Django JavaScript Integration: AJAX and jQuery

Develop AJAX applications using Django and jQuery

Jonathan Hayward
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Preface</strong></td>
<td>1</td>
</tr>
<tr>
<td><strong>Chapter 1: jQuery and Ajax Integration in Django</strong></td>
<td>7</td>
</tr>
<tr>
<td>Ajax and the XMLHttpRequest object</td>
<td>8</td>
</tr>
<tr>
<td>Human speech: An overlaid function</td>
<td>8</td>
</tr>
<tr>
<td>Ajax: Another overlaid function</td>
<td>8</td>
</tr>
<tr>
<td>The technologies Ajax is overlaid on</td>
<td>9</td>
</tr>
<tr>
<td>JavaScript</td>
<td>9</td>
</tr>
<tr>
<td>XMLHttpRequest</td>
<td>14</td>
</tr>
<tr>
<td>Methods</td>
<td>14</td>
</tr>
<tr>
<td>Properties</td>
<td>15</td>
</tr>
<tr>
<td>HTML/XHTML</td>
<td>17</td>
</tr>
<tr>
<td>XML</td>
<td>18</td>
</tr>
<tr>
<td>JSON</td>
<td>18</td>
</tr>
<tr>
<td>CSS</td>
<td>19</td>
</tr>
<tr>
<td>The DOM</td>
<td>19</td>
</tr>
<tr>
<td>iframes and other Ajax variations</td>
<td>20</td>
</tr>
<tr>
<td>JavaScript/Ajax Libraries</td>
<td>21</td>
</tr>
<tr>
<td>Server-side technologies</td>
<td>21</td>
</tr>
<tr>
<td><strong>A look at Django</strong></td>
<td>21</td>
</tr>
<tr>
<td>Django templating kickstart</td>
<td>22</td>
</tr>
<tr>
<td>A more complete glimpse at Django templating</td>
<td>23</td>
</tr>
<tr>
<td><strong>Setting JavaScript and other static content in place</strong></td>
<td>32</td>
</tr>
<tr>
<td><strong>Summary</strong></td>
<td>33</td>
</tr>
<tr>
<td><strong>Chapter 2: jQuery—the Most Common JavaScript Framework</strong></td>
<td>35</td>
</tr>
<tr>
<td>jQuery and basic Ajax</td>
<td>36</td>
</tr>
<tr>
<td>jQuery Ajax facilities</td>
<td>39</td>
</tr>
<tr>
<td>$.ajax()</td>
<td>39</td>
</tr>
<tr>
<td>context</td>
<td>40</td>
</tr>
<tr>
<td>data</td>
<td>42</td>
</tr>
</tbody>
</table>
Table of Contents

dataFilter 43
dataType 43
error(XMLHttpRequest, textStatus, errorThrown) 44
success(data, textStatus, XMLHttpRequest) 44
type 44
url 44
$.ajaxSetup() 45
Sample invocation 45
$.get() and $.post() 45
.load() 46

jQuery as a virtual higher-level language 48
The selectors 48

A closure-based example to measure clock skew 52

Case study: A more in-depth application 56
Chapter 3: Validating Form Input on the Server Side 56
Chapter 4: Server-side Database Search with Ajax 56
Chapter 5: Signing Up and Logging into a Website Using Ajax 57
Chapter 6: jQuery In-place Editing Using Ajax 57
Chapter 7: Using jQuery UI Autocomplete in Django Templates 57
Chapter 8: Django ModelForm: a CSS Makeover 57
Chapter 9: Database and Search Handling 57
Chapter 10: Tinkering Around: Bugfixes, Friendlier Password Input, and a Directory That Tells Local Time 58
Chapter 11: Usability for Hackers 58
Appendix: Debugging Hard JavaScript Bugs 58

Summary 58

Chapter 3: Validating Form Input on the Server Side 61

The standard lecture: low-level validation 62
Matching regular expressions 62
You cannot guarantee absolutely valid data 63
Validating can detect (some) malicious input 63

The Django way of validation 64
Django gives you some things for free 64
The steps in Django's validation 65
A more sensible and cruelty-free approach to validation 66
Things get murrker 67
The zero-one-infinity rule: a cardinal rule of thumb in usability 68
An improvement on Django's advertised approach 68
A validation example: GPS coordinates 70
Avoiding error messages that point fingers and say, "You're wrong!" 71
Table of Contents

### Validation as demanding that assumptions be met

- Old-school: conform to our U.S.-based assumptions! 72
- Adding the wrong kind of band-aid 74

### Making assumptions and demanding that users conform

- At least names are simple, right? 76
- Even in ASCII, things keep getting murkier 77

### Better validation may be less validation

- Caveat: English is something of a lingua franca 79
- We don't have to negotiate with pistols 80
- Doing our best to solve the wrong problem: a story 81
- It really does apply to validation 82
- Facebook and LinkedIn know something better 83

### Summary 83

#### Chapter 4: Server-side Database Search with Ajax 85

- Searching on the client side and server side 86
- Handling databases through Django models 86
- Models for an intranet employee photo directory 87
- Searching our database 95
- A tour of Django persistence facilities 100

### Summary 103

#### Chapter 5: Signing Up and Logging into a Website Using Ajax 105

- admin.py: administrative functions called once 107
- functions.py: project-specific functions, including our
  @ajax_login_required decorator 107
- views.py: functions that render web pages 108
- style.css: basic styling for usability 113
- search.html: a template for client-side Ajax 114
- The Django admin interface 122

### Summary 124

#### Chapter 6: jQuery In-place Editing Using Ajax 125

- Including a plugin 127
- How to make pages more responsive 127

### A template handling the client-side requirements 128

- The bulk of the profile 132
- Whitespace and delivery 133
- Page-specific JavaScript 136
- Support on the server side 137

### Summary 139
## Table of Contents

### Chapter 7: Using jQuery UI Autocomplete in Django Templates  141

- Adding autocomplete: first attempt  142
  - Progressive enhancement, a best practice  142
- A real-world workaround  146
  - "Interest-based negotiation": a power tool for problem solving  146
  - when plan A doesn't work  146
  - A first workaround  148
  - Boilerplate code from jQuery UI documentation  154
  - Turning on Ajax behavior (or trying to)  156
  - Code on the server side  156
- Refining our solution further  159
- Summary  163

### Chapter 8: Django ModelForm: a CSS Makeover  165

- "Hello, world!" in ModelForm  165
- Expanding and customizing the example  168
- Customizing ModelForm pages' appearance  170
- Going under ModelForm's hood  182
- An excellent "stupid" question: where's the e-mail slot?  184
- Summary  187

### Chapter 9: Database and Search Handling  189

- Moving forward to an AHAH solution  189
  - Django templates for simple AHAH  192
  - Templating for a list of search results  192
  - Template for an individual profile  195
- Views on the server side  202
  - Telling if the user is logged in  202
  - A view to support deletion  202
  - The AHAH view to load profiles  203
  - Helper functions for the AHAH view for searching  204
  - An updated model  206
  - An AHAH server-side search function  207
- Handling the client-side: A template for the main page  209
  - CSS for styling the directory  232
  - Our updated urlpatterns  241
- Summary  241

### Chapter 10: Tinkering Around: Bugfixes, Friendlier Password Input, and a Directory That Tells Local Time  243

- Minor tweaks and bugfixes  243
  - Setting a default name of "(Insert name here)"  244
  - Eliminating Borg behavior  244
<table>
<thead>
<tr>
<th>Table of Contents</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Firebug and Chrome developer tools</td>
<td>290</td>
</tr>
<tr>
<td>The basics across browsers</td>
<td>290</td>
</tr>
<tr>
<td>Zeroing in on Chrome</td>
<td>293</td>
</tr>
<tr>
<td>Summary</td>
<td>298</td>
</tr>
<tr>
<td>Index</td>
<td>299</td>
</tr>
</tbody>
</table>