

## Numerics

32-bit 362  
404 240–242, 247, 351, 354  
64-bit 357, 363

## A

abstract factories 325  
Accepting Input 54  
AccountController 212–213, 312–313  
action 8, 50–62, 64–65, 127, 129, 131, 133–135, 230–231, 233, 235–245, 313  
  filter 129–131, 283, 365, 379  
  testing 364, 376  
  invokers 283  
  methods 52  
  naming 369  
  parameters 203  
  results 127, 268, 276–277, 281  
  selectors 127, 131  
  single responsibility 51  
ActionExecutedContext 377  
ActionExecutingContext 377–379  
ActionFilter 129  
ActionFilterAttribute 129  
ActionLink 138, 141–142, 144, 148–149, 243–244, 305–307, 310  
ActionMethodSelector 131  
ActionMethodSelectorAttribute 369  
ActionNameAttribute 238, 370  
ActionNameSelectorAttribute 369  
ActionResult 50–51, 53–55, 58, 127, 130, 132–134, 238, 240–242, 304, 309–310, 369–370, 372–373  
Activator.CreateInstance 191  
Add Area wizard 302  
Add Controller 10  
Add View 11, 19–20  
Add View dialog 357, 359–360  
Add Web Site 82  
Add/Edit Application Extension Mapping 86, 88  
administration sections 301  
adonet.batch\_size 329  
aggregate 120, 122–126  
  boundaries 124  
  root 122–123  
AJAX 167–179, 181–182, 184–189, 236, 244  
  definition of 167  
  from scratch 168  
  helpers 173, 186  
  HttpHandler 171  
  no-touch 179  
  request has extra HTTP header 179  
  return values 172  
  simple example 168  
  with ASP.NET MVC 172  
  with JSON 181  
  with Web Forms 170  
  with XML 181  
Ajax.ActionLink 186  
Ajax.BeginForm 173, 187  
AjaxHelper 35, 138  
AjaxOptions 187–188  
  alternate path 52, 55, 58–59, 65  
  anonymous type 237, 240  
Apache 228, 230  
ApartmentState 287–289  
App\_GlobalResources 112  
App\_LocalResources 114  
application architects 278  
application bus 268, 278–282  
application logic 230–231  
application pool 83  
Application\_Start 183, 194, 198–199, 201, 220  
area 301–302, 312–313, 315–321  
  area registration 302–304  
  area registration class 302  
  AreaName 303, 307  
  AreaRegistration 303–304, 315  
  Areas folder 302  
ASP.NET Ajax  
  215, 221–222, 224  
ASP.NET MVC, alternative to Web Forms 95  
ASP.NET Routing  
  Debugger 355  
aspect-oriented  
  programming 129  
.aspx extension 84–86  
assembly 312–314, 316, 320–321  
assertion  
  negative 372  
assertions 364, 370  
AssociatedMetadataProvider 219

Asynchronous JavaScript and XML. *See* AJAX  
 attack 152, 155–160, 163  
 attribute 272  
 auditing 281  
 authentication 152–155, 166, 312  
 authorization 129, 152–155, 166  
 AuthorizationContext 377  
 Authorize 130, 153–154  
 AuthorizeAttribute 153–155, 161, 166  
 autocomplete 380  
 autocomplete plugin, filters  
   local data structures 382  
 AutoGenerateColumns 67–68, 77  
 AutoMapper 258–260, 262–267, 276–277  
 AutoMapperConfiguration-Tester 265  
 AutoMapperViewResult 277  
 automated deployment process 251  
 automated integration 252  
 automated test cases 364  
 automated testing 59  
 automated tests 5, 364, 379  
 automated user interface tests 284  
 automating deployment 252  
 automating the build 252  
 automation 251, 257, 284, 289, 356  
 autonumber 234

## B

---

ball of spaghetti 124  
 base controller 272  
 BaseFormatter 265–266  
 BeginForm 16, 243, 308  
 best practice 364  
 Bibeault, Bear 174  
 binary distribution 313  
 binding context 206  
 BindModel 204–207, 374–375  
 blog 231  
 boilerplate code 313, 317, 321  
 bootstrapper 194, 198  
 bootstrapping code 325  
 bounded contexts 120  
 breadcrumb path 115  
 brittle tests 283, 289  
 browser automation 283, 288

Build Action 314  
 build automation 251–252  
 bus 268, 278–280, 312, 315, 317, 320–321  
 business applications 124  
   long-lived 125  
 business domain 120  
 business logic 31, 49–52, 269–270, 280–281, 322, 326  
 business rule 52

## C

---

cache 95, 101–106  
   dependencies 103  
   wrapping in our interface 103  
 caching 101, 103–104, 116  
   making cache testable 101  
   output 103  
   page fragment 104  
 Calendar control 99  
 Castle Project 200  
 Castle Windsor 190–191, 200  
 catch-all, last route defined 241  
 change management 251  
 charset=UTF-8 228–229  
 check-in dance 252  
 CheckBoxFor 38–39  
 child action 136–137, 140, 142, 151  
 ChildActionOnly 130, 141–142  
 ChildActionOnlyAttribute 130, 154  
 Chrome 98, 159  
 Classic mode 83  
 client-side behavior 224  
 client-side validation 215, 221–222, 224  
 ClientID 99, 172  
 closure  
   negative 68  
 CLR constants 272  
 CLR objects 54  
 Code Camp Server, testing routes 246  
 code generation 310, 356, 360–361  
 code generators 361  
 code, test double 61  
 CodeCampServer 345  
 CodePlex 92–93, 356  
 codeplex.com 91  
 CodeTemplates 357, 359  
 COMB GUID 332  
 comma separated values 132  
 command message 278–279  
 CommandResult 280–281  
 common behavior 268  
 common controls 97  
 common view data 268, 271  
 complex models 28  
 complex types 68, 204  
 complexity 369  
 component  
   reusable 321  
 componentization 311  
 components 66, 69–70, 77  
 compose 312  
 configSections 70  
 configuration 327–330, 332–333, 335, 337, 344  
 configuration settings 251  
 configuration values 254–255  
 Confirm 187  
 connection strings 251  
 constructor injection 272, 325  
 content files 313–314  
 Content-Type 228–229  
 continuous integration 251–253, 257  
 control server 96  
 controller 8, 24, 50–52, 54, 56, 59–62, 64, 107, 110–111, 127–130, 230–231, 235–245, 284, 312–313, 315–316  
   adding alternate view formats 182  
   developer in control of implementing 51  
   extensibility 128, 135  
   factory 273, 275, 283  
   focus of MVC pattern 50  
   large number of 301  
   lightweight 268–269  
   maintainable 370  
   should be thin 62  
   simplifying 271  
   testing 59–65, 364  
   well-designed 59  
 ControllerActionInvoker 130, 273, 281, 338  
 ControllerBase 127–129, 377–379  
 ControllerBuilder 275  
 ControllerContext 101, 105–106, 110–111, 374–375  
 convention 51, 53–54, 258, 264  
 cookie 95, 101, 106, 155–157  
   *See also* HttpCookies  
 copy 254, 256–257

Copy Local 362  
 Core project 324, 326–327, 343  
 core, remain portable 324  
 coupling 125, 325, 345  
 CreateMetadata 219–220  
 cross-cutting 341  
 cross-cutting concerns  
   129, 135, 281  
 cross-site request forgery  
   152, 160  
 cross-site scripting 152, 155, 166  
 CruiseControl.net 252–253, 257  
 CSS 31, 66, 88, 100,  
   313, 383–385  
   styling autocomplete  
     results 384  
 CSV 132–134  
 CsvActionResult 132–133  
 culture 110–114  
 curl 228  
 CurrentUserFilter 377–379  
 custom action results 132  
 custom controller factories 190  
 custom extension 84, 86–87  
 custom project template 362  
 custom route 227, 283  
   designing 365  
 custom test project  
   templates 363  
 CustomRouteHandler 249–250,  
   352–355  
 cyclomatic complexity  
   270–271, 280  
 Cygwin 229

## D

---

dash vs. slash 232  
 data access 124–125, 193, 204,  
   208, 321–322, 324, 330, 335,  
   337, 343, 345  
   integration test concern 335  
 data access layer 125  
 data annotations 37, 215–216,  
   219, 224  
 data persistence 326  
 data store 322, 326  
 data type 52, 55–56  
 data-transfer objects 120  
 DataAnnotationsModelMetadata  
   Provider 219–220  
 database 326–327, 329, 331,  
   334, 336, 343, 345  
   administrator 234  
   IDs 231, 234  
   local 371  
   query 24  
   schema 334  
   server 256  
 DataConfig.cs 343  
 DataSource 99  
 DataType 36, 41, 44,  
   47, 216–217  
 DataTypeAttribute 217  
 DDD 120  
   divide domain model 122  
   inside onion architecture 324  
   repository for each  
     aggregate 124  
 debug flags 254  
 debugging 107  
 declarative programming 262  
 decoupling 324–325, 337, 344  
 deep linking 231  
 Default 237  
 default route 85  
 default template 313–314  
 DefaultControllerFactory  
   191–192, 196, 198, 201,  
   273, 281  
 DefaultModelBinder 28, 204,  
   209, 212–213, 374  
 DELETE 235–236  
 dependency 59–64, 125–126,  
   190–193, 195–197, 202, 272  
   hard to test 134  
 dependency graph 274  
 dependency injection 190–191,  
   193, 195, 202, 325  
 deploy.build 255  
 deploying 254  
 deployment 79, 251,  
   253, 256–257  
   batch script 255  
   bootstrapper 254  
   environment 251  
   installation strategy 80  
   package 253–254  
   scenario 252  
   script 253–254, 256  
   simplifying 251  
   URL rewriting 91  
   wildcard mapping 88  
 deployment.build 254–255  
 design  
   domain-driven. *See* DDD  
   hand in hand with testing 59  
   destination property 260, 264  
   destination type 277–278  
 developer friction 267

DevExpress 67  
 DI. *See* dependency injection  
 DisplayFor 39–40  
 DisplayForModel 40, 42  
 DisplayName 36, 41, 47, 216,  
   219–220  
 DisplayNameAttribute 218–219  
 DisplayTemplates 44  
 DisplayTextFor 39  
 distributed systems 278  
 Django 230  
 DLL 312–313, 320  
 document.getElementById  
   169–170  
*DOM Scripting: Web Design with  
 JavaScript and the Document  
 Object Model* 170  
*Domain Driven Design Quickly* 121  
 domain language 126  
 domain logic 270  
 domain model 119–122, 124  
   important to application 325  
 domain objects 260  
 domain service 193  
 domain-driven design. *See* DDD  
 Don't Repeat Yourself 65  
 DOS 80  
 DropDownListFor 39  
 DRY principle 65  
 duplicate code 129  
 duplication 136–137, 143, 151  
 durable 123  
 dynamic mocking 105  
 dynamic route 236

## E

---

editor templates 43–45  
 EditorFor 39–40, 42, 45–46  
 EditorForModel 40–42, 293  
 Eini, Oren 63, 371  
 email 233  
   sending 193  
 embedded resource 313–314  
 embedding views 312  
 EnableClientValidation 222  
 encoded user input 36  
 entity 119–120, 122–123  
   key objects 122  
 EntityModelBinder 206, 208  
 Enum.TryParse 376  
 environment configurations,  
   managing 254  
 environment settings 251, 254  
 environment variables 254–255

environments 79–80  
 error messages 52, 217, 221  
 Evans, Eric 120  
 event.preventDefault() 176–177  
 exception handling 375  
 ExceptionContext 377  
 exec 253–254  
 Execute() 128  
 exploit 155, 164  
 Export Template 362  
 expression-based helpers  
   283, 293  
 extensibility 4, 127, 129,  
   131, 204  
 extensibility points  
   126–129, 135  
 extension methods 308–310  
 external input 227  
 external libraries 324

## F

F4 key 314  
 factored 56, 62  
 Feather, Michael 63  
 file input 69, 74  
 file system 193  
 filter 154, 284, 372, 376–379  
   context objects 376  
 FinalBuilder 80  
 Firebug 169–170  
   invaluable in AJAX  
   development 178  
 Firefox 98, 113, 159  
 flexibility 127  
 flow 51–52, 55  
 fluent interface 67  
 Fluent NHibernate 328  
 FluentForm 294–295  
 FluentPage 299  
 folder 312–313  
   shared 45, 304  
   structure 228  
 form input 204  
 form posts 56  
 form values 54–55, 203, 210, 213  
 form, interacting with 293  
 FormatException 375–376  
 formatters 258, 260, 265–266  
 FormatValueCore 266  
 forms authentication 153  
 FormsAuthentication 134  
 FormsAuthentication.SignOut()  
   134  
 FormValueProviderFactory 210

Fowler, Martin 252  
 framework, ask for URL 243

## G

Gallio, external test runner is  
   Icarus 285  
 Garret, Jesse James 167  
 GET 164–165, 235–236  
 GetControllerInstance 192, 196,  
   199, 201  
 GetRouteData 353  
 GetValueProvider 210–211  
 global constants 370  
 global content 306, 311  
 global data 268  
 Global.asax 193–194, 196,  
   198, 220  
 Google Suggest 380  
 Grid 66–69, 77  
 GridView 96, 99–100  
 guestbook 5, 10  
 GuestBookController 10, 14,  
   17, 19  
 GUID 229  
 Guid.TryParse 376  
 GuidComb() 331

## H

Haack, Phil 355  
*Haacked* 355  
 hackable URLs 231  
 hackers 159  
 handlers 279–280  
 happy path 53  
 Hawley, Matt 114  
 health monitoring 95, 108  
 Helicon Tech 91  
 hibernate.cfg.xml 329–330, 333,  
   335, 344  
 HiddenFor 39  
 Hijax 174, 178, 187  
 HomeController 8–9  
 host application 320–321  
 hosted 78  
 hosting 78–79  
 HTML 31, 35, 37–41, 47  
   rendering 66  
   want more control over 99  
 HTML DOM 168  
 HTML encode 158–159, 166  
 html extension 228  
 HTML generators 295, 300  
 HTML helper 66, 162–163

HTML replacement 167, 189  
 Html.Action 138, 141  
 Html.AntiForgeryToken() 163  
 Html.BeginForm 16, 18  
 Html.CheckBoxFor 27  
 Html.DisplayForModel 20  
 Html.Encode 158–159  
 Html.LabelFor 27, 37–38  
 Html.RenderAction 104  
 Html.TextBoxFor 27, 37–38  
 HtmlHelper 35–38, 40, 43,  
   137–138, 141, 151, 217,  
   222, 243, 310, 318–319  
   extensions 137, 141, 151  
 HtmlHelperExtensions 316, 319  
 HTTP 203  
   GET 228  
   header 228  
   response 228–229  
   status code 242  
 HttpContext 54–55, 101–102,  
   105–107, 110–111  
 HttpContextBase 55, 245,  
   350, 352  
 HttpCookies 106  
   adding to the response 106  
 HttpFileCollectionValue-  
   ProviderFactory 210  
 HttpGetAttribute 132  
 HttpMethod 187–188  
 httpModules 248  
 HttpPost 17, 19, 130, 178  
 HttpPostAttribute 132  
 HttpRequestBase 55, 244–245  
 HttpResponse 106, 241  
 HttpResponseBase 55, 106, 241  
 HttpRuntime 101  
 HttpSessionState 105  
 HttpSessionStateBase 105  
 HttpUnauthorizedResult 154  
 Hu, Ying 121

## I

IActionFilter 129, 273, 275, 377  
 IActionInvoker 130, 191, 202  
 IAuthenticationFilter 154  
 IAuthorizationFilter  
   130, 154, 377  
 ICache 101–103  
 Icarus 285  
 IController 9, 52, 128, 135, 192,  
   196, 199, 201  
   implemented by a  
   controller 50

- IControllerFactory 192, 196
  - identifier, unique 230, 234
  - IEnumerable 67
  - IExceptionHandler 377
  - IFilteredModelBinder 204–206
  - IgnoreRoute 7, 236–237
  - IHttpHandler 71–72, 84, 171, 349
  - IHttpModule 70–72, 84, 91–92
  - IIRF.dll 92
  - IIS 78–80, 82–89, 91–93, 253–254
    - mapping new extension 86
  - IIS 6.0
    - configuring routes for 88
    - deploying to 84
    - URL rewriting 91
    - using a custom extension 86
    - with .aspx extension 85
  - IIS 7.0 79
    - application pool
      - configuration 82
      - deploying to 80
      - URL generation in 83
  - IIS6 83
  - IIS7 72
  - image files 313
  - IMessageProvider 194–199, 201–202
  - IModelBinder 374
  - imperative programming 262
  - Index 53–54, 56
  - Infrastructure project 325, 328, 330, 337, 341, 343–344
  - inheritance 272
  - input 25–30
    - data 54
    - element 37–40, 43
    - validation 159
  - InputBuilder.Bootstrap 315–316
  - inputClass 384
  - InsertionMode 187–188
  - instant messenger 233
  - Integrated mode 82
  - integration machine 252
  - integration, continuous 252
  - IntelliSense 24, 330
  - Internet Explorer 98, 229, 242
    - HTTP error messages 242
  - Internet Information Services. *See* IIS
  - Inversion of Control. *See* IoC
  - IoC 191, 274, 325
    - containers 191, 194, 325
    - framework 199, 202
    - tools provide factories
      - automatically 343
  - Ionic 91
  - IOrderRepository 124–125
  - IPrincipal 154
  - IRepository 206–208
  - IResultFilter 377
  - IRouteConstraint 240
  - IRouteHandler 8, 349–350, 352, 355
  - ISAPI 84–86, 88, 91–93
    - developing custom filters
      - requires C/C++ 84
    - filters 84
    - handlers 84
  - ISAPI Rewrite 91–93
  - IsapiRewrite4.dll 92
  - IsMatch 204–206
  - IsValid 56, 217–218
  - It works on my machine 252
  - IUserSession 377–379
  - IValueProvider 209–212
  - IView 32
  - IViewEngine 32
  - IVisitorRepository 326–328, 333, 339–341, 343
  - IWindsorContainer 200–201
- J**
- 
- JavaScript 31, 79, 88, 98–100, 155, 162, 164, 166–168, 170, 172–173, 175–177, 179, 182, 185, 187–189, 313, 382–383
    - canceling form
      - submissions 177
    - jQuery library 173
    - libraries 167, 173, 222
    - XML and JSON 179
  - JetBrains 67
  - JetBrains ReSharper
    - refactor code 28
  - jQuery 99–100, 168, 173–174, 176–177, 179, 185–186, 188–189, 221–222, 224, 380–381, 385
    - a must for web
      - developers 185
    - JavaScript library 173
    - Menu 100
    - Tabs 100
    - Treeview 100
    - UI 99
  - JSON 86, 152–153, 163–167, 172, 179, 181–185, 189, 222
    - better solution 172
    - consuming an action from the view 184
    - hijacking 152–153, 163–166
  - Json() 183
  - JsonRequestBehavior 165
  - JsonRequestBehavior.AllowGet 183
  - JsonResult 164–166, 183
- K**
- 
- Katz, Yehuda 174
  - Keith, Jeremy 170
    - Hijax technique 174
  - Kohari, Nate 197
- L**
- 
- LabelFor 39
  - lambda expression 27–28, 68, 245
    - aid in refactoring 28
  - large applications 301
  - large files 69–71, 77
  - late binding 69
  - launch 251
  - Law of Demeter 171
  - layer supertype 272–273
  - layouts 137, 139, 147, 149
  - level, value indicates difficulty of session 122
  - Lightweight Test Automation Framework 284
  - lineSeparator 385
  - links 136, 138, 148, 301, 305–307, 310
  - LINQ 53
  - LINQ to SQL 125
  - ListBoxFor 39
  - loadingClass 385
  - LoadingElementId 187
  - local build 252
  - Local Data Mode 382
  - localization 95, 109–110, 114
    - adding an additional culture 112
    - adding global resources 111
    - configuring Firefox to prefer a different language 113
    - enabling autoculture selection from the browser 113
    - getting localized strings 111

log4net 108  
 Log4Net.config 330, 333  
 logging 285  
 logging in 312, 320

## M

magic string 69, 307, 309–310  
 magic strings 308  
 maintainability 56, 269  
 malicious 155–156, 159  
 map 258–260, 264–265  
 mapping 258–260, 262–266, 322, 328, 330, 332, 335  
   manual 261–262  
   wildcard 88  
 mapping files 330, 333  
 mappings 330, 333–334  
 MapRoute 7, 237, 239–241, 303, 349  
 Marinescu, Floyd 121  
 Martin, Bob 269  
 master pages 10, 136–138, 140, 142, 147, 151, 301, 304, 306, 313  
 MbUnit 59, 362  
   unit testing framework 285  
 membership 109  
 menu control 97  
   renders in Firefox and IE 98  
 message handler 320–321  
 metadata 50, 56  
 micromessaging 233  
 MIME type 229  
 minChars 385  
 mocks 62–64  
 model 22–32, 34, 39–44, 46–47, 49, 51, 53–56, 58, 65, 204, 209  
   binder 202–206, 208–210, 283–284, 373–376, 379  
   custom, testing 364, 373  
   having confidence in 376  
   testing 376  
   binding 17, 21, 28, 30, 203, 208, 213–214  
   state 372  
   templates 217  
 Model-View-Controller  
   pattern 4  
   controller is focus 50  
 ModelBindingContext 205–207, 374–376  
 ModelMetadata 43–44, 46–48, 219–220

ModelMetadata.DataTypeName 44  
 ModelMetadata.Properties 43, 47–48  
 ModelMetadata.TemplateHint 44  
 ModelMetadataProvider 215, 218–220  
 ModelMetadataProviders.  
   Current 220  
 ModelState 55–56, 58, 372–373  
 ModelState.IsValid 55–56, 58  
 ModelType 205–207  
 MonoRail 145  
 Mozilla Firefox 113  
 MSBuild 80  
 MSDeploy 256  
 MSI 80  
 MSTest 59  
 .mvc extension 86–87, 92–93  
 MVC Futures 151  
 MVC pattern 4, 99  
 MVC website, files needed 79  
 MvcContrib 8, 66–69, 77, 244–247, 250, 279–280, 308, 312–313, 315–316, 320–321, 356, 362, 367–368, 372, 378–379  
   fluent route testing 245  
   makes testing routes easier 367  
   test helpers 369  
 MvcContrib.TestHelper 367, 372  
 MvcFutures 221–222  
 MvcHandler 350, 352  
 mvcHost 359–360  
 MvcRouteHandler 8, 350, 355  
 MvcTextTemplateHost 359–360

## N

NameFormatter 261, 263–264, 266  
 namespaces 312  
 naming conventions 260, 262  
 NAnt 80, 252–257  
   XCOPY deployment 79  
 NAntContrib 253  
 NBehave 59, 288  
 .NET 3.5 SP1 248  
 .NET 4 83, 96, 99  
 New Project 5  
 NewtonSoft 182

NHibernate 106, 321–322, 324–338, 341–345  
   initialization 343  
   library, not a framework 327  
   needs configuration 329  
 Nhibernate-  
   configuration.xsd 330  
 Nhibernate-mapping.xsd 330  
 NHibernateModule.cs 341  
 NHibernateSample 329, 334  
 Nilsson, Jimmy 121, 332  
 Ninject 190–191, 196–199, 201–202  
 NinjectBootstrapper 198–199  
 NinjectMessageProvider 197–198  
 NLog 108  
 NoScript 159  
 NUnit 59, 252, 257, 334–335, 362  
   testing a route 244

## O

object database 124  
 object-oriented principles 281  
 object-oriented software 22  
 object-to-object mapper 258  
 ObjectDataSource 99  
 ObjectFactory.GetInstance 196  
 ObjectFactory.Initialize() 194  
 objects, data-transfer 120  
 OnActionExecuted 377  
 OnActionExecuting 377–378  
 OnAuthorization 154, 377  
 OnBegin 187–188  
 onclick event 170  
 OnComplete 187  
 OnException 377  
 OnExecutingMethod 379  
 OnFailure 187  
 onion architecture 324  
 OnResultExecuted 377  
 OnResultExecuting 377  
 OnSuccess 187  
 Oracle 329  
   sequence functionality 331  
 organization 311  
 Oshero, Roy 61, 64  
 out-of-process call 333  
 output caching 103  
 OutputCache 103–104  
 OutputRouteDiagnostics 352

**P**


---

Page\_Load 268  
 page-centric request  
   lifecycle 168  
 parameter lists 136, 142–144  
 partial view 313, 338, 340–341  
 partials 136–137, 139–140,  
   151, 304  
 PasswordFor 38–39  
 patterns, Session-per-  
   Request 106  
 per-web request 195  
 Perl 230  
 permalink, keep simple and  
   clean 231  
 persist 324, 327–328, 339  
 persistence 119–120, 124–126,  
   326–328, 341  
 personalization 95, 109  
   building SQL tables for 109  
   configuring 109  
   displaying profile data 110  
   editing profile data 110  
 PHP 167, 230  
 placeholders 137–138  
 plugins 381  
 portable 324  
 portable area 311–321  
 PortableAreaRegistration  
   314–316  
 POST 164, 166, 235–236  
 post-build event 343  
 Post-Redirect-Get 18–19, 56  
 Powershell 253  
 preprocessor directives 355  
 presentation layer 124–125, 322  
 presentation model 22–25, 28,  
   30, 50, 53, 68, 119–120, 124,  
   258, 261, 263  
 PRG 21, 56  
 production environment  
   252, 257  
 Profile 32–36  
 profiles 263  
 progressive enhancement 176  
 project generators 356  
 projects 312, 314, 316  
 public 51, 53–55, 58, 60–61, 63  
 PUT 235–236  
 Python 230

**Q**


---

quality assurance 256  
 querystring 55, 203

QueryStringValueProvider-  
   Factory 210  
 QUnit 284

---

**R**

---

RadioButtonFor 39  
 Rails 230  
 RangeAttribute 216  
 RedirectController, testing 59  
 RedirectResult 132  
 RedirectToAction 309–310  
 RedirectToRouteResult  
   134, 373  
 Refactor Pro 67  
 refactoring 30, 270, 281,  
   308, 310  
 refactoring tool 68, 293  
 reference 120, 123, 324–325  
 reflection 333  
 regedit 362  
 RegisterAllAreas 315  
 RegisterArea 303  
 RegisterTheViewsInTheEmbedd-  
   edViewEngine 315, 317  
 registry 195  
 regression testing 284  
 regular expression 262  
 RegularExpressionAttribute 216  
 relational database  
   124–125, 322  
 remote deployment 251  
 RenderAction 130, 318–319  
 RenderPartial 104, 139–140  
 repository 60, 62–64, 120,  
   124–126, 326  
   Save method not called 372  
 representational state transfer.  
   *See* REST  
 request  
   forgery 153, 160  
   storage 106  
   values 206, 208  
 Request.Params 238  
 Request.QueryString 17  
 RequiredAttribute 40, 47,  
   216–217  
 RequireHttpsAttribute 154  
 ReSharper 67, 247, 285  
   refactor code 28  
 resource locator 227, 229, 241  
 resources 110  
 Response 106  
 Response.Write() 171–172  
 responsibility 54, 59, 120,  
   124–125, 269, 271–272, 274

REST 235  
 result filters 129  
 ResultExecutedContext 377  
 ResultExecutingContext 377  
 resultsClass 384  
 Rhino Mocks 63, 105, 362, 371  
   creates a subclass at  
     runtime 379  
   not always appropriate 64  
   supports dynamic stubs and  
     mocks 64  
   testing a route 244  
 route 6, 8, 79, 81, 84–86, 91,  
   93, 241, 284, 313, 317,  
   365–369, 379  
   catch-all 241  
   components 237  
   configuring to use .aspx  
     extension 85  
   configuring to use .mvc  
     extension 86  
   custom static 239  
   designing custom  
     handler 249  
   first matched, first used 355  
   generic 8  
   inspecting 352  
   testing 244, 364–365, 369  
   testing with NUnit 244  
   testing with Rhino Mocks 244  
   values 55  
 RouteCollection 236–237  
 RouteData 245, 352  
 RouteDataProviderFactory  
   210  
 RouteHandler  
   implementing 352  
 RouteLink 244  
 RouteValueDictionary  
   142–144, 307  
 routing 227–228, 230–233, 235,  
   239–244, 246–248, 250  
   custom dynamic 240  
   decouples URLs 231  
   generating URLs 243  
   IIS6 workaround 240  
   outbound 247  
   priorities 239  
   runtime diagnostics 352  
   with existing ASP.NET  
     projects 248  
 RSS 91, 316–319  
 RssWidget 312, 316–319  
 RssWidgetController 317–318  
 Ruby 230

Ruby on Rails 167, 236  
runtime errors 69

## S

Safari 98  
SchemaExport 334  
script 257  
    nefarious 156  
scripting 257  
scripting attacks 36  
search engine optimization 230  
search engines 230, 235, 243  
secure areas 155  
security 152  
selectors 127, 131  
Selenium 284  
separation of concerns  
    31, 62, 269  
    critical to long-term  
    maintainability 322  
server build 252–253  
server controls 96  
server environments 251  
service-oriented  
    architectures 278  
Session 209–214  
session 95, 104–106, 328–329,  
    332–334, 336, 341–343  
session factory 332–333,  
    341, 343  
    creates all sessions 332  
session state 101, 105  
Session-per-Request pattern 106  
SessionValueProvider 210–213  
SessionValueProviderFactory  
    210, 212, 214  
SetControllerFactory 193, 196,  
    199, 202, 275  
ShouldMapTo 367–368  
ShouldMapTo<TController>  
    245  
show\_sql 329  
Silverlight 124  
simulate 61–62  
single responsibility 51, 62  
singleton 195  
site maps 95, 114  
Site.css 13  
Site.Master 9–10, 12, 14, 18,  
    20, 306  
SiteMapPath 115–116  
SitemapProvider 116  
slash vs. dash 232

SlickUpload 66, 69–73, 75, 77  
SmartBinder 204–205, 208  
SMS 233  
SOAP 236  
software engineering 364  
Solution Explorer 314  
solution structure 324  
source code viscosity 270  
source control 252–254  
source type 259, 261, 263  
Spanish 112–113  
Spark 136, 144–151  
SparkViewFactory 146  
special-widget-route 244  
SQL 80  
SQL Server 329, 331, 334–335  
    identity functionality 331  
SQL Server 2005 329  
SqlDataSource 99  
SQLite 329  
SRP, common violation 269  
StartupModule 341  
state management 101  
state, session 105  
static files 88  
storage, request 106  
storyboard 50–52, 54–55, 59, 65,  
    270, 278, 280  
streaming 70  
StringLengthAttribute 216  
strongly typed views 31, 34,  
    293, 295  
StructureMap 190–191,  
    194–198, 202, 274, 325  
StructureMapControllerFactory  
    196  
StructureMapMessageProvider  
    195  
stubs 62–64  
Subversion 252  
SVN 252  
SyndicationService 316–317  
System.ComponentModel  
    namespace 218  
System.ComponentModel.DataAnnotations 215  
System.Web.Abstractions.dll 101  
System.Web.Mvc.Controller  
    9, 50–52  
System.Web.Mvc.MvcHandler 8  
System.Web.Routing 244–246,  
    248–249

## T

T4 308, 356–357, 360–361, 363  
    templates 356–357, 361, 363  
T4MVC 301, 307–311, 356  
TDD 121, 269  
Team Foundation Server 284  
technical analysis 269  
technical debt 125  
Telerik 100  
TempData 55–57, 128, 138  
templated helpers 20  
templates 31, 40–43, 45–49, 137,  
    139, 356, 359–361, 363  
templating 31, 45, 308, 311  
test  
    cases 364–365, 369, 379  
    double 61, 126, 371, 382  
    stubs and mocks 62  
    framework 361, 363  
    project template  
    356, 361, 363  
    running in parallel 285  
    setup 367  
test-driven development. *See*  
    TDD  
testability 4, 269, 283, 289, 300  
testable navigation 283  
TestFixtureAttribute 288  
TestHelper 368  
testing 50, 55, 59, 61, 63–65,  
    252, 257, 283–285, 364–365,  
    367–368, 370–374, 376  
    automated 59  
    hand in hand with design 59  
    manual 283–284, 289  
    regression 284  
    with WatiN 284  
Text Template Transformation  
    Toolkit. *See* T4  
text/html 228–229  
TextAreaFor 39  
TextBox 96–98, 110  
TextBoxFor 39  
third-party component 66  
ThreadStatic 106  
tooling 356  
ToSeparatedWords 220  
ToString 68  
Trace.axd 108, 237  
TraceContext 108  
tracing 95, 107–108  
trade-offs 56  
transient 123, 195, 201

TreeView 96, 99–100  
Twitter 233, 318

## U

UAC 362  
UI 22  
  project 325, 338,  
  341, 343–344  
  testing tools 284  
  tests 293, 295, 297, 300  
  automating 284  
unit test project 5  
unit testing 59–65, 125, 191,  
  284–285, 300, 365, 371, 373,  
  375–376, 378  
  calling action methods  
  directly 130  
  check only a single class 371  
  do not allow out-of-process  
  calls 64  
  do not call out of process 59  
  frameworks 284  
  no shared global variables 65  
  substitute object  
  provided 126  
Unix 229  
  curl command 228  
unnecessary information  
  231, 234  
untestable 64  
unzip 254  
UpdatePanel 179  
UpdateTargetId 187–188  
uploadButton 73  
UploadConnector 75  
uploading 66, 70, 74, 77  
UploadResult 72–73, 75  
UploadStatus 75–76  
URL 227–239, 241–245, 248,  
  250, 301, 303, 306–307,  
  310–311, 350  
  allow parameters to clash 233  
  designing schema 231–236  
  generation 307–308  
  make hackable 232  
  rewriting 84, 91–93, 230  
  schema 227–235, 238–239,  
  242–243, 250  
  take care when  
  restructuring 116  
  taking control with  
  routing 230  
  ugly and nonintuitive 86  
Url 187

Url.Action 27  
UrlAuthorizationModule 250  
UrlHelper 138  
UrlRouteModule 349  
UrlRoutingModule 248–250  
URLs 301, 307  
user input 22, 25–26, 28, 30, 32,  
  36, 51, 54–56  
  validation 215  
user interface 40, 47  
user interface layer 284  
user sessions 101  
UsersController 237–238

## V

ValidateAntiForgeryToken 162  
ValidateAntiForgeryTokenAttrib  
  ute 155, 162–163, 166  
ValidateFor 39  
ValidateInputAttribute 155  
validation 52, 55–56, 58, 204,  
  215–218, 220–224, 281  
  attributes 216–217  
  error 217, 224  
  if none 373  
  framework 215, 221  
  message 40, 48, 217, 221  
  model state errors 372  
ValidationMessage 217  
ValidationMessageFor  
  38–39, 217  
ValidationSummary 217  
validators 215, 223–224  
value assignment 260  
value objects 119–120, 122  
value providers 202–203,  
  209–210, 214  
ValueProvider 206–207  
ValueProviderFactories 210, 214  
ValueProviderFactory 209–211  
ValueProviderFactoryCollection  
  210  
ValueProviderResult 206–207,  
  209, 211–212  
VCS 360  
Verheul, Dylan 381  
VerifyRenderingInServerForm  
  97  
version control system 360  
View 51  
view 6, 9–10, 12, 14, 18, 20, 31,  
  34, 38, 40, 46, 48–49,  
  312–313, 316–317

engine 136, 144–147,  
  150–151  
helper 31, 96, 114  
model 53, 136, 142  
name  
  corresponds to action  
  names 369  
View() 51  
ViewContext 138, 222  
ViewData 8–10, 14, 16, 24–25,  
  33, 35, 46–47, 51, 99, 103,  
  109, 111, 114, 128, 130, 138,  
  149–150, 182, 271–272,  
  276–278, 365, 370–371,  
  377–379  
ViewData.Model 24–25, 35  
ViewDataDictionary 24, 32–35,  
  46, 140  
ViewMasterPage 137–138  
ViewModel 34–36, 38–40  
ViewPage 9, 12, 14, 17–18, 20,  
  34–36  
ViewResult 9, 32–33, 35, 72, 132,  
  277–278, 366, 370–373  
ViewState 96–97, 115  
ViewUserControl 140, 142  
virtual 326, 334  
Visitor 324–326, 328, 330–331,  
  336–337, 340  
VisitorAdditionFilter 339  
VisitorBuilder 339–340  
VisitorRepositoryFactory 327,  
  333, 339–340, 343  
VisitorRetrievalFilter 339–340  
Visual Studio 59, 314, 316, 330,  
  334, 343–345, 355–357,  
  361–363  
  Unit Test framework 361  
Visual Studio 2008 5  
Visual Studio 2010 5  
Visual Studio integration 150  
Visual T4 Editor 361  
Visual T4 Editor for Visual Stu  
  dio 2008 Community  
  Edition 361  
vulnerabilities 152, 155,  
  158–160, 162, 166  
vulnerable 155, 160, 162–165

## W

WatIN 284  
  testing with 284  
Watir 284

Web Application Testing In  
.NET. *See* WatiN  
Web Deploy 251, 256–257  
Web Forms 167–168, 170,  
227–228, 247–250  
  adding HttpCookies 106  
  ended URL 232  
web project 314  
web servers 257  
web services 193  
Web.config 153, 254  
WebFormViewEngine 31, 34,  
97, 114, 137, 144–146, 151  
WebResource.axd 237  
website, vulnerable 155  
WebTestBase 289–292, 294, 297  
wildcard mapping 83–84, 87–90  
  side effect 88

Windows 78–79  
Windows Server 2000 79  
Windows XP 79  
Windsor 199–202  
WindsorBootstrapper 200–202  
WinForms 124  
Word document 229  
Workflow Foundation 4  
Working Effectively with Legacy  
  Code 63  
WPF 124  
www.jeremyskinner.co.uk 69

---

**X**

XCOPY deployment 79–80,  
251, 253  
XHTML 99

XML 86, 167–168, 172, 179,  
181–183  
  files 124  
  manipulation 253  
  mapping 330  
XmlDataSource 99  
XMLHttpRequest 169–170  
xmlpoke 254, 256  
XSRF 159–161, 163–164  
XSS 153, 155, 158–159  
xUnit 59  
xUnit.NET 362

---

**Z**

zero-code AJAX 179  
zip file 320  
zip tasks 253