
django-embed-video Documentation

Release 1.4.5

Juda Kaleta

Sep 01, 2022

Contents

1	Contents	3
1.1	Installation & Setup	3
1.2	Examples	4
1.3	Example project	6
1.4	Development	6
1.5	Websites using django-embed-video	11
1.6	Changes	12
2	Library API	17
2.1	API reference	17
3	Indices and tables	25
	Python Module Index	27
	Index	29

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

Repository is located on GitHub:

<https://github.com/jazzband/django-embed-video>

1.1 Installation & Setup

1.1.1 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/jazzband/django-embed-video
```

1.1.2 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',  
)
```

1.2 Examples

1.2.1 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' %}
```

This usage has been added in version 0.7.

And use relative percentual size:

```
{% video my_video '100% x 50%' %}
```

It is possible to set backend options via parameters in template tag. It is useful for example to enforce HTTPS protocol or set different query appended to url.

```
{% video my_video query="rel=0&wmode=transparent" is_secure=True as my_video %}
    {{ my_video.url }} {# always with https #}
{% endvideo %}
```

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

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

Tip: You can overwrite default template of embed code located in `templates/embed_video/embed_code.html` or set own file for custom backend (`template_name`).

`template_name` has been added in version 0.9.

1.2.2 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()
```

1.2.3 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)
```

1.2.4 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}/'

    template_name = 'embed_video/custombackend_embed_code.html' # added in v0.9
```

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',
)
```

1.2.5 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')
```

1.3 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.

1.3.1 Running example project

1. Install requirements:

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

2. Create database:

```
python manage.py migrate
```

3. Load sample data:

```
python manage.py loaddata initial_data
```

4. Run testing server:

```
python manage.py runserver
```

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

1.3.2 Testing HTTPS

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

1.4 Development

1.4.1 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/jazzband/django-embed-video>

1.4.2 Testing

Requirements

The library needs Django and requests and nose, mock, south and testfixtures libraries to run tests. They will be installed automatically when running tests via setup.py

Running tests

Run tests with this command:

```
python setup.py nosetests
```

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

Run tests with coverage:

```
python setup.py nosetests --with-coverage --cover-package=embed_video
```

1.4.3 Changelog

1.4.4 Changes

Release 1.4.5. (September 1, 2022)

- Add support for Django 4.1

Release 1.4.4 (April 18, 2022)

- Add EMBED_VIDEO_YOUTUBE_CHECK_THUMBNAIL to settings.
- Drop support for EOL Django 2.2.

Release 1.4.3 (April 8, 2022)

- Add support for YouTube short URLs.

Release 1.4.2 (March 17, 2022)

- Add Polish translation

Release 1.4.1 (January 6, 2022)

- Add Code of Conduct to repository
- Add support for Python 3.10
- Add support for Django 3.2 and Django 4.0
- Drop support for EOL Python 3.6
- Drop support for EOL Django 3.0 and EOL Django 3.1

- Remove Nose from dependencies due to version incompatibility

Release 1.4.0 (December 18, 2020)

- Add lazy loading for video template tags.
- Handle Vimeo admin “manage” URLs
- Migrate from Travis CI to GitHub Actions.
- Drop unsupported Django versions prior to 2.2.
- Add support for Python up to 3.9.
- Add support for Django up to 3.1.
- Improve code formatting.

Release 1.3.3 (June 10, 2020)

- Fix admin UI exception with form validation.

Release 1.3.2 (March 29, 2020)

- Add support for Soundcloud mobile URLs.

Release 1.3.1 (January 6, 2020)

- Add support for Vimeo review video URLs.
- Update example project code.

Release 1.3 (August 30, 2019)

- Drop unsupported Python version prior to 3.6.
- Drop unsupported Django versions prior to 1.11.
- Add support for Python up to 3.8 including PyPy.
- Add support for Django up to 2.2.
- Improve project structure, docs and language.
- Improve CI and CD infrastructure and automation.
- Move project to Jazzband organization.

Release 1.2.0 (October 04, 2018)

- hotfix docs directive (#94)
- update docs (#92)
- use tests_require and setup_requires for nose testing requirements (#91)
- add renderer kwarg to Widget render method to support Python 2.1 and later (#88)

- enable default HTTPS support for YouTube, VimeoBackend, SoundCloudBackend (#86)
- added syntax highlight in README.rst (#81)
- updating requests >=2.19

Release 1.1.2 (April 27, 2017)

- fix pypi

Release 1.1.1 (March 24, 2017)

- updates for Django 1.10 and 1.11 (#73)
- update requirements for installation of the example project (#72)
- use secure connection to query soundcloud endpoint (#68)

Release 1.1.0 (Jan 19, 2016)

- added support for Django 1.9 (#52)
- if possible YouTube thumbnails are returned in better resolution (#43)

Release 1.0.0 (May 01, 2015)

Backward incompatible changes:

- filter `embed_video_tags.embed` has been removed
- changed behaviour of extra params in video tag (#34, #36)

Backward compatible changes:

- added support for Django 1.7 and Django 1.8
- added support for Vimeo channels (#47)
- fix resizing of SoundCloud iframe (#41)

Release 0.11 (July 26, 2014)

- add support for YouTube mobile urls (#27)
- fix passing parameters in calling request library (#28)
- fix validation of urls (#31)

Release 0.10 (May 24, 2014)

- `video` tag accepts kwargs (#20)
- `video` tag will not crash anymore with `None` passed as url (#24)

Release 0.9 (Apr. 04, 2014)

- Add `VideoBackend.template_name` and rendering embed code from file.
- Allow relative sizes in template tag (#19).
- Fix handling invalid urls of SoundCloud. (#21).
- Catch `VideoDoesNotExistException` and `UnknownBackendException` in template tags and admin widget.
- Add base exception `EmbedVideoException`.

Release 0.8 (Feb. 22, 2014)

- 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...](#))

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)

Release 0.6 (Oct. 04, 2013)

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

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.

Release 0.4 (Aug. 22, 2013)

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

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 %}
```

Release 0.2 (June 25, 2013)

- Support of SoundCloud

Release 0.1 (June 1, 2013)

- Initial release

1.5 Websites using django-embed-video

- [Al.ta Cucina](#)
- [Bici.news](#)
- [Outdoor Passion](#)
- [Snow Passion](#)
- [Running Passion](#)

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

1.6 Changes

1.6.1 Release 1.4.5. (September 1, 2022)

- Add support for Django 4.1

1.6.2 Release 1.4.4 (April 18, 2022)

- Add `EMBED_VIDEO_YOUTUBE_CHECK_THUMBNAIL` to settings.
- Drop support for EOL Django 2.2.

1.6.3 Release 1.4.3 (April 8, 2022)

- Add support for YouTube short URLs.

1.6.4 Release 1.4.2 (March 17, 2022)

- Add Polish translation

1.6.5 Release 1.4.1 (January 6, 2022)

- Add Code of Conduct to repository
- Add support for Python 3.10
- Add support for Django 3.2 and Django 4.0
- Drop support for EOL Python 3.6
- Drop support for EOL Django 3.0 and EOL Django 3.1
- Remove Nose from dependencies due to version incompatibility

1.6.6 Release 1.4.0 (December 18, 2020)

- Add lazy loading for video template tags.
- Handle Vimeo admin “manage” URLs
- Migrate from Travis CI to GitHub Actions.
- Drop unsupported Django versions prior to 2.2.
- Add support for Python up to 3.9.
- Add support for Django up to 3.1.
- Improve code formatting.

1.6.7 Release 1.3.3 (June 10, 2020)

- Fix admin UI exception with form validation.

1.6.8 Release 1.3.2 (March 29, 2020)

- Add support for Soundcloud mobile URLs.

1.6.9 Release 1.3.1 (January 6, 2020)

- Add support for Vimeo review video URLs.
- Update example project code.

1.6.10 Release 1.3 (August 30, 2019)

- Drop unsupported Python version prior to 3.6.
- Drop unsupported Django versions prior to 1.11.
- Add support for Python up to 3.8 including PyPy.
- Add support for Django up to 2.2.
- Improve project structure, docs and language.
- Improve CI and CD infrastructure and automation.
- Move project to Jazzband organization.

1.6.11 Release 1.2.0 (October 04, 2018)

- hotfix docs directive (#94)
- update docs (#92)
- use tests_require and setup_requires for nose testing requirements (#91)
- add renderer kwargs to Widget render method to support Python 2.1 and later (#88)
- enable default HTTPS support for YouTube, VimeoBackend, SoundCloudBackend (#86)
- added syntax highlight in README.rst (#81)
- updating requests >=2.19

1.6.12 Release 1.1.2 (April 27, 2017)

- fix pypi

1.6.13 Release 1.1.1 (March 24, 2017)

- updates for Django 1.10 and 1.11 (#73)
- update requirements for installation of the example project (#72)
- use secure connection to query soundcloud endpoint (#68)

1.6.14 Release 1.1.0 (Jan 19, 2016)

- added support for Django 1.9 (#52)
- if possible YouTube thumbnails are returned in better resolution (#43)

1.6.15 Release 1.0.0 (May 01, 2015)

Backward incompatible changes:

- filter `embed_video_tags.embed` has been removed
- changed behaviour of extra params in video tag (#34, #36)

Backward compatible changes:

- added support for Django 1.7 and Django 1.8
- added support for Vimeo channels (#47)
- fix resizing of SoundCloud iframe (#41)

1.6.16 Release 0.11 (July 26, 2014)

- add support for YouTube mobile urls (#27)
- fix passing parameters in calling request library (#28)
- fix validation of urls (#31)

1.6.17 Release 0.10 (May 24, 2014)

- `video` tag accepts kwargs (#20)
- `video` tag will not crash anymore with `None` passed as url (#24)

1.6.18 Release 0.9 (Apr. 04, 2014)

- Add `VideoBackend.template_name` and rendering embed code from file.
- Allow relative sizes in template tag (#19).
- Fix handling invalid urls of SoundCloud. (#21).
- Catch `VideoDoesNotExistException` and `UnknownBackendException` in template tags and admin widget.
- Add base exception `EmbedVideoException`.

1.6.19 Release 0.8 (Feb. 22, 2014)

- 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...](#))

1.6.20 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)

1.6.21 Release 0.6 (Oct. 04, 2013)

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

1.6.22 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.

1.6.23 Release 0.4 (Aug. 22, 2013)

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

1.6.24 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 %}
```

1.6.25 Release 0.2 (June 25, 2013)

- Support of SoundCloud

1.6.26 Release 0.1 (June 1, 2013)

- Initial release

2.1 API reference

2.1.1 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)
```

formfield_for_dbfield (*db_field*, ***kwargs*)

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`.

render (*name*, *value=""*, *attrs=None*, *size=(420, 315)*, *renderer=None*)

2.1.2 Backends

exception `embed_video.backends.EmbedVideoException`
 Parental class for all `embed_video` exceptions

class `embed_video.backends.SoundCloudBackend(url)`
 Backend for SoundCloud URLs.

get_code()
 Returns video code matched from given url by `re_code`.

Return type `str`

get_embed_code(width, height)
 Returns embed code rendered from template `template_name`.

Return type `str`

get_info()
Return type `dict`

get_thumbnail_url()
 Returns thumbnail URL folded from `pattern_thumbnail_url` and parsed code.

Return type `str`

get_url()
 Returns URL folded from `pattern_url` and parsed code.

height
Return type `str`

is_secure = True
 Decides if secured protocol (HTTPS) is used.

Type `bool`

width
Return type `str`

exception `embed_video.backends.UnknownBackendException`
 Exception thrown if video backend is not recognized.

exception `embed_video.backends.UnknownIdException`
 Exception thrown if backend is detected, but video ID cannot be parsed.

class `embed_video.backends.VideoBackend(url)`
 Base class used as parental class for backends.

Backend variables:

<code>url</code>	URL of video.
<code>code</code>	Code of video.
<code>thumbnail</code>	URL of video thumbnail.
<code>query</code>	String transformed to QueryDict appended to url.
<code>info</code>	Additional information about video.
<code>is_secure</code>	Decides if secured protocol (HTTPS) is used.
<code>protocol</code>	Protocol used to generate URL.

Continued on next page

Table 1 – continued from previous page

<i>template_name</i>	Name of embed code template used by <i>get_embed_code()</i> .
<pre>class MyBackend(VideoBackend): ...</pre>	
allow_https = True	Sets if HTTPS version allowed for specific backend.
Type <code>bool</code>	
code	Code of video.
default_query = ''	Default query string or <i>QueryDict</i> appended to url
Type <code>str</code>	
get_code()	Returns video code matched from given url by <i>re_code</i> .
Return type <code>str</code>	
get_embed_code(<i>width</i>, <i>height</i>)	Returns embed code rendered from template <i>template_name</i> .
Return type <code>str</code>	
get_info()	
Return type <code>dict</code>	
get_thumbnail_url()	Returns thumbnail URL folded from <i>pattern_thumbnail_url</i> and parsed code.
Return type <code>str</code>	
get_url()	Returns URL folded from <i>pattern_url</i> and parsed code.
info	Additional information about video. Not implemented in all backends.
is_secure = False	Decides if secured protocol (HTTPS) is used.
Type <code>bool</code>	
classmethod is_valid(<i>url</i>)	Class method to control if passed url is valid for current backend. By default it is done by <i>re_detect</i> regex.
pattern_thumbnail_url = None	Pattern in which the code is inserted to get thumbnail url.
	Example: <code>http://static.myvideo.com/thumbs/%s</code>
Type <code>str</code>	

pattern_url = None

Pattern in which the code is inserted.

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

Type `str`

protocol

Protocol used to generate URL.

query

String transformed to QueryDict appended to url.

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

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

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

set_options (options)

template_name = 'embed_video/embed_code.html'

Name of embed code template used by `get_embed_code()`.

Passed template variables: `{{ backend }}` (instance of `VideoBackend`), `{{ width }}`, `{{ height }}`

Type `str`

thumbnail

URL of video thumbnail.

url

URL of video.

exception embed_video.backends.VideoDoesntExistException

Exception thrown if video doesn't exist

class embed_video.backends.VimeoBackend (url)

Backend for Vimeo URLs.

get_info ()

Return type `dict`

get_thumbnail_url ()

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

Return type `str`

is_secure = True

Decides if secured protocol (HTTPS) is used.

Type `bool`

class embed_video.backends.YouTubeBackend (url)

Backend for YouTube URLs.

get_code ()

Returns video code matched from given url by `re_code`.

Return type `str`

get_thumbnail_url()

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

Return type `str`

is_secure = True

Decides if secured protocol (HTTPS) is used.

Type `bool`

`embed_video.backends.detect_backend(url)`

Detect the right backend for given URL.

Goes over backends in `settings.EMBED_VIDEO_BACKENDS`, calls `is_valid()` and returns backend instance.

Parameters `url (str)` – URL which is passed to `is_valid` methods of VideoBackends.

Returns Returns recognized VideoBackend

Return type `VideoBackend`

2.1.3 Fields

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

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

formfield (`**kwargs`)

Return a `django.forms.Field` instance for this field.

class `embed_video.fields.EmbedVideoFormField(**kwargs)`

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

2.1.4 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

EMBED_VIDEO_YOUTUBE_DEFAULT_QUERY

Sets default *query* appended to YouTube url. Can be string or `QueryDict` instance.

Default: "wmode=opaque"

EMBED_VIDEO_YOUTUBE_CHECK_THUMBNAIL

Sets whether to check thumbnail for YouTube. If `False`, it uses high resolution as it's guaranteed to exist.

Default: `True`

2.1.5 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] [key1=value1, key2=value2...] %}
```

Or as a block:

```
{% video URL [SIZE] [key1=value1, key2=value2...] as VAR %}
...
{% endvideo %}
```

Examples:

```
{% video item.video %}
{% video item.video "large" %}
{% video item.video "340x200" %}
{% video item.video "100% x 300" query="rel=0&wmode=opaque" %}

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

classmethod `embed` (*url, size, context=None, **options*)

Direct render of embed video.

Parameters

- **url** (*str*) – URL to embed video
- **size** (*str*) – Size of rendered block
- **context** (*django.template.RequestContext* | *None*) – Django template `RequestContext`

static `get_backend(backend_or_url, context=None, **options)`

Returns instance of VideoBackend. If context is passed to the method and request is secure, than the `is_secure` mark is set to backend.

A string or VideoBackend instance can be passed to the method.

Parameters

- **backend** – Given instance inherited from VideoBackend or url
- **context** (*django.template.RequestContext* | *None*) – Django template RequestContext

Return type *VideoBackend*

classmethod `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).

Returns Returns tuple with (width, height) values.

Return type tuple[int, int]

render (*context*)

Returns generated HTML.

Parameters **context** (*django.template.RequestContext*) – Django template RequestContext

Returns Rendered HTML with embed video.

Return type django.utils.safestring.SafeText | str

render_block (*context, backend*)

Parameters

- **context** (*django.template.RequestContext*) – Django template RequestContext
- **backend** (*VideoBackend*) – Given instance inherited from VideoBackend

Return type django.utils.safestring.SafeText

resolve_options (*context*)

Parameters **context** (*django.template.RequestContext*) – Django template RequestContext

2.1.6 Utils

CHAPTER 3

Indices and tables

- `genindex`
- `modindex`
- `search`

e

`embed_video.admin`, [17](#)
`embed_video.backends`, [18](#)
`embed_video.fields`, [21](#)
`embed_video.templatetags.embed_video_tags`,
[22](#)

A

AdminVideoMixin (class in *embed_video.admin*), 17
AdminVideoWidget (class in *embed_video.admin*), 17
allow_https (*embed_video.backends.VideoBackend* attribute), 19

C

code (*embed_video.backends.VideoBackend* attribute), 19

D

default_query (*embed_video.backends.VideoBackend* attribute), 19
detect_backend() (in module *embed_video.backends*), 21

E

embed() (*embed_video.templatetags.embed_video_tags.VideoNode* class method), 22
embed_video.admin (module), 17
embed_video.backends (module), 18
embed_video.fields (module), 21
embed_video.templatetags.embed_video_tags (module), 22
EMBED_VIDEO_BACKENDS setting, 21
EMBED_VIDEO_TIMEOUT setting, 21
EMBED_VIDEO_YOUTUBE_CHECK_THUMBNAIL setting, 22
EMBED_VIDEO_YOUTUBE_DEFAULT_QUERY setting, 21
EmbedVideoException, 18
EmbedVideoField (class in *embed_video.fields*), 21
EmbedVideoFormField (class in *embed_video.fields*), 21

F

formfield() (*embed_video.fields.EmbedVideoField* method), 21
formfield_for_dbfield() (*embed_video.admin.AdminVideoMixin* method), 17

G

get_backend() (*embed_video.templatetags.embed_video_tags.VideoNode* static method), 22
get_code() (*embed_video.backends.SoundCloudBackend* method), 18
get_code() (*embed_video.backends.VideoBackend* method), 19
get_code() (*embed_video.backends.YoutubeBackend* method), 20
get_embed_code() (*embed_video.backends.SoundCloudBackend* method), 18
get_embed_code() (*embed_video.backends.VideoBackend* method), 19
get_info() (*embed_video.backends.SoundCloudBackend* method), 18
get_info() (*embed_video.backends.VideoBackend* method), 19
get_info() (*embed_video.backends.VimeoBackend* method), 20
get_size() (*embed_video.templatetags.embed_video_tags.VideoNode* class method), 23
get_thumbnail_url() (*embed_video.backends.SoundCloudBackend* method), 18
get_thumbnail_url() (*embed_video.backends.VideoBackend* method), 19
get_thumbnail_url() (*embed_video.backends.VimeoBackend* method),

20
[get_thumbnail_url\(\)](#) (*embed_video.backends.YoutubeBackend* method), 21
[get_url\(\)](#) (*embed_video.backends.SoundCloudBackend* method), 18
[get_url\(\)](#) (*embed_video.backends.VideoBackend* method), 19

H

[height](#) (*embed_video.backends.SoundCloudBackend* attribute), 18

I

[info](#) (*embed_video.backends.VideoBackend* attribute), 19
[is_secure](#) (*embed_video.backends.SoundCloudBackend* attribute), 18
[is_secure](#) (*embed_video.backends.VideoBackend* attribute), 19
[is_secure](#) (*embed_video.backends.VimeoBackend* attribute), 20
[is_secure](#) (*embed_video.backends.YoutubeBackend* attribute), 21
[is_valid\(\)](#) (*embed_video.backends.VideoBackend* class method), 19

P

[pattern_thumbnail_url](#) (*embed_video.backends.VideoBackend* attribute), 19
[pattern_url](#) (*embed_video.backends.VideoBackend* attribute), 19
[protocol](#) (*embed_video.backends.VideoBackend* attribute), 20

Q

[query](#) (*embed_video.backends.VideoBackend* attribute), 20

R

[re_code](#) (*embed_video.backends.VideoBackend* attribute), 20
[re_detect](#) (*embed_video.backends.VideoBackend* attribute), 20
[render\(\)](#) (*embed_video.admin.AdminVideoWidget* method), 17
[render\(\)](#) (*embed_video.templatetags.embed_video_tags.VideoNode* method), 23
[render_block\(\)](#) (*embed_video.templatetags.embed_video_tags.VideoNode* method), 23

[resolve_options\(\)](#) (*embed_video.templatetags.embed_video_tags.VideoNode* method), 23

S

[set_options\(\)](#) (*embed_video.backends.VideoBackend* method), 20
[setting](#)
[EMBED_VIDEO_BACKENDS](#), 21
[EMBED_VIDEO_TIMEOUT](#), 21
[EMBED_VIDEO_YOUTUBE_CHECK_THUMBNAIL](#), 22
[EMBED_VIDEO_YOUTUBE_DEFAULT_QUERY](#), 21
[SoundCloudBackend](#) (class in *embed_video.backends*), 18

T

[template_name](#) (*embed_video.backends.VideoBackend* attribute), 20
[thumbnail](#) (*embed_video.backends.VideoBackend* attribute), 20

U

[UnknownBackendException](#), 18
[UnknownIdException](#), 18
[url](#) (*embed_video.backends.VideoBackend* attribute), 20

V

[VideoBackend](#) (class in *embed_video.backends*), 18
[VideoDoesntExistException](#), 20
[VideoNode](#) (class in *embed_video.templatetags.embed_video_tags*), 22
[VimeoBackend](#) (class in *embed_video.backends*), 20

W

[width](#) (*embed_video.backends.SoundCloudBackend* attribute), 18

Y

[YoutubeBackend](#) (class in *embed_video.backends*), 20