
Manifesto Documentation

Release 0.5.0

Timothée Peignier

June 11, 2016

1	Table Of Contents	3
1.1	Installation	3
1.2	Configuration	3
1.3	Usage	3
2	Indices and tables	5

Manifesto provides per-app pluggable cache-manifest declaration for Django.

Table Of Contents

1.1 Installation

1. Either check out Manifesto from [GitHub](#) or pull a release off [PyPI](#)

```
pip install manifesto
```

2. Add 'compress' to your INSTALLED_APPS

```
INSTALLED_APPS = (  
    'manifesto',  
)
```

1.2 Configuration

Configuration and list of available settings for Manifesto.

1.2.1 Excluding manifests

You can exclude manifest from the final cache-manifest with `MANIFESTO_EXCLUDED_MANIFESTS`

```
MANIFESTO_EXCLUDED_MANIFESTS = (  
    'randomapp.manifest.WrongManifest',  
)
```

Default: []

1.3 Usage

Describes how to use Manifesto when it is installed and configured.

1.3.1 Creating a manifest

To add a manifest in your application, you need to add a `manifest` module in it. Inside this module you can create any number of manifest as long as they inherit from `Manifest` class.

```
from manifesto import Manifest

class StaticManifest(Manifest):
    def cache(self):
        return [
            '/static/js/application.js',
            '/static/css/screen.css',
        ]

    def network(self):
        return ['*']

    def fallback(self):
        return [
            ('/', '/offline.html'),
        ]
```

Note: The meaning of each method is corresponding to each section of a cache-manifest, see [Offline Web applications documentation](#) for reference.

By default, a revision is calculated for each manifest, if you want to override the way we calculate revision, you just need to add a revision method to your manifest

```
import time
from manifesto import Manifest

class StaticManifest(Manifest):
    def revision(self):
        return int(time.time())
```

1.3.2 Access to the final cache-manifest

Manifesto provides a default ManifestView and urls to link to the final cache-manifest

```
from django.conf.urls.defaults import *
from manifesto.views import ManifestView

urlpatterns = patterns('',
    url(r'^manifest\.appcache$', ManifestView.as_view(), name="cache_manifest"),
)
```

Then from your template, you can link to your cache-manifest

```
<!doctype html>
<html manifest="{% url cache_manifest %}">
<head>
<meta charset="utf-8">
```

Indices and tables

- search