Django & Twitter bootstrap in the workplace: build 'em fast and furious

Eliezer Kanal
Copyright 2017 Carnegie Mellon University. All Rights Reserved.

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

Internal use:* Permission to reproduce this material and to prepare derivative works from this material for internal use is granted, provided the copyright and “No Warranty” statements are included with all reproductions and derivative works.

External use:* This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other external and/or commercial use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

* These restrictions do not apply to U.S. government entities.


DM17-0178
Outline

* Quick note on Microservices and API Endpoints

- Framework Selection
- Set up for success
- Django
- Django extras
- Bootstrap
Django & Twitter bootstrap in the workplace: build 'em fast and furious

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

- Too many decisions
- Too many options
- Too many variables
Don’t…

- Roll your own
- Choose “no framework”
- Just choose your personal favorite

- Technical debt
- Framework/base development vs. Actual development
- Security bugs!!!

- You are on a team
- Each project is unique
Web Framework Benefits

Backend Framework will make you:
• Faster
• Capable
• Secure

Frontend Framework (layout) will make you:
• Look Professional

Frontend Framework (interactivity) will make you:
• Interactive
Framework Selection

Critical:

• What will the organization support?
• What does your team already know?
• What best suits your needs?
  - Size of community, support

Less critical:

• Ability to scale?
• Framework speed?
Set up for Success: Minimal training

• **Django:**
  1. Do the tutorial
  2. Google EVERYTHING

• **Bootstrap:**
  1. Explain the grid
  2. Scan through [docs](#)
     
     Resize the window while browsing!

Time investment: 1 day
Django – projects, apps, URLs

Terminal:

$ django-admin startproject mail_clone
$ django-admin startapp email
$ django-admin startapp calendar
$ django-admin startapp todo

settings.py:

```python
INSTALLED_APPS = (  
    'email',  
    'calendar',  
    'todo',  
)
```

urls.py:

```python
urlpatterns = [  
    url(r'^email/', include('email.urls')),  
    url(r'^calendar/', include('calendar.urls')),  
    url(r'^todo/', include('todo.urls')),  
]
```
Django – Database, ORM

models.py:

```python
from django.db import models

class Email(models.Model):
    to = models.CharField()
    from = models.CharField()
    date = models.DateField()
    subject = models.CharField()
    body = models.TextField()
```

Terminal:

```bash
$ python manage.py makemigrations
$ python manage.py migrate
```

- Create database changes file
- Update database structure
Django – Database, ORM

```python
>> Email.objects.create(
    to="alice@example.com",
    from="bob@example.com",
    date="2017-04-01 13:00:00",
    subject="Howdy!",
    body="Created an email!"
)

>> e = Email.objects.filter(to="alice@example.com")
>> e.subject = "Hello!"
>> e.save()
>> e.delete()
```

models.py:
```
from django.db import models

class Program(models.Model):
    name = models.CharField(max_length=500)
    description = models.TextField()
    date_added = models.DateField(auto_now_add=True)
```
Django & Twitter bootstrap in the workplace: build ‘em fast and furious

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.
Django – Other fun stuff

views.py
- Logic for each page
- Submitted form processing

forms.py
- Forms framework... cleans, error checks, etc. for you

Templates
- HTML, with custom syntax
- very similar to most other template languages

tests.py
- Built-in testing framework
Django – DRF

Django Rest Framework (DRF)

- Sophisticated API design framework
- From their website:

Some reasons you might want to use REST framework:

- The [Web browsable API](#) is a huge usability win for your developers.
- [Authentication policies](#) including packages for [OAuth1a](#) and [OAuth2](#).
- [Serialization](#) that supports both [ORM](#) and [non-ORM](#) data sources.
Bootstrap – Why

• UI Consistency
• Proven UI concepts
• Professional look
• Excellent community support:
  • Themes
  • Extensions
Bootstrap – Grid

- Think newspapers
- Columnar content
- Bootstrap deals with padding, margin
- “Just Add Classes”
### Django & Twitter Bootstrap in the Workplace: Build ‘Em Fast and Furious

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

```html
<div class="row">
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
</div>
```

```html
<div class="row">
  <div class="col-md-8">.col-md-8</div>
  .col-md-4
</div>
```

```html
<div class="row">
  .col-md-4
  .col-md-4
  .col-md-4
</div>
```

```html
<div class="row">
  .col-md-6
  .col-md-6
</div>
```
<table>
<thead>
<tr>
<th>Bootstrap – Everything else</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Forms</td>
</tr>
<tr>
<td>• Buttons</td>
</tr>
<tr>
<td>• Wells</td>
</tr>
<tr>
<td>• Tables</td>
</tr>
<tr>
<td>• Images</td>
</tr>
<tr>
<td>• Typography</td>
</tr>
<tr>
<td>• Navbars</td>
</tr>
<tr>
<td>• Breadcrumbs</td>
</tr>
<tr>
<td>• Alerts</td>
</tr>
<tr>
<td>• Thumbnails</td>
</tr>
<tr>
<td>• Progress bars</td>
</tr>
<tr>
<td>• Labels</td>
</tr>
<tr>
<td>• Badges</td>
</tr>
<tr>
<td>• Panels</td>
</tr>
<tr>
<td>• Lists</td>
</tr>
<tr>
<td>• Dropdowns</td>
</tr>
<tr>
<td>• List groups</td>
</tr>
<tr>
<td>• Glyphicons</td>
</tr>
<tr>
<td>• Tooltips</td>
</tr>
<tr>
<td>• Carousels</td>
</tr>
<tr>
<td>• Responsive menus</td>
</tr>
<tr>
<td>• Tabbed lists</td>
</tr>
<tr>
<td>• Modal boxes</td>
</tr>
<tr>
<td>• Popovers</td>
</tr>
<tr>
<td>• Transitions</td>
</tr>
<tr>
<td>• Plugins</td>
</tr>
<tr>
<td>• Themes</td>
</tr>
</tbody>
</table>
Bootstrap – Implementation

- Almost all through `<div>`’s and classes
- Encourages valid HTML
- Note: Poor bug handling (just markup, no “debugger”)

```html
<form class="form-inline">
  <div class="form-group">
    <label class="sr-only" for="exampleInputAmount">Amount (in dollars)</label>
    <div class="input-group">
      <div class="input-group-addon">$
        <input type="text" class="form-control" id="exampleInputAmount" placeholder="Amount">
      </div>
      .00
    </div>
    <button type="submit" class="btn btn-primary">Transfer cash</button>
  </div>
</form>
```
### Dashboard Version 2.0

#### CPU TRAFFIC
- 90%

#### LIKES
- 41,410

#### SALES
- 760

#### NEW MEMBERS
- 2,000

---

#### Monthly Recap Report

**Sales: 1 Jan, 2014 - 30 Jul, 2014**

<table>
<thead>
<tr>
<th>Month</th>
<th>Revenue</th>
<th>Cost</th>
<th>Profit</th>
</tr>
</thead>
<tbody>
<tr>
<td>January</td>
<td>$35,210.43</td>
<td></td>
<td></td>
</tr>
<tr>
<td>February</td>
<td>$10,390.90</td>
<td></td>
<td></td>
</tr>
<tr>
<td>March</td>
<td>$24,813.53</td>
<td></td>
<td></td>
</tr>
<tr>
<td>April</td>
<td></td>
<td></td>
<td>$1200</td>
</tr>
</tbody>
</table>

#### Goal Completion

- Add Products to Cart: 160/200
- Complete Purchase: 310/400
- Visit Premium Page: 480/800
- Send Inquiries: 250/500

---

**Inventory**

- 5,200
Django – Admin tools

- Built-in interface for oft-used functionality
- Needs only model definition
- Highly customizable UI
- Limited functionality extensibility
Takeaway

• Frameworks save time, effort
• Roll-your-own is bad
• Django (and other modern tools) have powerful features
• Bootstrap is a really cheap way to look super professional
Contact Info

Eliezer Kanal
Technical Manager & Principle Researcher
Telephone: +1 412.268.5204
Email: ekanal@cert.org

blog.erikdev.com
@eykanal

@eykanal