
django-embed-video Documentation

Release 0.8.stable

Juda Kaleta

February 22, 2014

Django app for easy embedding YouTube and Vimeo videos and music from SoundCloud.

Repository is located on GitHub: <https://github.com/yetty/django-embed-video>

Installation & Setup

1.1 Installation

The simplest way is to use `pip` to install package:

```
pip install django-embed-video
```

If you want latest version, you may use `Git`. It is fresh, but unstable.

```
pip install git+https://github.com/yetty/django-embed-video.git
```

1.2 Setup

Add `embed_video` to `INSTALLED_APPS` in your Django settings.

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

To detect HTTP/S you must use `request` context processor:

```
TEMPLATE_CONTEXT_PROCESSORS = (  
    ...  
    'django.core.context_processors.request',  
)
```

Examples

2.1 Template examples

First you have to load the `embed_video_tags` template tags in your template:

```
{% load embed_video_tags %}
```

Embedding of video:

```
{# you can just embed #}
{% video item.video 'small' %}

{# or use variables (with embedding, too) #}
{% video item.video as my_video %}
    URL: {{ my_video.url }}
    Thumbnail: {{ my_video.thumbnail }}
    Backend: {{ my_video.backend }}
    {% video my_video 'small' %}
{% endvideo %}
```

Default sizes are `tiny` (420x315), `small` (480x360), `medium` (640x480), `large` (960x720) and `huge` (1280x960). You can set your own size:

```
{% video my_video '800x600' %}
```

Tip: We recommend to use `sorl-thumbnail` to [change](#) thumbnail size.

Tip: To speed up your pages, consider [template fragment caching](#).

2.2 Model examples

Using the `EmbedVideoField` provides you validation of URLs.

```
from django.db import models
from embed_video.fields import EmbedVideoField

class Item(models.Model):
    video = EmbedVideoField() # same like models.URLField()
```

2.3 Admin mixin examples

Use `AdminVideoMixin` in `admin.py`.

```
from django.contrib import admin
from embed_video.admin import AdminVideoMixin
from .models import MyModel

class MyModelAdmin(AdminVideoMixin, admin.ModelAdmin):
    pass

admin.site.register(MyModel, MyModelAdmin)
```

2.4 Custom backends

If you have specific needs and default backends don't suits you, you can write your custom backend.

`my_project/my_app/backends.py`:

```
from embed_video.backends import VideoBackend

class CustomBackend(VideoBackend):
    re_detect = re.compile(r'http://myvideo\.com/[0-9]+')
    re_code = re.compile(r'http://myvideo\.com/(?P<code>[0-9]+)')

    allow_https = False
    pattern_url = '{protocol}://play.myvideo.com/c/{code}/'
    pattern_thumbnail_url = '{protocol}://thumb.myvideo.com/c/{code}/'
```

You can also overwrite `VideoBackend` methods, if using regular expressions isn't well enough.

`my_project/my_project/settings.py`:

```
MBED_VIDEO_BACKENDS = (
    'embed_video.backends.YoutubeBackend',
    'embed_video.backends.VimeoBackend',
    'embed_video.backends.SoundCloudBackend',
    'my_app.backends.CustomBackend',
)
```

2.5 Low level API examples

You can get instance of `VideoBackend` in your python code thanks to `detect_backend()`:

```
from embed_video.backends import detect_backend

my_video = detect_backend('http://www.youtube.com/watch?v=H4tAOexHdR4')
```

Example project

For easy start with using django-embed-video, you can take a look at example project. It is located in `example_project` directory in root of repository.

3.1 Running example project

1. Install Django and PyYAML:

```
pip install Django
pip install pyyaml
```

2. Create database:

```
python manage.py syncdb --noinput
```

3. Run testing server:

```
python manage.py runserver
```

4. Take a look at <http://localhost:8000> . You can log in to administration with username `admin` and password `admin`.

3.2 Testing HTTPS

To test HTTPS on development server, [follow this instructions](#).

Development

4.1 Contributing

I will be really pleased if you will provide patch to this Django app. Feel free to change whatever, but keep [PEP8](#) rules and [Zen](#).

It is a good habit to cover your patches with *tests*.

Repository is hosted on Github: <https://github.com/yetty/django-embed-video>

4.2 Testing

4.2.1 Requirements

The library needs Django and requests and nose, mock, south and textfixtures libraries to run tests.

```
pip install Django
pip install requests
pip install nose
pip install mock
pip install south
pip install textfixtures
```

4.2.2 Running tests

Run tests with this command:

```
nosetests
```

Be sure to run it before each commit and fix broken tests.

Run tests with coverage:

```
pip install coverage
nosetests --with-coverage --cover-package=embed_video
```

4.3 Changelog

4.3.1 Release 0.8 (dev)

- Add `EMBED_VIDEO_TIMEOUT` to settings.
- Fix rendering template tag if no url is provided (#18)
- If `EMBED_VIDEO_TIMEOUT` timeout is reached in templates, no exception is raised, error is just logged.
- Fix default size in template tag. (See [more...](#))

4.3.2 Release 0.7 (Dec. 21, 2013)

- Support for sites running on HTTPS
- `embed` filter is deprecated and replaced by `video` filter.
- caching for whole backends was removed and replaced by caching properties
- minor improvements on example project (fixtures, urls)

4.3.3 Release 0.6 (Oct. 04, 2013)

- Ability to overwrite embed code of backend
- Caching backends properties
- PyPy compatibility
- Admin video mixin and video widget

4.3.4 Release 0.5 (Sep. 03, 2013)

- Added Vimeo thumbnails support
- Added caching of results
- Added example project
- Fixed template tag embed
- Fixed raising `UnknownIdException` in YouTube detecting.

4.3.5 Release 0.4 (Aug. 22, 2013)

- Documentation was rewritten and moved to <http://django-embed-video.rtd.org/>.
- Custom backends (<http://django-embed-video.rtd.org/en/latest/examples.html#custom-backends>).
- Improved YouTube and Vimeo regex.
- Support for Python 3.
- Renamed `base` to `backends`.

4.3.6 Release 0.3 (Aug. 20, 2013)

- Security fix: faked urls are treated as invalid. See [this page](#) for more details.
- Fixes:
 - allow of empty video field.
 - requirements in setup.py
- Added simpler way to embed video in one-line template tag:

```
{{ 'http://www.youtube.com/watch?v=guXyvo2FfLs' | embed:'large' }}
```

- backend variable in video template tag.

Usage:

```
{% video item.video as my_video %}
  Backend: {{ my_video.backend }}
{% endvideo %}
```

4.3.7 Release 0.2 (June 25, 2013)

- Support of SoundCloud

4.3.8 Release 0.1 (June 1, 2013)

- Initial release

4.4 TODOs list

Todo

Django 1.6 provides better parent for this widget - `django.forms.URLInput`.

(The *original entry* is located in `/var/build/user_builds/django-embed-video/checkouts/v0.8/embed_video/admin.py:docstring of embed_video.admin.AdminVideoWidget, line 5`.)

Websites using django-embed-video

- [Tchorici.cz](#)

Are you using django-embed-video? Send pull request!

6.1 API reference

6.1.1 Admin

class `embed_video.admin.AdminVideoMixin`

Mixin using `AdminVideoWidget` for fields with `EmbedVideoField`.

Usage:

```
from django.contrib import admin
from embed_video.admin import AdminVideoMixin
from .models import MyModel

class MyModelAdmin(AdminVideoMixin, admin.ModelAdmin):
    pass

admin.site.register(MyModel, MyModelAdmin)
```

class `embed_video.admin.AdminVideoWidget` (*attrs=None*)

Widget for video input in administration. If empty it works just like `django.forms.TextInput`. Otherwise it renders embedded video together with input field.

Todo

Django 1.6 provides better parent for this widget - `django.forms.URLInput`.

6.1.2 Backends

class `embed_video.backends.SoundCloudBackend` (*url, is_secure=False*)

Backend for SoundCloud URLs.

class `embed_video.backends.VideoBackend` (*url, is_secure=False*)

Base backend, good to inherit.

allow_https = True

Sets if HTTPS version allowed for specific backend.

get_embed_code (*width, height*)

Returns embed code.

get_thumbnail_url()

Returns thumbnail URL folded from `pattern_thumbnail_url` and parsed code.

get_url()

Returns URL folded from `pattern_url` and parsed code.

classmethod is_valid(url)

Class method to control if passed url is valid for current backend. By default it is done by `re_detect` regex.

pattern_thumbnail_url = None

Pattern in which the code is inserted to get thumbnail url.

Example: `http://static.myvideo.com/thumbs/%s`

pattern_url = None

Pattern in which the code is inserted.

Example: `http://myvideo.com?code=%s`

re_code = None

Compiled regex (`re.compile()`) to search code in URL.

Example: `re.compile(r'myvideo\.com/\?code=(?P<code>\w+)')`

re_detect = None

Compile regex (`re.compile()`) to detect, if input URL is valid for current backend.

Example: `re.compile(r'^http://myvideo\.com/.*)`

class embed_video.backends.VimeoBackend(url, is_secure=False)

Backend for Vimeo URLs.

class embed_video.backends.YoutubeBackend(url, is_secure=False)

Backend for YouTube URLs.

embed_video.backends.detect_backend(url)

Detect the right backend for given URL.

6.1.3 Fields

class embed_video.fields.EmbedVideoField(verbose_name=None, name=None, **kwargs)

Model field for embeded video. Descendant of `django.db.models.URLField`.

class embed_video.fields.EmbedVideoFormField(max_length=None, min_length=None, *args, **kwargs)

Form field for embeded video. Descendant of `django.forms.URLField`

6.1.4 Settings

EMBED_VIDEO_BACKENDS

List of backends to use.

Default:

```
EMBED_VIDEO_BACKENDS = (
    'embed_video.backends.YoutubeBackend',
    'embed_video.backends.VimeoBackend',
```

```
'embed_video.backends.SoundCloudBackend',
)
```

EMBED_VIDEO_TIMEOUT

Sets timeout for GET requests to remote servers.

Default: 10

6.1.5 Template tags

You have to load template tag library first.

```
{% load embed_video_tags %}
```

class `embed_video.templatetags.embed_video_tags.VideoNode` (*parser, token*)
 Template tag video. It gives access to all VideoBackend variables.

Usage (shortcut):

```
{% video URL [SIZE] %}
```

Or as a block:

```
{% video URL as VAR %}
...
{% endvideo %}
```

Example:

```
{% video item.video "large" %}
{% video item.video "340x200" %}

{% video item.video as my_video %}
  URL: {{ my_video.url }}
  Thumbnail: {{ my_video.thumbnail }}
  Backend: {{ my_video.backend }}
{% endvideo %}
```

static `embed(url, size, context=None)`

Direct render of embed video.

static `get_size(value)`

Predefined sizes:

size	width	height
tiny	420	315
small	480	360
medium	640	480
large	960	720
huge	1280	960

You can also use custom size - in format WIDTHxHEIGHT (eg. 500x400).

```
embed_video.templatetags.embed_video_tags.embed(backend, size='small')
```

Warning: Deprecated since version 0.7: Use `VideoNode.embed()` instead.

Same like `VideoNode.embed()` tag but **always uses in secure HTTP protocol**.

Usage:

```
{{ URL|embed:SIZE }}
```

Example:

```
{{ 'http://www.youtube.com/watch?v=guXyvo2FfLs'|embed:'large' }}
```

6.1.6 Utils

`embed_video.utils.import_by_path(dotted_path, error_prefix='')`

Import a dotted module path and return the attribute/class designated by the last name in the path. Raise `ImportError` if something goes wrong.

Warning: Deprecated since version Django: 1.6

Function `django.utils.module_loading.import_by_path()` has been added in Django 1.6.

Indices and tables

- *genindex*
- *modindex*
- *search*

e

embed_video.admin, ??
embed_video.backends, ??
embed_video.fields, ??
embed_video.templatetags.embed_video_tags,
 ??
embed_video.utils, ??