

## Day 8 - 构建前端

338次阅读

虽然我们跑通了一个最简单的MVC，但是页面效果肯定不会让人满意。

对于复杂的HTML前端页面来说，我们需要一套基础的CSS框架来完成页面布局和基本样式。另外，jQuery作为操作DOM的JavaScript库也必不可少。

从零开始写CSS不如直接从一个已有的功能完善的CSS框架开始。有很多CSS框架可供选择。我们这次选择uikit这个强大的CSS框架。它具备完善的响应式布局，漂亮的UI，以及丰富的HTML组件，让我们能轻松设计出美观而简洁的页面。

可以从uikit首页下载打包的资源文件。

所有的静态资源文件我们统一放到www/static目录下，并按照类别归类：

```
static/
+- css/
|   +- addons/
|   |   +- uikit.addons.min.css
|   |   +- uikit.almost-flat.addons.min.css
|   |   +- uikit.gradient.addons.min.css
|   +- awesome.css
|   +- uikit.almost-flat.addons.min.css
|   +- uikit.gradient.addons.min.css
|   +- uikit.min.css
+- fonts/
|   +- fontawesome-webfont.eot
|   +- fontawesome-webfont.ttf
|   +- fontawesome-webfont.woff
|   +- FontAwesome.otf
+- js/
|   +- awesome.js
|   +- html5.js
|   +- jquery.min.js
|   +- uikit.min.js
```

由于前端页面肯定不止首页一个页面，每个页面都有相同的页眉和页脚。如果每个页面都是独立的HTML模板，那么我们在修改页眉和页脚的时候，就需要把每个模板都改一遍，这显然是没有效率的。

常见的模板引擎已经考虑到了页面上重复的HTML部分的复用问题。有的模板通过include把页面拆成三部分：

```
<html>
  <% include file="inc_header.html" %>
  <% include file="index_body.html" %>
  <% include file="inc_footer.html" %>
</html>
```

这样，相同的部分inc\_header.html和inc\_footer.html就可以共享。

但是include方法不利于页面整体结构的维护。jinja2的模板还有另一种“继承”方式，实现模板的复用更简单。

“继承”模板的方式是通过编写一个“父模板”，在父模板中定义一些可替换的block（块）。然后，编写多个“子模板”，每个子模板都可以只替换父模板定义的block。比如，定义一个最简单的父模板：

```
<!-- base.html -->
<html>
  <head>
    <title>{% block title%} 这里定义了一个名为title的block {% endblock %}</title>
  </head>
  <body>
    {% block content %} 这里定义了一个名为content的block {% endblock %}
  </body>
</html>
```

对于子模板a.html，只需要把父模板的title和content替换掉：

```
{% extends 'base.html' %}

{% block title %} A {% endblock %}

{% block content %}
<h1>Chapter A</h1>
<p>blablabla...</p>
{% endblock %}
```

对于子模板b.html，如法炮制：

```
{% extends 'base.html' %}

{% block title %} B {% endblock %}

{% block content %}
<h1>Chapter B</h1>
```

```
<ul>
  <li>list 1</li>
  <li>list 2</li>
</ul>
{% endblock %}
```

这样，一旦定义好父模板的整体布局和CSS样式，编写子模板就会非常容易。

让我们通过uikit这个CSS框架来完成父模板\_\_base\_\_.html的编写：

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  {% block meta %}<!-- block meta -->{% endblock %}
  <title>{% block title %} ? {% endblock %} - Awesome Python Webapp</title>
  <link rel="stylesheet" href="/static/css/uikit.min.css">
  <link rel="stylesheet" href="/static/css/uikit.gradient.min.css">
  <link rel="stylesheet" href="/static/css/awesome.css" />
  <script src="/static/js/jquery.min.js"></script>
  <script src="/static/js/md5.js"></script>
  <script src="/static/js/uikit.min.js"></script>
  <script src="/static/js/awesome.js"></script>
  {% block beforehead %}<!-- before head -->{% endblock %}
</head>
<body>
  <nav class="uk-navbar uk-navbar-attached uk-margin-bottom">
    <div class="uk-container uk-container-center">
      <a href="/" class="uk-navbar-brand">Awesome</a>
      <ul class="uk-navbar-nav">
        <li data-url="blogs"><a href="/"><i class="uk-icon-home"></i> 日志</a></li>
        <li><a target="_blank" href="#"><i class="uk-icon-book"></i> 教程</a></li>
        <li><a target="_blank" href="#"><i class="uk-icon-code"></i> 源码</a></li>
      </ul>
      <div class="uk-navbar-flip">
        <ul class="uk-navbar-nav">
          {% if user %}
            <li class="uk-parent" data-uk-dropdown>
              <a href="#0"><i class="uk-icon-user"></i> {{ user.name }}</a>
              <div class="uk-dropdown uk-dropdown-nav">
                <ul class="uk-nav uk-nav-navbar">
                  <li><a href="/signout"><i class="uk-icon-sign-out"></i> 登出</a></li>
                </ul>
              </div>
            </li>
          {% else %}
            <li><a href="/signin"><i class="uk-icon-sign-in"></i> 登陆</a></li>
            <li><a href="/register"><i class="uk-icon-edit"></i> 注册</a></li>
          {% endif %}
        </ul>
      </div>
    </div>
  </nav>

  <div class="uk-container uk-container-center">
    <div class="uk-grid">
      <!-- content -->
      {% block content %}
      {% endblock %}
      <!-- // content -->
    </div>
  </div>

  <div class="uk-margin-large-top" style="background-color:#eee; border-top:1px solid #ccc;">
    <div class="uk-container uk-container-center uk-text-center">
      <div class="uk-panel uk-margin-top uk-margin-bottom">
        <p>
          <a target="_blank" href="#" class="uk-icon-button uk-icon-weibo"></a>
          <a target="_blank" href="#" class="uk-icon-button uk-icon-github"></a>
          <a target="_blank" href="#" class="uk-icon-button uk-icon-linkedin-square"></a>
          <a target="_blank" href="#" class="uk-icon-button uk-icon-twitter"></a>
        </p>
        <p>Powered by <a href="#">Awesome Python Webapp</a>. Copyright &copy; 2014. [<a href="/manage/" target="_blank">Manage</a>]</p>
        <p><a href="http://www.liaoxuefeng.com/" target="_blank">www.liaoxuefeng.com</a>. All rights reserved.</p>
        <a target="_blank" href="#"><i class="uk-icon-html5" style="font-size:64px; color: #444;"></i></a>
      </div>
    </div>
  </div>
</body>
</html>
```

\_\_base\_\_.html定义的几个block作用如下：

用于子页面定义一些meta，例如rss feed：

```
{% block meta %} ... {% endblock %}
```

覆盖页面的标题：

```
{% block title %} ... {% endblock %}
```

子页面可以在标签关闭前插入JavaScript代码:

```
{% block beforehead %} ... {% endblock %}
```

子页面的content布局和内容:

```
{% block content %}
...
{% endblock %}
```

我们把首页改造一下, 从\_\_base\_\_.html继承一个blogs.html:

```
{% extends '__base__.html' %}

{% block title %} 日志 {% endblock %}

{% block content %}

<div class="uk-width-medium-3-4">
{% for blog in blogs %}
  <article class="uk-article">
    <h2><a href="/blog/{{ blog.id }}">{{ blog.name }}</a></h2>
    <p class="uk-article-meta">发表于{{ blog.created_at }}</p>
    <p>{{ blog.summary }}</p>
    <p><a href="/blog/{{ blog.id }}">继续阅读 <i class="uk-icon-angle-double-right"></i></a></p>
  </article>
  <hr class="uk-article-divider">
{% endfor %}
</div>

<div class="uk-width-medium-1-4">
  <div class="uk-panel uk-panel-header">
    <h3 class="uk-panel-title">友情链接</h3>
    <ul class="uk-list uk-list-line">
      <li><i class="uk-icon-thumbs-o-up"></i> <a target="_blank" href="#">编程</a></li>
      <li><i class="uk-icon-thumbs-o-up"></i> <a target="_blank" href="#">读书</a></li>
      <li><i class="uk-icon-thumbs-o-up"></i> <a target="_blank" href="#">Python教程</a></li>
      <li><i class="uk-icon-thumbs-o-up"></i> <a target="_blank" href="#">Git教程</a></li>
    </ul>
  </div>
</div>

{% endblock %}
```

相应地, 首页URL的处理函数更新如下:

```
@view('blogs.html')
@get('/')
def index():
    blogs = Blog.find_all()
    # 查找登陆用户:
    user = User.find_first('where email=?', 'admin@example.com')
    return dict(blogs=blogs, user=user)
```

往MySQL的blogs表中手动插入一些数据, 我们就可以看到一个真正的首页了。但是Blog的创建日期显示的是一个浮点数, 因为它是由这段模板渲染出来的:

```
<p class="uk-article-meta">发表于{{ blog.created_at }}</p>
```

解决方法是通过jinja2的filter (过滤器), 把一个浮点数转换成日期字符串。我们来编写一个datetime的filter, 在模板里用法如下:

```
<p class="uk-article-meta">发表于{{ blog.created_at|datetime }}</p>
```

filter需要在初始化jinja2时设置。修改wsgiapp.py相关代码如下:

```
# wsgiapp.py:

...

# 定义datetime_filter, 输入是t, 输出是unicode字符串:
def datetime_filter(t):
    delta = int(time.time() - t)
    if delta < 60:
        return u'1分钟前'
    if delta < 3600:
        return u'%s分钟前' % (delta // 60)
    if delta < 86400:
        return u'%s小时前' % (delta // 3600)
    if delta < 604800:
        return u'%s天前' % (delta // 86400)
    dt = datetime.fromtimestamp(t)
    return u'%s年%s月%s日' % (dt.year, dt.month, dt.day)

template_engine = Jinja2TemplateEngine(os.path.join(os.path.dirname(os.path.abspath(__file__)), 'templates'))
# 把filter添加到jinja2, filter名称为datetime, filter本身是一个函数对象:
```

```
template_engine.add_filter('datetime', datetime_filter)

wsgi.template_engine = template_engine
```

现在，完善的首页显示如下：

