

Preface

You want to create an AJAX application. Why would you use Django? Why would you use jQuery? Why would you use both together? Enter Django JavaScript Integration: AJAX and jQuery – your comprehensive answer to all these questions and the only extensive, practical, and hands-on guide to developing any AJAX application with Django and jQuery.

Gone are the days when you used to lament over the lack of official documentation on AJAX with Django. This book will teach you exactly why Django is called The web framework for perfectionists with deadlines, how jQuery – the "write less do more" JavaScript library – is practically a virtual higher-level language, and why they both deserve to be integrated with AJAX.

This hands-on-guide shows you how to put Django and jQuery together in the process of creating an AJAX application. In this book, they are brought together in a real-world scenario, with attention to usability, to build and develop an AJAX application.

The first two chapters provide a short and necessary introduction to the world of Django, jQuery, and AJAX; the remaining chapters are based on a case study that will make you realize the immense potential and benefits of integrating Django and jQuery with your AJAX application.

By the time you are done with this book, you'll be developing your AJAX applications with Django and jQuery in less time than you can say "integrate".

You will cover the basics of AJAX; use jQuery, the most common JavaScript library, on the client side, and learn form validation with an eye towards usability, build things with Django on the server side, handle login and authentication via Django-based AJAX, and then dip into the rich jQuery plugin ecosystem to build in-place editing into your pages.

You will add auto-complete functionality courtesy of jQuery UI, easily build forms with Django ModelForm, and then look at a client-side search implementation that can look things up without network access after initial download. You will learn to implement a simple, expandable undo system, and offer more full-blooded account management, tinker, fix some bugs, offer a more usable way to handle password input, add local time support for people who are not in your time zone, look at usability, and finally take a look at debugging.

After working through this book, you will have both an AJAX application: a Web 2.0 employee intranet photo directory, and with it a deep understanding that you can use to customize, extend, and further develop it in your organization.

What this book covers

This book covers Django JavaScript integration and building an Ajax application with Django on the server side and jQuery on the client side. It provides first an overview, then a first Ajax application, and introduces jQuery; discusses form validation, server-side database search; Ajax login facilities; jQuery in-place editing and autocomplete, Django Modelform, and how to give auto-generated forms a transformational CSS makeover. It also discusses client-side functionality, customization, and further development with tinkering and added features, before a grand finale exploring usability, and an appendix on debugging hard JavaScript bugs.

Chapter 1, jQuery and Ajax Integration in Django lays a solid foundation and introduces you to the working pieces of Django Ajax to be explored in the rest of the book.

Chapter 2, jQuery – the Most Common JavaScript Framework explores the "higher-level" way of doing things in jQuery. You will learn how jQuery is not Python and does not look like Python, but how there is something "Pythonic in spirit" about how it works.

Chapter 3, Validating Form Input on the Server Side will teach you how to send an Ajax request to the server via jQuery, and validate it on the server side based on the principle that all input is guilty until proven innocent of being malicious, malformed, incomplete, or otherwise invalid.

Chapter 4, Server-side Database Search with Ajax looks both at the merits of handling searching and other backend functions with the full power of a backend environment, and explores why, on the client side, you should work hard to be as lazy as possible in doing network-related work.

Chapter 5, Signing-up and Logging into a Website Using Ajax introduces Django authentication facilities and account management and includes both server-side and client-side code.

Chapter 6, jQuery In-place Editing Using Ajax goes from a basic foundation to a continuing practical application. It will show a way to use jQuery to make an in-place replacement of a table that allows in-place editing, which communicates with the server in the background, adding persistence to changes.

Chapter 7, Using jQuery UI Autocomplete in Django Templates tells you what you need on the client side and server side to get autocomplete working with jQuery UI. It also includes creative problem solving when something goes wrong. This chapter will tell you why it is not uncommon for programmers to write plugins their first day doing jQuery.

Chapter 8, Django ModelForm: a CSS Makeover explores Django ModelForm and how to use it.

Chapter 9, Database and Search Handling covers all the bases for a simple, AHAH solution. In addition to showing "lazy" best practices, it also showcases a JavaScript in-memory database, with an application designed, at the developer's preference, to either always perform lazy handling of search and other requests, or start loading an in-memory database and falling back to lazy handling until the in-memory database is available.

Chapter 10, Tinkering Around: Bugfixes, Friendlier Password Input, and a Directory That Tells Local Time covers some tinkering and tweaks, and bugfixes along the way

Chapter 11, Usability for Hackers steps back from your application and takes a look at usability and the bedrock competencies hackers can leverage to do usability.

Appendix, Debugging Hard JavaScript Bugs looks at the state of mind that is needed to debug difficult bugs.

What you need for this book

This book assumes a broad technical maturity and an ability to learn and integrate different skills. It helps to be a Pythonista with the usual strengths that come with Python, and knowledge of the Web. A basic understanding of, or the ability to learn, Django and JavaScript will be helpful.

If you're a good generalist programmer who wants to learn Django JavaScript integration, this book is for you.

Who this book is for

This book is for people looking to integrate AJAX / JavaScript functionality into their web applications. It is for Django users who are looking to easily integrate AJAX features into their applications. Conversely, it will also be a priceless companion for users familiar with Django and jQuery who are looking to integrate them in their AJAX applications. A working knowledge of Django and basic familiarity with AJAX and jQuery are assumed.

Conventions

In this book, you will find a number of styles of text that distinguish between different kinds of information. Here are some examples of these styles, and an explanation of their meaning.

Code words in text are shown as follows: "You can override the empty string by setting `TEMPLATE_STRING_IF_INVALID` in your `settings.py` file".

A block of code is set as follows:

```
function outer()
{
    result = 0;
    for(i = 0; i < 100; ++i)
    {
        result += inner(i);
    }
    return result
}
```

When we wish to draw your attention to a particular part of a code block, the relevant lines or items are set in bold:

```
def ajax_profile(request, id):
    entity = directory.models.Entity.objects.filter(id = int(id))[0]
    if entity.is_invisible:
        return HttpResponse(u'<h2>People, etc.</h2>')
```

New terms and **important words** are shown in bold. Words that you see on the screen, in menus or dialog boxes for example, appear in the text like this: "We can then click on **Entity** (or **Locations**), and add an entity".



Warnings or important notes appear in a box like this.

Reader feedback

Feedback from our readers is always welcome. Let us know what you think about this book — what you liked or may have disliked. Reader feedback is important for us to develop titles that you really get the most out of.

To send us general feedback, simply send an e-mail to feedback@packtpub.com, and mention the book title via the subject of your message.

If there is a book that you need and would like to see us publish, please send us a note in the **SUGGEST A TITLE** form on www.packtpub.com or e-mail suggest@packtpub.com.

If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, see our author guide on www.packtpub.com/authors.

Customer support

Now that you are the proud owner of a Packt book, we have a number of things to help you to get the most from your purchase.



Downloading the example code for the book

You can download the example code files for all Packt books you have purchased from your account at <http://www.PacktPub.com>. If you purchased this book elsewhere, you can visit <http://www.PacktPub.com/support> and register to have the files e-mailed directly to you.

Errata

Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you find a mistake in one of our books — maybe a mistake in the text or the code — we would be grateful if you would report this to us. By doing so, you can save other readers from frustration and help us improve subsequent versions of this book. If you find any errata, please report them by visiting <http://www.packtpub.com/support>, selecting your book, clicking on the **errata submission form** link, and entering the details of your errata. Once your errata are verified, your submission will be accepted and the errata will be uploaded on our website, or added to any list of existing errata, under the Errata section of that title. Any existing errata can be viewed by selecting your title from <http://www.packtpub.com/support>.

Piracy

Piracy of copyright material on the Internet is an ongoing problem across all media. At Packt, we take the protection of our copyright and licenses very seriously. If you come across any illegal copies of our works, in any form, on the Internet, please provide us with the location address or website name immediately so that we can pursue a remedy.

Please contact us at copyright@packtpub.com with a link to the suspected pirated material.

We appreciate your help in protecting our authors, and our ability to bring you valuable content.

Questions

You can contact us at questions@packtpub.com if you are having a problem with any aspect of the book, and we will do our best to address it.