PROJECT-NAME Documentation

Release 0.1

Author

January 20, 2017

Contents

Contents:

CHAPTER 1

Settings Configuring Installation

- enable `’django.template.loaders.eggs.Loader’` in `TEMPLATE_LOADERS` in your `settings.py` file

To set this application up the right way, you will need to do some things to your project configuration, and your app_folder configuration. You can use the demo_project or use your own if you have already started a main project

- add `crudstuff.context_processprs.admin_data` to TEMPLATE_CONTEXT_PROCESSORS on admin
- (optional) add `DEBUG_CRUD = True` to enable logging to stderr and stdout

Here is an example of my context processors entry on settings.py:

```
TEMPLATE_CONTEXT_PROCESSORS = ("django.contrib.auth.context_processors.auth",
"django.core.context_processors.debug",
"django.core.context_processors.i18n",
"django.core.context_processors.media",
"django.core.context_processors.static",
"django.core.context_processors.tz",
"django.contrib.messages.context_processors.messages",
"crudstuff.context_processors.admin_data",)
```

1.1 Bind the models and forms

You will need to specify the models, the app name where that model class lives, and the form name associated with that model. The “name” field of the model is a lowercase string representation of the class name so that we can then perform a model_name search via the context_processors. See the code in context_processors.py

modify `settings.py` and change the class-table names and forms inside of data dictionary. See below:

```
CRUDSTUFF_MODELS = {
      'school': 'app_folder_name',
      'student': 'app_folder_name'
}
CRUDSTUFF_FORMS = {
        'school': 'SchoolForm',
        'student': 'StudentsForm'
}
```

Projects Configuration

modify your PROJECTS (see `demo_project` folder) urls.py file to include the app's `urls.py` file. See sample:

```
from django.conf.urls import patterns, include, url
from django.contrib.auth.views import login,logout
# Uncomment the next two lines to enable the admin:
# from django.contrib import admin
# admin.autodiscover()
from settings import PROJECT_ROOTDIR
urlpatterns = patterns('',
    # Examples:
    url(r'^crudstuff/', include('crudstuff.urls')),


    url(r'^login/', login),
    url(r'^logout/', logout, {'template_name': 'registration/login.html'}),
    url(r'^accounts/login', login),
    url(r'media/(?P<path>.*)$', 'django.views.static.serve', {'document_root':'%s/static/' % (PROJECT

    # Uncomment the admin/doc line below to enable admin documentation:
    # url(r'^admin/doc/', include('django.contrib.admindocs.urls')),

    # Uncomment the next line to enable the admin:
    #url(r'^admin/', include(admin.site.urls)),
)
```

Some notes if you're copying from the project settings file

- modify `demo_project/settings.py` (or your custom projects), replace "demo_project" with your project name in `ROOT_URLCONF` and `WSGI_APPLICATION` variables, and update any other settings you would like here.
- modify `demo_project/wsgi.py` and replace demo_project with your project name (if using demo_project)
- add any other custom settings you might add to a project and enjoy!

CHAPTER 3

Usage

Using this application. Add the following to your URLS.py file:: This application is made up of three components: Main-Project, Main reusable-app, Other-reusable-apps

CHAPTER 4

Main-Project

The main project will be your django-project (see demo_project in github repo) This project will include one URL entry, which is called the "root-url" string This will tell it that any regex caught after that URL prefix, will be processed via the main reusable app's controller.

CHAPTER 5

Main-reusable-app

The main reusable app is the application that will handle the main crud events (add, write, edit, delete) Any other reusable apps that are built can be used just for their model and their internal API urls. If you decide to use internal api urls for the “other” reusable apps (which will be utilzed by the main app) then you will need to add those url entries into your PROJECTS urls.py configuration. (api/other-reusable/show-users)

CHAPTER 6

Other-reusable-apps

Other reusable apps will be used for their model definitions, custom forms, or API views. They will have to be coded into the main-reusable-app views, with a try/except import. If those imports are found, the data for the models and forms will then be added to the main `data` dictionary for the controller.

CHAPTER 7

NOTE ON Overwriting Templates

By default, it comes with a clean set of templates with Twitter Bootstrap. If you want to override the templates with a custom dashboard template you can refer to the applications sample temptaes to see what data you have to play with and implement your own designs. This can be done similar to overriding django’s login template. You can just create a folder named “crudstuff” in your TEMPLATE_DIR location and overwrite the filenames listed in the project repository

CHAPTER 8

ChangeLog for crudstuff

8.1 0.2.0

- Restructure and Implemented context processor based view
- Using bindmodels class to bind models, apps, to their forms
- Bindmodels read from settings for pluggable app
- Version bump

8.2 0.1.2

- implemented dynamic app to project
- Added main-index ui view with dynamic app
- modified urls structure to support crud functions
- added fixed-top navigation

8.3 0.1.1

- implemented login page
- index view of customer data

8.4 0.1

- initial release

CHAPTER 9

Indices and tables

- genindex
- modindex
- search