Django-EpicEd Documentation

Release 0.4.2

Belug

Mar 08, 2018

Contents

EpicEd is a markdown field with the default widjet using EpicEditor.js http://epiceditor.com/

This plugin is made to keep a much as possible the possibilities to overide EpicEditor.js configurations the only settings you'll not be able to edit are 'container' and 'textarea' since they are rewriten by the wideget and are needed to save the data to the database.

Contents

1.1 About

1.1.1 Installation

To install with pip

```
pip install django-epiceditor
```

To install from source

```
git clone https://github.com/belug23/django-epiced.git
cd django-epiced
python setup.py install
```

From source:

```
wget https://github.com/belug23/django-epiced/archive/0.4.3.tar.gz
tar -zxvf 0.4.3.tar.gz
cd django-epiced
python setup.py install
```

1.1.2 Usage

Add it to your installed apps:

```
INSTALLED_APPS = (
    ...
    'epiced',
)
```

and then use it to your model:

```python
from epiced.models import EpicEditorField

class Post(models.Model):
    ...
    article = EpicEditorField()
    ...
```

by default the field will escape all the HTML saved, to use it with HTML you need to add a safe_mode parameter:

```python
from epiced.models import EpicEditorField

class Post(models.Model):
    article = EpicEditorField(safe_mode=False)
```

for more information look at http://pythonhosted.org/Markdown/reference.html#safe_mode

1.2 Configurations

To know the option please visite http://epiceditor.com/#options

There is only two options you can't edit

container:

```
It's setted to the ID of the field so you can use it on multiple fields in the same
→time.
```

textarea:

```
It's used with the ID to synchronize automatically multipls fields
```

1.2.1 Default Settings

Those are the settings used by default:

```python
DEFAULT_EPICEDITOR_CONFIG = {
    "basePath": settings.STATIC_URL + 'epiced',
    "clientSideStorage": False,
    "localStorageName": 'epiceditor',
    "useNativeFullscreen": True,
    "parser": 'marked',
    "file": {
        "name": 'epiceditor',
        "defaultContent": '',
        "autoSave": 100
    },
    "theme": {
        "base": '/themes/base/epiceditor.css',
        "preview": '/themes/preview/preview-dark.css',
        "editor": '/themes/editor/epic-dark.css'
    },
    "button": {
        "preview": True,
        "fullscreen": True
    },
```

```
    "focusOnLoad": True,
    "shortcut": {
        "modifier": 18,
        "fullscreen": 70,
        "preview": 80
    },
    "string": {
        "togglePreview": 'Toggle Preview Mode',
        "toggleEdit": 'Toggle Edit Mode',
        "toggleFullscreen": 'Enter Fullscreen'
    }
}
```

1.2.2 Overwriting the Settings

You can partially overwrite the config by adding EPICEDITOR_CONFIGS your settings.py:

```
EPICEDITOR_CONFIG = {
    "basePath": settings.STATIC_URL + 'personal',
    "button": {
        "preview": True,
        "fullscreen": True
    },
}
```

You can also do it by using adding a "configs" parameter to your widget or EpicEditorField:

```
text = EpicEditorField(configs={"focusOnLoad": False})


content = forms.CharField(widget=EpicEditorWidget(configs={"focusOnLoad": False}))
```

1.3 Widget Usage

Insted of using the EpicEditorField you can go the hard way by using TextField and applying the widget directly

```
from django import forms
from epiced.widgets import EpicEditorWidget


class FlatPageForm(forms.ModelForm):
    ...
    content = forms.CharField(widget=EpicEditorWidget())
    ...
```

You don't forget that you can overwrite the configs directly from the widget

```
from django import forms
from epiced.widgets import EpicEditorWidget


class FlatPageForm(forms.ModelForm):
    ...
    content = forms.CharField(widget=EpicEditorWidget(configs={"focusOnLoad": False}))
    ...
```

you also need to add the JS:

```
<script src="{% static 'epiced/js/epiceditor.js' %}"></script>
```