

# OpenLayers 3 Beginner's Guide

Get started with OpenLayers 3 and enhance your web pages by creating and displaying dynamic maps

Thomas Gratier

Paul Spencer

Erik Hazzard

[  
PUBLISHING



1 open source  
community experience distilled

ji.

BIRMINGHAM - MUMBAI

# Table of Contents

|  |          |
|--|----------|
| <b><u>Preface</u></b>                                    | <b>1</b> |
| <b><u>Chapter 1: Getting Started with OpenLayers</u></b> | <b>7</b> |
| Introducing OpenLayers                                   | 8        |
| Advantages of using OpenLayers                           | 9        |
| What, technically, is OpenLayers?                        | 10       |
| Client side  | 10       |
| Library  | 10       |
| Anatomy of a web mapping application                     | 11       |
| Web map client   | 12       |
| Web map server   | 12       |
| Connecting to Google, Bing Maps, and other mapping APIs  | 13       |
| Layers in OpenLayers                                     | 14       |
| Understanding a layer                                    | 14       |
| The OpenLayers website                                   | 14       |
| Time for action – downloading OpenLayers                 | 17       |
| Time for action – creating your first map                | 19       |
| Where to go for help                                     | 26       |
| API docs   | 26       |
| This book's website                                      | 27       |
| Mailing lists  | 27       |
| Other online resources                                   | 27       |
| OpenLayers issues  | 27       |
| IRC  | 28       |
| OpenLayers source code repository                        | 28       |
| Getting live news from RSS and social networks           | 28       |
| Summary  | 29       |

**Chapter 2: Key Concepts in OpenLayers**



|   |           |
|---|-----------|
| OpenLayers' key components                      | 32        |
| It's all about the map                          | 32        |
| Time for action – creating a map                | 34        |
| Time for action – using the JavaScript console  | 38        |
| Controlling the map's view                      | 47        |
| Displaying map content                          | 47        |
| Time for action – overlaying information        | 48        |
| Interacting with the map                        | 51        |
| Using interactions                              | 51        |
| Controls  | 51        |
| OpenLayers' super classes                       | 52        |
| Event management with the Observable class      | 53        |
| Working with events                             | 54        |
| Key-Value Observing with the Object class       | 56        |
| Time for action – using bindTo                  | 58        |
| Transforming values with bindTo                 | 60        |
| More about KVO properties                       | 60        |
| Working with collections                        | 61        |
| Creating a collection                           | 62        |
| Collection properties                           | 62        |
| Collection events                               | 62        |
| Collection methods                              | 63        |
| Summary   | 64        |
| <b><u>Chapter 3: Charting the Map Class</u></b> | <b>65</b> |
| Understanding the Map class                     | 66        |
| Time for action – creating a map                | 66        |
| Map Tenderers                                   | 70        |
| The Canvas renderer                             | 70        |
| The WebGL renderer                              | 71        |
| The DOM renderer                                | 71        |
| Time for action – rendering a masterpiece       | 72        |
| Map properties                                  | 74        |
| Time for action – target practice               | 74        |
| Map methods                                     | 76        |
| Control methods                                 | 77        |
| Interaction methods                             | 77        |
| Layer methods                                   | 77        |
| Overlay methods                                 | 78        |
| Map rendering methods                           | 78        |
| Animation functions                             | 78        |

|   |            |
|---|------------|
| <b>Time for action – creating animated maps</b>               | <b>80</b>  |
| <b>Conversion methods</b>                                     | <b>83</b>  |
| <b>Other methods</b>  | <b>84</b>  |
| <b>Events</b>   | <b>85</b>  |
| <b>Browser events</b>   | <b>85</b>  |
| <b>Map events</b>   | <b>86</b>  |
| <b>Render events</b>  | <b>86</b>  |
| <b>Views</b>  | <b>87</b>  |
| <b>The view Class</b>   | <b>87</b>  |
| <b>View options</b>   | <b>87</b>  |
| Understanding resolution                                      | 89         |
| <b>View KVO properties</b>                                    | <b>90</b>  |
| View methods  | 90         |
| <b>Time for action – linking two views</b>                    | <b>92</b>  |
| <b>Summary</b>  | <b>94</b>  |
| <b><u>Chapter 4: interacting with Raster Data Source</u></b>  | <b>95</b>  |
| <b>Introducing layers</b>                                     | <b>96</b>  |
| <b>Layers in OpenLayers 3</b>                                 | <b>96</b>  |
| <b>The base layer</b>   | <b>97</b>  |
| <b>Overlay layers</b>   | <b>97</b>  |
| <b>Types of layers</b>  | <b>97</b>  |
| <b>Common operations on layers</b>                            | <b>99</b>  |
| <b>Time for action – changing layer properties</b>            | <b>100</b> |
| <b>Tiled versus untiled layers</b>                            | <b>103</b> |
| <b>Types of raster sources</b>                                | <b>105</b> |
| <b>Defining a source</b>                                      | <b>105</b> |
| <b>A quick look at the history of API and tiles providers</b> | <b>106</b> |
| <b>Map mashups</b>  | <b>107</b> |
| <b>OpenLayers and third-party APIs</b>                        | <b>107</b> |
| <b>Tiled images' layers and their sources</b>                 | <b>108</b> |
| <b>The OpenStreetMap layer</b>                                | <b>109</b> |
| Accessing your own OSM tiles                                  | 109        |
| Understanding OSM tiling                                      | 110        |
| OpenStreetMap source class properties                         | 111        |
| <b>The MapQuest layer</b>                                     | <b>112</b> |
| MapQuest source class properties                              | 112        |
| <b>Stamen layers</b>  | <b>113</b> |
| <b>Time for action – creating a Stamen layer</b>              | <b>114</b> |
| <b>The Bing Maps layer</b>                                    | <b>116</b> |
| <b>Time for action – creating a Bing Maps layer</b>           | <b>116</b> |
| <b>The TileJSON layer</b>                                     | <b>118</b> |
| TileJSON source class properties                              | 119        |

## Table of Contents

|  |            |
|--|------------|
| <b>WMTS layers</b>   | <b>119</b> |
| WMTS source class properties   | 120        |
| <b>The DebugTileSource source</b>  | <b>120</b> |
| TileDebugTile source class properties  | 120        |
| <b>OpenLayers tiled WMS</b>  | <b>121</b> |
| Tiled WMS source class properties  | 122        |
| <b>OpenLayers Zoomify</b>  | <b>122</b> |
| <b>Time for action – creating tiles and adding Zoomify layer</b>                 | <b>123</b> |
| <b>Image layers and their sources</b>  | <b>125</b> |
| OpenLayers <sup>1</sup> image WMS layer  | 125        |
| <b>Using Spherical Mercator raster data with other layers</b>                    | <b>127</b> |
| <b>Time For action – playing with various sources and layers together</b>        | <b>128</b> |
| OpenLayers image for MapGuide  | 131        |
| Inserting raw images using ImageStatic class                                     | 132        |
| <b>Time For action – applying Zoomify sample knowledge to a single raw image</b> | <b>132</b> |
| <b>Summary</b>   | <b>134</b> |
| <b><u>Chanter 5: Using Vector Layers</u></b>                                     | <b>135</b> |
| <b>Understanding the vector layer</b>  | <b>136</b> |
| <b>Features of the vector layer</b>  | <b>136</b> |
| The vector layer is client side  | 137        |
| Performance considerations   | 137        |
| <b>The difference between raster and vector</b>                                  | <b>138</b> |
| <b>Time for action – creating a vector layer</b>                                 | <b>139</b> |
| <b>How the vector layer works</b>  | <b>141</b> |
| How the vector layer is rendered   | 141        |
| <b>The vector layer class</b>  | <b>142</b> |
| Creating a vector layer  | 142        |
| Vector layer methods   | 143        |
| <b>Vector sources</b>  | <b>143</b> |
| The vector source class  | 145        |
| The cluster source   | 147        |
| <b>Time for action – using the cluster source</b>                                | <b>148</b> |
| <b>The format sources</b>  | <b>150</b> |
| What are formats?  | 151        |
| The StaticVector source  | 154        |
| The ServerVector source  | 158        |
| <b>Time for action – creating a loader function</b>                              | <b>158</b> |
| <b>Time for action – working with the TileVector source</b>                      | <b>165</b> |
| <b>Time for action – a drag and drop viewer for vector files</b>                 | <b>168</b> |
| <b>Features and geometries</b>   | <b>170</b> |
| <b>The Geometry class</b>  | <b>170</b> |
| Coordinates  | 171        |

|  |            |
|--|------------|
| <b>Geometry methods</b>  | <b>171</b> |
| <b>Geometry subclasses</b>                                     | <b>172</b> |
| The SimpleGeometry class and subclasses                        | 172        |
| <b>Time for action – geometries in action</b>                  | <b>174</b> |
| <b>The Feature class</b>                                       | <b>176</b> |
| Creating a feature   | 176        |
| The Feature class properties                                   | 177        |
| Feature methods  | 177        |
| <b>Time for action – interacting with features</b>             | <b>179</b> |
| <b>Summary</b>   | <b>181</b> |
| <b><u>Chapter 6: Styling Vector Layers</u></b>                 | <b>183</b> |
| <b>What are vector styles?</b>                                 | <b>184</b> |
| What is a style function?                                      | 185        |
| <b>Time for action – basic styling</b>                         | <b>185</b> |
| <b>The style class</b>   | <b>189</b> |
| Fill styles  | 190        |
| Stroke styles  | 191        |
| Image styles   | 193        |
| <b>Time for action – using the icon style</b>                  | <b>195</b> |
| <b>Have a go hero – using the circle style</b>                 | <b>198</b> |
| Text styles  | 198        |
| <b>Multiple styles</b>   | <b>199</b> |
| <b>Time for action – using multiple styles</b>                 | <b>200</b> |
| <b>Style functions</b>   | <b>202</b> |
| <b>Time for action – using properties to style features</b>    | <b>203</b> |
| <b>Interactive styles</b>                                      | <b>207</b> |
| The feature overlays   | 208        |
| <b>Time for action – creating interactive styles</b>           | <b>209</b> |
| <b>Summary</b>   | <b>215</b> |
| <b><u>Chapter 7: Wrapping Our Heads Around Projections</u></b> | <b>217</b> |
| <b>Map projections</b>   | <b>218</b> |
| Why on earth are projections used?                             | 218        |
| Projection characteristics                                     | 218        |
| Area   | 219        |
| Scale  | 219        |
| Shape  | 220        |
| Other characteristics  | 221        |
| Types of projections   | 223        |
| EPSG codes   | 224        |
| <b>Time for action – using different projection codes</b>      | <b>224</b> |
| Latitude/longitude   | 227        |
| Latitude   | 227        |

|  |            |
|--|------------|
| Longitude  | 228        |
| <b>Time for action – determining coordinates</b>   | <b>228</b> |
| <b>OpenLayers projection class</b>   | <b>229</b> |
| Creating a projection object   | 229        |
| Functions  | 230        |
| <b>Transforming coordinates</b>  | <b>230</b> |
| <b>Time for action – coordinate transforms</b>   | <b>231</b> |
| The Proj4js library  | 232        |
| <b>Time for action – setting up Proj4js.org</b>  | <b>233</b> |
| Proj4js custom projections   | 233        |
| Adding custom projections  | 234        |
| OpenLayers 3 custom projections use cases  | 234        |
| <b>Time for action – reprojecting extent</b>   | <b>235</b> |
| Using raster layers with projections   | 235        |
| <b>Time for action – using custom projection with WMS sources</b>                                    | <b>235</b> |
| <b>Time for action – reprojecting geometries in vector layers</b>                                    | <b>238</b> |
| <b>Summary</b>   | <b>243</b> |
| <b><u>Chapter 8: Interacting with Your Map</u></b>   | <b>245</b> |
| <b>Selecting features with OpenLayers 3</b>  | <b>246</b> |
| Using, creating, and converting your own data  | 246        |
| <b>Time for action – converting your local or national authorities data into web mapping formats</b> | <b>248</b> |
| <b>Time for action – testing the use cases for ol.interaction.Select</b>                             | <b>251</b> |
| <b>Time for action – more options with ol.interaction.Select</b>                                     | <b>255</b> |
| <b>Introducing methods to get information from your map</b>  | <b>257</b> |
| Getting features information from your map vector layers   | 258        |
| <b>Time for action – understanding the forEachFeatureAtPixel method</b>                              | <b>258</b> |
| The getGetFeatureInfoUrl method – an alternative way of getting information from a map               | 261        |
| Basics of the WMS standard   | 261        |
| Using the getGetFeatureInfoUrl method to get information from your map                               | 262        |
| <b>Time for action – understanding the getGetFeatureInfoUrl method</b>                               | <b>262</b> |
| <b>Adding a pop-up on your map</b>   | <b>266</b> |
| The ol.Overlay reference   | 266        |
| <b>Time for action – introducing ol.Overlay with a static example</b>                                | <b>267</b> |
| Combining ol.Overlay with ol.Map features methods  | 270        |
| <b>Time for action – using ol.Overlay dynamically with layers information</b>                        | <b>270</b> |
| Creating or updating content on your map   | 274        |
| Drawing features on map  | 274        |
| <b>Time for action – using ol.interaction.Draw to share new information on the Web</b>               | <b>274</b> |

|  |            |
|--|------------|
| Time for action – using ol.interaction.Modify to update drawing                    | 278        |
| Understanding interactions and their architecture                                  | 280        |
| The short story of interactions  | 280        |
| Inspecting the ol.interaction.defaults function                                    | 281        |
| Time for action – configuring default interactions                                 | 282        |
| A functional view for the nine default interactions                                | 283        |
| Discovering the other interactions   | 284        |
| Time for action – using ol.interaction.DragRotateAndZoom                           | 284        |
| Time for action – making a rectangle export to GeoJSON with ol.interaction.DragBox | 286        |
| Summary  | 289        |
| <b><u>Chapter 9; Taking Control of Controls</u></b>                                | <b>291</b> |
| Introducing controls   | 292        |
| Using controls in OpenLayers   | 292        |
| Adding controls to your map  | 292        |
| Time for action – starting with the default controls                               | 293        |
| Controls overview  | 296        |
| The ol.control.Control class   | 297        |
| Control options  | 297        |
| The ol.control.Attribution control   | 297        |
| Attribution options  | 297        |
| Time for action – changing the default attribution styles                          | 298        |
| The ol.control.Zoom control  | 300        |
| Zoom options   | 300        |
| The ol.control.Rotate control  | 301        |
| Rotate options   | 302        |
| The ol.control.Fullscreen control  | 302        |
| Fullscreen options   | 303        |
| The ol.control.MousePosition control   | 303        |
| MousePosition options  | 303        |
| Time for action – finding your mouse position                                      | 304        |
| The ol.control.ScaleLine control   | 306        |
| ScaleLine options  | 306        |
| The ol.control.ZoomSlider control  | 308        |
| ZoomSlider options   | 308        |
| The ol.control.ZoomToExtent control  | 309        |
| Time for action – configuring ZoomToExtent and manipulate controls                 | 309        |
| Creating a custom control  | 311        |
| Time for action – extending ol.control.Control to make your own control            | 312        |
| Summary  | 315        |



|  |            |
|--|------------|
| <b><u>Chapter 10: OpenLayers Goes Mobile</u></b> | <b>317</b> |
| Touch support in OpenLayers                      | 318        |
| Using a web server                               | 318        |
| Finding your IP address on Windows               | 318        |
| Finding your IP address on OSX                   | 319        |
| Finding your IP address on Linux                 | 320        |
| Testing your IP address                          | 321        |
| Time for action – go mobile!                     | 321        |
| The Geolocation class                            | 324        |
| Limitations of the Geolocation class             | 325        |
| Using the Geolocation class                      | 325        |
| Time for action – location, location, location   | 325        |
| The Geolocation class in detail                  | 327        |
| Geolocation constructor options                  | 328        |
| Geolocation KVO properties                       | 328        |
| The DeviceOrientation class                      | 329        |
| Time for action – a sense of direction           | 331        |
| DeviceOrientation constructor options            | 332        |
| DeviceOrientation KVO property methods           | 333        |
| Debugging mobile web applications                | 333        |
| Debugging on iOS                                 | 333        |
| Debugging on Android                             | 336        |
| Debug anywhere – WEb INspector REmote (WEINRE)   | 340        |
| Getting started with WEINRE                      | 341        |
| Going offline                                    | 346        |
| The HTML 5 ApplicationCache interface            | 346        |
| Creating an ApplicationCache MANIFEST file       | 347        |
| Referencing a MANIFEST file in a web page        | 348        |
| Time for action – MANIFEST destiny               | 349        |
| Going native with web applications               | 351        |
| Time for action – track me                       | 352        |
| Summary  | 354        |
| <b><u>Chapter 11: Creating Web Map Apps</u></b>  | <b>355</b> |
| Development strategies                           | 355        |
| Using geospatial data from Flickr                | 356        |
| Note on APIs                                     | 356        |
| Accessing the Flickr public data feeds           | 356        |
| How we'll do it                                  | 357        |
| Time for action – getting Flickr data            | 357        |
| A simple application                             | 359        |
| Time for Action – adding data to your map        | 359        |
| Styling the features                             | 361        |

|  |            |
|--|------------|
| Time for action – creating a style function                                | 361        |
| Creating a thumbnail style   | 362        |
| Switch to JSON   | 363        |
| Time for action – switching to JSON data                                   | 363        |
| Time for action – creating a thumbnail style                               | 366        |
| Turning our example into an application                                    | 367        |
| Adding interactivity   | 367        |
| Time for action – adding the select interaction                            | 368        |
| Time for action – handling selection events                                | 370        |
| Displaying photo information   | 371        |
| Time for action – displaying photo information                             | 372        |
| Using real time data   | 375        |
| Time for action – getting dynamic data                                     | 375        |
| Wrapping up the application  | 376        |
| The plan   | 377        |
| Changing the URL   | 377        |
| Time for action – adding dynamic tags to your map                          | 377        |
| Deploying an application   | 379        |
| Creating custom builds   | 379        |
| Benefits of serving small files  | 380        |
| Two approaches to optimization   | 380        |
| What does the compiler do?   | 381        |
| Rewriting code   | 382        |
| Removing unused code   | 382        |
| Renaming objects, functions, and properties                                | 383        |
| Creating a combined build  | 384        |
| Time for action – creating a combined build                                | 384        |
| Creating a separate build  | 388        |
| Time for action – creating a separate build                                | 389        |
| Summary  | 392        |
| <b>Appendix A: Object-oriented Programming – Introduction and Concepts</b> | <b>393</b> |
| What is object-oriented programming?                                       | 394        |
| What is an object?   | 394        |
| What is a class?   | 394        |
| What is a constructor?   | 395        |
| What is inheritance?   | 395        |
| What is an abstract class?   | 396        |
| What is a namespace?   | 397        |
| What are getters and setters?  | 397        |
| Going further  | 400        |

**Appendix B: More details on Closure Tools and Code Optimization**

|  |            |
|--|------------|
| <b><u>Techniques</u></b>   | <b>401</b> |
| The Closure Tools philosophy   | 402        |
| Ensuring optimum performance   | 402        |
| Introducing Closure Library, yet another JavaScript library                      | 404        |
| The basics   | 404        |
| Time for action – first steps with Closure Library                               | 404        |
| Custom components  | 409        |
| Inheritance, dependencies, and annotations                                       | 409        |
| Making custom build for optimizing performance                                   | 414        |
| Time for action – playing with Closure Compiler                                  | 414        |
| Applying your knowledge to the OpenLayers case                                   | 417        |
| Installing the OpenLayers development environment                                | 417        |
| Installing Node.js   | 418        |
| Installing Java  | 419        |
| Installing Git   | 420        |
| Microsoft Windows  | 420        |
| Local OpenLayers development reloaded  | 421        |
| Time for action - running official examples with the internal OpenLayers toolkit | 421        |
| Time for action - building your custom OpenLayers library                        | 424        |
| Syntax and styles  | 431        |
| Time for action – using Closure Linter to fix JavaScript                         | 431        |
| Coding styles alternatives and tools   | 433        |
| Summary  | 434        |
| <b><u>Appendix C: Squashing Bugs with Web Debuggers</u></b>                      | <b>435</b> |
| Introducing Chrome Developer Tools   | 436        |
| Getting started with Chrome Developer Tools                                      | 436        |
| Time for action – opening Chrome Developer Tools                                 | 437        |
| Explaining Chrome Developer debugging controls                                   | 439        |
| Panels   | 440        |
| Time for action – using DOM manipulation with OpenStreetMap map images           | 442        |
| Time for action – using breakpoints to explore your code                         | 446        |
| Time for action – playing with zoom button and map copyrights                    | 449        |
| Panel conclusion   | 452        |
| Using the Console panel  | 452        |
| Time for action – executing code in the Console                                  | 453        |
| Time for action – creating object literals                                       | 454        |
| Object literals  | 455        |
| Time for action – interacting with a map   | 456        |
| The API documentation  | 458        |

|  |                   |
|--|-------------------|
| Improving Chrome and Developer Tools with extensions                       | 458               |
| JSONView   | 458               |
| Dealing with color with ColorZilla   | 459               |
| Debugging in other browsers  | 460               |
| Debugging in Microsoft Internet Explorer                                   | 461               |
| Debugging in Mozilla Firefox   | 462               |
| Summary  | 464               |
| <b><u>Appendix D: Pop Quiz Answers</u></b>                                 | <b><u>465</u></b> |
| Chapter 3, Charting the Map Class  | 465               |
| Chapter 5, Using Vector Layers   | 465               |
| Chapter 7, Wrapping Our Heads Around Projections                           | 465               |
| Chapter 8, Interacting with Your Map                                       | 466               |
| Chapter 9, Taking Control of Controls                                      | 466               |
| Chapter 10, OpenLayers Goes Mobile   | 466               |
| Appendix B, More details on Closure Tools and Code Optimization Techniques | 466               |
| Appendix C, Squashing Bugs with Web Debuggers                              | 467               |
| <b><u>Index</u></b>  | <b>463</b>        |