

Numerics

500 Internal Server Error 246
7zip 5

A

AcceptVerbsAttribute 63, 206
act 21
action 8, 115
 single responsibility 45
action filter 138
 built-in 138
 requires SSL 139
 testing 301
 using to load common data 165
 when to create 274
action naming 292
action parameters 59
ActionController 264
ActionFilterAttribute 63, 166
ActionLink 80
ActionResult 44–46
 ContentResult 134
 FileResult 134
 JsonResult 134
 ViewResult 134
ActionView 264
Active Record pattern 239, 248, 260
Active Server Pages. *See* ASP
ActiveRecord 248
 built on top of NHibernate 248
ActiveRecordBase 249, 259
ActiveRecordMediator 249, 252
aggregates 27
 boundaries 28
 root 27
Agile 34
AJAX 47, 195
 from scratch 196
 helpers 201, 213
 Hijax technique 201
 HttpHandler 198
 request has extra HTTP header 206
 return values 200
 simple example 196
 with ASP.NET MVC 200
 with JSON 208
 with Web Forms 198
 with XML 207
Ajax.ActionLink 213
Ajax.BeginForm 201, 214
AjaxHelper 80
alternating row styles 350
Apache 94
Apache Velocity 141
ApartmentState 305
ApartmentState.STA 305
Application_Start 8
area. *See* controller
arrange 21
ASP 1
ASP.NET 2
 convention for locating views 16
 first MVC controller from scratch 14
 preferred way to write 3
 request pipeline 9
 simplified life of a request 10
 supports complex web applications 2
 using without SP1 4

ASP.NET Dynamic Data 282

ASP.NET MVC

- actions not same as in MonoRail 240
- alternative to Web Forms 174
- does not replace core ASP.NET libraries 3
- framework is open-ended 270
- views differ from Web Forms 66

ASP.NET MVC Framework

- ask for URL 107
- filter attributes 63
- not all-or-nothing 3

assert 21

AssertActionRedirect 297–298

Asynchronous JavaScript and XML. *See* AJAX

AuthorizationContext 273

AuthorizeAttribute 63

autocomplete 286

autocomplete plugin, filters local data
structures 314

AutoMapper 5, 87, 281

B

BarCamp 26

belongs_to 262

Bibeault, Bear 203

binder, smart 274

binding 59

Boo 141, 253

Brail 69, 141

- uses Boo 141

breadcrumb path 193

brittle tests 311

Builder pattern 252

business applications, long-lived 30

C

cache

- dependencies 181
- wrapping in our interface 181

caching 179

- making Cache testable 180

output 181

- page fragment 182

Calendar control 178

calendar picker 286

Castle Project 5, 87, 239, 242, 248

- ActiveRecord. *See* ActiveRecord
- frameworks and components collection 130
- MonoRail 141, 167

Validators 282

Windsor 49

- arranges the dependency chain 131
- ControllerFactory 130

catch-all, last route defined 105

CGI 1

check-in dance steps 232

Chrome 177, 307

Classic ASP 65

client-side validation 318

- never rely on 324

closed generic type 277

code

- brittle and unmaintainable 279
- test double 52

code blocks, do not mix within strings 157

Code Camp 26

Code Camp Server 92

AttendeesController 135

Conference model 98

creating routes for 108

formatting 157

Hijax technique 201

testing routes 113

workflow example 153

code-behind 2, 66

- markup files, do not use 11

MVC Views, do not use 67

CodeCampServer 23, 26, 47, 67

aggregates 27

benefits from Layer Supertype 272

thin controllers 31

codecampserver.org 34

COM, IE wrapper not thread-safe 305

Common Gateway Interface. *See* CGI

CompanyControllerFactory, two constructors 130

component, creating 163

Conery, Rob 159

Conference 26

continuous integration 232

control

- responsibility 66
- server 175

Controller

custom FormattableController 135

overridable methods 134

returning JSON results 139

returning XML results 138

controller 3, 8, 10

action 35

action meaning lost in noise 162

adding alternate view formats 208

and dependencies 48

in charge 3

default 9

developer in control of implementing 45

extending 134

flow to view 17

focus of MVC pattern 44

controller (*continued*)
 keep small 156
 maintainable 295
 notion of an action method selector 63
 order of execution 63
 organizing into areas 167
 responsibility in ViewData 73
 should be thin 52
 taming large actions 153
 well-designed 50
 without a view 47
 controller action 45–47
 controller classes, testing 20
 controller code, testing 13
 controller factory, StructureMap 133
 ControllerActionInvoker 167
 ControllerBase 134
 ControllerBuilder 127, 171
 ControllerFactory
 Castle Windsor 130
 creating custom 125, 128
 controllers 271–283
 defining additional namespaces for 171
 testing 50–55
 convention over configuration 256, 259
 cookies. *See* HttpCookies
 Core, must remain portable 327
 cross-cutting 341
 CruiseControl.net 233
 CSS 71
 classes to report form validation 85
 styling autocomplete results 315
 curl 92
 custom route, designing 289
 Cygwin 92

D

dash vs. slash 96
 data access integration test concern 336
 data access layer 30
 data-transfer objects 25
 DDD 24
 basics 25
 divide domain model 27
 inside Onion Architecture 327
 references 25
 repository for each aggregate 29
 debugging 185
 default application, picking apart 3–14
 Demeter, Law of explained 14
 dependencies 30, 37
 dependency injection. *See* DI
 deployment
 automation 232–236

batch script 235
 bootstrapper 235
 installation strategy 219
 URL rewriting 229
 wildcard mapping 226
 design, domain-driven. *See* DDD
 design, hand in hand with testing 50
 DI 48
 dictionary
 single location for key 279
 useful 279
 dispatch web request to a controller 7
 Django 94
 domain model
 important to application 328
 why important 31
 domain-driven design. *See* DDD
 domain-specific language. *See* DSL
 DRY principle 256, 259
 DSL 253, 258
 duplication, fighting 284

E

Eini, Oren 38, 53
 entity, key objects 26
 entropy 22
 environment configurations, managing 234
 error page, custom 246
 error reporting 85
 Evans, Eric 24–25
 expressions 286
 pointing to a property 288
 Extreme Programming 34

F

Factory pattern 252
 Feather, Michael 53
 filter
 action 138, 274, 301
 authorization 272
 context objects 301
 two classes may be overkill 274
 filtering, four interfaces provide support 62
 filters 62–64
 FinalBuilder 218
 Firebug 197, 307
 invaluable in AJAX development 205
 Firefox 177, 191, 307
 FireFox 2+ 66
 Flash 245
 contents short-lived 244
 folder structure basics 67–69

form
 HTML 84
 rendering 79
 form values 57
 processed first 58
 Fowler, Martin 35, 232

G

GAC 4
 Gallio 5
 external test runner is Icarus 304
 Garret, Jesse James 195
 Global.asax 8
 Google Suggest 313
 GridView 178

H

Haack, Phil 125
 HandleErrorAttribute 63
 Happy Path 6
 Hawley, Matt 191
 health monitoring 186
 Helicon Tech 229
 Hello World 14, 16
 HTML, more control over 177
 Html.BeginForm() 84
 HtmlHelper 283
 contains no view helpers 88
 extending 88
 HttpCookies 184
 adding to the response 184
 HTX 1, 65
 Hu, Ying 25

I

IActionFilter 62, 301
 IActionInvoker 167
 IAuthorizationFilter 62, 272–273
 IBuildManager 69
 Icarus 304
 IController 14
 implemented by a controller 44
 IControllerFactory 8, 127
 IDC 1, 65
 IE 66, 177, 305, 307
 browser discarded at teardown 309
 HTTP error messages 107
 managed by base test class 308
 IE8 307
 IExceptionHandler 62
 IHttpHandler 120

IIS 216
 mapping new extension 225
 IIS 6.0
 configuring routes for 226
 deploying to 223
 URL rewriting 229
 using a custom extension 225
 with .aspx extension 224
 IIS 7.0 217
 application pool configuration 222
 deploying to 219
 URL generation in 222
 IModelBinder 36, 59, 84
 input validation 282
 integration, continuous 232
 IntelliSense 142, 284, 289, 333
 Internet Explorer. *See* IE
 Internet Information Services. *See* IIS
 inversion of control. *See* IoC
 IoC 37, 49, 129
 constructor injection 131
 container to construct a controller 50
 in controllers 130
 tools provide factories automatically 343
 Ionic 229
 IResultFilter 62
 IRouteHandler 8, 120
 ISAPI
 developing custom filters requires C/C++ 223
 filters 223
 Rewrite 229–231
 IUserSession 273, 301
 IView 68, 142
 extensibility point 69
 single Render operation 68
 IViewDataContainer, extensibility point 69
 IViewEngine 142
 custom implementation 142
 extensibility point 69
 locating a view 70
 supports many master pages 71
 IViewLocationCache, extensibility point 69

J

JavaScript
 canceling form submissions 204
 jQuery library 200
 unobtrusive 198
 XML and JSON 207
 JavaScript Object Notation. *See* JSON
 JetBrains 21
 ReSharper 38, 70
 refactor code 164

- jQuery 71, 196
 - a must for web developers 212
 - autocomplete text box 312
 - JavaScript library 200
 - primer 203
- JSON 47, 135
 - better solution 200
 - consumed via JavaScript 138
 - consuming an action from the view 210

K

- Katz, Yehuda 203
- Keith, Jeremy 198
 - Hijax technique 201

L

- lambda expression 88, 113, 283
 - aid in refactoring 164
 - syntax not in ActionLink 157
- Law of Demeter 14
 - explained 199
- Layer Supertype 35, 271
 - create your own 11
 - gathers common filters 272
- layering 45
- layout 284
 - master page in ASP.NET 71
- Level, value indicates difficulty of session 27
- Lieberherr, Karl 199
- Link Helpers 157
- LINQ to SQL 30
- localization 187
 - adding an additional culture 190
 - adding global resources 189
 - configuring Firefox to prefer a different language 191
 - enabling autoculture selection from the browser 191
 - getting localized strings 189
- log4net 186

M

- magic strings 286
- maintainable 18
- manual testing 303
- mapping, wildcard 227
- MapRoute 120
- Marinescu, Floyd 25
- Martin, Bob 155
- master page 12, 71, 267
 - can be nested 284

- rendering only responsibility 19
 - strongly typed view 19
- MbUnit 50
 - imbed images into test report 307
 - unit testing framework 304
- menu control 176
 - renders in Firefox and IE 177
- MicroKernel. *See* Windsor
- Microsoft
 - and lambda expression syntax 157
 - Patterns and Practices 282
 - Reference Symbol Servers, debug framework code 128
- migration script 257
- MIME type 93
- mocks 53
- Model 75
- model binder 303
 - having confidence in 300
 - loading persistent objects 276
 - replace default 275
 - testing 298, 301
- model presentation 31
- ModelBinding 36
- ModelBindingContext, retrieve request value from 277
- ModelState 283, 297
- ModelStateDictionary 85
- Model-View-Controller pattern. *See* MVC pattern
- MonoRail 58, 167, 239, 248–255
 - actions not same as in ASPNET MVC 240
 - configuring to use Windsor integration 254
 - FilterAttribute 243
 - filters 242
 - Flash 244
 - layouts 241
 - NVelocity 240
 - PropertyBag 240
 - rescues 246
- Mozilla Firefox 191
- MSTest 13, 50
 - remove references 20
- MVC application hosting requirements 217
- MVC Futures 286
 - magic string usage remains 289
- MVC pattern
 - controller is focus 44
 - separation of concerns 294
- MVC View, renders top to bottom 66
- MVC website, files needed 218
- MvcContrib 5, 7–8, 17, 289, 292
 - alternatives to view engines 12
 - base classes 11
 - fluent route testing 112

MvcContrib (*continued*)
 makes testing routes easier 290
 SimplyRestful 100
 test helpers 292
 view helpers 13
 ViewDataExtensions 46
 MvcRouteHandler 7–8, 120

N

Naak 5
 naming, action 292
 NAnt 5, 233–236
 XCOPY deployment 218
 NAntContrib 233
 NBehave 5, 50
 NestedAttribute 250
 .NET 3.5
 LINQ expressions 88
 SPI 115
 NewtonSoft 208
 NHibernate 5, 184, 248
 configuration options exposed 250
 data access with 325
 library, not a framework 330
 needs configuration 331
 Nilsson, Jimmy 25, 334
 NLog 186
 NonActionAttribute 63
 NUnit 5, 13, 20, 36, 50
 NVelocity 17, 69, 141
 drawbacks 142

O

object-object mapper. *See* OOM
 object-relational mapping. *See* ORM
 objects, data-transfer 25
 Onion Architecture 327
 OOM 281
 Oracle, sequence functionality 334
 ORM 248
 Oshero, Roy 55
 output caching 181
 OutputCacheAttribute 63

P

Page_Load, large files 2
 parameters
 action 59
 querystring 58
 parameters, action 59

partial view
 defined 76
 who decides which to use 78
 partials
 creating 159
 perfect for small markup 285
 patterns
 Active Record 248
 Builder 252
 Factory 252
 Layer Supertype 271
 MVC 3
 Server Page 65
 Service Locator 130
 Session-per-Request 184
 Perl 94
 permalink, keep simple and clean 95
 persistence 29
 personalization 187
 building SQL tables for 187
 configuring 187
 display profile data 188
 editing profile data 188
 PHP 94
 pit of success 66
 posting HTML forms to server 84
 presentation layer, book focus on 30
 presentation model 31, 33
 key element of presentation layer 33
 object model that serves a screen 32
 part of presentation layer 32
 responsibilities 31
 for specific needs 31
 presentation object 33
 Principle of Least Knowledge. *See* Law of Demeter
 progressive enhancement 203
 projecting from domain model 33
 property bag, ViewData 73
 PropertyBag 239, 244
 Python 94

Q

querystring parameters 58
 QUnit 304

R

RAD frameworks 256
 Rails 94
 See also Ruby on Rails
 Rails Components 165
 rapid application development. *See* RAD
 Red Gate, .Net Reflector 11

- RedirectController, testing 50
 - reference implementation 326
 - Reflector 127
 - regression testing 303
 - RenderAction 163
 - alternative to filter and partial combo 285
 - rendering, places to customize behavior 69
 - repository 29, 297, 329
 - common base class 278
 - Save method not called 297
 - use to load entity 276
 - representational state transfer. *See* REST
 - request storage 184
 - request, partial 284
 - rescue 246
 - rescues 246
 - ReSharper 21, 38, 70, 304
 - refactor code 164
 - resources 188
 - REST 100
 - RESTful 293
 - Rhino Mocks 5, 53, 183, 296
 - creates a subclass at runtime 303
 - CreateStub method 303
 - generates derived classes on the fly 38
 - not always appropriate 54
 - supports dynamic stubs and mocks 54
 - testing a route 111
 - route 7, 35, 105, 109, 115, 289, 294
 - components 101
 - configuring to use .aspx extension 224
 - configuring to use .mvc extension 225
 - custom static 103
 - designing custom handler 116
 - first matched, first used 124
 - front door of web application 92
 - generic 8
 - testing with NUnit 111
 - testing with Rhino Mocks 111
 - RouteConfigurator 35
 - RouteData 122
 - RouteHandler, implementing 122
 - RouteTable 35
 - Routing 4
 - routing 95
 - available to all ASP.NET applications 7
 - custom dynamic 103
 - decouples URLs 95
 - default values 109
 - extending 120
 - generating URLs 107
 - IIS6 workaround 104
 - inbound 94
 - outbound 94, 115
 - regular expressions 114
 - runtime diagnostics 122
 - with existing ASP.NET projects 115
 - row style, alternating 350
 - Ruby 94
 - Ruby on Rails 58, 135, 239, 255–268
 - ActionPack 239, 264
 - Active Record 260
 - CodeCampServer 264
 - console is a sandbox 259
 - creating application from the command line 256
 - generators 257
 - harvested framework 255
 - migration 258
 - removed Components from Core 165
-
- ## S
- Safari 177
 - Sanderson, Steve 325
 - screenshot, capture if test failing 309
 - Search Engine Optimization. *See* SEO
 - Selenium 13, 70, 304
 - SEO 99
 - separated view models 280
 - separation of concerns. *See* SoC
 - server controls 175
 - Server Page pattern 65
 - Service Locator pattern 130
 - need to fetch dependencies 166
 - session factory, creates all sessions 334
 - session state 183
 - Session-per-Request pattern 184
 - Silverlight 29
 - SimplyRestful 100
 - single responsibility principle. *See* SRP
 - single-threaded apartment. *See* STA
 - site maps 192
 - slash vs. dash 96
 - Smalltalk 238
 - smart binders 274
 - SmartController 271
 - filters 272
 - SmartDispatcherController 240, 243
 - SoC 8, 52
 - critical to long-term maintainability 326
 - key best practice 17
 - solution templates 5
 - Spark 69
 - advantage over Web Forms 349
 - Spark view engine 345
 - SQL Server 218
 - identity functionality 334
 - SQL Server 2005 332

SRP 8, 152
 common violation 155
 STA 305
 starter project 9–10
 basic layout and CSS 9
 master page 12
 objects are simple strings 11
 state management 179
 state session 183
 storage request 184
 string, hardcoded 279–280
 strings, magic 286
 StructureMap 5, 49
 stubs 53
 subcontrollers 284
 System.Web.Abstractions 4
 System.Web.Abstractions.dll 4
 System.Web.Mvc 7–8, 19

T

T4 templates 145
 Tarantino 5, 59
 TDD 22, 25, 36
 Team Foundation Server 304
 TechFest 26
 technical debt 30
 templates 145
 solution 5
 test
 adding 299
 brittle 311
 UI 303
 test double 52, 296
 stubs and mocks 53
 test suite 22
 test-driven development. *See* TDD
 TestHelper 292
 testing 20, 289, 294, 303–311
 automated 50
 hand in hand with design 50
 login screen 305
 manual 303
 regression 303
 with WatiN 304
 testing routes 289
 tests, running in parallel 304
 TextBox 175
 TextBoxWithLabelFor 288
 Trace.axd 186
 tracing 185
 TreeView control 178
 Twitter 97
 type
 closed generic 277
 open generic 277

U

UI testing tools 303
 UI tests 303
 automating 304
 underscore, why use 161
 unit test 295, 298, 301
 arrange, act, assert flow 21
 check only a single class 295
 creating automated 21
 do not call out of process 50
 no shared global variables 55
 NUnit test fixture 36
 a simple controller 14
 substitute object provided 31
 unit testing 2, 31
 calling action methods directly 167
 do not allow out-of-process calls 55
 frameworks 274
 templates 13
 unit testing frameworks 304
 Unix curl command 92
 URL
 allow parameters to clash 96
 designing schema 95–99
 extending routing 120
 make hackable 96
 rewriting 94, 229
 slug 209
 take care when restructuring 193
 taking control with routing 94
 ugly and nonintuitive 224
 UrlRouteModule 120
 UrlRoutingModule 3
 user story 34

V

ValidateAntiForgeryTokenAttribute 63
 ValidateInputAttribute 63
 validation 85, 281, 283
 Castle Project 87
 form errors 85
 keep away from controller 86
 model state errors 296
 run against a model 283
 Validation Application Block 282
 validation error
 handling in controller action 283
 if none 298
 value objects 26
 Verheul, Dylan 313
 view 283–289
 affected by code choice 141
 difficult to unit test 81
 enforce strongly typed 284

- view (*continued*)
 - how selected 73
 - lives with parent module 67
 - master pages can be nested 71
 - multiplies do not share ViewDataDictionary instances 76
 - needs ViewData to get the information to render 76
 - partials help decompose complex screen 76
 - responsibility 12
 - segment into partials 156
 - simple or complex 56
 - strongly typed 283
 - volume of string identifies used 88
 - view basics, overview 69–79
 - view engine
 - adding additional 171
 - Brail 141
 - creating custom 141
 - in MonoRail 240
 - main components 142
 - not all support view specifying master 19
 - NVelocity 141
 - replacing 16
 - Spark 345
 - Web Forms 12
 - view helpers 13, 79
 - possibilities are endless 159
 - referencing CSS stylesheets 158
 - referencing javascript includes 158
 - strongly typed lambda helpers 157
 - using and creating 156
 - view model objects, data oriented 318
 - view model, separated 280
 - view name
 - corresponds to action names 293
 - same as action name 46
 - View Page 16
 - ViewData 12, 45–46, 66, 272, 279, 296
 - options for passing 61
 - property a dictionary 61
 - sending objects to a view 73
 - ViewEngineResult 168
 - ViewModel 281
 - advantage of separated 280
 - ViewPage 67
 - ViewResult 68, 278
 - ViewState 2
 - no longer necessary 2
 - Visual Studio 2, 50
 - customizing 145
 - T4 templates 145
 - Unit Test framework 149
 - WebFormViewEngine 70
 - Visual Studio 2008 Professional 13
 - Visual Studio 2008 SP1 4
 - Visual T4 Editor for Visual Studio 2008 Community Edition 148
 - Visual Web Developer Express SP1 4
- ## W
-
- WatiN 5, 13, 70, 304
 - communicating with IE 305
 - IE 305
 - testing with 304
 - Watir 13, 70, 304
 - Web Application Projects 4
 - Web Application Testing In .NET. *See* WatiN
 - Web Developer Toolbar 307
 - Web Forms 2
 - adding HttpCookies 184
 - ASP.NET MVC views differ 66
 - built on concept of a control 66
 - emphasizes web controls 79
 - ended URL 95
 - mandate strict convention for URLs 7
 - master pages 72
 - porting from 67
 - use regular expressions 306
 - view engine 12
 - Web Forms Framework 2
 - WebFormView 68–69
 - WebFormViewEngine 67, 159, 168
 - support out of the box 70
 - wildcard mapping, side effect 227
 - Windsor 239, 248
 - Castles IoC container 253
 - MonoRail integration 253
 - See also* Castle Windsor
 - WinForms 29
 - won 282
 - WPF 29
- ## X
-
- XCOPY deployment 218, 233
 - xUnit 50
 - xVal Validation Framework 325
- ## Y
-
- yellow screen of death 246, 248