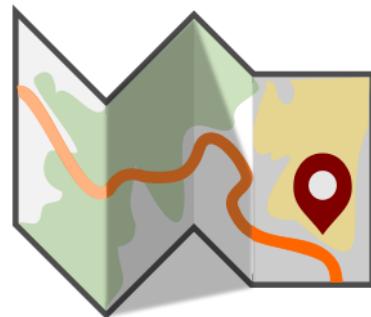


Web Mapping with QGIS2Web

QGIS Tutorials and Tips



Author

Ujaval Gandhi

<http://www.spatialthoughts.com>

Web Mapping with QGIS2Web

Web mapping is a great medium to publish your GIS data to the web and make it accessible by other users. Creating a web map is a very different process than creating one in a GIS. GIS users are typically aren't web programmers and it presents a challenge when one needs to create a web map that is of the same quality as a map creating in a GIS. Fortunately, there are tools available to easily translate your work in QGIS to web maps. In this tutorial, you will learn how to use the **QGIS2Web** plugin to create a web map using OpenLayers or Leaflet libraries from your QGIS project.

Overview of the task

We will create a openlayers web map of world's airports.

Other skills you will learn

- How to use Edit Widgets in QGIS to hide certain fields and set custom types.
- How to create a virtual field using Field Calculator.
- Creating labels for features that appear only at certain scale.

Get the data

We will use the [Airports](#) dataset from Natural Earth.

Download the [Airports shapefile](#).

For convenience, you may directly download a copy of the datasets from the links below:

[ne_10m_airports.zip](#)

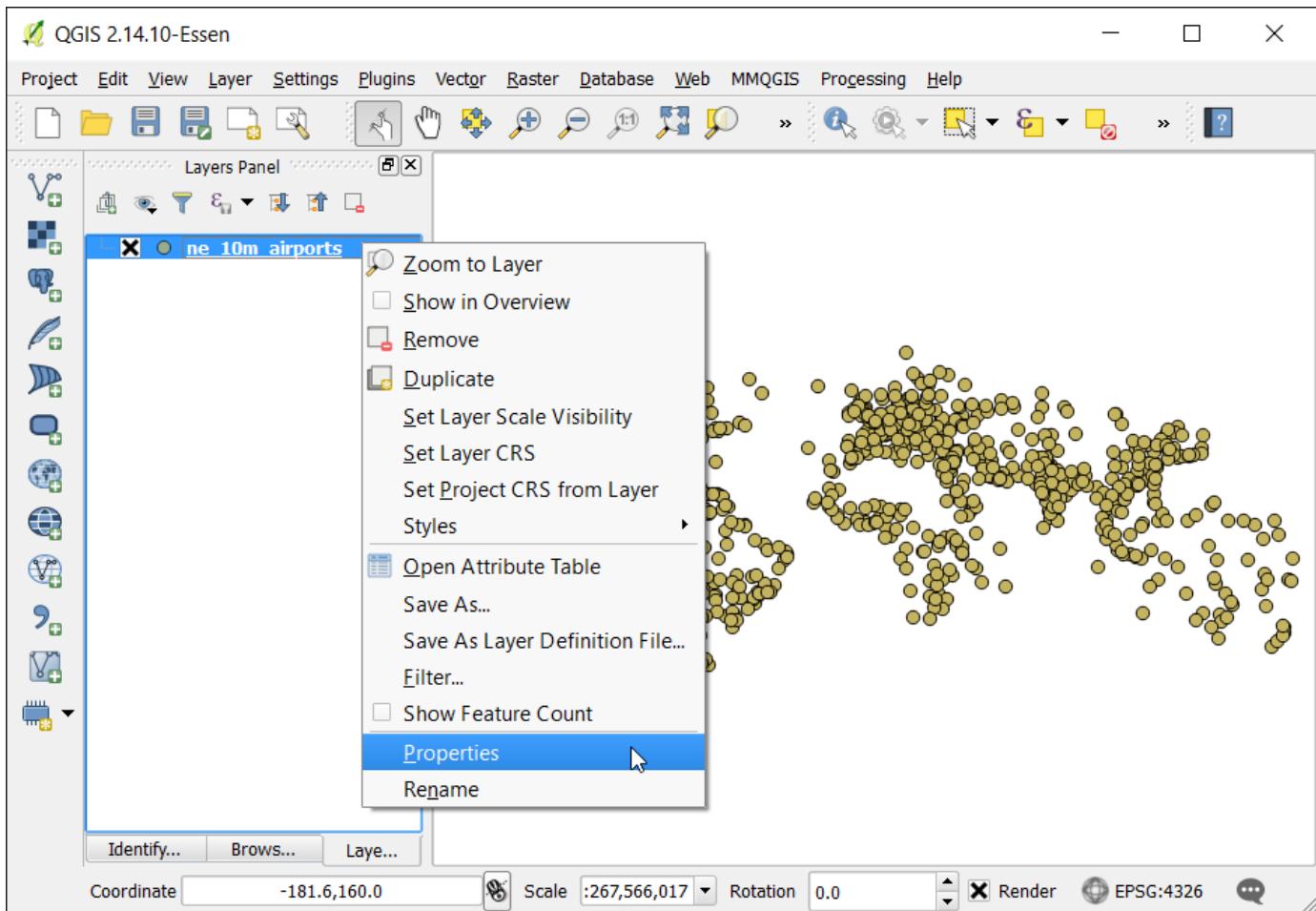
Data Source [NATURALEARTH]

Procedure

1. Open QGIS and go to Layer ▶ Add Vector Layer. Browse to the location of the downloaded file and select `ne_10m_airports.zip`. Click OK.



2. We will now create a map in QGIS that looks and behaves just like we would like in the web map. The plugin `qgis2web` will use replicate the QGIS settings and automatically create the web map without us knowing about web mapping libraries. When a user clicks on a airport marker, we want an info-window to display useful information about the airport. This information is already present in the attribute table of the `ne_10m_airports` layers. Right-click on the `ne_10m_airports` layer and select Properties.



3. Switch to the Fields tab. You will notice the different attributes present in the layer. Some of these aren't relevant to the users of our web map, so we can choose to hide these. We will keep type, name, iata_code and wikipedia fields and hide the others. Click on Text Edit button under the Edit widget column for `scalerank` field.

Layer Properties - ne_10m_airports | Fields

Attribute editor layout: Autogenerate Python Init function

Fields

ID	Name	Type	Type name	Length	Precision	Comment	Edit widget	Alias
123 0	scalerank	int	Integer	4	0		Text Edit	
abc 1	featurecla	QString	String	80	0		Text Edit	
abc 2	type	QString	String	50	0		Text Edit	
abc 3	name	QString	String	200	0		Text Edit	
abc 4	abbrev	QString	String	4	0		Text Edit	
abc 5	location	QString	String	50	0		Text Edit	
abc 6	gps_code	QString	String	254	0		Text Edit	
abc 7	iata_code	QString	String	254	0		Text Edit	
abc 8	wikipedia	QString	String	254	0		Text Edit	
1.2 9	natlscal	double	Real	7	3		Text Edit	

Relations

Suppress attribute form pop-up after feature creation

Style

OK Cancel Apply Help

The screenshot shows the 'Fields' tab in the QGIS Layer Properties dialog. The 'Edit widget' column for the first field ('scalerank') has a red box around it. The 'Name' column for the second field ('featurecla'), the third field ('type'), and the eighth field ('wikipedia') also have red boxes around them. The table lists various fields with their names, types, lengths, and precisions. The bottom right corner of the table has a blue highlight.

4. In the Edit Widget Properties dialog, choose `Hidden` as the type. Click OK.



5. Similarly set other fields to Hidden type. As you may have notices, there are other field types available that allow us to set how the fields appear to the users of our map. Click Edit Widget for wikipedia field.

Layer Properties - ne_10m_airports | Fields

Attribute editor layout: Autogenerate Python Init function

Fields

ID	Name	Type	Type name	Length	Precision	Comment	Edit widget	Alias
123 0	scalerank	int	Integer	4	0		Hidden	
abc 1	featurecla	QString	String	80	0		Hidden	
abc 2	type	QString	String	50	0		Text Edit	
abc 3	name	QString	String	200	0		Text Edit	
abc 4	abbrev	QString	String	4	0		Hidden	
abc 5	location	QString	String	50	0		Hidden	
abc 6	gps_code	QString	String	254	0		Hidden	
abc 7	iata_code	QString	String	254	0		Text Edit	
abc 8	wikipedia	QString	String	254	0		Text Edit	
1.2 9	natlscalerank	double	Real	7	3		Hidden	

Relations

Suppress attribute form pop-up after feature creation Default

Style

OK Cancel Apply Help

The screenshot shows the 'Fields' tab in the QGIS Layer Properties dialog. The 'Edit widget' column for the 'wikipedia' field is highlighted with a red box, indicating it is selected or being modified. The 'wikipedia' field has a type of 'QString' and a length of 254, and its current edit widget is 'Text Edit'. Other fields like 'name' and 'type' also have 'Text Edit' as their edit widget.

6. Select `Web View` as the field type. This type indicates that the value contained in this field is a URL.



7. We can also use the Alias column to indicate an alternate name for the fields without actually changing the underlying data table. This is useful to give more user-friendly field names to the users of our map. Add aliases as per your choices and click Ok.

Layer Properties - ne_10m_airports | Fields

Attribute editor layout: Autogenerate Python Init function

Fields

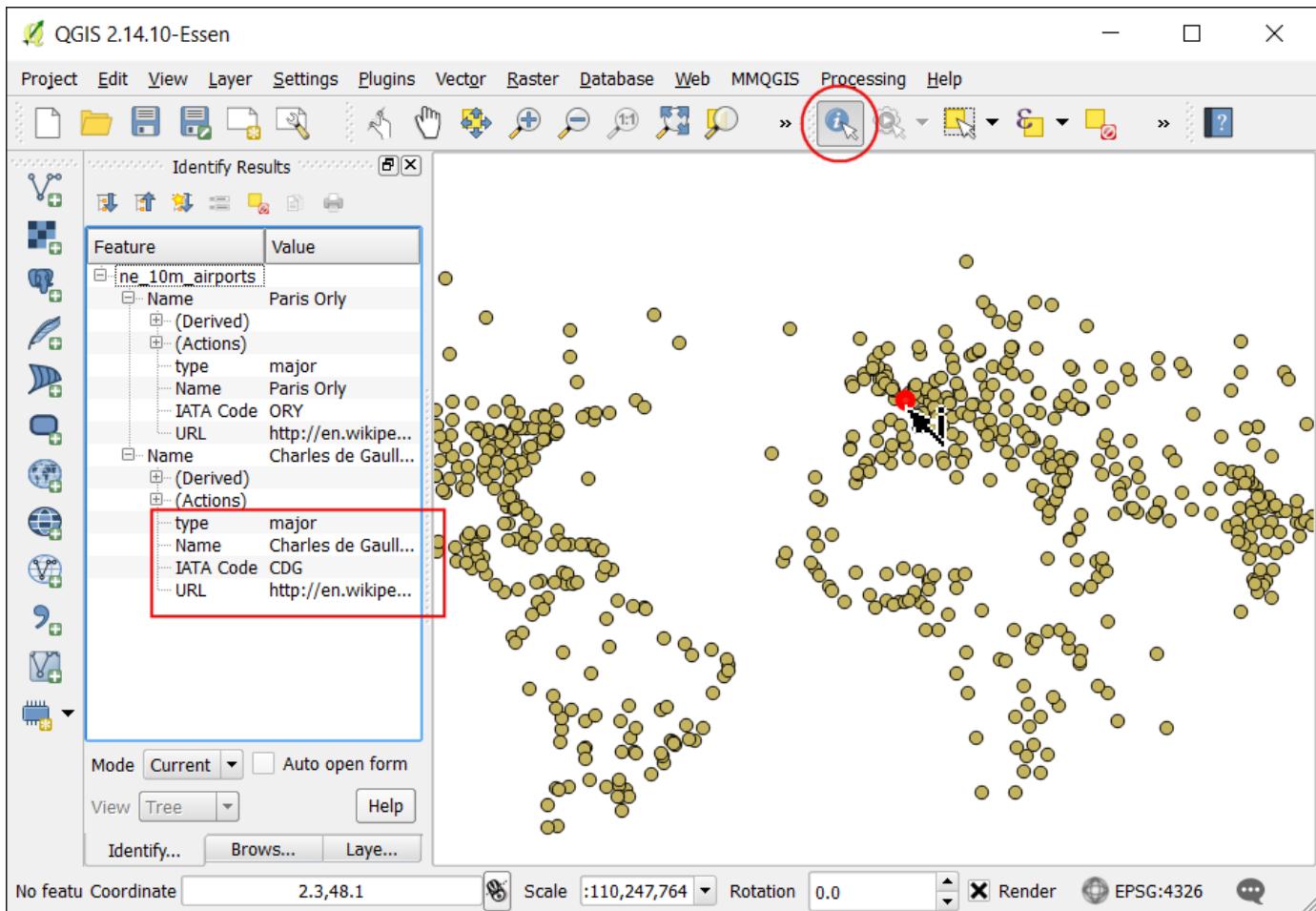
Name	Type	Type name	Length	Precision	Comment	Edit widget	Alias	WM
scalerank	int	Integer	4	0		Hidden		X
featurecla	QString	String	80	0		Hidden		X
type	QString	String	50	0		Text Edit		X
name	QString	String	200	0		Text Edit	Name	X
abbrev	QString	String	4	0		Hidden		X
location	QString	String	50	0		Hidden		X
gps_code	QString	String	254	0		Hidden		X
iata_code	QString	String	254	0		Text Edit	IATA Code	X
wikipedia	QString	String	254	0		Web View	URL	X
natlscalerank	double	Real	7	3		Hidden		X

Relations

Suppress attribute form pop-up after feature creation Default

Style OK Cancel Apply Help

- Back in the main QGIS window, choose the Identify tool and click on any point. The Identify Results panel will display the nicely formatted attributes with the newly added aliases. You will notice that the hidden fields do not appear in the results.



9. While this method is useful, there is one limitation. We are not able to change the order of the fields. One way to overcome this limitation is to create a Virtual Field. In our case, if we wanted the `type` field to appear at the end of the info window, we can simply add a new virtual field at the end and hide the original `type` field. While we are at it - we can also use an expression to better format the `type` values. Right-click the `ne_10m_airports` layer and choose Properties. Go to the Fields tab and click Field Calculator.

Layer Properties - ne_10m_airports | Fields

Attribute editor layout: Autogenerate Python Init function

Fields

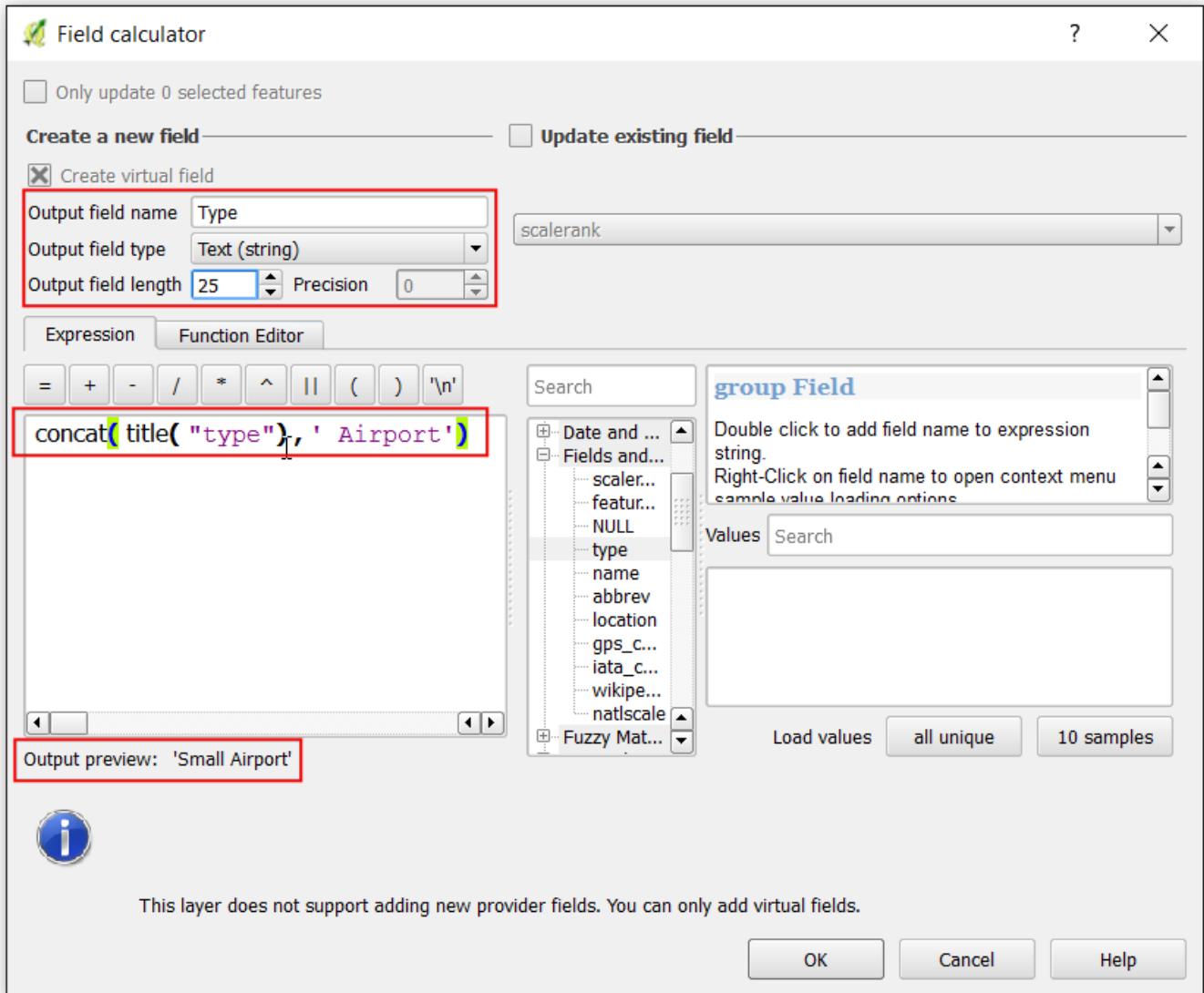
Type	Type	Field calculator	Length	Precision	Comment	Edit widget	Alias	WMS	WFS
int	Integer		4	0		Hidden		X	X
QString	String		80	0		Hidden		X	X
QString	String		50	0		Text Edit		X	X
QString	String		200	0		Text Edit	Name	X	X
QString	String		4	0		Hidden		X	X
QString	String		50	0		Hidden		X	X
QString	String		254	0		Hidden		X	X
QString	String		254	0		Text Edit	IATA Code	X	X
QString	String		254	0		Web View	URL	X	X
double	Real		7	3		Hidden		X	X

Relations

Suppress attribute form pop-up after feature creation

10. As the field names need to be unique, use `Type` as the new field name. Set the field type to `Text (String)` with a length of 25 characters. The field `type` contains values such as `small`, `mid`, `large` etc. We can add an expression to change the case of the words to sentence case and append the word `airport` for better readability. Enter the following expression in the Expression box and click OK.

```
concat( title("type"), ' Airport')
```



11. Now that we have a much better looking `Type` field, you can go ahead and set the Edit Widget for `Type` field to `Hidden`.

Layer Properties - ne_10m_airports | Fields

Attribute editor layout: Autogenerate Python Init function

Fields

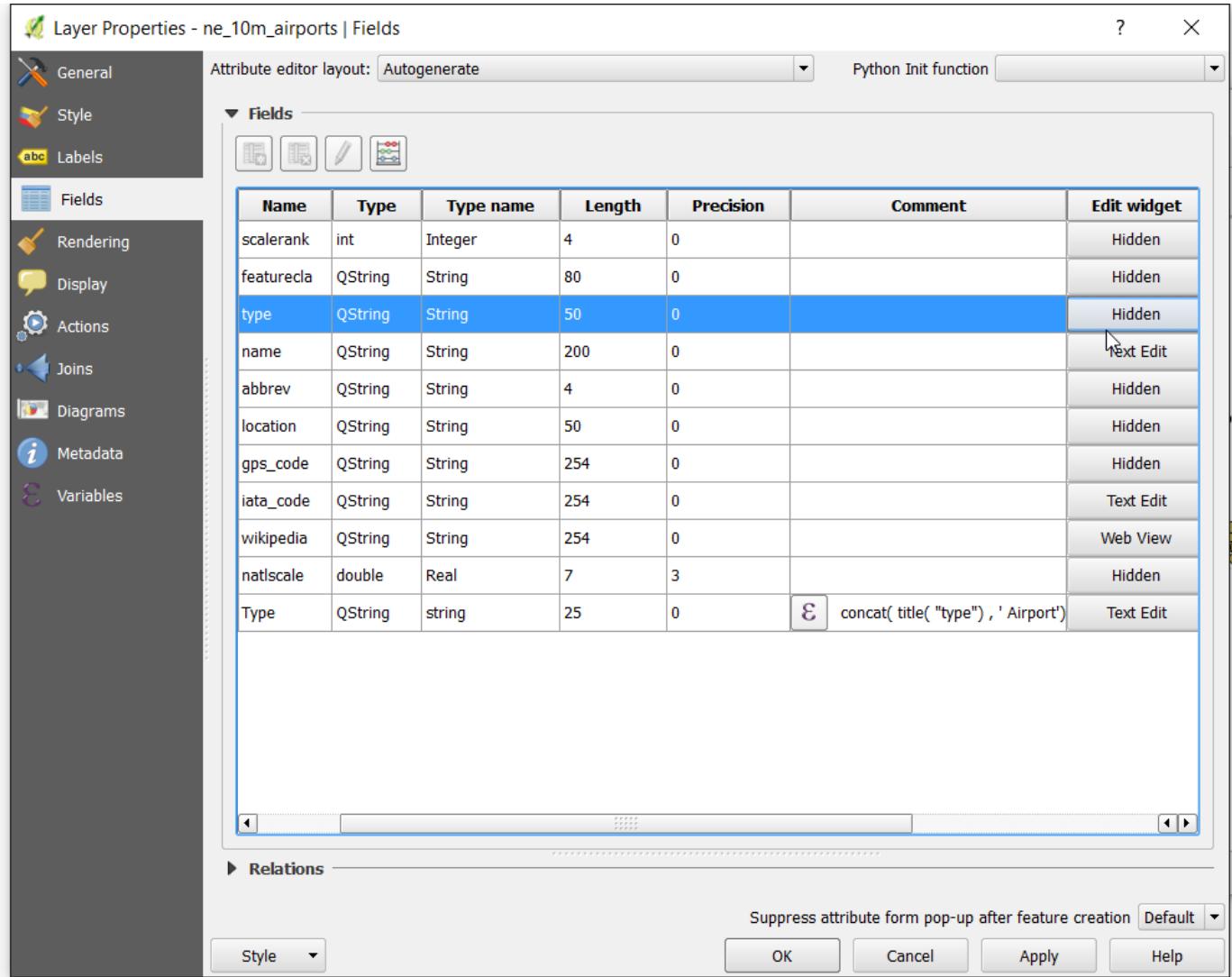
Name	Type	Type name	Length	Precision	Comment	Edit widget
scalerank	int	Integer	4	0		Hidden
featurecla	QString	String	80	0		Hidden
type	QString	String	50	0		Hidden
name	QString	String	200	0		Next Edit
abbrev	QString	String	4	0		Hidden
location	QString	String	50	0		Hidden
gps_code	QString	String	254	0		Hidden
iata_code	QString	String	254	0		Text Edit
wikipedia	QString	String	254	0		Web View
natlscalerank	double	Real	7	3		Hidden
Type	QString	string	25	0	 concat(title("type"), ' Airport')	Text Edit

Relations

Suppress attribute form pop-up after feature creation

Style

OK Cancel Apply Help



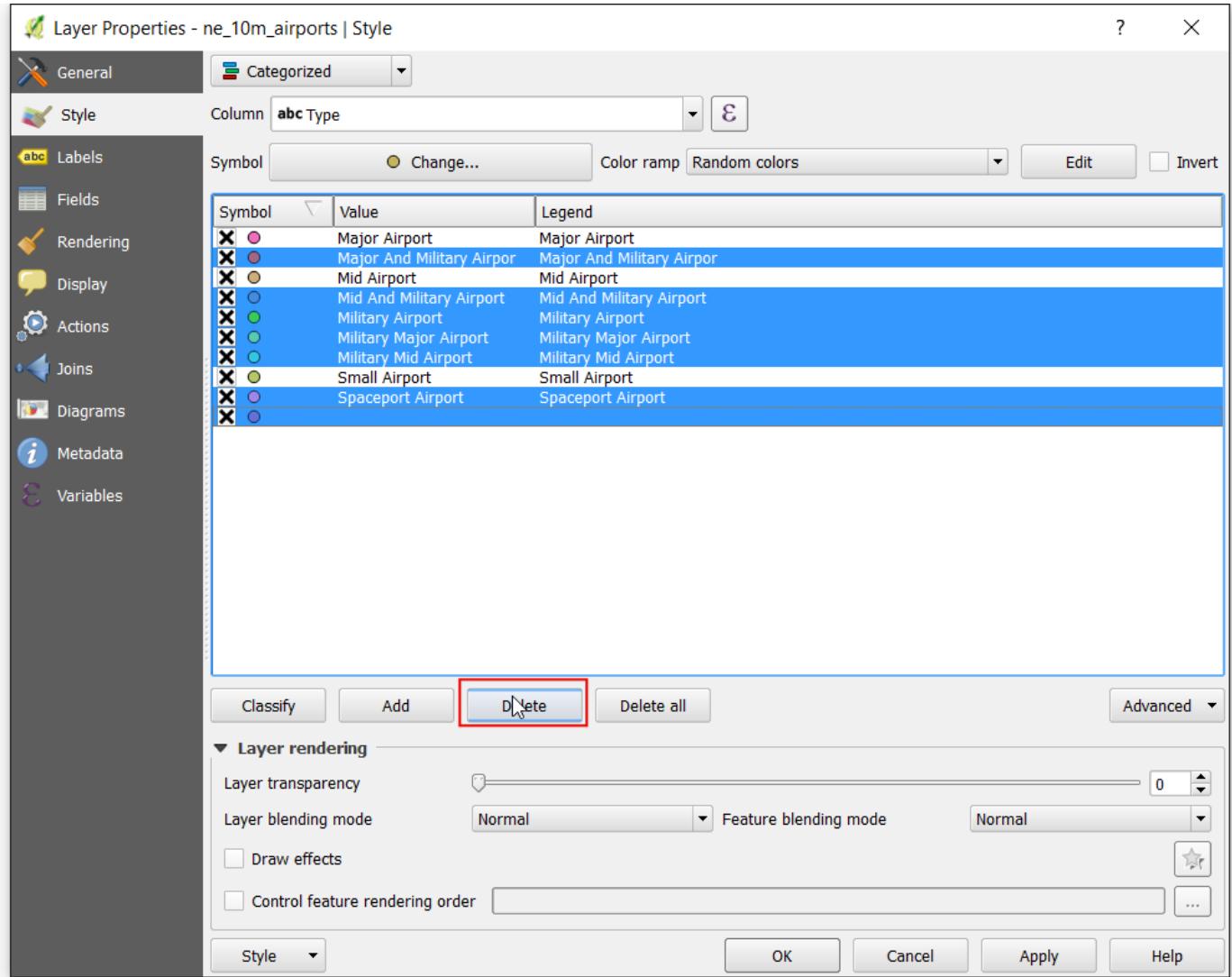
12. Use the Identify tool to verify that the attributes appear as expected.



13. Now let's style our layer to be more visually appealing and informative. Right-click the `ne_10m_airports` layer and select Properties. Switch to the Style tab. Choose Categorized style and our virtual field `Type` as the Column. Click Classify.



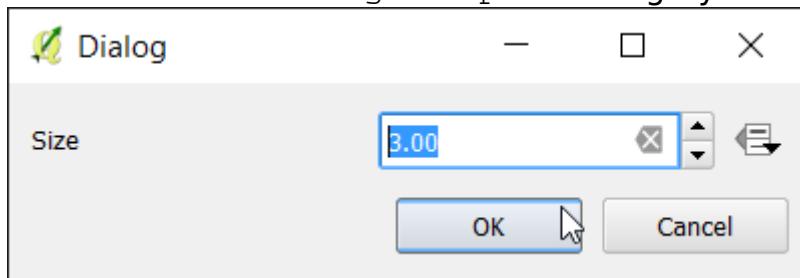
14. You will see a different colored circle gets assigned to a differnt type of airport. For the purpose of this tutorial, we will restrict the map to civilian airports. Hold the Ctrl key and select all categories for military airports. Once selected, click Delete.



15. Apart from assigning a different color to the category, we can change the size of the symbol to visually help our users distinguish different type of airports. Right-click on a category and select Change size.



16. Set the Size value to 3 for the Large Airport category.



17. Similarly, set the Size to 2 for Mid Airport and 1 for Small Airport.



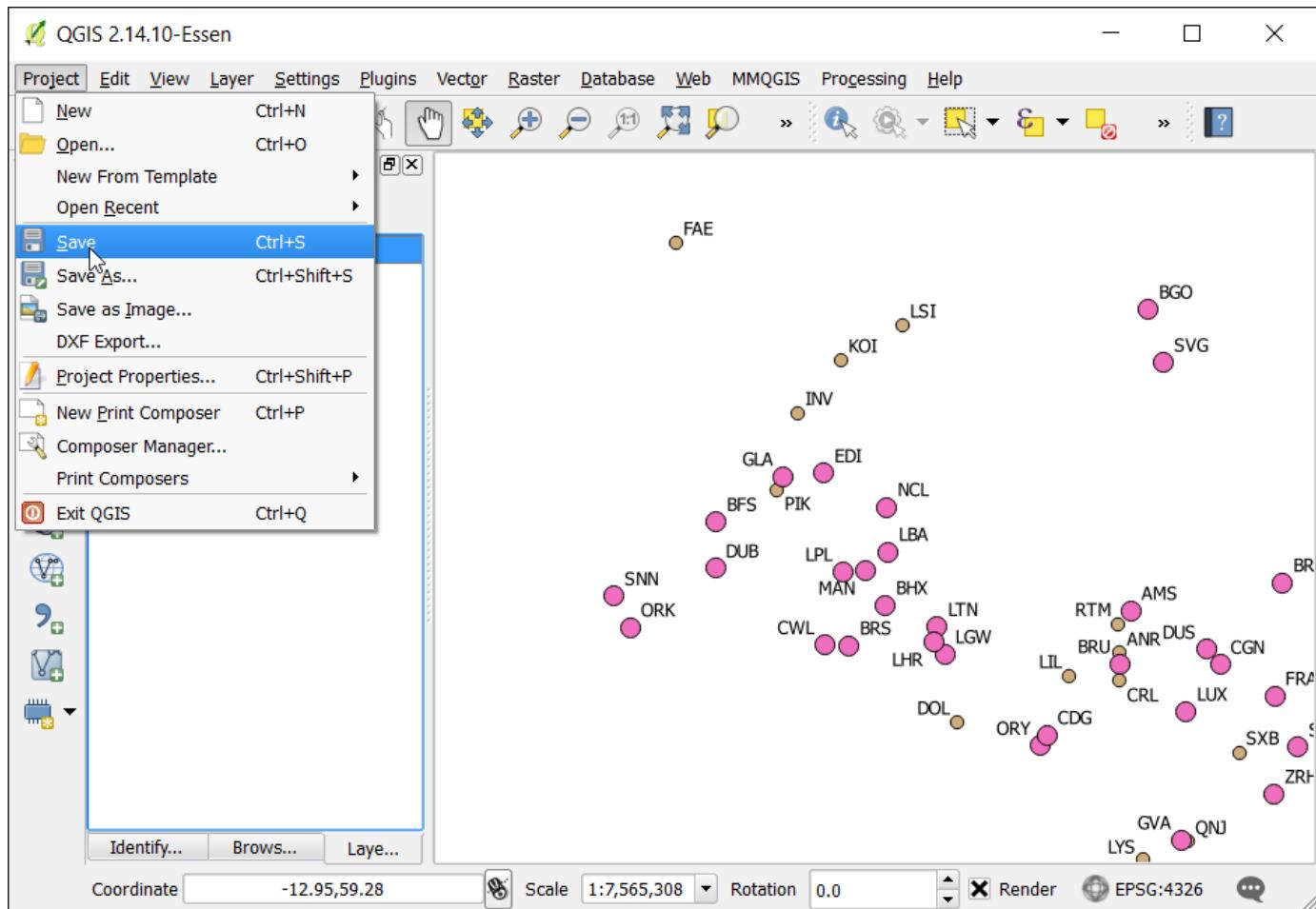
18. For a complete map, we also need to label each airport. Switch to the Labels tab in the Properties dialog. Select Show labels for this layer and choose `iata_code` as the value for Label with. We will also set Rendering option so that the labels only appear when the user is sufficiently zoomed in. Check Scale-based visibility under Label options. Enter 1 as the Minimum scale and 10000000 as the maximum scale. This setting will render the labels only after the user has zoomed in more than 1:10000000 scale and will be visible till 1:1 scale.



19. As we are using circles to depict the airports, we need to ensure that the labels don't overlap with the circles. Go to the Placement tab in the Labels dialog and set the Placement to Cartographic. Select From symbol bounds as Distance offset from. Click OK.



20. Our map is now ready. This is a good time to save our work. Go to Project ▶ Save. Enter Airports as the name of the project.



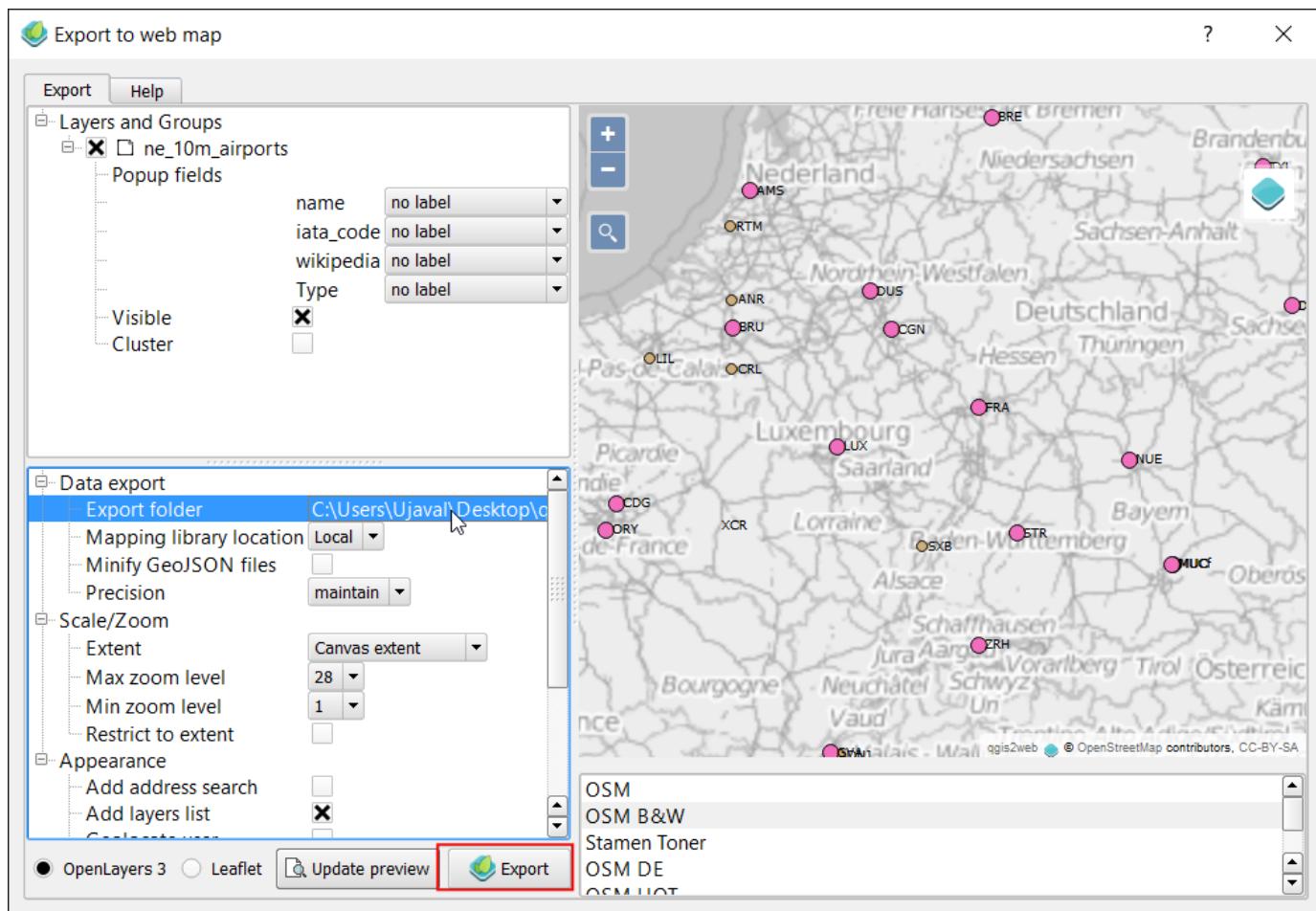
21. Now we are ready to export our project to a web map. Install the `qgis2web` plugin by going to Plugins > Manage and Install Plugin (See [Using Plugins](#) for more details on installing plugins in QGIS). Once the plugin is installed, go to Web > `qgis2web` > Create a web map.



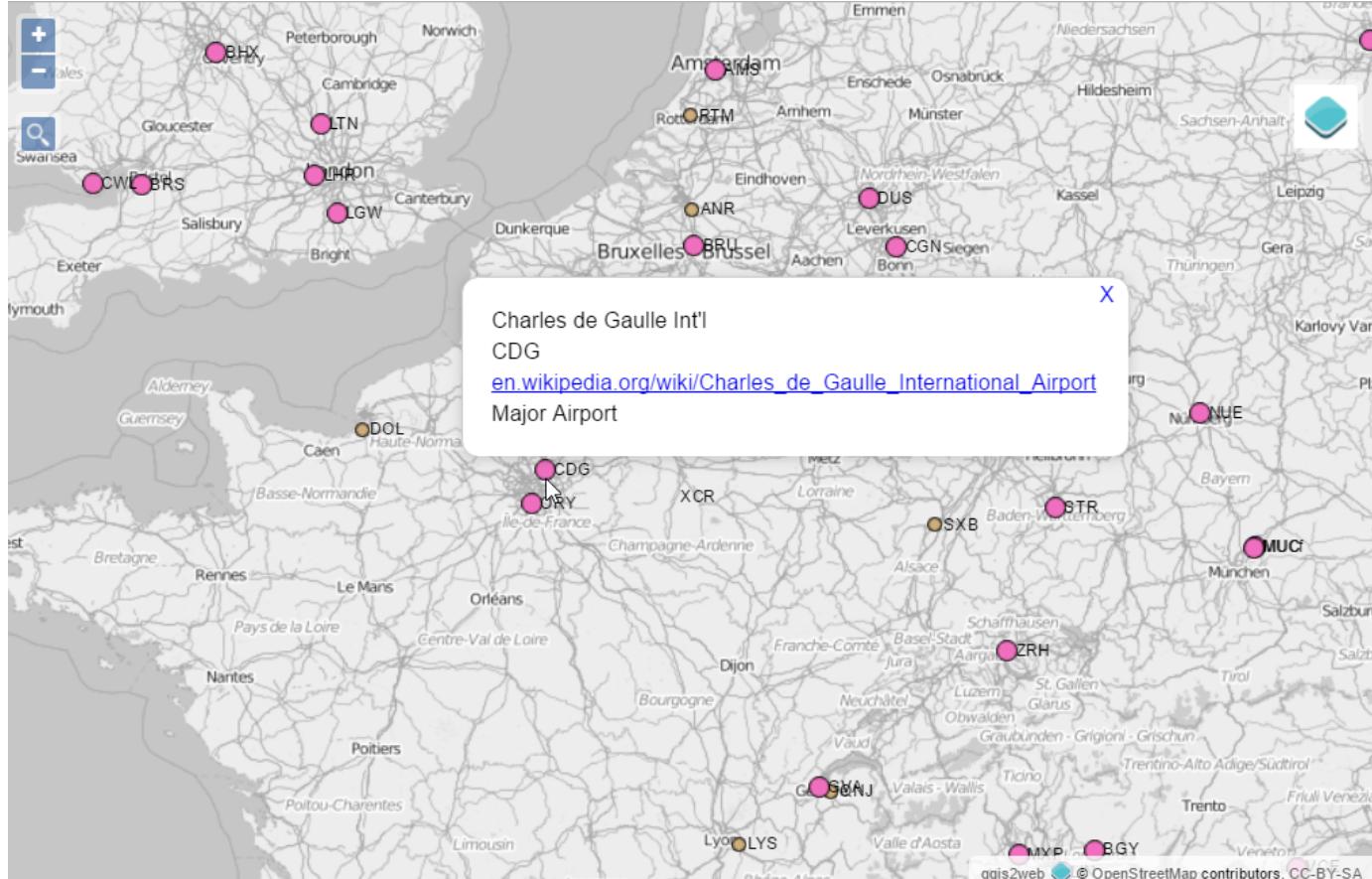
22. In the Export to web map dialog, check Add layers list in the bottom panel under the Appearance section. Also select ne_10m_airports: iata_code as the field for Label search. Check the Show popups on hover to allow display of info-windows on hover. We can also set a basemap so the users have more context when looking at the airports layer. Select OSM B&W to use a black-and-white themed basemap create using OpenStreetMap data. You also have an option to choose from either OpenLayers or Leaflet as the web mapping library. We will restrict this tutorial to use the OpenLayers library. Click Update Preview` to see how your exported map will look like. Before we do the actual export, we need to set the Export folder. You can select a folder of your choice and click Export.



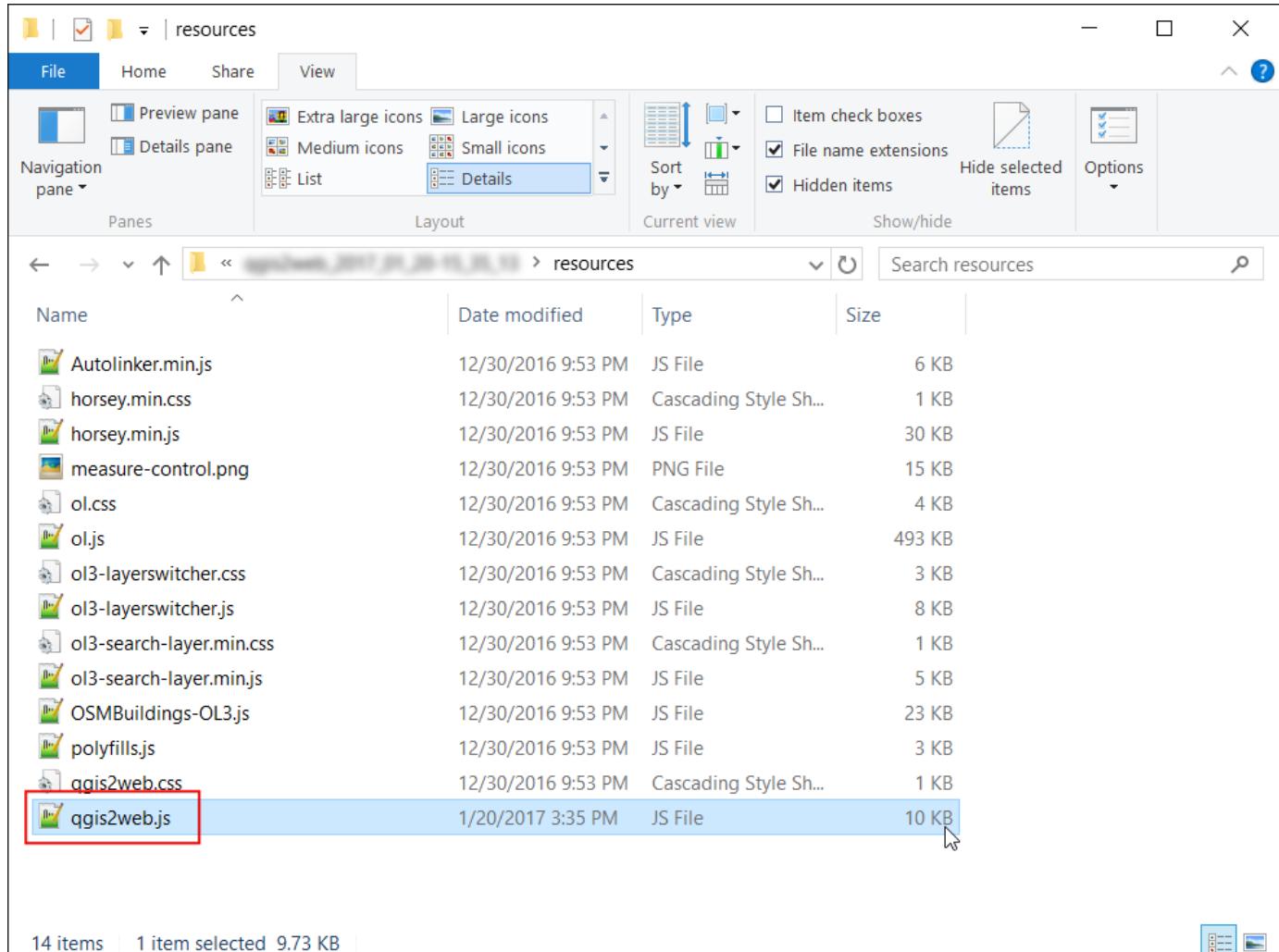
23. Once the export is complete, the default browser for your computer will open and show the interactive web map.



24. Your web map is now ready for publishing.



25. The `qgis2web` plugin has many limitations and it cannot do everything that the powerful web mapping libraries OpenLayers and Leaflet can do. This process can act as the starting point in your web mapping process and save you valuable time by creating a basic template from which you can further customize the web map. To highlight the fact that the output created from this process can be readily changed to suit your requirement - we will make a simple change to the web map to zoom to a particular airport when the user initially loads the map. On your computer, go to the folder where the web map was exported. Locate the `resources` folder and open `qgis2web.js` file in a text editor.



26. Locate the line where the `map.getView().fit()` function is called and add the following code after that. This new line of code instructs the web browser to center the map on the coordinates of Paris. Save the changes to the `qgis2web.js` file.

```
map.getView().setCenter.ol.proj.fromLonLat([2.35, 48.85]))
```

The screenshot shows a Notepad++ window with the file "qgis2web.js" open. The code is written in JavaScript and defines a search layer for airports. The center of the map is set to Paris at coordinates [2.35, 48.85].

```
resources\qgis2web.js - Notepad++  
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?  
qgis2web.js  
29     })  
30  } );  
31  
32  var searchLayer = new ol.SearchLayer({  
33      layer: lyr_ne_10m_airports,  
34      colName: 'iata_code',  
35      zoom: 10,  
36      collapsed: true,  
37      map: map  
38  });  
39  
40  map.addControl(searchLayer);  
41  map.getView().fit([-1519627 389624, 5724361 148262, 1068882 161610, 9457802.6828]  
42  map.getView().setCenter(ol.proj.fromLonLat([2.35, 48.85]))  
43  
44  var NO_POPUP = 0  
45  var ALL_FIELDS = 1  
46  
47  /**  
48   * Returns either NO_POPUP, ALL_FIELDS or the name of a single field to use for  
49   * a given layer  
50   * @param {Array} layerList List of ol.Layer instances  
51   */  
52  function getPopupField(layerList) {  
53    if (layerList.length === 1) {  
54      return layerList[0].get('field') || NO_POPUP;  
55    } else {  
56      return ALL_FIELDS;  
57    }  
58  }  
59  
60  // Add a search control to the map.  
61  var searchControl = new ol.control.Search({  
62    layer: lyr_ne_10m_airports,  
63    title: 'Search Airports'  
64  });  
65  map.addControl(searchControl);  
66  
67  // Set the map's center to Paris.  
68  map.setView({  
69    center: [2.35, 48.85],  
70    zoom: 10  
71  });  
72  
73  // Set the map's projection to WGS 84.  
74  map.setProjection('EPSG:4326');  
75  
76  // Set the map's extent to cover the world.  
77  map.setExtent([-180, -85, 180, 85]);  
78  
79  // Set the map's resolution.  
80  map.setResolution(150000);  
81  
82  // Set the map's rotation.  
83  map.setRotation(0);  
84  
85  // Set the map's scale.  
86  map.setScale(1);  
87  
88  // Set the map's style.  
89  map.setStyle('OpenStreetMap');  
90  
91  // Set the map's attribution.  
92  map.setAttribution('OpenStreetMap');  
93  
94  // Set the map's controls.  
95  map.setControls(true);  
96  
97  // Set the map's interactions.  
98  map.setInteractions(true);  
99  
100 // Set the map's layers.  
101 map.setLayers([lyr_ne_10m_airports]);  
102  
103 // Set the map's view.  
104 map.setView({  
105   center: [2.35, 48.85],  
106   zoom: 10  
107 });  
108  
109 // Set the map's projection.  
110 map.setProjection('EPSG:4326');  
111  
112 // Set the map's extent.  
113 map.setExtent([-180, -85, 180, 85]);  
114  
115 // Set the map's resolution.  
116 map.setResolution(150000);  
117  
118 // Set the map's rotation.  
119 map.setRotation(0);  
120  
121 // Set the map's scale.  
122 map.setScale(1);  
123  
124 // Set the map's style.  
125 map.setStyle('OpenStreetMap');  
126  
127 // Set the map's attribution.  
128 map.setAttribution('OpenStreetMap');  
129  
130 // Set the map's controls.  
131 map.setControls(true);  
132  
133 // Set the map's interactions.  
134 map.setInteractions(true);  
135  
136 // Set the map's layers.  
137 map.setLayers([lyr_ne_10m_airports]);  
138  
139 // Set the map's view.  
140 map.setView({  
141   center: [2.35, 48.85],  
142   zoom: 10  
143 });  
144  
145 // Set the map's projection.  
146 map.setProjection('EPSG:4326');  
147  
148 // Set the map's extent.  
149 map.setExtent([-180, -85, 180, 85]);  
150  
151 // Set the map's resolution.  
152 map.setResolution(150000);  
153  
154 // Set the map's rotation.  
155 map.setRotation(0);  
156  
157 // Set the map's scale.  
158 map.setScale(1);  
159  
160 // Set the map's style.  
161 map.setStyle('OpenStreetMap');  
162  
163 // Set the map's attribution.  
164 map.setAttribution('OpenStreetMap');  
165  
166 // Set the map's controls.  
167 map.setControls(true);  
168  
169 // Set the map's interactions.  
170 map.setInteractions(true);  
171  
172 // Set the map's layers.  
173 map.setLayers([lyr_ne_10m_airports]);  
174  
175 // Set the map's view.  
176 map.setView({  
177   center: [2.35, 48.85],  
178   zoom: 10  
179 });  
180  
181 // Set the map's projection.  
182 map.setProjection('EPSG:4326');  
183  
184 // Set the map's extent.  
185 map.setExtent([-180, -85, 180, 85]);  
186  
187 // Set the map's resolution.  
188 map.setResolution(150000);  
189  
190 // Set the map's rotation.  
191 map.setRotation(0);  
192  
193 // Set the map's scale.  
194 map.setScale(1);  
195  
196 // Set the map's style.  
197 map.setStyle('OpenStreetMap');  
198  
199 // Set the map's attribution.  
200 map.setAttribution('OpenStreetMap');  
201  
202 // Set the map's controls.  
203 map.setControls(true);  
204  
205 // Set the map's interactions.  
206 map.setInteractions(true);  
207  
208 // Set the map's layers.  
209 map.setLayers([lyr_ne_10m_airports]);  
210  
211 // Set the map's view.  
212 map.setView({  
213   center: [2.35, 48.85],  
214   zoom: 10  
215 });  
216  
217 // Set the map's projection.  
218 map.setProjection('EPSG:4326');  
219  
220 // Set the map's extent.  
221 map.setExtent([-180, -85, 180, 85]);  
222  
223 // Set the map's resolution.  
224 map.setResolution(150000);  
225  
226 // Set the map's rotation.  
227 map.setRotation(0);  
228  
229 // Set the map's scale.  
230 map.setScale(1);  
231  
232 // Set the map's style.  
233 map.setStyle('OpenStreetMap');  
234  
235 // Set the map's attribution.  
236 map.setAttribution('OpenStreetMap');  
237  
238 // Set the map's controls.  
239 map.setControls(true);  
240  
241 // Set the map's interactions.  
242 map.setInteractions(true);  
243  
244 // Set the map's layers.  
245 map.setLayers([lyr_ne_10m_airports]);  
246  
247 // Set the map's view.  
248 map.setView({  
249   center: [2.35, 48.85],  
250   zoom: 10  
251 });  
252  
253 // Set the map's projection.  
254 map.setProjection('EPSG:4326');  
255  
256 // Set the map's extent.  
257 map.setExtent([-180, -85, 180, 85]);  
258  
259 // Set the map's resolution.  
260 map.setResolution(150000);  
261  
262 // Set the map's rotation.  
263 map.setRotation(0);  
264  
265 // Set the map's scale.  
266 map.setScale(1);  
267  
268 // Set the map's style.  
269 map.setStyle('OpenStreetMap');  
270  
271 // Set the map's attribution.  
272 map.setAttribution('OpenStreetMap');  
273  
274 // Set the map's controls.  
275 map.setControls(true);  
276  
277 // Set the map's interactions.  
278 map.setInteractions(true);  
279  
280 // Set the map's layers.  
281 map.setLayers([lyr_ne_10m_airports]);  
282  
283 // Set the map's view.  
284 map.setView({  
285   center: [2.35, 48.85],  
286   zoom: 10  
287 });  
288  
289 // Set the map's projection.  
290 map.setProjection('EPSG:4326');  
291  
292 // Set the map's extent.  
293 map.setExtent([-180, -85, 180, 85]);  
294  
295 // Set the map's resolution.  
296 map.setResolution(150000);  
297  
298 // Set the map's rotation.  
299 map.setRotation(0);  
300  
301 // Set the map's scale.  
302 map.setScale(1);  
303  
304 // Set the map's style.  
305 map.setStyle('OpenStreetMap');  
306  
307 // Set the map's attribution.  
308 map.setAttribution('OpenStreetMap');  
309  
310 // Set the map's controls.  
311 map.setControls(true);  
312  
313 // Set the map's interactions.  
314 map.setInteractions(true);  
315  
316 // Set the map's layers.  
317 map.setLayers([lyr_ne_10m_airports]);  
318  
319 // Set the map's view.  
320 map.setView({  
321   center: [2.35, 48.85],  
322   zoom: 10  
323 });  
324  
325 // Set the map's projection.  
326 map.setProjection('EPSG:4326');  
327  
328 // Set the map's extent.  
329 map.setExtent([-180, -85, 180, 85]);  
330  
331 // Set the map's resolution.  
332 map.setResolution(150000);  
333  
334 // Set the map's rotation.  
335 map.setRotation(0);  
336  
337 // Set the map's scale.  
338 map.setScale(1);  
339  
340 // Set the map's style.  
341 map.setStyle('OpenStreetMap');  
342  
343 // Set the map's attribution.  
344 map.setAttribution('OpenStreetMap');  
345  
346 // Set the map's controls.  
347 map.setControls(true);  
348  
349 // Set the map's interactions.  
350 map.setInteractions(true);  
351  
352 // Set the map's layers.  
353 map.setLayers([lyr_ne_10m_airports]);  
354  
355 // Set the map's view.  
356 map.setView({  
357   center: [2.35, 48.85],  
358   zoom: 10  
359 });  
360  
361 // Set the map's projection.  
362 map.setProjection('EPSG:4326');  
363  
364 // Set the map's extent.  
365 map.setExtent([-180, -85, 180, 85]);  
366  
367 // Set the map's resolution.  
368 map.setResolution(150000);  
369  
370 // Set the map's rotation.  
371 map.setRotation(0);  
372  
373 // Set the map's scale.  
374 map.setScale(1);  
375  
376 // Set the map's style.  
377 map.setStyle('OpenStreetMap');  
378  
379 // Set the map's attribution.  
380 map.setAttribution('OpenStreetMap');  
381  
382 // Set the map's controls.  
383 map.setControls(true);  
384  
385 // Set the map's interactions.  
386 map.setInteractions(true);  
387  
388 // Set the map's layers.  
389 map.setLayers([lyr_ne_10m_airports]);  
390  
391 // Set the map's view.  
392 map.setView({  
393   center: [2.35, 48.85],  
394   zoom: 10  
395 });  
396  
397 // Set the map's projection.  
398 map.setProjection('EPSG:4326');  
399  
400 // Set the map's extent.  
401 map.setExtent([-180, -85, 180, 85]);  
402  
403 // Set the map's resolution.  
404 map.setResolution(150000);  
405  
406 // Set the map's rotation.  
407 map.setRotation(0);  
408  
409 // Set the map's scale.  
410 map.setScale(1);  
411  
412 // Set the map's style.  
413 map.setStyle('OpenStreetMap');  
414  
415 // Set the map's attribution.  
416 map.setAttribution('OpenStreetMap');  
417  
418 // Set the map's controls.  
419 map.setControls(true);  
420  
421 // Set the map's interactions.  
422 map.setInteractions(true);  
423  
424 // Set the map's layers.  
425 map.setLayers([lyr_ne_10m_airports]);  
426  
427 // Set the map's view.  
428 map.setView({  
429   center: [2.35, 48.85],  
430   zoom: 10  
431 });  
432  
433 // Set the map's projection.  
434 map.setProjection('EPSG:4326');  
435  
436 // Set the map's extent.  
437 map.setExtent([-180, -85, 180, 85]);  
438  
439 // Set the map's resolution.  
440 map.setResolution(150000);  
441  
442 // Set the map's rotation.  
443 map.setRotation(0);  
444  
445 // Set the map's scale.  
446 map.setScale(1);  
447  
448 // Set the map's style.  
449 map.setStyle('OpenStreetMap');  
450  
451 // Set the map's attribution.  
452 map.setAttribution('OpenStreetMap');  
453  
454 // Set the map's controls.  
455 map.setControls(true);  
456  
457 // Set the map's interactions.  
458 map.setInteractions(true);  
459  
460 // Set the map's layers.  
461 map.setLayers([lyr_ne_10m_airports]);  
462  
463 // Set the map's view.  
464 map.setView({  
465   center: [2.35, 48.85],  
466   zoom: 10  
467 });  
468  
469 // Set the map's projection.  
470 map.setProjection('EPSG:4326');  
471  
472 // Set the map's extent.  
473 map.setExtent([-180, -85, 180, 85]);  
474  
475 // Set the map's resolution.  
476 map.setResolution(150000);  
477  
478 // Set the map's rotation.  
479 map.setRotation(0);  
480  
481 // Set the map's scale.  
482 map.setScale(1);  
483  
484 // Set the map's style.  
485 map.setStyle('OpenStreetMap');  
486  
487 // Set the map's attribution.  
488 map.setAttribution('OpenStreetMap');  
489  
490 // Set the map's controls.  
491 map.setControls(true);  
492  
493 // Set the map's interactions.  
494 map.setInteractions(true);  
495  
496 // Set the map's layers.  
497 map.setLayers([lyr_ne_10m_airports]);  
498  
499 // Set the map's view.  
500 map.setView({  
501   center: [2.35, 48.85],  
502   zoom: 10  
503 });  
504  
505 // Set the map's projection.  
506 map.setProjection('EPSG:4326');  
507  
508 // Set the map's extent.  
509 map.setExtent([-180, -85, 180, 85]);  
510  
511 // Set the map's resolution.  
512 map.setResolution(150000);  
513  
514 // Set the map's rotation.  
515 map.setRotation(0);  
516  
517 // Set the map's scale.  
518 map.setScale(1);  
519  
520 // Set the map's style.  
521 map.setStyle('OpenStreetMap');  
522  
523 // Set the map's attribution.  
524 map.setAttribution('OpenStreetMap');  
525  
526 // Set the map's controls.  
527 map.setControls(true);  
528  
529 // Set the map's interactions.  
530 map.setInteractions(true);  
531  
532 // Set the map's layers.  
533 map.setLayers([lyr_ne_10m_airports]);  
534  
535 // Set the map's view.  
536 map.setView({  
537   center: [2.35, 48.85],  
538   zoom: 10  
539 });  
540  
541 // Set the map's projection.  
542 map.setProjection('EPSG:4326');  
543  
544 // Set the map's extent.  
545 map.setExtent([-180, -85, 180, 85]);  
546  
547 // Set the map's resolution.  
548 map.setResolution(150000);  
549  
550 // Set the map's rotation.  
551 map.setRotation(0);  
552  
553 // Set the map's scale.  
554 map.setScale(1);  
555  
556 // Set the map's style.  
557 map.setStyle('OpenStreetMap');  
558  
559 // Set the map's attribution.  
560 map.setAttribution('OpenStreetMap');  
561  
562 // Set the map's controls.  
563 map.setControls(true);  
564  
565 // Set the map's interactions.  
566 map.setInteractions(true);  
567  
568 // Set the map's layers.  
569 map.setLayers([lyr_ne_10m_airports]);  
570  
571 // Set the map's view.  
572 map.setView({  
573   center: [2.35, 48.85],  
574   zoom: 10  
575 });  
576  
577 // Set the map's projection.  
578 map.setProjection('EPSG:4326');  
579  
580 // Set the map's extent.  
581 map.setExtent([-180, -85, 180, 85]);  
582  
583 // Set the map's resolution.  
584 map.setResolution(150000);  
585  
586 // Set the map's rotation.  
587 map.setRotation(0);  
588  
589 // Set the map's scale.  
590 map.setScale(1);  
591  
592 // Set the map's style.  
593 map.setStyle('OpenStreetMap');  
594  
595 // Set the map's attribution.  
596 map.setAttribution('OpenStreetMap');  
597  
598 // Set the map's controls.  
599 map.setControls(true);  
600  
601 // Set the map's interactions.  
602 map.setInteractions(true);  
603  
604 // Set the map's layers.  
605 map.setLayers([lyr_ne_10m_airports]);  
606  
607 // Set the map's view.  
608 map.setView({  
609   center: [2.35, 48.85],  
610   zoom: 10  
611 });  
612  
613 // Set the map's projection.  
614 map.setProjection('EPSG:4326');  
615  
616 // Set the map's extent.  
617 map.setExtent([-180, -85, 180, 85]);  
618  
619 // Set the map's resolution.  
620 map.setResolution(150000);  
621  
622 // Set the map's rotation.  
623 map.setRotation(0);  
624  
625 // Set the map's scale.  
626 map.setScale(1);  
627  
628 // Set the map's style.  
629 map.setStyle('OpenStreetMap');  
630  
631 // Set the map's attribution.  
632 map.setAttribution('OpenStreetMap');  
633  
634 // Set the map's controls.  
635 map.setControls(true);  
636  
637 // Set the map's interactions.  
638 map.setInteractions(true);  
639  
640 // Set the map's layers.  
641 map.setLayers([lyr_ne_10m_airports]);  
642  
643 // Set the map's view.  
644 map.setView({  
645   center: [2.35, 48.85],  
646   zoom: 10  
647 });  
648  
649 // Set the map's projection.  
650 map.setProjection('EPSG:4326');  
651  
652 // Set the map's extent.  
653 map.setExtent([-180, -85, 180, 85]);  
654  
655 // Set the map's resolution.  
656 map.setResolution(150000);  
657  
658 // Set the map's rotation.  
659 map.setRotation(0);  
660  
661 // Set the map's scale.  
662 map.setScale(1);  
663  
664 // Set the map's style.  
665 map.setStyle('OpenStreetMap');  
666  
667 // Set the map's attribution.  
668 map.setAttribution('OpenStreetMap');  
669  
670 // Set the map's controls.  
671 map.setControls(true);  
672  
673 // Set the map's interactions.  
674 map.setInteractions(true);  
675  
676 // Set the map's layers.  
677 map.setLayers([lyr_ne_10m_airports]);  
678  
679 // Set the map's view.  
680 map.setView({  
681   center: [2.35, 48.85],  
682   zoom: 10  
683 });  
684  
685 // Set the map's projection.  
686 map.setProjection('EPSG:4326');  
687  
688 // Set the map's extent.  
689 map.setExtent([-180, -85, 180, 85]);  
690  
691 // Set the map's resolution.  
692 map.setResolution(150000);  
693  
694 // Set the map's rotation.  
695 map.setRotation(0);  
696  
697 // Set the map's scale.  
698 map.setScale(1);  
699  
700 // Set the map's style.  
701 map.setStyle('OpenStreetMap');  
702  
703 // Set the map's attribution.  
704 map.setAttribution('OpenStreetMap');  
705  
706 // Set the map's controls.  
707 map.setControls(true);  
708  
709 // Set the map's interactions.  
710 map.setInteractions(true);  
711  
712 // Set the map's layers.  
713 map.setLayers([lyr_ne_10m_airports]);  
714  
715 // Set the map's view.  
716 map.setView({  
717   center: [2.35, 48.85],  
718   zoom: 10  
719 });  
720  
721 // Set the map's projection.  
722 map.setProjection('EPSG:4326');  
723  
724 // Set the map's extent.  
725 map.setExtent([-180, -85, 180, 85]);  
726  
727 // Set the map's resolution.  
728 map.setResolution(150000);  
729  
730 // Set the map's rotation.  
731 map.setRotation(0);  
732  
733 // Set the map's scale.  
734 map.setScale(1);  
735  
736 // Set the map's style.  
737 map.setStyle('OpenStreetMap');  
738  
739 // Set the map's attribution.  
740 map.setAttribution('OpenStreetMap');  
741  
742 // Set the map's controls.  
743 map.setControls(true);  
744  
745 // Set the map's interactions.  
746 map.setInteractions(true);  
747  
748 // Set the map's layers.  
749 map.setLayers([lyr_ne_10m_airports]);  
750  
751 // Set the map's view.  
752 map.setView({  
753   center: [2.35, 48.85],  
754   zoom: 10  
755 });  
756  
757 // Set the map's projection.  
758 map.setProjection('EPSG:4326');  
759  
760 // Set the map's extent.  
761 map.setExtent([-180, -85, 180, 85]);  
762  
763 // Set the map's resolution.  
764 map.setResolution(150000);  
765  
766 // Set the map's rotation.  
767 map.setRotation(0);  
768  
769 // Set the map's scale.  
770 map.setScale(1);  
771  
772 // Set the map's style.  
773 map.setStyle('OpenStreetMap');  
774  
775 // Set the map's attribution.  
776 map.setAttribution('OpenStreetMap');  
777  
778 // Set the map's controls.  
779 map.setControls(true);  
780  
781 // Set the map's interactions.  
782 map.setInteractions(true);  
783  
784 // Set the map's layers.  
785 map.setLayers([lyr_ne_10m_airports]);  
786  
787 // Set the map's view.  
788 map.setView({  
789   center: [2.35, 48.85],  
790   zoom: 10  
791 });  
792  
793 // Set the map's projection.  
794 map.setProjection('EPSG:4326');  
795  
796 // Set the map's extent.  
797 map.setExtent([-180, -85, 180, 85]);  
798  
799 // Set the map's resolution.  
800 map.setResolution(150000);  
801  
802 // Set the map's rotation.  
803 map.setRotation(0);  
804  
805 // Set the map's scale.  
806 map.setScale(1);  
807  
808 // Set the map's style.  
809 map.setStyle('OpenStreetMap');  
810  
811 // Set the map's attribution.  
812 map.setAttribution('OpenStreetMap');  
813  
814 // Set the map's controls.  
815 map.setControls(true);  
816  
817 // Set the map's interactions.  
818 map.setInteractions(true);  
819  
820 // Set the map's layers.  
821 map.setLayers([lyr_ne_10m_airports]);  
822  
823 // Set the map's view.  
824 map.setView({  
825   center: [2.35, 48.85],  
826   zoom: 10  
827 });  
828  
829 // Set the map's projection.  
830 map.setProjection('EPSG:4326');  
831  
832 // Set the map's extent.  
833 map.setExtent([-180, -85, 180, 85]);  
834  
835 // Set the map's resolution.  
836 map.setResolution(150000);  
837  
838 // Set the map's rotation.  
839 map.setRotation(0);  
840  
841 // Set the map's scale.  
842 map.setScale(1);  
843  
844 // Set the map's style.  
845 map.setStyle('OpenStreetMap');  
846  
847 // Set the map's attribution.  
848 map.setAttribution('OpenStreetMap');  
849  
850 // Set the map's controls.  
851 map.setControls(true);  
852  
853 // Set the map's interactions.  
854 map.setInteractions(true);  
855  
856 // Set the map's layers.  
857 map.setLayers([lyr_ne_10m_airports]);  
858  
859 // Set the map's view.  
860 map.setView({  
861   center: [2.35, 48.85],  
862   zoom: 10  
863 });  
864  
865 // Set the map's projection.  
866 map.setProjection('EPSG:4326');
```

27. Refresh your browser and see that the web map will load with Paris at the center. This is a trivial example, but you can see how you can use any function available in the OpenLayers or Leaflet libraries to customize the web map.



28. The exported map resides on your computer. While you can see it in action, it is not very useful since you cannot share it anyone. For others to be able to see the map, you need to upload it to a web server. While the upload process will vary on the type of server you have access to - a cheap and easy way to publish your map on the web would be to use any of the public cloud storage services. [Amazon S3](#) is a popular storage service. You will need to sign up for an account and follow the instructions to create a bucket. Once a bucket is created, you can upload the contents of your exported folder to the bucket and set it to public. Here I created a bucket named `qgis-tutorials` and uploaded the contents of my exported folder to a sub-folder named `qgis2web`. You can access the resulting map at <http://s3.amazonaws.com/qgis-tutorials/qgis2web/index.html>

The screenshot shows the AWS S3 console interface. At the top, there are navigation links for 'Services' and 'Resource Groups', and a search bar labeled 'Search by prefix'. Below the header, there are buttons for 'Upload', 'Create Folder', 'Actions', and a dropdown set to 'None'. The main area displays a list of objects in the 'qgis2web' folder of the 'qgis-tutorials' bucket. The table has columns for 'Name', 'Storage Class', and 'S' (Size). The objects listed are:

	Name	Storage Class	S
<input type="checkbox"/>	index.html	Standard	1.9
<input type="checkbox"/>	layers	--	--
<input type="checkbox"/>	resources	--	--
<input type="checkbox"/>	styles	--	--

29. Similarly, Google also offers a cloud storage service called [Google Cloud Storage](#). Once you have created an account and enable billing, you can create a bucket and upload objects to the bucket. I create a bucket and sub-folder similar to Amazon and set the folder to public. The resulting map can be viewed at <https://storage.googleapis.com/qgis-tutorials/qgis2web/index.html>

Storage



Browser

Transfer

Settings

Browser

UPLOAD FILES



Buckets / qgis-tutorials / qgis2web

<input type="checkbox"/>	Name	Size	Type	Storage class
<input type="checkbox"/>	index.html	1.99 KB	text/html	Multi-Regional
<input type="checkbox"/>	layers/	—	Folder	—
<input type="checkbox"/>	resources/	—	Folder	—
<input type="checkbox"/>	styles/	—	Folder	—

