

# Towards GeoExt 3

Supporting both OpenLayers 3 and ExtJS 6

---



Marc Jansen & Christian Mayer

FOSS4G 2015, Seoul, South Korea, 2015-09-18

# Outline

- Intro & Meta
- A (short) history of GeoExt
- So what is GeoExt 3?
  - Objectives
  - State
  - Examples & features
- Outlook

# Intro & Meta

# Marc Jansen

- Since 2007 developer & project lead @ terrestris
- Core developer & member of PSC GeoExt
- Core developer OpenLayers
- Author of the German "OpenLayers" book
- OSGeo Foundation Charter Member
- ♥ OpenSource & GIS / Spatial



# terrestris



 @terrestris  
 @terrestrisde

- [terrestris.de](https://terrestris.de)
- OpenSource GIS from Bonn, Germany
- Development, Projects & Support/Teaching
- Consulting, Planning, Implementation & Maintenance

# Christian Mayer

- Software developer & architect
- Especially GIS / SDI
- Founder of meggsimum
- Core developer & member of PSC GeoExt
- OSGeo Foundation Charter Member
- Speaker at nat. & intern. conferences
- ♥ OpenSource & GIS / Spatial



# meggsum



- [meggsum.de](http://meggsum.de)
- Services around GIS
- Based in Germany
- Webmapping Solutions
- Software Planning and Development
- Consulting and Trainings

 [@meggsum](https://github.com/meggsum)

 [@meggsum](https://twitter.com/meggsum)

# GeoExt

- JavaScript framework for sophisticated WebGIS
- Based on OpenLayers and ExtJS
- Extends ExtJS with spatial components
- Embedding of spatial formats in ExtJS data-components
- Rich webmapping interfaces
- © OSGeo, OpenSource
- [First commit](#) on Mar 25, 2009

**GeoExt...**

**...is the marriage of ExtJS and OpenLayers**

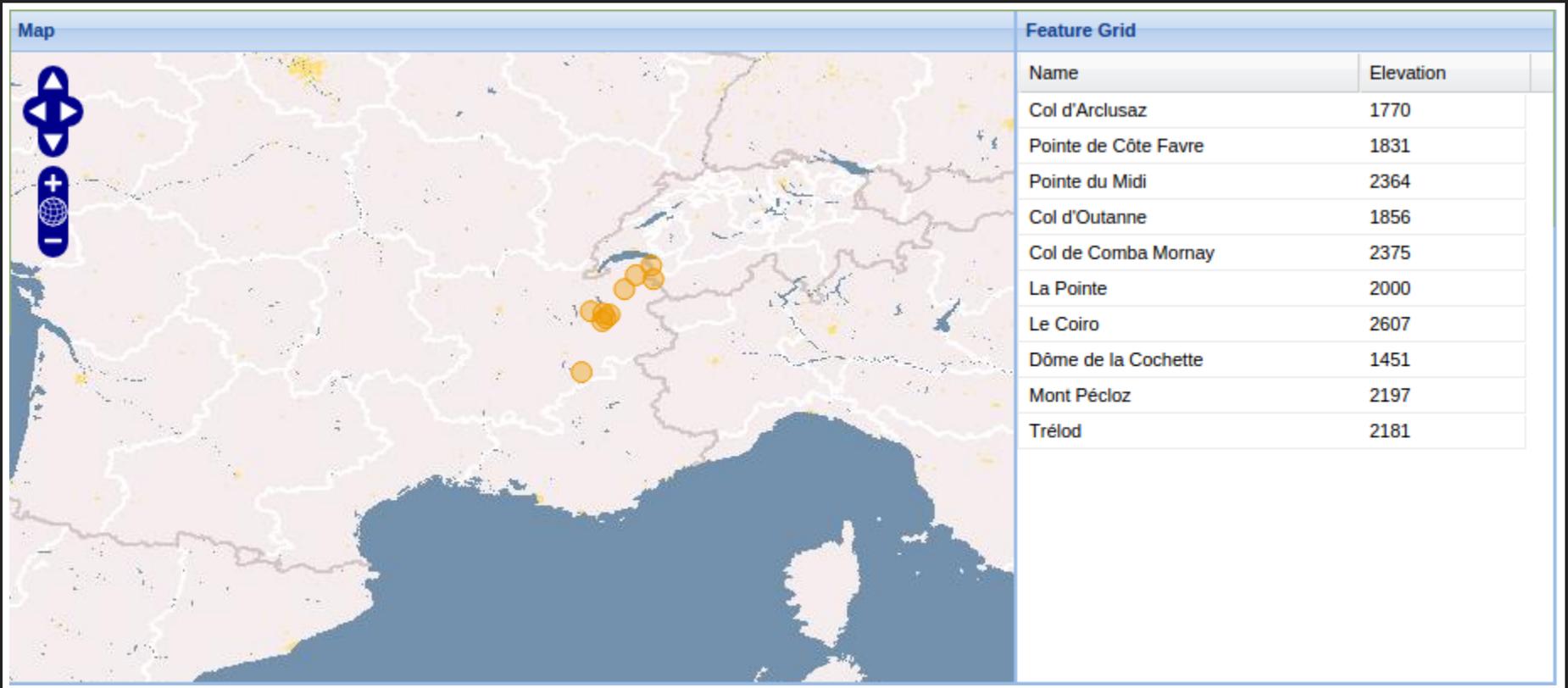
**...is the child of ExtJS and OpenLayers**

**...enhances both ExtJS and OpenLayers**

# A (short) history of GeoExt

# GeoExt 1.x

- Based on ExtJS 3.x & OpenLayers 2.x
- [geoext.org](http://geoext.org)



The screenshot displays the GeoExt 1.x web application interface. On the left, a map shows the Alpine region with several orange circular markers indicating specific locations. On the right, a 'Feature Grid' table lists the names and elevations of these locations. The map includes navigation controls such as a compass and zoom in/out buttons.

Name	Elevation
Col d'Arclusaz	1770
Pointe de Côte Favre	1831
Pointe du Midi	2364
Col d'Outanne	1856
Col de Comba Mornay	2375
La Pointe	2000
Le Coiro	2607
Dôme de la Cochette	1451
Mont Pécloz	2197
Trélod	2181

2007



2009



2012



2013



2014



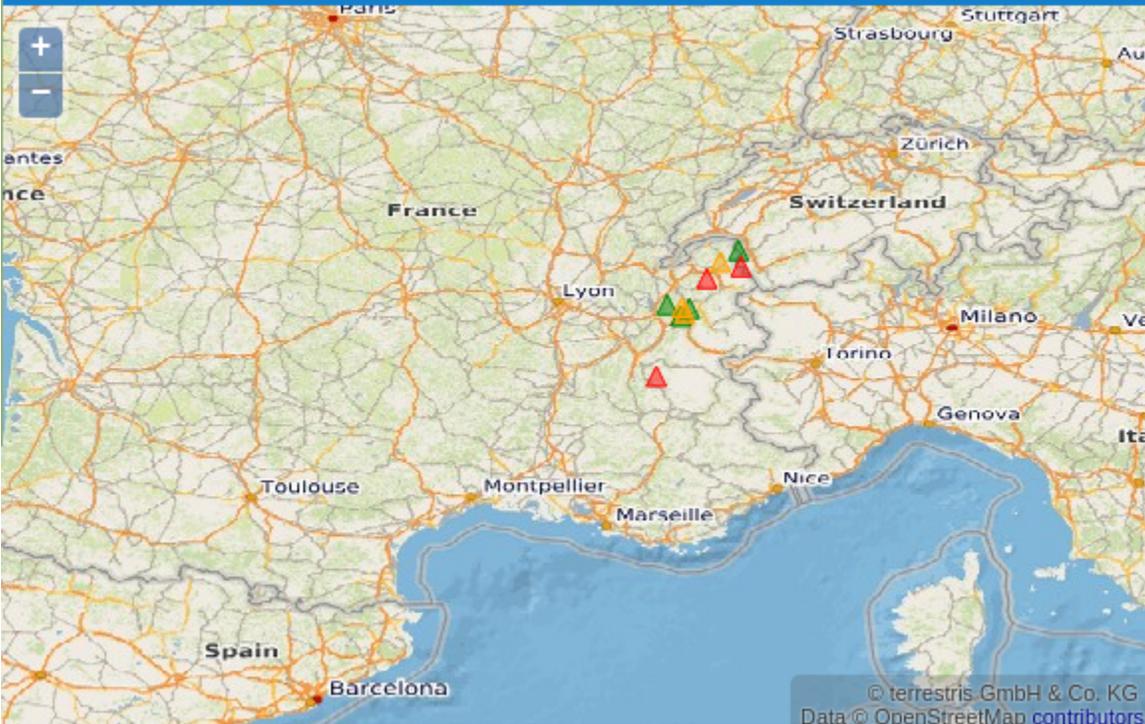
2015



# GeoExt 2.0.x

- Based on ExtJS 4.x & OpenLayers 2.x
- [geoext.github.io/geoext2](http://geoext.github.io/geoext2)

Map



The map displays the Alpine region between France, Switzerland, and Italy. Several elevation points are marked with colored triangles: green for higher elevations, red for medium, and yellow for lower. The points are clustered in the central Alps, near the French-Swiss border.

Feature Grid

	Name	Elevation
▲	Col d'Arclusaz	1770
▲	Pointe de Côte Favre	1831
▲	Pointe du Midi	2364
▲	Col d'Outanne	1856
▲	Col de Comba Mornay	2375
▲	La Pointe	2000
▲	Le Coiro	2607
▲	Dôme de la Cochette	1451
▲	Mont Pécloz	2197
▲	Trélod	2181

© terrestris GmbH & Co. KG  
Data © OpenStreetMap contributors

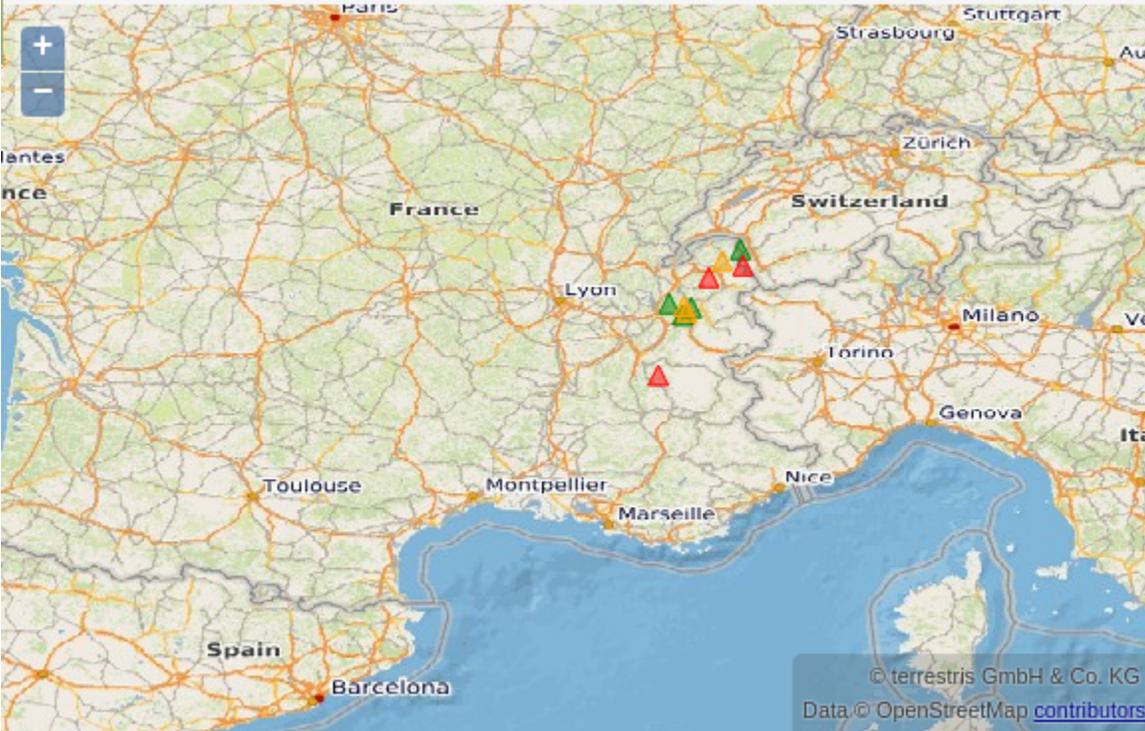
# GeoExt 2.0.x

- Major improvements
  - Support for new creation syntax
  - MVC support
  - Advanced app theming
  - Improved API-Docs
  - Compatibility to Sencha's build tools (dependency resolving, ...)

# GeoExt 2.1.x (beta)

- Based on ExtJS 4.2.x / ExtJS 5.1.x OpenLayers 2.x
- [geoext.github.io/geoext2](http://geoext.github.io/geoext2)

Map



The map displays the Alpine region between France, Switzerland, and Italy. Several elevation points are marked with colored triangles: green for higher elevations and red for lower ones. The points are clustered in the central Alps, near the French-Swiss border. Major cities like Paris, Lyon, Marseille, Nice, Strasbourg, Zürich, and Milan are labeled. The map includes a zoom control on the left and a copyright notice at the bottom: © terrestris GmbH & Co. KG, Data © OpenStreetMap contributors.

Feature Grid

	Name	Elevation
▲	Col d'Arclusaz	1770
▲	Pointe de Côte Favre	1831
▲	Pointe du Midi	2364
▲	Col d'Outanne	1856
▲	Col de Comba Mornay	2375
▲	La Pointe	2000
▲	Le Coiro	2607
▲	Dôme de la Cochette	1451
▲	Mont Pécloz	2197
▲	Trélod	2181

# GeoExt 2.1.x

- Major improvements
  - Support for two major ExtJS versions
  - MVVM support
  - Two-way-binding
  - Improved mobile / touch support
  - Responsive design

**In the meantime...**

**OpenLayers 3**

**&**

**ExtJS 6**

**...were born**

# GeoExt 3 Codesprint



- Jun 17 - Jun 19, 2015 in Bonn
- 10 developers from 4 countries
- Built foundation for GeoExt 3 with
  - OpenLayers 3
  - ExtJS 6



# Sponsors

- Bistum Eichstätt
- Boundless
- Bundesamt für Strahlenschutz
- Compass Informatics Ltd
- ISB AG
- Landesamt für Geoinformation und Landentwicklung  
Baden Württemberg
- Landplan AG
- meggsimum
- terrestris GmbH & Co. KG

**So what is GeoExt 3?**

# Objectives

- Start from scratch
- Benefit from sencha tooling (build / packaging)
- Benefit OpenLayers feature galore
- Unbiased about medium (desktop / mobile)
- Enable access of library objects (e.g. ol3)
- More examples, improved tests and documentation

# State

- [github.com/geoext/geoext3](https://github.com/geoext/geoext3)
- > 300 commits ✓
- 7 contributors ✓
- Build and packaging ✓
- 82% test-coverage ✓
- Nice API-docs ✓
- Some examples ✓
- BSD to GPLv3 ✓
- Universal app example ✗
- 0 releases ✗

# State

- [Homepage](#)
- latest [API Docs](#)
- latest [API Docs with ExtJS](#)
- latest Examples (linked on Homepage)
- latest Sencha Package

```
# Once on the commandline
sencha package repo add \
  GeoExt http://geoext.github.io/geoext3/cmd/pkgs
```

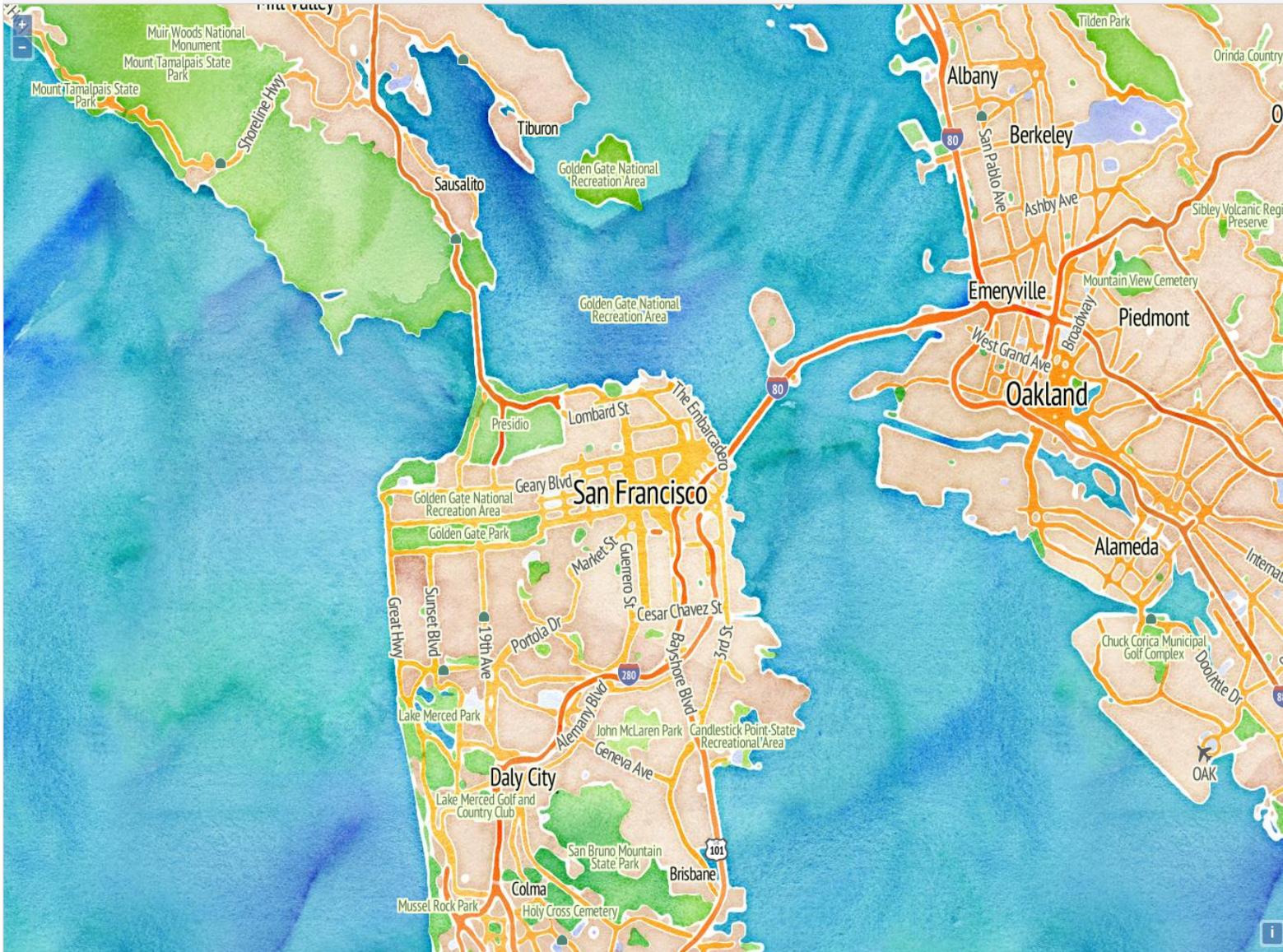
```
// in app.json
"requires": [
  "GeoExt"
],
```

# Examples & features

# Basic MapComponent example

GeoExt.component.Map Example

Description



This example shows how to use the `GeoExt.component.Map` class. Have a look at [map.js](#) to see how this is done.

```
var olMap = new ol.Map({
  layers: [
    // ...
  ],
  view: new ol.View({
    // ...
  })
});

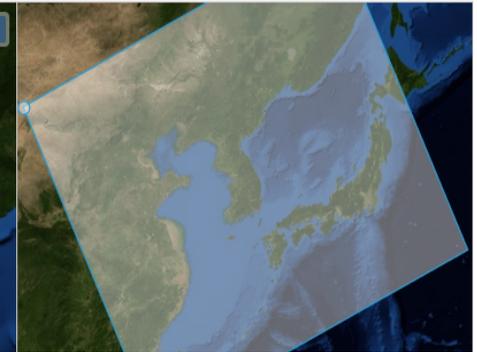
var mapComponent = Ext.create('GeoExt.component.Map', {
  map: olMap
});
```

# OverviewComponent example

GeoExt.component.OverviewMap Example



OverviewMap (default)



## Description

This example shows how to use the `GeoExt.component.OverviewMap` class.

The overviewmap will visualize the extent of the main map with a rectangle. The main map can be rotated (using SHIFT & drag), and the overviewmap will adjust the rotation of the rectangle. The top-left corner is visualized with a circle in the overviewmap.

Have a look at [overviewMap.js](#) to see how this is done.

OverviewMap (configured)



# LayerTree with legends example

**Legends in tree panel**

- Vector
- MapQuest Hybrid
- ol.layer.Group
  - MapQuest OSM
  - MapQuest Satellite

**Description**

This example shows how to use the GeoExt . tree . Panel class and shows two methods how to include legends for every treenode.

Have a look at [tree-legend-simple.js](#) to see how this is done.

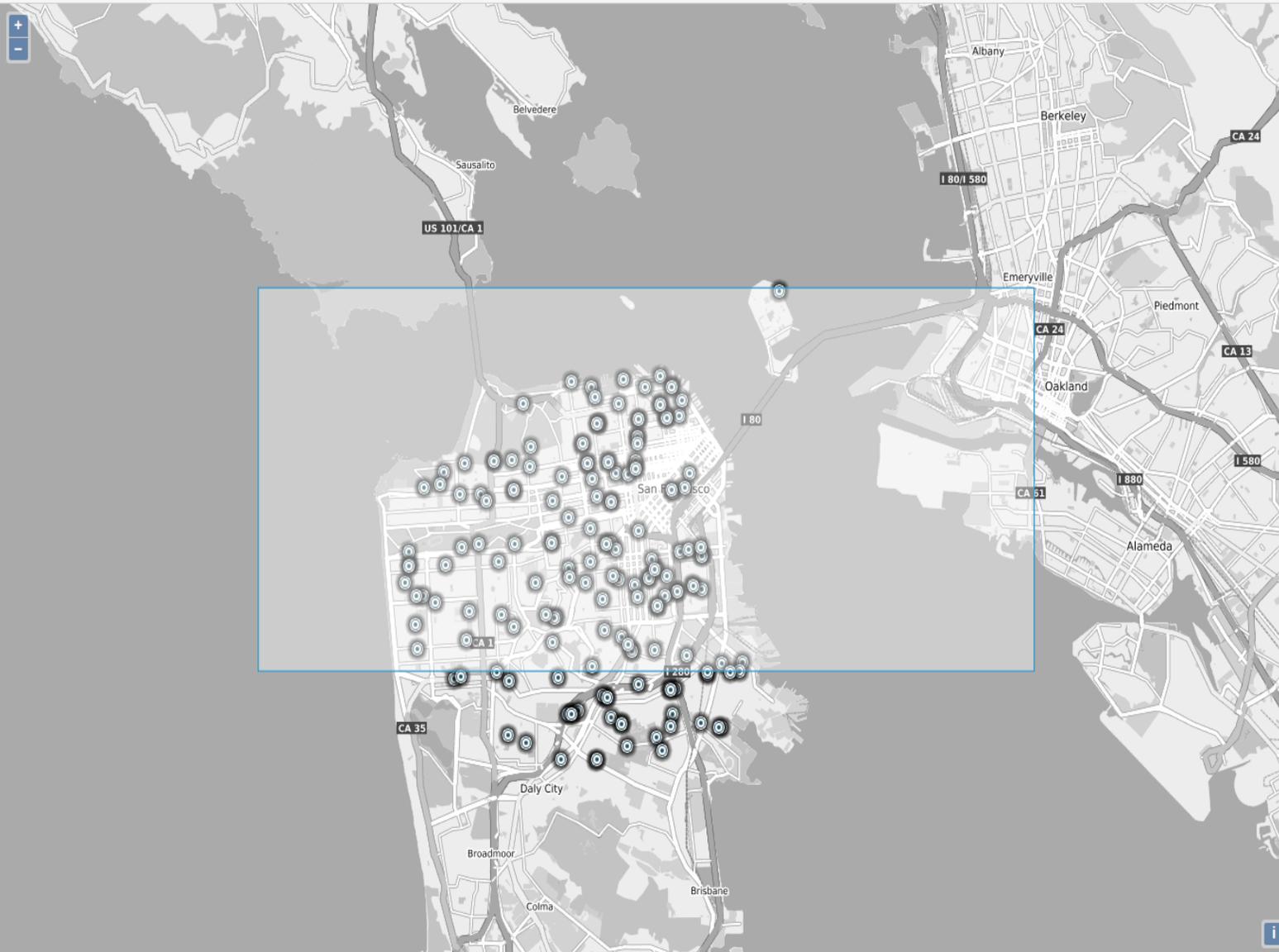
**treePanel**

- Vector
- MapQuest Hybrid
- +  ol.layer.Group

# Printing example (with MapFish v3)

GeoExt.data.model.print.Capability Example

Description



Print

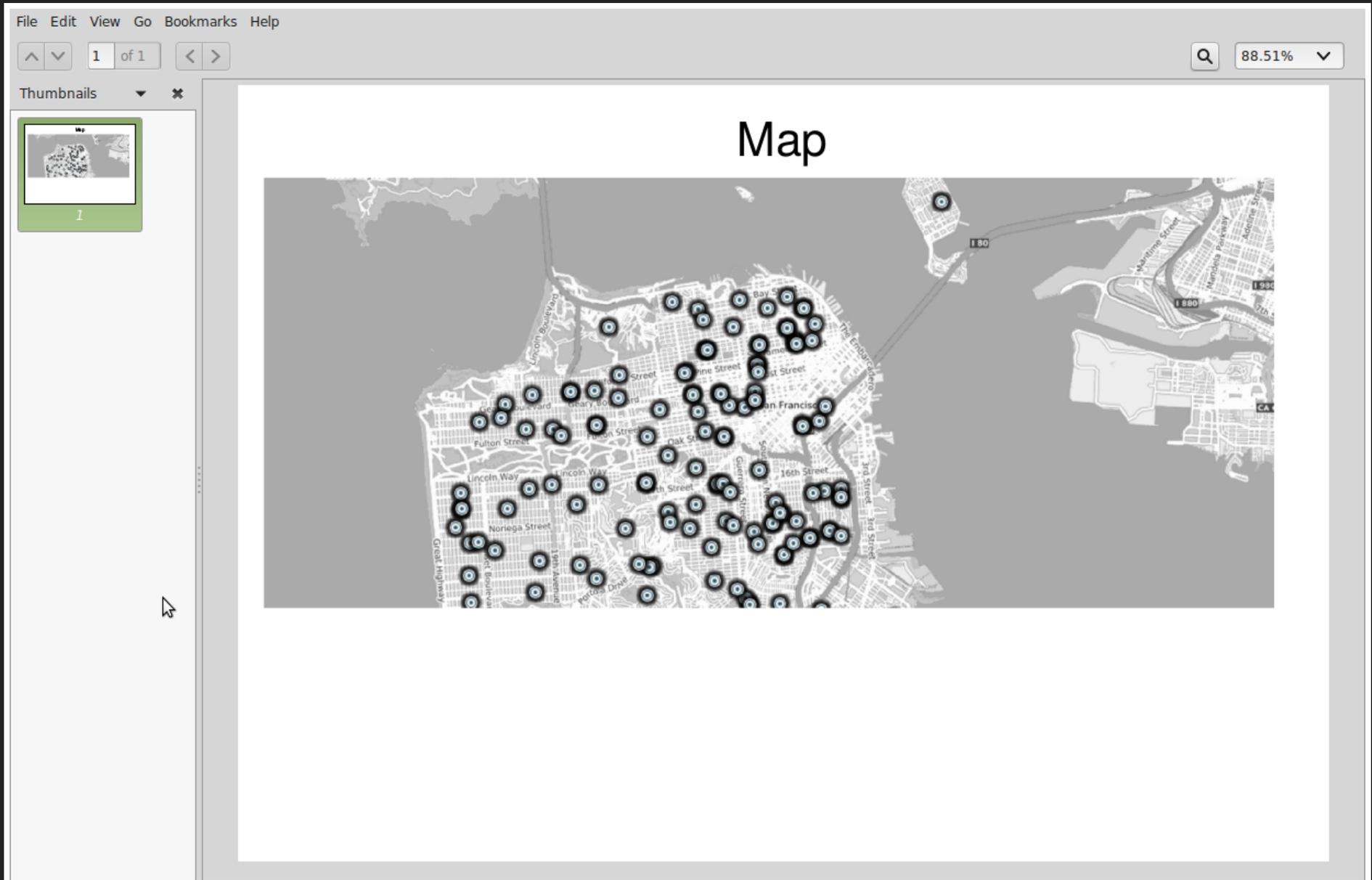
This example shows how to use the `GeoExt.data.MapfishPrintProvider` class to talk to a Mapfish Print Server (v3.x).

Afterwards we have all information to create a valid POST to the servlet. The printed extent is highlighted as vector layer. If you move around or change the zoom, the extent will adjust accordingly.

Click the button labelled 'Print' to actually create a PDF for the displayed extent.

Have a look at [basic-mapfish.js](#) to see how this is done.

# Resulting PDF of printing example

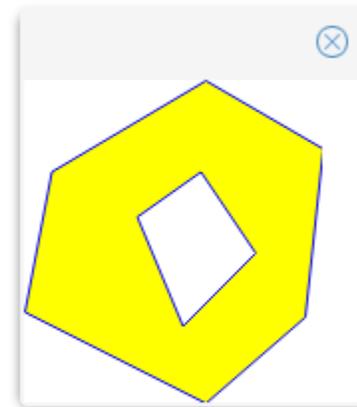


# GeoExt popups and pointerrest



# FeatureRenderer

	point	line	polygon	text
default				n/a
red				n/a
custom				Ab
stacked				Ab



## Text-graphic

Text and graphic combined



Text only

Ab

Graphic only



# FeatureStore in ExtJS Grid

**Feature Grid**

	Name	Population
●	Hamburg	1,700,000
●	Frankfurt / ...	700,000
●	Berlin	3,500,000
●	München	1,400,000

**Feature Grid with selection**

	Name
●	Dortmund
●	Köln
●	Kaiserslautern
●	Bonn
●	Stuttgart

## Description

This example shows how to display features in grids.

The grid on the left side is created by passing an OpenLayers collection (`ol.Collection`) with feature objects (`ol.Feature`)

The grid on the right side is created from an existing vector layer and also highlights the selected feature in the grid on the map.

Have a look at [grid.js](#) to see how this is done.

# MapView-form



**View**

Resolution (m): 38.22

Rotation (rad): 0.9

**Layer**

Opacity: 1

Brightness: -0.125

Contrast: 1.25

Hue: 0.5

Saturation: 2.125

**Description**

This example shows how to wrap OpenLayers classes as GeoExt.data.model.Objects. Changes on the Ext.data.Model are forwarded to the OpenLayers object and vice versa.

More layer settings like saturation and hue are available if WebGL is supported by your browser.

Have a look at [mapviewform.js](#) to see how this is done.

# Outlook

# Outlook / future

- Release betas / previews
- Develop roadmap
- Attract more people
- Universal application
- Possible restructuring: GeoExt-base, -modern, -classic
- Continuous deployment: first steps done
- ... then release it as 3.0.0

Thanks!

---

Questions &  
Remarks?

---

Imprint

# Imprint

## Authors

**Marc Jansen**

terrestris GmbH & Co. KG

Pützchens Chaussee 56

53227 Bonn

Germany

[jansen@terrestris.de](mailto:jansen@terrestris.de)

**Christian Mayer**

meggsum

Hauptstraße 165a

67125 Dannstadt-Schauernheim

Germany

[chris@meggsum.de](mailto:chris@meggsum.de)

## License

The slides of this talk are licensed under [CC BY-SA](#)



[Slides \(HTML\)](#), [Slides \(PDF\)](#), [git repository](#)