Flask-Genshi Documentation

Release 0.1

Dag Odenhall

September 14, 2011

CONTENTS

Flask-Genshi is an extension to Flask that allows you to easily use Genshi for templating. It knows how to render a template based on the file extension and can create `Response` objects with mimetype set accordingly.

Source code and issue tracking at Bitbucket.

CHAPTER

ONE

INSTALLATION

Just grab it from PyPI with *easy_install* or *pip*, for example:

```
$ easy_install Flask-Genshi
```

If you're starting a new project you don't need to explicitly install Flask as Flask-Genshi depends on it already.

CHAPTER

TWO

HOW TO USE

You need to construct a `Genshi` with your `Flask` instance.

```
from flaskext.genshi import Genshi

app = Flask(__name__)
genshi = Genshi(app)
```

The best way to render templates is to use `render_response()`. This ensures that the proper mimetype is sent if you render XHTML, XML or text, and sets the right doctype for you.

Use it like so:

```
from flaskext.genshi import render_response

@app.route('/')
def index():
    title = 'Genshi + Flask, a match made in heaven!'
    return render_response('index.html', dict(title=title))
```

CHAPTER

THREE

USING METHODS

Methods control things such as the doctype and how end tags are rendered, and with `render_response()` also the mimetype. Unless overridden the method used is decided by the template's filename extension.

By default HTML renders as strict HTML 4.01. This is how you change it to HTML5:

```
genshi.extensions['html'] = 'html5'
```

I recommend against this but of course you can also change it to XHTML:

```
genshi.extensions['html'] = 'xhtml'
```

You can also override the default with a parameter to the templating functions:

```
render_response('video.html', method='html5')
```

The extensions *html*, *xml*, *txt*, *js* and *css* are recognized, but you can add any extension and method you like. Note that *txt*, *js* and *css* templates are rendered with `genshi.template.NewTextTemplate` which is not XML-based. Rendering javascript with templates gives you tools like `flask.url_for()` and rendering CSS with templates gives you dynamic stylesheets with things like variables.

CHAPTER

FOUR

API REFERENCE

class `flaskext.genshi.`**`Genshi`**(*app*)

`extensions`
What method is used for an extension.

`filter`(**methods*)
Decorator that adds a function to apply filters to templates by rendering method.

Example:

```
from genshi.filters import Transformer

@genshi.filter('html')
def prepend_title(template):
    return template | Transformer('head/title').prepend('MySite - ')
```

See the Genshi documentation for more filters you can use. New in version 0.3.

`filters`
Filter functions to be applied to templates. New in version 0.3.

`methods`
Render methods.

`template_loader`
A `genshi.template.TemplateLoader` that loads templates from the same place as Flask.

`flaskext.genshi.`**`select_method`**(*template*, *method=None*)
Selects a method from `Genshi.methods` based on the file extension of `template` and `Genshi.extensions`, or based on `method`.

`flaskext.genshi.`**`generate_template`**(*template*, *context=None*, *method=None*)
Creates a Genshi template stream that you can run filters and transformations on.

`flaskext.genshi.`**`render_template`**(*template*, *context=None*, *method=None*)
Renders a template to a string.

`flaskext.genshi.`**`render_response`**(*template*, *context=None*, *method=None*)
Renders a template and wraps it in a `response_class` with mimetype set according to the rendering method.

PYTHON MODULE INDEX