1. Record Nr. UNINA9911006804003321

Autore	Hayward Jonathan
Titolo	Django JavaScript integration : AJAX and JQuery / / Jonathan Hayward
Pubbl/distr/stampa	Birmingham, U.K., : Packt Open Source, 2011, c2010
ISBN	1-62198-894-5 1-282-96669-3 9786612966699 1-84951-035-0
Edizione	[1st edition]
Descrizione fisica	1 online resource (324 p.)
Disciplina	006.7
Soggetti	JavaScript (Computer program language) Ajax (Web site development technology) Web site development Electronic books.
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Includes index.
Nota di contenuto	Cover; Copyright; Credits; Foreword; About the Author; About the Reviewers; Table of Contents; Preface; Chapter 1: jQuery and Ajax Integration in Django; Ajax and the XMLHttpRequest object; Human speech: An overlaid function; Ajax: Another overlaid function; The technologies Ajax is overlaid on; JavaScript; XMLHttpRequest; Methods; Properties; HTML/XHTML; XML; JSON; CSS; The DOM; iframes and other Ajax variations; JavaScript/Ajax Libraries; Server-side technologies; A look at Django; Django templating kickstart; A more complete glimpse at Django templating Setting JavaScript and other static content in placeSummary; Chapter 2: jQuery-the Most Common JavaScript Framework; jQuery and basic Ajax; jQuery Ajax facilities; .ajax(); context; data; dataFilter; dataType; error (XMLHttpRequest, textStatus, errorThrown); success(data, textStatus, XMLHttpRequest); type; url; .aj0axSetup(); Sample invocation; .get() and .post(); .load(); jQuery as a virtual higher-level language; The selectors; A closure-based example to measure clock skew; Case study: A more in-depth application; Chapter 3: Validating Form Input on the Server Side

Chapter 4: Server-side Database Search with AjaxChapter 5: Signing Up and Logging into a Website Using Ajax; Chapter 6: jQuery In-place Editing Using Ajax; Chapter 7: Using jQuery UI Autocomplete in Django Templates; Chapter 8: Django ModelForm: a CSS Makeover; Chapter 9: Database and Search Handling; Chapter 10: Tinkering Around: Bugfixes, Friendlier Password Input, and a Directory That Tells Local Time; Chapter 11: Usability for Hackers; Appendix: Debugging Hard JavaScript Bugs; Summary; Chapter 3: Validating Form Input on the Server Side; The standard lecture: low-level validation
Matching regular expressionsYou cannot guarantee absolutely valid data; Validating can detect (some) malicious input; The Django way of validation; Django gives you some things for free; The steps in Django's validation; A more sensible and cruelty-free approach to validation; Things get murkier; The zero-one-infinity rule: a cardinal rule of thumb in usability; An improvement on Django's advertised approach; A validation example: GPS coordinates; Avoiding error messages that point fingers and say, ""You're wrong!""; Validation as demanding that assumptions be met
Old-school: conform to our U.S.-based assumptions!Adding the wrong kind of band-aid; Making assumptions and demanding that users conform; At least names are simple, right?; Even in ASCII, things keep getting murkier; Better validation may be less validation; Caveat: English is something of a lingua franca; We don't have to negotiate with pistols; Doing our best to solve the wrong problem: a story; It really does apply to validation; Facebook and LinkedIn know something better; Summary; Chapter 4: Server-side Database Search with Ajax; Searching on the client side and server side
Handling databases through Django models

Sommario/riassunto

Develop AJAX applications using Django and jQuery