

Table of Contents

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

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 murkier	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

Validation as demanding that assumptions be met	72
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	76
At least names are simple, right?	76
Even in ASCII, things keep getting murkier	77
Better validation may be less validation	78
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

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	
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

Confusing jQuery's load() with html()	245
Preventing display of deleted instances	246
Adding a favicon.ico	249
Handling password input in a slightly different way	250
A directory that includes local timekeeping	252
Summary	260
Chapter 11: Usability for Hackers	261
Usability begins with anthropology... and Django hackers have a good start on anthropology	262
Anthropological usability techniques	263
An introductory example: card sorting	263
Focus groups: cargo cult research for usability	265
Anthropological observation: the bedrock of usability	265
More than one way to see the same situation	266
Applying this foundation to usability	268
It's just like (hard) debugging	271
Lessons from other areas	272
Live cross-cultural encounters	272
History	273
Old books and literature	274
The last other area: whatever you have	277
Understanding the user	278
A lesson from optimization	278
What's wrong with scratching an itch, or you are not your user	279
Worst practices from the jargon file	279
Python and usability	280
It's not all about the computer!	280
What to do in the concrete	282
Further reading	283
Summary	284
Appendix: Debugging Hard JavaScript Bugs	285
"Just fiddling with Firebug" is considered harmful	285
Cargo cult debugging at your fingertips	285
The scientific method of debugging	286
Exhausting yourself by barking up the wrong tree	287
The humble debugger	289
The value of taking a break	289
Two major benefits to asking for help	290

Firebug and Chrome developer tools	290
The basics across browsers	290
Zeroing in on Chrome	293
Summary	298
Index	299
