django-amp-tools Documentation

Release latest

February 07, 2017

Contents

django-amp-tools (amp_tools) provides a simple way to work with Accelerated mobile pages (AMP) in django and gives you tools at your hand to render some different templates to deliver an AMP version of your site to the user.

The idea (from django-mobile app) is to keep your views exactly the same but to transparently interchange the templates used to render a response. This is done in two steps:

1. A middleware determines the client's preference to view your site. E.g. if he wants to use the AMP or the standart version.
2. The template loader takes then care of choosing the correct templates based on the GET param detected in the middleware.

The **source code** for this app is hosted at https://github.com/shtalinberg/django-amp-tools.

Contents:

CHAPTER 1

Installation

Pre-Requirements: `django-amp-tools` depends on django's sites framework. So before you try to use `django-amp-tools` make sure that the sites framework is enabled and working.

1. Install `django-amp-tools` with your favourite python tool, e.g. with `easy_install django-amp-tools` or `pip install django-amp-tools`.
2. Add `amp_tools` to your `INSTALLED_APPS` setting in the `settings.py`.
3. Add `amp_tools.middleware.AMPDetectionMiddleware` to end of your `MIDDLEWARE_CLASSES` setting.
4. Add `amp_tools.loader.Loader` as first item to your `loaders` list for `TEMPLATES` setting in `settings.py`.

CHAPTER 2

Usage

2.1 Usage app

The concept of **django-amp-tools** is render *amp* for your site from save view as desktop version.

This makes it possible to provide many possible designs instead of just differentiating between a full desktop experience and one AMP version.

After the correct version is somehow chosen by the middlewares, it's assigned to the `request.is_amp_detect` attribute. You can use this in your views to provide separate logic.

```
<html>
<head>
    <title>My site {% if request.is_amp_detect %}(amp version){% endif %}</title>
</head>
<body>
    ...
</body>
</html>
```

You can create own AMP base.html.

My example base_amp.html

```
<!doctype html>{% load i18n amp_tags %}{% load static from staticfiles %}
<html amp lang="{{ LANGUAGE_CODE }}">
  <head>
    <meta charset="utf-8">
    <script async src="https://cdn.ampproject.org/v0.js"></script>
    <title>BERGSOFT+</title>
    {% block amp-link %}{% amp_canonical_link request %}{% endblock %}
    <meta name="viewport" content="width=device-width,minimum-scale=1,initial-scale=1">
    <style amp-custom>
    {% block amp-custom %}
        {% include "amp/includes/style_amp_custom.css" %}
    {% endblock %}
    </style>
    {% block amp-ld-json %}{% endblock %}
    <style amp-boilerplate>body{-webkit-animation:-amp-start 8s steps(1,end) 0s 1 normal both;-moz-an
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.
  </head>
  <body>
    <header id="#top" class="amp-site-header">
        <div>
```

```
      <a href="{% url 'home_page' %}">
        <amp-img src="{% static "images/v-2/logo.png" %}" width="145" height="35" class="amp-
        {% trans 'Web site development' %}
      </a>
    </div>
  </header>
  {% block content %}
  <h1>{% trans "Bergsoft+ company" %}</h1>
  {% endblock %}

  <footer class="amp-site-footer">
    <div>
      <h2>{% trans 'Web Development' %}</h2>
      <p>
        &copy;&nbsp;{% now "Y" %} {{ site.name }}. {% trans "All rights reserved" %}.
      </p>
      <a href="#top" class="back-to-top">{% trans "Back to top" %}</a>
    </div>
  </footer>
 </body>
</html>
```

In your app template folder (*blog*, for example) create folder *amp*

for blog detail page `blog/tempplates/blog/post_detail.html` create AMP version `blog/tempplates/blog/amp/post_detail.html` with content

```
{% extends "base_amp.html" %}
{% load i18n blog_tags %}
{% load thumbnail staticfiles %}

{% block content %}
<article class="amp-post-article">
  <header class="amp-post-article-header">
    <h1 class="amp-post-title">{{ object.title|safe }}</h1>
    <div class="amp-post-meta amp-post-byline">
      {% if post.author.avatar %}
        <amp-img src="{{ post.author.avatar.url }}" width="24" height="24" layout="fixed"></a
      {% endif %}
      <span class="amp-post-author author vcard"><i class="fa fa-user-o"></i>  {% trans "Poste
    </div>

    <div class="amp-post-meta amp-post-posted-on">
      <i class="fa fa-calendar"></i> {{ post.published_at|date:"Y-m-d" }}
    </div>
  </header>
  {% if object.picture %}
  <figure class="amp-post-article-featured-image amp-caption">
    <amp-img src="{{ object.picture.url }}" width="738" height="430" alt="{{ object.title|stripta
    {#<p class="amp-caption-text">{{ object.picture_caption|striptags }}</p>#}
  </figure>
  {% endif %}

  <div class="amp-post-article-content">
    {{ post.body_html|safe }}
  </div>
  <footer class="amp-post-article-footer">
    <div class="amp-post-meta amp-post-tax-category">
      <i class="fa fa-folder-o"></i>
```

```
            <a href="{{ post.categories.all.0.get_absolute_url }}"> {{ post.categories.all.0 }}</a>
        </div>
        {% if post.tags %}
        <div class="amp-wp-meta amp-wp-tax-tag">
            <i class="fa fa-tags"></i>
            {% for tag in post.tags %}
                <a href="{{ tag.get_absolute_url }}"> {{ tag }}</a>
            {% endfor %}
        </div>
        {% endif %}
    </footer>

</article>
{% endblock %}
```

See result of this code here:: http://bergsoftplus.com/weblog/2016/11/28/internet-bezopasnost-pri-onlajn-pokupkah-20/?amp-content=amp

2.1.1 Settings

Here is a list of settings that are used by **django-amp-tools** and can be changed in your own `settings.py`:

`AMP_TOOLS_GET_PARAMETER` The name of GET parameter which check in `AMPDetectionMiddleware`.

Default: `'amp-content'`

`AMP_TOOLS_GET_VALUE` The value of GET parameter which check in `AMPDetectionMiddleware`. select AMP version.

Default: `'amp'`

`AMP_TOOLS_TEMPLATE_PREFIX` This string will be prefixed to the template names when searching for AMP templates. This is useful if you have many flavours and want to store them in a common subdirectory. Example:

```
from django.template.loader import render_to_string
from amp_tools import set_amp_detect

set_amp_detect(is_amp_detect=True, request=request)
render_to_string('index.html') # will render 'amp/index.html'

# now add this to settings.py
AMP_TOOLS_TEMPLATE_PREFIX = 'amp-new/'

# and try again

set_amp_detect(is_amp_detect=True, request=request)
render_to_string('index.html') # will render 'amp-new/amp/index.html'
```

Default: `''` (empty string)

`AMP_TOOLS_TEMPLATE_LOADERS` **django-mobile**'s template loader can load templates prefixed with the current flavour. Specify with this setting which loaders are used to load flavoured templates.

Default: same as `TEMPLATE_LOADERS` setting but without `'amp_tools.loader.Loader'`.

CHAPTER 3

Contributing

Here are the steps needed to set up a development and testing environment.

WARNING

This app use *git flow* for branching strategy and release management.

Please, change code and submit all pull requests into branch *develop*

Thanks for contributing, and have fun!

CHAPTER 4

Source code and contacts

4.1 Repository and bugs

The **source code** for this app is hosted on https://github.com/shtalinberg/django-amp-tools.

To file **bugs and requests**, please use https://github.com/shtalinberg/django-amp-tools/issues.

4.2 Contacts

Oleksandr Shtalinberg

- Email: `o.shtalinberg at gmail.com`