ocr zlib1g-dev -y
```

Create an user account for the installation:

```
sudo adduser mayan --disabled-password --disabled-login --no-create-home --gecos ""
```

Create the parent directory where the project will be deployed:

```
sudo mkdir /opt
```

Create the Python virtual environment for the installation:

```
sudo virtualenv /opt/mayan-edms
```

Make the mayan user the owner of the installation directory:

```
sudo chown mayan:mayan /opt/mayan-edms -R
```

Install Mayan EDMS from PyPI:

```
sudo -u mayan /opt/mayan-edms/bin/pip install --no-cache-dir mayan-edms
```

Install the Python client for PostgreSQL and Redis:

```
sudo -u mayan /opt/mayan-edms/bin/pip install --no-cache-dir psycopg2==2.7.3.2_
↳redis==2.10.6
```

Create the database for the installation:

```
sudo -u postgres psql -c "CREATE USER mayan WITH password 'mayanuserpass';"
sudo -u postgres createdb -O mayan mayan
```

Initialize the project:

```
sudo -u mayan MAYAN_DATABASE_ENGINE=django.db.backends.postgresql MAYAN_DATABASE_
↳NAME=mayan \
MAYAN_DATABASE_PASSWORD=mayanuserpass MAYAN_DATABASE_USER=mayan \
MAYAN_DATABASE_HOST=127.0.0.1 MAYAN_MEDIA_ROOT=/opt/mayan-edms/media \
/opt/mayan-edms/bin/mayan-edms.py initialsetup
```

Collect the static files:

```
sudo -u mayan MAYAN_MEDIA_ROOT=/opt/mayan-edms/media \
/opt/mayan-edms/bin/mayan-edms.py collectstatic --noinput
```

Create the supervisor file at /etc/supervisor/conf.d/mayan.conf:

```
[supervisord]
environment=
    MAYAN_ALLOWED_HOSTS="*", # Allow access to other network hosts other than_
↳localhost
    MAYAN_CELERY_RESULT_BACKEND="redis://127.0.0.1:6379/0",
    MAYAN_BROKER_URL="redis://127.0.0.1:6379/0",
    PYTHONPATH=/opt/mayan-edms/lib/python2.7/site-packages:/opt/mayan-edms/data,
    MAYAN_MEDIA_ROOT=/opt/mayan-edms/media,
    MAYAN_DATABASE_ENGINE=django.db.backends.postgresql,
    MAYAN_DATABASE_HOST=127.0.0.1,
    MAYAN_DATABASE_NAME=mayan,
    MAYAN_DATABASE_PASSWORD=mayanuserpass,
    MAYAN_DATABASE_USER=mayan,
    MAYAN_DATABASE_CONN_MAX_AGE=60,
    DJANGO_SETTINGS_MODULE=mayan.settings.production

[program:mayan-gunicorn]
autorestart = true
autostart = true
command = /opt/mayan-edms/bin/gunicorn -w 2 mayan.wsgi --max-requests 500 --max-
↳requests-jitter 50 --worker-class gevent --bind 0.0.0.0:8000
user = mayan

[program:mayan-worker-fast]
autorestart = true
autostart = true
```

(continues on next page)

(continued from previous page)

```

command = nice -n 1 /opt/mayan-edms/bin/mayan-edms.py celery worker -Ofair -l ERROR -
↳Q converter -n mayan-worker-fast.%%h --concurrency=1
killasgroup = true
numprocs = 1
priority = 998
startsecs = 10
stopwaitsecs = 1
user = mayan

[program:mayan-worker-medium]
autorestart = true
autostart = true
command = nice -n 18 /opt/mayan-edms/bin/mayan-edms.py celery worker -Ofair -l ERROR -
↳Q checkouts_periodic,documents_periodic,indexing,metadata,sources,sources_periodic,
↳uploads,documents -n mayan-worker-medium.%%h --concurrency=1
killasgroup = true
numprocs = 1
priority = 998
startsecs = 10
stopwaitsecs = 1
user = mayan

[program:mayan-worker-slow]
autorestart = true
autostart = true
command = nice -n 19 /opt/mayan-edms/bin/mayan-edms.py celery worker -Ofair -l ERROR -
↳Q mailing,tools,statistics,parsing,ocr -n mayan-worker-slow.%%h --concurrency=1
killasgroup = true
numprocs = 1
priority = 998
startsecs = 10
stopwaitsecs = 1
user = mayan

[program:mayan-celery-beat]
autorestart = true
autostart = true
command = nice -n 1 /opt/mayan-edms/bin/mayan-edms.py celery beat --pidfile= -l ERROR
killasgroup = true
numprocs = 1
priority = 998
startsecs = 10
stopwaitsecs = 1
user = mayan

```

Enable and restart the services [1]:

```

systemctl enable supervisor
systemctl restart supervisor

```

[1]: <https://bugs.launchpad.net/ubuntu/+source/supervisor/+bug/1594740>

Mayan EDMS is under active development, and contributions are welcome.

If you have a feature request, suggestion or bug report, please open a new issue on the [GitLab issue tracker](#). To submit patches, please send a merge request on [GitLab](#).

25.1 Project philosophies

How to think about Mayan EDMS when doing changes or adding new features; why things are the way they are in Mayan EDMS:

- Functionality must be as market/sector independent as possible, code for the 95% of use cases.
- Each user must be able to configure and customize it to their needs after install.
- Abstract as much as possible, each app must be an expert in just one thing, for other things they should use the API/classes/functions of other apps.
- Assume as little as possible about anything outside the project (hardware, OS, storage).
- Provide Python based abstraction so that a default install runs with a single step.
- No hard dependencies on binaries unless there is no other choice.
- Provide “drivers” or switchable backends to allow users to fine tune the installation.
- Call to binaries only when there is no other choice or the Python choices are not viable/mature/efficient.
- Each app is as independent and self contained as possible. Exceptions, the basic requirements: navigation, permissions, common, main.
- If an app is meant to be used by more than one other app, it should be as generic as possible in regard to the project and another app will bridge the functionality.
 - Example: since indexing (`document_indexing`) only applies to documents, the app is specialized and depends on the documents app.

25.2 Coding conventions

25.2.1 Follow PEP8

Whenever possible, but don't obsess over things like line length:

```
$ flake8 --ignore=E501,E128,E122 |less
```

To perform automatic PEP8 checks, install flake8's git hook using:

```
$ flake8 --install-hook git
```

25.2.2 Imports

Import order should be:

- Standard Python modules
- Installed Python modules
- Core Django modules
- Installed Django modules
- Mayan EDMS modules
- Local imports

Example:

```
from __future__ import absolute_import

# Standard Python library
import base64

# 3rd party installed Python libraries
import requests

# Django core modules
from django.db.models import Q
from django.template.defaultfilters import slugify
from django.utils.translation import ugettext
from django.utils.translation import ugettext_lazy as _

# 3rd party installed Django libraries
from rest_framework import APIView

# Mayan apps
from metadata.classes import MetadataClass

# Local app imports (relative)
from .conf.settings import (
    AVAILABLE_INDEXING_FUNCTIONS,
    MAX_SUFFIX_COUNT, SLUGIFY_PATHS
)
from .exceptions import MaxSuffixCountReached
from .filesystem import (
```

(continues on next page)

(continued from previous page)

```

    fs_create_index_directory, fs_create_document_link,
    fs_delete_document_link, fs_delete_index_directory,
    assemble_suffixed_filename
)
from .models import Index, IndexInstanceNode, DocumentRenameCount

```

All local app module imports are in relative form. Local app module name is to be referenced as little as possible, unless required by a specific feature, trick, restriction (e.g., Runtime modification of the module's attributes).

Incorrect:

```

# documents app views.py model
from documents.models import Document

```

Correct:

```

# documents app views.py model
from .models import Document

```

25.2.3 Dependencies

Mayan EDMS apps follow a hierarchical model of dependency. Apps import from their parents or siblings, never from their children. Think plugins. A parent app must never assume anything about a possible existing child app. The documents app and the Document model are the basic entities; they must never import anything else. The common and main apps are the base apps.

25.2.4 Variables

Naming of variables should follow a Major to Minor convention, usually including the purpose of the variable as the first piece of the name, using underscores as spaces. camelCase is not used in Mayan EDMS.

Examples:

Links:

```

link_document_page_transformation_list = ...
link_document_page_transformation_create = ...
link_document_page_transformation_edit = ...
link_document_page_transformation_delete = ...

```

Constants:

```

PERMISSION_SMART_LINK_VIEW = ...
PERMISSION_SMART_LINK_CREATE = ...
PERMISSION_SMART_LINK_DELETE = ...
PERMISSION_SMART_LINK_EDIT = ...

```

Classes:

```

class Document(models.Model):
class DocumentPage(models.Model):
class DocumentPageTransformation(models.Model):
class DocumentType(models.Model):
class DocumentTypeFilename(models.Model):

```

25.2.5 Strings

Quotation character used in Mayan EDMS for strings is the single quote. Double quote is used for multiple line comments or HTML markup.

25.2.6 Migrations

Migrations should do only one thing (eg: either create a table, move data to a new table or remove an old table) to aid retrying on failure.

25.2.7 General

Code should appear in their modules in alphabetic order or in their order of importance if it makes more sense for the specific application. This makes visual scanning easier on modules with a large number of imports, views or classes. Class methods that return a value should be pretended with a `get_` to differentiate from an object's properties. When a variable refers to a file it should be named as follows:

- `filename`: The file's name and extension only.
- `filepath`: The entire path to the file including the filename.
- `path`: A path to a directory.

Flash messages should end with a period as applicable for the language. Only exception is when the tail of the message contains an exceptions message as passed directly from the exception object.

25.3 Source Control

Mayan EDMS source is controlled with `Git`.

The project is publicly accessible, hosted and can be cloned from **GitLab** using:

```
$ git clone https://gitlab.com/mayan-edms/mayan-edms.git
```

25.4 Git branch structure

Mayan EDMS follows a simplified model layout based on Vincent Driessen's [Successful Git Branching Model](#) blog post.

develop The "next release" branch, likely unstable.

master Current production release (3.0.1).

feature/ Unfinished/unmerged feature.

series/ Released versions.

Each release is tagged separately.

When submitting patches, please place your code in its own `feature/` branch prior to opening a Merge Request on [GitLab](#).

25.5 Steps to deploy a development version

```
$ git clone https://gitlab.com/mayan-edms/mayan-edms.git
$ cd mayan-edms
$ git checkout development
$ virtualenv venv
$ source venv/bin/activate
$ pip install -r requirements.txt
$ ./manage.py initialsetup
$ ./manage.py runserver
```

25.6 Contributing changes

Follow the latest contributing guidelines outlined here: <https://gitlab.com/mayan-edms/mayan-edms/blob/master/CONTRIBUTING.md>

25.7 Debugging

Mayan EDMS makes extensive use of Django's new [logging capabilities](#). By default debug logging for all apps is turned on. If you wish to customize how logging is managed turn off automatic logging by setting `COMMON_AUTO_LOGGING` to `False` and add the following lines to your `settings/local.py` file:

```
LOGGING = {
    'version': 1,
    'disable_existing_loggers': True,
    'formatters': {
        'verbose': {
            'format': '%(levelname)s %(asctime)s %(name)s %(process)d %(thread)d
↪ %(message)s'
        },
        'intermediate': {
            'format': '%(name)s <%(process)d> [%(levelname)s] "%(funcName)s()"
↪ %(message)s"'
        },
        'simple': {
            'format': '%(levelname)s %(message)s'
        },
    },
    'handlers': {
        'console': {
            'level': 'DEBUG',
            'class': 'logging.StreamHandler',
            'formatter': 'intermediate'
        },
    },
    'loggers': {
        'documents': {
            'handlers': ['console'],
            'propagate': True,
            'level': 'DEBUG',
        },
        'common': {
```

(continues on next page)

(continued from previous page)

```
        'handlers':['console'],
        'propagate': True,
        'level':'DEBUG',
    },
}
```

Likewise, to see the debug output of the `tags` app, just add the following inside the `loggers` block:

```
'tags': {
    'handlers':['console'],
    'propagate': True,
    'level':'DEBUG',
},
```

25.8 Documentation

The documentation is written in `reStructured Text` format, processed with `Sphinx`, and resides in the `docs` directory. In order to build it, you will first need to install the documentation editing dependencies with:

```
$ pip install -r requirements/documentation.txt
```

Then, to build an HTML version of the documentation, run the following command from the `docs` directory:

```
$ make docs-serve
```

The generated documentation can be viewed by browsing to <http://127.0.0.1:8000> or by browsing to the `docs/_build/html` directory.

You can also generate the documentation in formats other than HTML. Consult the `Sphinx` documentation for more details.

25.9 Installable package

25.9.1 Source file package

This is the sequence of step used to produce an installable package:

1. Generate the packaged version (will produce `dist/mayan-edms-x.y.z.tar.gz`):

```
$ make sdist
```

2. Do a test install:

```
$ cd /tmp
$ virtualenv venv
$ source venv/bin/activate
$ pip install <path of the Git repository>/dist/mayan-edms-x.y.z.tar.gz
$ mayan-edms.py initialsetup
$ mayan-edms.py runserver
```

25.9.2 Wheel package

1. Install the development requirements:

```
$ pip install -r requirements/development.txt
```

2. Create wheel package using the makefile:

```
$ make wheel
```

3. Do a test install:

```
$ cd /tmp
$ virtualenv venv
$ source venv/bin/activate
$ pip install <path of the Git repository>/dist/mayan_edms-x.y.z-py2-none-any.whl
$ mayan-edms.py initialsetup
$ mayan-edms.py runserver
```

25.9.3 Version numbering

Mayan EDMS uses the Semantic Versioning (<http://semver.org/>) method to choose version numbers along with Python's PEP-0440 (<https://www.python.org/dev/peps/pep-0440/>) to format them.

X.YaN # Alpha release X.YbN # Beta release X.YrcN # Release Candidate X.Y # Final release

25.9.4 Release checklist

1. Check for missing migrations:

```
make check-missing-migrations
```

2. Synchronize translations:

```
make translations-pull
```

3. Compile translations:

```
make translations-compile
```

4. Write release notes.
5. Update changelog.
6. Scan the code with flake8 for simple style warnings.
7. Check README.rst format with:

```
python setup.py check -r -s
```

or with:

```
make check-readme
```

8. Bump version in *mayan/__init__.py* and in *docker/version*.
9. Update requirements version in *setup.py* using:

```
make generate-setup
```

10. Build source package and test:

```
make test-sdist-via-docker-ubuntu
```

11. Build wheel package and test:

```
make test-wheel-via-docker-ubuntu
```

12. Tag version:

```
git tag -a vX.Y.Z -m "Version X.Y.Z"
```

13. Push tag upstream:

```
git push --tags
```

14. Build and upload a test release:

```
make release-test-via-docker-ubuntu
```

15. Build and upload a final release:

```
make release-via-docker-ubuntu
```

Mayan EDMS apps are essentially Django app with some extra code to register navigation, permissions and other relationships.

26.1 App modules

- `__init__.py`
Should be empty if possible. No initialization code should be here, use the `ready()` method of the `MayanAppConfig` class in the `apps.py` module.
- `admin.py`
Standard Django app module to define how models are to be presented in the admin interface.
- `api_views.py`
REST API views go here. Mayan EDMS uses Django REST Framework API view classes.
- `apps.py`
Contains the `MayanAppConfig` subclass as required by Django 1.7 and up. This is a place to define the app name and translatable verbose name as well as code to be execute when the modules of the app are ready.
- `classes.py`
Hold python classes to be used internally or externally. Any class defined by the app that is not a model.
- `events.py`
Define event class instances that are later committed to a log by custom code.
- `exceptions.py`
Custom exceptions defined by the app.
- `fields.py`
Place any custom form field classed you define here.

- `forms.py`

Standard Django app module that hold custom form classes.
- `handlers.py`

Contains the signal handlers, functions that will process a given signal emitted from this or other apps. Connect the handler functions to the corresponding signal in the `ready()` method of the `MayanAppConfig` subclass in `apps.py`
- `links.py`

Defines the links to be used by the app. Import only from the navigation app and the local `permissions.py` file.
- `literals.py`

Stores magic numbers, module choices (if static), settings defaults, and constants. Should contain all capital case variables. Must not import from any other module.
- `managers.py`

Standard Django app module that hold custom model managers. These act as model class method to performs actions in a series of model instances or utilitarian actions on external models instances.
- `models.py`

Standard Django app module that defines ORM persistent data schema.
- `permissions.py`

Defines the permissions to be used to validate user access by links and views. Imports only from the permissions app. Link or view conditions such as testing for `is_staff` or `is_superuser` flag are defined in this same module.
- `runtime.py`

Use this module when you need the same instance of a class for the entire app. This module acts as a shared memory space for the other modules of the app or other apps.
- `serializers.py`

Hold Django REST Framework serializers used by the `api_views.py` module.
- `settings.py`

Define the configuration settings instances that the app will use.
- `signals.py`

Any custom defined signal goes here.
- `statistics.py`

Provides functions that will compute any sort of statistical information on the app's data.
- `tasks.py`

Code to be execute in the background or as an out-of-process action.
- `tests/` directory

Hold test modules. There should be one `test_*.py` module for each aspect being tested, examples: `test_api.py`, `test_views.py`, `test_parsers.py`, `test_permissions.py` Any shared constant data used by the tests should be added to `tests/literals.py`
- `utils.py`

Holds utilitarian code that doesn't fit on any other app module or that is used by several modules in the app. Anything used internally by the app that is not a class or a literal (should be as little as possible)

- widgets.py

HTML widgets go here. This should be the only place with presentation directives in the app (aside the templates).

26.2 Views

The module `common.generics` provides custom generic class based views to be used. The basic views used to create, edit, view and delete objects in Mayan EDMS are: `SingleObjectCreateView`, `SingleObjectDetailView`, `SingleObjectEditView`, and `SingleObjectListView`

These views handle aspects relating to view permissions, object permissions, post action redirection and template context generation.

27.1 Release blockers

These are errors or issues that are blocking a release.

- None

27.2 Pending work

These are tasks that need to be completed but are missing a dependency or a design decision. As more information is added to each, they should be converted into a MERC.

27.2.1 API

- User API edit view: Should not be able to add or remove groups without corresponding group access.
- User group list API get & post views: Should adding a group to an user via the API return 201 or 200. Currently returns 201.
- Consistent API return code for delete views without access. Some views return 403 other return 404.
- Update API docstrings. The upgrade to the latest DRF broke all formatting.
- Make views smaller. Much as much as possible to serializers.
- Switch to ViewSets.
- Add API filtering. Example unread notifications.

27.2.2 Documents

- Navigating to the interactive document page image is not triggering the document view event.

27.2.3 Events

- New event: document emailed.

27.2.4 Forms processing

- Remove usage of `self.cleaned_data`. Use `self.clean_data` instead.

27.2.5 Permissions

- Permission should be reciprocal. Example: To be able to add a tag to a document, the user must hold the tag add permission for the document and for the tag to be added. To be able to enable a metadata type to a document type, the user must hold the metadata add permissions for the metadata type and for the document type.
- Edit type permissions should only grant the ability to edit the properties of an object. To modify its relationship with other objects a reciprocal permission check should be instead.

27.2.6 Search

- Rename `SearchModel.pk` to `id`

27.2.7 Sources

- Add ACLs support to sources.
- Provide error message/feedback when scanning from a remote scanner fails.
- Require a permission for document types to avoid a user that has the workflow creation permission to attach a workflow to a document type they don't control.
- Research making `APIWorkflowDocumentTypeList` a subclass of `documents.api_views.APIDocumentTypeList`
- A POST request to `APIWorkflowDocumentTypeList` should require some permission on the document type part to avoid adding non controlled document types to a new workflow.
- To transition a workflow, the transition permission is only needed for the workflow. Make it necessary to have the same permission for the document of document type.
- To view the transition log, the workflow view permission is only needed for the document. Make it necessary to have the same permission for the workflow or for the transition and the states.
- Render date time of scanned documents using SANE to a better output (like document versions).

27.2.8 Testing

- Add document test mixin that creates documents types and documents (to be used in `dynamic_search.test_api`).
- Update all API tests using `self.client` to just `self`. and the HTTP method.
- Add test for searches for each app that uses search.

27.2.9 UI - Frontend

- Fix menu not collapsing at the same width of nav parent.
- Move direct CSS style from code into base.css. grep 'style' * -R. Style code in:
 - appearance/templates/appearance/generic_list_items_subtemplate.html
 - appearance/templates/appearance/base.html
 - appearance/templates/appearance/generic_list_subtemplate.html
 - appearance/templates/navigation/generic_link_instance.html
- Check if location is found in partial and remove it. Avoid circular loading.
- Add location to history after a form submit redirect.

27.2.10 UI

- Shift click select to select multiple documents.
- During the document upload wizard and the option to double click to select document type and submit the form. The purpose is to speed up the step with less mouse travel since this is a common screen.
- Add metadata to the Menu class to allow UI code to decide where and how to display each menu.
- Reduce the facet buttons paddings.
- Make facet action button smaller. Reduce margin-bottom.
- Remove previous from the views.py. It is not longer used by the form's cancel button.

27.2.11 Workflows

- Workflow trigger filters. Example: `{{ document.document_type.name = 'invoice' }}` or same UI as the smart links app. Will allow restricting the firing of workflow actions by a user defined filter criteria.
- Require a permission for document types to avoid a user that has the workflow creation permission to attach a workflow to a document type they don't control.
- Research making `APIWorkflowDocumentTypeList` a subclass of `documents.api_views.APIDocumentTypeList`
- A POST request to `APIWorkflowDocumentTypeList` should require some permission on the document type part to avoid adding non controlled document types to a new workflow.
- To transition a workflow, the transition permission is only needed for the workflow. Make it necessary to have the same permission for the document of document type.
- To view the transition log, the workflow view permission is only needed for the document. Make it necessary to have the same permission for the workflow or for the transition and the states.

27.3 New features

27.3.1 API

- Add converter API
- Document signatures API

- Smart settings API
- Use REST_API's 'mayan_permission_attribute_check' to simplify API views access checking and filtering.
- DRF filtering using the URL query.

27.3.2 Converter

- New zoom transformation. Resample, not just bigger final size but do a `resize * zoom multiple` before. Produces a bigger image or higher quality than the original.

27.3.3 Caching

- Size limited caching. A new model in the common app will keep track of all cache files. A manager method will be provided that will return the cache files in other of age to be deleted.

27.3.4 Distribution

- Python based Javascript package manager. Each app specifies what library and version needs. The common app (or a new app) will add all the JS loading lines automatically so that compress can detect them.

27.3.5 Metadata

- Metadata lookup memory. Add a select2 style widget that will query a new metadata API endpoint that will return all used values so far.
- Metadata validation_choices and parser_choices as static model methods
- Metadata.api as Metadata.utils and manager

27.3.6 Search

- Add support for highlighting the search results in pages.

27.3.7 Settings

- Database based settings.

27.3.8 Sources

- UI improvement for staging folders files selection. GitLab issue.

27.3.9 UI

- Upgrade to Bootstrap 4.
- Upgrade to Flatly 4.
- Better workflow transition UI. Instead of a dropdown show all the available transitions as buttons.

27.3.10 Workflows

- Workflow trigger filters. Example: `{{ document.document_type.name = 'invoice' }}` or same UI as the smart links app. Will allow restricting the firing of workflow actions by an user defined filter criteria.
- New workflow action: send email. Subject and content are templates.

27.3.11 Other

- Python based Javascript package manager. Each app specifies what library and version needs. The common app (or a new app) will add all the JS loading lines automatically so that compress can detect them.
- When moving documents to the trash update the message to “submitted” and not “moved” or “deleted” since this is handled by a task queue and is not immediate and doesn’t delete the document.
- When emptying the trash update the message to “submitted” since this is handled by a task queue and is not immediate.
- New app that allows creating user document filters. Will provide the same service as the document filters class. Interface can be made using the template language or the same UI as the smart links.
- Allow add queue metadata that can be exported via a management command. This will allow creating supervisor templates without all the worker entries being hardcoded.
- Delete `.gitignore` files from copied packages. Include `.gitignore` files keep compiled or distributable files from being included in the main repository. Temporary measure until a Javascript library manager is added.
- Automatically capture license information from installed Javascript libraries.
- Automatically capture license information from installed Python packages.
- Update celery to 4.1.0
- Update to use the new class based views in the authentication app. `password_change()`, `password_change_done()`, `password_reset()`, `password_reset_done()`, `password_reset_confirm()`, and `password_reset_complete()` function-based views are deprecated in favor of new class-based views `PasswordChangeView`, `PasswordChangeDoneView`, `PasswordResetView`, `PasswordResetDoneView`, `PasswordResetConfirmView`, and `PasswordResetCompleteView`.
- `django.utils.translation.string_concat()` is deprecated in favor of `django.utils.text.format_lazy()`. `string_concat(strings)` can be replaced by `format_lazy('{{' x len(strings), strings)`. Found in `converter/classes.py` and `metadata/forms.py`.
- Fix warnings in preparation for Django 2.0.
- Update all `tempfile.mkstemp()` to `tempfile.mkstemp(dir=setting_temporary_directory.value)`
- Get rid of `common.utils.get_descriptor` only used by `common.utils.copyfile`
- Update `common.utils.copyfile` to use only file objects.
- Change metadata label column from `CharField` to `Label`
- Start testing to Python 3 compatibility.
- Unify all `RelationshipForms` into a common class.
- Add test for event subscription view.
- Repeated templates: `password_reset_confirm.html` and `password_reset_form.html`
- Remove unused `text=get_notification_count` from `events.links`
- Reduce number of languages so dropzone view starts faster.

- In DocumentComment model, rename comment field to text.
- class MultipleInstanceActionMixin is deprecated. Replace views using this with MultipleObjectFormActionView or MultipleObjectConfirmActionView
- Subscribe to tag events
- Subscribe to workflow events
- Subscribe to workflow state events
- Link to go from event notification to view. Example document workflow notification to document workflow views.
- Find replacement for `python-gnupg`. Unstable & inconsistent API.
- New app that allows creating user document filters. Will provide the same service as the document filters class. Interface can be made using the template language or the same UI as the smart links.
- Allow add queue metadata that can be exported via a management command. This will allow creating supervisor templates without all the worker entries being hardcoded.
- Automatically capture license information from installed Javascript libraries.
- Automatically capture license information from installed Python packages.
- Finish and merge improved compressed file branch.
- Improve and merge PCL support branch.
- Switch to self hosted documentation.
- Unify error logs in a common model. Fields: Datetime, namespace, message, content type, object id.
- Export documents as PDF. Each document image is used to create a PDF dynamically.
- Document splitting. Only for PDF files first. A document versions relationship between the documents has to be designed.
- Manually linking documents.
- Migrate settings/base.py to Django's 1.11 format.
- Rename model methods to use `get_` or `do_`
- Hunt TODO
- Hunt FIXME
- Convert `SETTING_GPG_BACKEND` into a setting option similar to `converter` and `converter` options.
- Reorganize modelForms Meta class and methods.
- Periodic messages, like notifications. Using an AJAX worker.
- Find replacement for `python-gnupg`. Unstable & inconsistent API.
- Google docs integration. Upload document from Google Drive.
- Get `dumpdata` and `loaddata` working flawlessly. Will allow for easier backups, restores and database backend migrations.
- Add generic list ordering. `django.views.generic.list.MultipleObjectMixin` (<https://docs.djangoproject.com/en/1.8/ref/class-based-views/mixins-multiple-object/#django.views.generic.list.MultipleObjectMixin>) now supports an `ordering` parameter.
- Add support to convert any document to PDF. <https://gitlab.mister-muffin.de/josch/img2pdf>
- Add support for combining documents.

- Add support for splitting documents.
- Add new document source to get documents from an URL.
- Add support for metadata mapping files. CSV file containing filename to metadata values mapping, useful for bulk upload and migrations.
- Add support for registering widgets to the home screen.
- Merge mimetype and converter apps.
- Metadata widgets (Date, time, timedate).
- Datetime widget: <https://github.com/smalot/bootstrap-datetimepicker>
- Add events for document signing app (uploaded detached signature, signed document, deleted signature)
- A configurable conversion process. Being able to invoke different binaries for file conversion, as opposed to the current libreoffice only solution.
- A tool in the admin interface to mass (re)convert the files (basically the page count function, but then applied on all documents).

As of Wed Mar 14 19:31:49 2018 (commit f3ba6ee2cca99acbbb2403d69cc3f922b0f2b0a2)

App name: acls

Views

module: mayan/apps/acls/views.py class based views: 4 function based views: 0

API Views

module: mayan/apps/acls/api_views.py class based views: 4 function based views: 0

Tests

module: mayan/apps/acls/tests/test_models.py tests: 8

module: mayan/apps/acls/tests/test_actions.py tests: 2

module: mayan/apps/acls/tests/test_links.py tests: 4

module: mayan/apps/acls/tests/test_api.py tests: 9

module: mayan/apps/acls/tests/test_views.py tests: 8

App name: appearance

Views No views

API Views No API views

Tests No tests

App name: authentication

Views

module: mayan/apps/authentication/views.py class based views: 0 function based views: 7

API Views No API views

Tests

module: mayan/apps/authentication/tests/test_views.py tests: 10

App name: cabinets

Views

module: mayan/apps/cabinets/views.py class based views: 9 function based views: 0

API Views

module: mayan/apps/cabinets/api_views.py class based views: 5 function based views: 0

Tests

module: mayan/apps/cabinets/tests/test_models.py tests: 5

module: mayan/apps/cabinets/tests/test_events.py tests: 2

module: mayan/apps/cabinets/tests/test_api.py tests: 13

module: mayan/apps/cabinets/tests/test_views.py tests: 15

App name: checkouts

Views

module: mayan/apps/checkouts/views.py class based views: 4 function based views: 0

API Views

module: mayan/apps/checkouts/api_views.py class based views: 2 function based views: 0

Tests

module: mayan/apps/checkouts/tests/test_models.py tests: 10

module: mayan/apps/checkouts/tests/test_api.py tests: 10

module: mayan/apps/checkouts/tests/test_views.py tests: 7

App name: common

Views

module: mayan/apps/common/views.py class based views: 17 function based views: 1

API Views

module: mayan/apps/common/api_views.py class based views: 1 function based views: 0

Tests

module: mayan/apps/common/tests/test_api.py tests: 1

module: mayan/apps/common/tests/test_commands.py tests: 1

module: mayan/apps/common/tests/test_views.py tests: 3

App name: converter

Views

module: mayan/apps/converter/views.py class based views: 4 function based views: 0

API Views No API views

Tests

module: mayan/apps/converter/tests/test_classes.py tests: 6

module: mayan/apps/converter/tests/test_views.py tests: 6

App name: django_gpg

Views

module: mayan/apps/django_gpg/views.py class based views: 9 function based views: 0

API Views

module: mayan/apps/django_gpg/api_views.py class based views: 2 function based views: 0

Tests

module: mayan/apps/django_gpg/tests/test_models.py tests: 15

module: mayan/apps/django_gpg/tests/test_api.py tests: 6

module: mayan/apps/django_gpg/tests/test_views.py tests: 4

App name: document_comments

Views

module: mayan/apps/document_comments/views.py class based views: 3 function based views: 0

API Views

module: mayan/apps/document_comments/api_views.py class based views: 2 function based views: 0

Tests

module: mayan/apps/document_comments/tests/test_api.py tests: 8

App name: document_indexing

Views

module: mayan/apps/document_indexing/views.py class based views: 13 function based views: 0

API Views

module: mayan/apps/document_indexing/api_views.py class based views: 6 function based views: 0

Tests

module: mayan/apps/document_indexing/tests/test_models.py tests: 4

module: mayan/apps/document_indexing/tests/test_views.py tests: 8

App name: document_parsing

Views

module: mayan/apps/document_parsing/views.py class based views: 6 function based views: 0

API Views

module: mayan/apps/document_parsing/api_views.py class based views: 1 function based views: 0

Tests

module: mayan/apps/document_parsing/tests/test_parsers.py tests: 1

module: mayan/apps/document_parsing/tests/test_events.py tests: 2

module: mayan/apps/document_parsing/tests/test_api.py tests: 2

module: mayan/apps/document_parsing/tests/test_views.py tests: 4

App name: document_signatures

Views

module: mayan/apps/document_signatures/views.py class based views: 8 function based views: 0

API Views No API views

Tests

module: mayan/apps/document_signatures/tests/test_models.py tests: 13

module: mayan/apps/document_signatures/tests/test_links.py tests: 4

module: mayan/apps/document_signatures/tests/test_views.py tests: 12

App name: document_states

Views

module: mayan/apps/document_states/views.py class based views: 29 function based views: 0

API Views

module: mayan/apps/document_states/api_views.py class based views: 12 function based views: 0

Tests

module: mayan/apps/document_states/tests/test_models.py tests: 3

module: mayan/apps/document_states/tests/test_actions.py tests: 0

module: mayan/apps/document_states/tests/test_api.py tests: 66

module: mayan/apps/document_states/tests/test_views.py tests: 33

App name: documents

Views No views

API Views

module: mayan/apps/documents/api_views.py class based views: 16 function based views: 0

Tests

module: mayan/apps/documents/tests/test_document_type_views.py tests: 12

module: mayan/apps/documents/tests/test_duplicated_document_views.py tests: 4

module: mayan/apps/documents/tests/test_document_views.py tests: 31

module: mayan/apps/documents/tests/test_utils.py tests: 1

module: mayan/apps/documents/tests/test_models.py tests: 12

module: mayan/apps/documents/tests/test_events.py tests: 4

module: mayan/apps/documents/tests/test_widgets.py tests: 1

module: mayan/apps/documents/tests/test_links.py tests: 6

module: mayan/apps/documents/tests/test_search.py tests: 4

module: mayan/apps/documents/tests/test_api.py tests: 39

module: mayan/apps/documents/tests/test_deleted_document_views.py tests: 8

module: mayan/apps/documents/tests/test_document_version_views.py tests: 4

module: mayan/apps/documents/tests/test_document_page_views.py tests: 2

App name: dynamic_search

Views

module: mayan/apps/dynamic_search/views.py class based views: 4 function based views: 0

API Views

module: mayan/apps/dynamic_search/api_views.py class based views: 3 function based views: 0

Tests

module: mayan/apps/dynamic_search/tests/test_models.py tests: 4

module: mayan/apps/dynamic_search/tests/test_api.py tests: 3

module: mayan/apps/dynamic_search/tests/test_views.py tests: 1

App name: events

Views

module: mayan/apps/events/views.py class based views: 9 function based views: 0

API Views

module: mayan/apps/events/api_views.py class based views: 7 function based views: 0

Tests

module: mayan/apps/events/tests/test_api.py tests: 1

module: mayan/apps/events/tests/test_views.py tests: 2

App name: linking

Views

module: mayan/apps/linking/views.py class based views: 11 function based views: 0

API Views

module: mayan/apps/linking/api_views.py class based views: 7 function based views: 0

Tests

module: mayan/apps/linking/tests/test_models.py tests: 1

module: mayan/apps/linking/tests/test_api.py tests: 34

module: mayan/apps/linking/tests/test_views.py tests: 8

App name: lock_manager

Views No views

API Views No API views

Tests

module: mayan/apps/lock_manager/tests/test_backends.py tests: 6

App name: mailer

Views

module: mayan/apps/mailer/views.py class based views: 10 function based views: 0

API Views No API views

Tests

module: mayan/apps/mailer/tests/test_models.py tests: 5

module: mayan/apps/mailer/tests/test_views.py tests: 12

App name: mayan_statistics

Views

module: mayan/apps/mayan_statistics/views.py class based views: 4 function based views: 0

API Views No API views

Tests

module: mayan/apps/mayan_statistics/tests/test_views.py tests: 4

App name: metadata

Views

module: mayan/apps/metadata/views.py class based views: 9 function based views: 0

API Views

module: mayan/apps/metadata/api_views.py class based views: 6 function based views: 0

Tests

module: mayan/apps/metadata/tests/test_models.py tests: 14

module: mayan/apps/metadata/tests/test_wizard_steps.py tests: 1

module: mayan/apps/metadata/tests/test_api.py tests: 35

module: mayan/apps/metadata/tests/test_views.py tests: 8

App name: mimetype

Views No views

API Views No API views

Tests No tests

App name: mirroring

Views No views

API Views No API views

Tests No tests

App name: motd

Views

module: mayan/apps/motd/views.py class based views: 4 function based views: 0

API Views

module: mayan/apps/motd/api_views.py class based views: 2 function based views: 0

Tests

module: mayan/apps/motd/tests/test_models.py tests: 4

module: mayan/apps/motd/tests/test_api.py tests: 10

App name: navigation

Views No views

API Views No API views

Tests

module: mayan/apps/navigation/tests/test_classes.py tests: 9

App name: ocr

Views

module: mayan/apps/ocr/views.py class based views: 8 function based views: 0

API Views

module: mayan/apps/ocr/api_views.py class based views: 3 function based views: 0

Tests

module: mayan/apps/ocr/tests/test_models.py tests: 2

module: mayan/apps/ocr/tests/test_events.py tests: 2

module: mayan/apps/ocr/tests/test_api.py tests: 6

module: mayan/apps/ocr/tests/test_views.py tests: 4

App name: permissions

Views

module: mayan/apps/permissions/views.py class based views: 7 function based views: 0

API Views

module: mayan/apps/permissions/api_views.py class based views: 3 function based views: 0

Tests

module: mayan/apps/permissions/tests/test_models.py tests: 2

module: mayan/apps/permissions/tests/test_api.py tests: 17

module: mayan/apps/permissions/tests/test_views.py tests: 3

App name: rest_api

Views No views

API Views

module: mayan/apps/rest_api/api_views.py class based views: 1 function based views: 0

Tests

App name: smart_settings

Views

module: mayan/apps/smart_settings/views.py class based views: 2 function based views: 0

API Views No API views

Tests

module: mayan/apps/smart_settings/tests/test_classes.py tests: 1

module: mayan/apps/smart_settings/tests/test_view_permissions.py tests: 2

App name: sources

Views

module: mayan/apps/sources/views.py class based views: 10 function based views: 0

API Views

module: mayan/apps/sources/api_views.py class based views: 4 function based views: 0

Tests

module: mayan/apps/sources/tests/test_models.py tests: 3

module: mayan/apps/sources/tests/test_classes.py tests: 1

module: mayan/apps/sources/tests/test_views.py tests: 15

App name: storage

Views No views

API Views No API views

Tests No tests

App name: tags

Views

module: mayan/apps/tags/views.py class based views: 8 function based views: 0

API Views

module: mayan/apps/tags/api_views.py class based views: 5 function based views: 0

Tests

module: mayan/apps/tags/tests/test_models.py tests: 1

module: mayan/apps/tags/tests/test_actions.py tests: 2

module: mayan/apps/tags/tests/test_wizard_steps.py tests: 1

module: mayan/apps/tags/tests/test_api.py tests: 28

module: mayan/apps/tags/tests/test_views.py tests: 18

App name: task_manager

Views

module: mayan/apps/task_manager/views.py class based views: 4 function based views: 0

API Views No API views

Tests

module: mayan/apps/task_manager/tests/test_views.py tests: 8

App name: user_management

Views

module: mayan/apps/user_management/views.py class based views: 11 function based views: 0

API Views

module: mayan/apps/user_management/api_views.py class based views: 6 function based views: 0

Tests

module: mayan/apps/user_management/tests/test_api.py tests: 31

module: mayan/apps/user_management/tests/test_views.py tests: 10

Totals: Tests: 753 Function based API views: 0 Function based views: 8 Apps: 34 Class based API views: 98 Class based views: 207

These are the defined views and tests defined. Actual executable views or tests could be higher if subclassed.

Actual test executed: 757 as lock manager app subclasses its 6 tests once

CHAPTER 29

Translations

Translations are handled online via the **Transifex** website: <https://www.transifex.com/projects/p/mayan-edms/>. To create a translation team for a new language or contribute to an already existing language translation, create a **Transifex** account and contact the team coordinator of the respective language in which you are interested.

Feel free to open translation issues inside **Transifex** itself if you have a question about the usage or meaning of a source text string. If you open a translation issue, it will be your responsibility to close it after you get an answers that satisfies your question. Administrator will not close new issues as they have no way to determine if your question has been properly answered. However to avoid clutter, answered questions will be scanned periodically and closed if no activity is observed from the original poster in a period of time.

Mayan EDMS is distributed under the [Apache 2.0 License](#). The complete license terms are included below.

30.1 License terms

Copyright 2011-2015 Roberto Rosario

Licensed under the Apache License, Version 2.0 (the “License”); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Frequently asked questions and solutions

Q: PostgreSQL vs. MySQL

Since Django abstracts database operations from a functional point of view Mayan EDMS will behave exactly the same either way. The only concern would be that MySQL doesn't support transactions for schema modifying commands. The only moment this could cause problems is when running South migrations during upgrades, if a migration fails the database structure is left in a transitory state and has to be reverted manually before trying again.

Q: `_mysql_exceptions. OperationalError: (1267, "Illegal mix of collations (latin1_swedish_ci, IMPLICIT) and (utf8_general_ci, COERCIBLE) for operation '=')`

- Solution:

```
$ mayan-edms.py shell
```

```
>>> from django.db import connection
>>> cursor = connection.cursor()
>>> cursor.execute('SHOW TABLES')
>>> results=[]
>>> for row in cursor.fetchall(): results.append(row)
>>> for row in results: cursor.execute('ALTER TABLE %s CONVERT TO CHARACTER SET_
↳utf8 COLLATE utf8_general_ci;' % (row[0]))
```

- References:

- <http://www.djangoshmango.com/?p=99>
- <http://stackoverflow.com/questions/1073295/django-character-set-with-mysql-weirdness>

Q: Incorrect string value: “‘xE2x80x95rs6...” for column “content” at row 1

When using MySQL and doing OCR on languages other than English

- Solution:

- Use utf-8 collation on MySQL server, or at least in table ‘documents_documentpage’, ‘content’ field

- Ref: 1- http://groups.google.com/group/django-users/browse_thread/thread/429447086fca6412
- Ref: 2- <http://markmail.org/message/bqajx2utvmtriixi>

Q: Error “django.db.utils.IntegrityError IntegrityError: (1452, ‘Cannot add or update a child row: a foreign key constraint fails (‘...‘:‘...‘, CONSTRAINT ‘..._refs_id_b0252274‘ FOREIGN KEY (‘...‘) REFERENCES ‘...‘ (‘...‘))’)

- Solution:
 - Convert all MySQL tables to the same type, either all MyISAM or InnoDB

Q: File system links not showing when serving content with “Samba“

- Solution:
 - Disable unix extensions in the [global] section and enable wide links for the file serving share
 - Example:

```
[global]
    unix extensions = no
    ...

[digitalizacion]
    path = /var/local/mayan
    guest ok = yes
    read only = yes
    wide links = yes
    follow symlinks = yes
```

- Ref: 1- <http://www.samba.org/samba/docs/man/manpages-3/smb.conf.5.html>

Q: How do you upload a new version of an existing file?

- Solution:
 - Choose a document, and go to the versions tab, on the right menu at the bottom under Other available action there is Upload new version. Clicking it will take you to a very similar view as the Upload new document but you will be able to specify version number and comments for the new version being uploaded.

Q: Is virtualenv required as specified in the documentation?

- It is not necessary, but it’s a strong recommendation mainly to reduce dependency conflicts by isolation from the main Python system install. If not using a virtualenv, pip would install Mayan’s dependencies globally coming in conflict with the distribution’s prepackaged Python libraries messing other Django projects or Python programs, or another later Python/Django project dependencies coming into conflict causing Mayan to stop working for no apparent reason.

Q: Mayan EDMS installed correctly and works, but static files are not served

Django’s development server doesn’t serve static files unless the DEBUG option is set to True, this mode of operation should only be used for development or testing. For production deployments the management command:

```
$ mayan-edms.py collectstatic
```

should be used and the resulting static folder served from a webserver. For more information, read <https://docs.djangoproject.com/en/dev/howto/static-files/> and <https://docs.djangoproject.com/en/1.2/howto/static-files/> or http://mayan-edms-ru.blogspot.com/2011/11/blog-post_09.html

Q: Can you change the display order of documents...i.e can they be in alphabetical order?

A the moment no, but it is something being considered.

Q: Does Mayan EDMS work with Python 3?

Not at the moment. When all the projects and libraries upon which Mayan is dependent support Python 3 then will the project move to fully support Python 3. Update 2018-06-04: Python 3 will be required from Django 2.0 onwards. As such this is now a bigger priority and the transition to Python 3 has begun.

Q: The watched folder feature is not working

Make sure that the Celery BEAT scheduler is running correctly as it is the element that triggers the periodics tasks.

Q: How to do LDAP authentication

A sample settings file called `ldap_connection_settings.py` is included in the `contrib/settings/` folder of the repository showing how to setup LDAP authentication.

32.1 FAQ

- Try the [FAQ](#) – it’s got answers to many common questions.

32.2 Mailing list

Search for information in the [archives of the mayan-edms mailing list](#), or [post a question](#). If you prefer news servers, use the gateway provided by [Gmane](#).

Mayan EDMS community developers do their best to reply to basic questions. Be sure to check the list archives as it may already contain the answers to your questions.

32.3 Twitter

Mayan EDMS has an official Twitter account, [@mayanedms](#), which is used for announcements and occasional related news tidbits.

32.4 Bugs/ticket tracker

Report bugs with Mayan EDMS or search existing ones using Gitlab’s [ticket tracker](#).

Mayan EDMS Request For Comment documents index.

Table of Contents

- *Status*
 - *Accepted*
 - *Draft*
- *Type*
 - *Process*
 - *Feature*

33.1 Status

33.1.1 Accepted

MERC 1: Purpose and Guidelines

MERC 1

Author Michael Price

Status Accepted

Type Process

Created 2018-02-17

Last-Modified 2018-02-17

Table of Contents

- *What is a MERC?*
- *MERC Types*
- *MERC submission workflow*
 - *Pre-proposal*
 - *Submitting the draft*
 - *Implementation*
- *MERC format*
 - *MERC Metadata*
 - *Auxiliary Files*
- *Reporting MERC Bugs, or Submitting MERC Updates*

What is a MERC?

A Mayan EDMS Request For Comment document or MERC document is a design document providing information to the Mayan EDMS community, or describing a new feature or process for Mayan EDMS. MERCs provide concise technical specifications of features, along with rationales.

MERC Types

There are three kinds of MERCs:

1. A **Feature** MERC describes a new feature or implementation for Mayan EDMS. Most MERCs will be Feature MERCs.
2. An **Informational** MERC describes a Mayan EDMS design issue, or provides general guidelines or information to the Mayan EDMS community, but does not propose a new feature. Informational MERCs do not necessarily represent a community consensus or recommendation, so users and implementers are free to ignore Informational MERCs or follow their advice.
3. A **Process** MERC describes a process surrounding Mayan EDMS, or proposes a change to (or an event in) a process. Process MERCs are like Feature MERCs but apply to areas other than the Mayan EDMS framework itself. They may propose an implementation, but not to Mayan EDMS's codebase; they often require community consensus; unlike Informational MERCs, they are more than recommendations, and users are typically not free to ignore them. Examples include procedures, guidelines, changes to the decision-making process, and changes to the tools or environment used in Mayan EDMS development. Any meta-MERC is also considered a Process MERC. (So this document is a Process MERC).

MERC submission workflow

Pre-proposal

The MERC process begins with a new idea for Mayan EDMS. It is highly recommended that a single MERC contain a single key proposal or new idea. Small enhancements or patches usually don't need a MERC and follow Mayan EDMS's normal contribution process.

MERCs should be focused on a single topic. If in doubt, split your MERC into several well-focused ones.

Once the idea's been vetted, a draft MERC should be presented to the Mayan EDMS mailing list. This gives the author a chance to flesh out the draft MERC to make sure it's properly formatted, of high quality, and to address initial concerns about the proposal.

The Core Developers will be responsible for accepting or rejecting the MERC proposal.

Submitting the draft

Following the discussion on Mayan EDMS mailing list, the proposal should be sent as a merge request to the Mayan EDMS repository. The draft must be written in MERC style; if it isn't the merge request may be rejected until proper formatting rules are followed.

Implementation

Finally, once a MERC has been accepted, the implementation must be completed. In many cases some (or all) implementation will actually happen during the MERC process: Feature MERCs will often have fairly complete implementations before being reviewed. When the implementation is complete and incorporated into the main source code repository, the status will be changed to "Final".

MERC format

MERCs need to follow a common format and outline; this section describes that format.

MERCs must be written in [reStructuredText](#) (the same format as Mayan EDMS's documentation).

Each MERC should have the following parts:

1. A short descriptive title (e.g. "User document filters"), which is also reflected in the MERC's filename (e.g. `0002-user-document-filters.rst`).
2. A preamble – a rST [field list](#) containing metadata about the MERC, including the MERC number and so forth. See [MERC Metadata](#) below for specific details.
3. Abstract – a short (~200 word) description of the technical issue being addressed.
4. Specification – The technical specification should describe the syntax and semantics of any new feature. The specification should be detailed enough to allow implementation – that is, developers other than the author should (given the right experience) be able to independently implement the feature, given only the MERC.
5. Motivation – The motivation is critical for MERCs that want to add substantial new features or materially refactor existing ones. It should clearly explain why the existing solutions are inadequate to address the problem that the MERC solves. MERC submissions without sufficient motivation may be rejected outright.
6. Rationale – The rationale fleshes out the specification by describing what motivated the design and why particular design decisions were made. It should describe alternate designs that were considered and related work.

The rationale should provide evidence of consensus within the community and discuss important objections or concerns raised during discussion.

7. Backwards Compatibility – All MERCs that introduce backwards incompatibilities must include a section describing these incompatibilities and their severity. The MERC must explain how the author proposes to deal with these incompatibilities. MERC submissions without a sufficient backwards compatibility treatise may be rejected outright.

8. Reference Implementation – The reference implementation must be completed before any MERC is given status “Final”, but it need not be completed before the MERC is accepted. While there is merit to the approach of reaching consensus on the specification and rationale before writing code, the principle of “rough consensus and running code” is still useful when it comes to resolving many discussions of API details.

The final implementation must include tests and documentation, per Mayan EDMS development guide.

MERC Metadata

Each MERC must begin with some metadata given as an rST [field list](#). The headers must contain the following fields:

MERC The MERC number. In an initial merge request, this can be left out or given as XXXX; the reviewer who merges the pull request will assign the MERC number.

Type Feature, Informational, or Process

Status Draft, Accepted, Rejected, Withdrawn, Final, or Superseded

Created Original creation date of the MERC (in yyyy-mm-dd format)

Last-Modified Date the MERC was last modified (in yyyy-mm-dd format)

Author The MERC’s author(s).

Implementation-Team The person/people who have committed to implementing this MERC

Requires If this MERC depends on another MERC being implemented first, this should be a link to the required MERC.

Mayan EDMS-Version (optional) For Feature MERCs, the version of Mayan EDMS (e.g. 2.7.3) that this feature will be released in.

Replaces and Superseded-By (optional) These fields indicate that a MERC has been rendered obsolete. The newer MERC must have a `Replaces` header containing the number of the MERC that it rendered obsolete; the older MERC has a `Superseded-By` header pointing to the newer MERC.

Resolution (optional) For MERCs that have been decided upon, this can be a link to the final rationale for acceptance/rejection. It’s also reasonable to simply update the MERC with a “Resolution” section, in which case this header can be left out.

Auxiliary Files

MERCs may include auxiliary files such as diagrams. Such files must be named `XXXX-descriptive-title.ext`, where “XXXX” is the MERC number, “descriptive-title” is a short slug indicating what the file contains, and “ext” is replaced by the actual file extension (e.g. “png”).

Reporting MERC Bugs, or Submitting MERC Updates

How you report a bug, or submit a MERC update depends on several factors, such as the maturity of the MERC, the preferences of the MERC author, and the nature of your comments. For the early draft stages of the MERC, it’s probably best to send your comments and changes directly to the MERC author. For more mature, or finished MERCs you can submit corrections as repository issues or merge requests against the git repository.

When in doubt about where to send your changes, please check first with the MERC author and/or a core developer.

MERC authors with git push privileges for the MERC repository can update the MERCs themselves.

MERC 2: Test writing

MERC 2

Author Michael Price

Status Accepted

Type Feature

Created 2018-02-22

Last-Modified 2018-04-01

Table of Contents

- *Abstract*
- *Motivation*
- *Specification*

Abstract

This MERC proposes a standard methodology for writing tests for Mayan EDMS.

Motivation

Having a standard methodology for writing tests has the following advantages:

1. Scaffolding can be reduced by providing the most frequently used paradigms as methods or helper functions.
2. Reduce the probabilities of errors slipping through poorly written tests.

Specification

1. Tests must test each view in at least two ways:
 - (a) Object creations views must be tested with and without permissions.
 - (b) Object detail, list and delete views must be tested with and without object access.
2. Tests must assert the status code of the response even when the expected status is HTTP 200.
3. The actual request performed must be enclosed in a private methods so that the fail and pass tests use the same HTTP request.
4. Test must verify that changes happened and didn't happened in the database regardless of the return code. Even is an edit view returns and error 4XX (404-Not found, 403-Forbidden, etc), the test must ensure that the data was not indeed modified.
5. All tests must use the test user created by the BaseAPITestCase and not an super user unless absolutely required by the test.
6. Each test must test just one thing.
7. If a test object needs to be created before the execution of a request this object must be created by a private method.

Example:

```
def _request_tag_create(self):
    return self.post(
        viewname='rest_api:tag-list', data={
            'label': TEST_TAG_LABEL, 'color': TEST_TAG_COLOR
        }
    )

def test_tag_create_view_no_permission(self):
    response = self._request_tag_create()
    self.assertEqual(response.status_code, status.HTTP_403_FORBIDDEN)
    self.assertEqual(Tag.objects.count(), 0)

def test_tag_create_view_with_permission(self):
    self.grant_permission(permission=permission_tag_create)
    response = self._request_tag_create()
    self.assertEqual(response.status_code, status.HTTP_201_CREATED)

    tag = Tag.objects.first()
    self.assertEqual(response.data['id'], tag.pk)
    self.assertEqual(response.data['label'], TEST_TAG_LABEL)
    self.assertEqual(response.data['color'], TEST_TAG_COLOR)

    self.assertEqual(Tag.objects.count(), 1)
    self.assertEqual(tag.label, TEST_TAG_LABEL)
    self.assertEqual(tag.color, TEST_TAG_COLOR)
```

MERC 3: Using javascript libraries

MERC 3

Author Eric Riggs

Status Accepted

Type Feature

Created 2018-03-08

Last-Modified 2018-06-04

Table of Contents

- *Abstract*
- *Rationale*
- *Motivation*
- *Backwards Compatibility*
- *Specification*

Abstract

This MERC proposes a standard way to use javascript libraries.

Rationale

Mayan EDMS uses several javascript libraries for user interface features. Currently, the libraries are not installed using any javascript package manager but copied uncompressed. Installing the libraries in this manner carries some disadvantages.

Motivation

The inclusion of the libraries in source form is required by many licenses if the library is not installed by a package manager in distributable form. There are several disadvantages with the current approach:

1. Having the library in source form means that the entire weight of the library's size carries over the overall size of the Mayan EDMS distribution files. The justification for not doing this is the same as with the Python libraries which are not copied with the code but downloaded upon installation.
2. Upgrading the libraries means manually examining the version of the installed in the project and manually searching, downloading, compressing and adding the files to the repository.
3. The source form of the libraries includes normal and minified versions of the code and the accompanying CSS files. There is no define preference and through the project both versions of the libraries are loaded interchangeably. Using a packager manager the minified version would be used of a pipeline to minify the installed libraries should be added.

Backwards Compatibility

There are no backwards compatibility issues with this proposal.

Specification

Changes needed:

1. Python based javascript package manager. Alternatively a Python wrapper for a javascript package manager could be used.
2. Package manifest for the javascript libraries used.
3. Installation pipeline to install the javascript libraries during the installation and setup of the project.

References:

- <https://github.com/JDeuce/powser>
- <https://github.com/javrasya/version-manager>
- <https://github.com/inveniosoftware-attic/setuptools-bower>
- <https://pypi.python.org/pypi/django-bower-cache/0.5.0>
- <http://django-pipeline.readthedocs.io/en/latest/index.html>
- <https://github.com/nvbn/django-bower>

33.1.2 Draft

MERC XX: Unify Roles and Groups

MERC XX

Author Michael Price

Status Draft

Type Feature

Created 2018-02-27

Last-Modified 2018-02-27

Table of Contents

- *Abstract*
- *Rationale*
- *Motivation*
- *Backwards Compatibility*
- *Specification*

Abstract

This MERC proposes the merging of the Roles and Group models.

Rationale

Mayan EDMS uses Groups as units of users that are meant to mirror an organization's actual user hierarchy. Roles are used as permission units.

Separation of concerns is a concept Mayan EDMS executes very successfully but when it comes to the Roles/Groups relationship that execution causes overheads without providing advantages in the day to day operations.

In reality there is almost a 1 to 1 correlation between Roles and Groups. Other permissions systems already use Groups as permission units without disadvantages. An example of this is LDAP and its commercial counterpart Active Directory.

Motivation

Merging the Role and Group model will reduce some complexity when initially setting up Mayan EDMS. The merge allows removing a Mayan EDMS model in favor of using a native Django model for the same task.

Merging the Role and Group models will also provide a speed boost in every permission check and queryset filtering. These checks are nested in nature. Since the access checks are performed for every view and for every link in the view the performance gain should be substantial.

Backwards Compatibility

To avoid loss of role configuration a data migration will be needed to convert existing roles to groups.

Specification

Changes needed:

1. Data migration to convert existing roles to groups.
2. Prepend or append an identifier to the migrated roles.
3. Intermediate model to map permissions to a group. This will substitute the Role model's permissions many to many field.
4. Update the `AccessControlList` models roles field to point to the group models.
5. Update the role checks in the `check_access` and `filter_by_access` `AccessControlList` model manager methods.

MERC XX: Support forum

MERC XX

Author Michael Price

Status Draft

Type Process

Created 2018-02-27

Last-Modified 2018-02-27

Table of Contents

- *Abstract*
- *Motivation*
- *Specification*

Abstract

This MERC proposes the move of the official community meeting place from Google Group to a self hosted forum platform.

Motivation

Google Groups is not a proper forum platform and as such is lacking many features that would increase participation. Google Groups has not had any significant update in years and the age of the platform is visible. There are no official mobile apps for Google Groups, no responsible template is not provided. There are not integration options. These factors contribute to the problems of knowledge dilution and one time participation being experienced in the Mayan EDMS community.

Other concerns for moving the community to a self hosted forum solution are that Google Groups presents a single point of failure by relying on a free product hosted by a commercial company with no continuation plan for the product. Google Groups doesn't provide archiving features and the current archive solution relies on other third party services, one of which (GMANE) has stopped working. Recent user privacy and censorship issues regarding Google, reinforce the need for a self hosted solution.

Specification

The proposed solution is to use the bbPress (<https://wordpress.org/plugins/bbpress/>) forum plugin for Wordpress. After 14 years, Wordpress dominance continues unrivaled. The wealth of material, themes, templates and professional avaiable make it a safe option.

33.2 Type

33.2.1 Process

33.2.2 Feature

Mayan EDMS Individual Contributor Assignment Agreement

Thank you for your interest in contributing to Mayan EDMS (“We” or “Us”).

This contributor agreement (“Agreement”) documents the rights granted by contributors to Us. To make this document effective, please print it, sign it (by copyright holder or authorized party) and send it to Us by email to caa@mayan-edms.com. This is a legally binding document, so please read it carefully before agreeing to it. The Agreement may cover more than one software project managed by Us.

34.1 1. Definitions

“You” means the individual who Submits a Contribution to Us.

“Contribution” means any work of authorship that is Submitted by You to Us in which You own or assert ownership of the Copyright. We cannot accept contributions for which you do not own the Copyright or for which you don’t have the necessary legal power to transfer.

“Copyright” means all rights protecting works of authorship owned or controlled by You, including copyright, moral and neighboring rights, as appropriate, for the full term of their existence including any extensions by You.

“Material” means the work of authorship which is made available by Us to third parties. When this Agreement covers more than one software project, the Material means the work of authorship to which the Contribution was Submitted. After You Submit the Contribution, it may be included in the Material.

“Submit” means any form of electronic, verbal, or written communication sent to Us or our representatives, including but not limited to electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, Us for the purpose of discussing and improving the Material, but excluding communication that is conspicuously marked or otherwise designated in writing by You as “Not a Contribution.”

“Submission Date” means the date on which You Submit a Contribution to Us.

“Effective Date” means the date You execute this Agreement or the date You first Submit a Contribution to Us, whichever is earlier.

34.2 2. Grant of Rights

34.2.1 2.1 Copyright Assignment

1. At the time the Contribution is Submitted, You assign to Us all right, title, and interest worldwide in all Copyright covering the Contribution; provided that this transfer is conditioned upon compliance with Section 2.3.
2. To the extent that any of the rights in Section 2.1(a) cannot be assigned by You to Us, You grant to Us a perpetual, worldwide, exclusive, royalty-free, transferable, irrevocable license under such non-assigned rights, with rights to sublicense through multiple tiers of sublicensees, to practice such non-assigned rights, including, but not limited to, the right to reproduce, modify, display, perform and distribute the Contribution; provided that this license is conditioned upon compliance with Section 2.3.
3. To the extent that any of the rights in Section 2.1(a) can neither be assigned nor licensed by You to Us, You irrevocably waive and agree never to assert such rights against Us, any of our successors in interest, or any of our licensees, either direct or indirect; provided that this agreement not to assert is conditioned upon compliance with Section 2.3.
4. Upon such transfer of rights to Us, the Contribution will be licenses under the terms of the Material.

34.2.2 2.2 Patent License

For patent claims including, without limitation, method, process, and apparatus claims which You own, control or have the right to grant, now or in the future, You grant to Us a perpetual, worldwide, non-exclusive, transferable, royalty-free, irrevocable patent license, with the right to sublicense these rights to multiple tiers of sublicensees, to make, have made, use, sell, offer for sale, import and otherwise transfer the Contribution and the Contribution in combination with the Material (and portions of such combination). This license is granted only to the extent that the exercise of the licensed rights infringes such patent claims; and provided that this license is conditioned upon compliance with Section 2.3.

34.2.3 2.3 Outbound License

As a condition on the grant of rights in Sections 2.1 and 2.2, We agree to license the Contribution only under the terms of the license or licenses which We are using on the Submission Date for the Material (including any rights to adopt any future version of a license if permitted).

34.2.4 2.4 Moral Rights

If moral rights apply to the Contribution, to the maximum extent permitted by law, You waive and agree not to assert such moral rights against Us or our successors in interest, or any of our licensees, either direct or indirect.

34.2.5 2.5 Our Rights

You acknowledge that We are not obligated to use Your Contribution as part of the Material and may decide to include any Contribution We consider appropriate.

34.2.6 2.6 Reservation of Rights

Any rights not expressly assigned or licensed under this section are expressly reserved by You.

34.3 3. Agreement

You confirm that:

1. You have the legal authority to enter into this Agreement.
2. You own the Copyright and patent claims covering the Contribution which are required to grant the rights under Section 2.
3. The grant of rights under Section 2 does not violate any grant of rights which You have made to third parties, including Your employer. If You are an employee, You have had Your employer approve this Agreement or sign the Entity version of this document. If You are less than eighteen years old, please have Your parents or guardian sign the Agreement.

34.4 4. Disclaimer

EXCEPT FOR THE EXPRESS WARRANTIES IN SECTION 3, THE CONTRIBUTION IS PROVIDED “AS IS”. MORE PARTICULARLY, ALL EXPRESS OR IMPLIED WARRANTIES INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT ARE EXPRESSLY DISCLAIMED BY YOU TO US AND BY US TO YOU. TO THE EXTENT THAT ANY SUCH WARRANTIES CANNOT BE DISCLAIMED, SUCH WARRANTY IS LIMITED IN DURATION TO THE MINIMUM PERIOD PERMITTED BY LAW.

34.5 5. Consequential Damage Waiver

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT WILL YOU OR US BE LIABLE FOR ANY LOSS OF PROFITS, LOSS OF ANTICIPATED SAVINGS, LOSS OF DATA, INDIRECT, SPECIAL, INCIDENTAL, CONSEQUENTIAL AND EXEMPLARY DAMAGES ARISING OUT OF THIS AGREEMENT REGARDLESS OF THE LEGAL OR EQUITABLE THEORY (CONTRACT, TORT OR OTHERWISE) UPON WHICH THE CLAIM IS BASED.

34.6 6. Miscellaneous

34.6.1 6.1 Jurisdiction

This Agreement will be governed by and construed in accordance with the laws of Puerto Rico excluding its conflicts of law provisions. Under certain circumstances, the governing law in this section might be superseded by the United Nations Convention on Contracts for the International Sale of Goods (“UN Convention”) and the parties intend to avoid the application of the UN Convention to this Agreement and, thus, exclude the application of the UN Convention in its entirety to this Agreement.

34.6.2 6.2 Acceptance

This Agreement sets out the entire agreement between You and Us for Your Contributions to Us and overrides all other agreements or understandings.

34.6.3 6.3 Third parties

If You or We assign the rights or obligations received through this Agreement to a third party, as a condition of the assignment, that third party must agree in writing to abide by all the rights and obligations in the Agreement.

34.6.4 6.4 Unmet responsibilities

The failure of either party to require performance by the other party of any provision of this Agreement in one situation shall not affect the right of a party to require such performance at any time in the future. A waiver of performance under a provision in one situation shall not be considered a waiver of the performance of the provision in the future or a waiver of the provision in its entirety.

34.6.5 6.5 Continuation

If any provision of this Agreement is found void and unenforceable, such provision will be replaced to the extent possible with a provision that comes closest to the meaning of the original provision and which is enforceable. The terms and conditions set forth in this Agreement shall apply notwithstanding any failure of essential purpose of this Agreement or any limited remedy to the maximum extent possible under law.

Name: _____
Email: _____
Address: _____
Address (cont): _____
Country: _____
Phone number: _____
Fax number: _____

Signature

Mayan EDMS Individual CAA Version 1.0 April 8, 2018

Mayan EDMS Entity Contributor Assignment Agreement

Thank you for your interest in contributing to Mayan EDMS (“We” or “Us”).

This contributor agreement (“Agreement”) documents the rights granted by contributors to Us. To make this document effective, please print it, sign it (by copyright holder or authorized party) and send it to Us by email to caa@mayan-edms.com. This is a legally binding document, so please read it carefully before agreeing to it. The Agreement may cover more than one software project managed by Us.

35.1 1. Definitions

“You” means any Legal Entity on behalf of whom a Contribution has been received by Us. “Legal Entity” means an entity which is not a natural person. “Affiliates” means other Legal Entities that control, are controlled by, or under common control with that Legal Entity. For the purposes of this definition, “control” means (i) the power, direct or indirect, to cause the direction or management of such Legal Entity, whether by contract or otherwise, (ii) ownership of fifty percent (50%) or more of the outstanding shares or securities which vote to elect the management or other persons who direct such Legal Entity or (iii) beneficial ownership of such entity.

“Contribution” means any work of authorship that is Submitted by You to Us in which You own or assert ownership of the Copyright. We cannot accept contributions for which you do not own the Copyright or for which you don’t have the necessary legal power to transfer.

“Copyright” means all rights protecting works of authorship owned or controlled by You or Your Affiliates, including copyright, moral and neighboring rights, as appropriate, for the full term of their existence including any extensions by You.

“Material” means the work of authorship which is made available by Us to third parties. When this Agreement covers more than one software project, the Material means the work of authorship to which the Contribution was Submitted. After You Submit the Contribution, it may be included in the Material.

“Submit” means any form of electronic, verbal, or written communication sent to Us or our representatives, including but not limited to electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, Us for the purpose of discussing and improving the Material, but excluding communication that is conspicuously marked or otherwise designated in writing by You as “Not a Contribution.”

“Submission Date” means the date on which You Submit a Contribution to Us.

“Effective Date” means the date You execute this Agreement or the date You first Submit a Contribution to Us, whichever is earlier.

35.2 2. Grant of Rights

35.2.1 2.1 Copyright Assignment

1. At the time the Contribution is Submitted, You assign to Us all right, title, and interest worldwide in all Copyright covering the Contribution; provided that this transfer is conditioned upon compliance with Section 2.3.
2. To the extent that any of the rights in Section 2.1(a) cannot be assigned by You to Us, You grant to Us a perpetual, worldwide, exclusive, royalty-free, transferable, irrevocable license under such non-assigned rights, with rights to sublicense through multiple tiers of sublicensees, to practice such non-assigned rights, including, but not limited to, the right to reproduce, modify, display, perform and distribute the Contribution; provided that this license is conditioned upon compliance with Section 2.3.
3. To the extent that any of the rights in Section 2.1(a) can neither be assigned nor licensed by You to Us, You irrevocably waive and agree never to assert such rights against Us, any of our successors in interest, or any of our licensees, either direct or indirect; provided that this agreement not to assert is conditioned upon compliance with Section 2.3.
4. Upon such transfer of rights to Us, the Contribution will be licenses under the terms of the Material.

35.2.2 2.2 Patent License

For patent claims including, without limitation, method, process, and apparatus claims which You or Your Affiliates own, control or have the right to grant, now or in the future, You grant to Us a perpetual, worldwide, non-exclusive, transferable, royalty-free, irrevocable patent license, with the right to sublicense these rights to multiple tiers of sublicensees, to make, have made, use, sell, offer for sale, import and otherwise transfer the Contribution and the Contribution in combination with the Material (and portions of such combination). This license is granted only to the extent that the exercise of the licensed rights infringes such patent claims; and provided that this license is conditioned upon compliance with Section 2.3.

35.2.3 2.3 Outbound License

As a condition on the grant of rights in Sections 2.1 and 2.2, We agree to license the Contribution only under the terms of the license or licenses which We are using on the Submission Date for the Material (including any rights to adopt any future version of a license if permitted).

35.2.4 2.4 Moral Rights

If moral rights apply to the Contribution, to the maximum extent permitted by law, You waive and agree not to assert such moral rights against Us or our successors in interest, or any of our licensees, either direct or indirect.

35.2.5 2.5 Our Rights

You acknowledge that We are not obligated to use Your Contribution as part of the Material and may decide to include any Contribution We consider appropriate.

35.2.6 2.6 Reservation of Rights

Any rights not expressly assigned or licensed under this section are expressly reserved by You.

35.3 3. Agreement

You confirm that:

1. You have the legal authority to enter into this Agreement.
2. You or Your Affiliates own the Copyright and patent claims covering the Contribution which are required to grant the rights under Section 2.
3. The grant of rights under Section 2 does not violate any grant of rights which You or Your Affiliates have made to third parties.

35.4 4. Disclaimer

EXCEPT FOR THE EXPRESS WARRANTIES IN SECTION 3, THE CONTRIBUTION IS PROVIDED “AS IS”. MORE PARTICULARLY, ALL EXPRESS OR IMPLIED WARRANTIES INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT ARE EXPRESSLY DISCLAIMED BY YOU TO US AND BY US TO YOU. TO THE EXTENT THAT ANY SUCH WARRANTIES CANNOT BE DISCLAIMED, SUCH WARRANTY IS LIMITED IN DURATION TO THE MINIMUM PERIOD PERMITTED BY LAW.

35.5 5. Consequential Damage Waiver

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT WILL YOU OR US BE LIABLE FOR ANY LOSS OF PROFITS, LOSS OF ANTICIPATED SAVINGS, LOSS OF DATA, INDIRECT, SPECIAL, INCIDENTAL, CONSEQUENTIAL AND EXEMPLARY DAMAGES ARISING OUT OF THIS AGREEMENT REGARDLESS OF THE LEGAL OR EQUITABLE THEORY (CONTRACT, TORT OR OTHERWISE) UPON WHICH THE CLAIM IS BASED.

35.6 6. Miscellaneous

35.6.1 6.1 Jurisdiction

This Agreement will be governed by and construed in accordance with the laws of Puerto Rico excluding its conflicts of law provisions. Under certain circumstances, the governing law in this section might be superseded by the United Nations Convention on Contracts for the International Sale of Goods (“UN Convention”) and the parties intend to avoid the application of the UN Convention to this Agreement and, thus, exclude the application of the UN Convention in its entirety to this Agreement.

35.6.2 6.2 Acceptance

This Agreement sets out the entire agreement between You and Us for Your Contributions to Us and overrides all other agreements or understandings.

35.6.3 6.3 Third parties

If You or We assign the rights or obligations received through this Agreement to a third party, as a condition of the assignment, that third party must agree in writing to abide by all the rights and obligations in the Agreement.

35.6.4 6.4 Unmet responsibilities

The failure of either party to require performance by the other party of any provision of this Agreement in one situation shall not affect the right of a party to require such performance at any time in the future. A waiver of performance under a provision in one situation shall not be considered a waiver of the performance of the provision in the future or a waiver of the provision in its entirety.

35.6.5 6.5 Continuation

If any provision of this Agreement is found void and unenforceable, such provision will be replaced to the extent possible with a provision that comes closest to the meaning of the original provision and which is enforceable. The terms and conditions set forth in this Agreement shall apply notwithstanding any failure of essential purpose of this Agreement or any limited remedy to the maximum extent possible under law.

Name: _____
Email: _____
Address: _____
Address (cont): _____
Country: _____
Phone number: _____
Fax number: _____

Signature

Mayan EDMS Entity CAA Version 1.0 April 8, 2018