GeoServer Orientation

FOSDEM 2020
GeoServer Basics

FOSDEM 2020
Introductions

Jody Garnett
Technical Director
jody.garnett@gmail.com
@jodygarnett

GeoCat
Netherlands open-source company focused on spatial data publication and discovery.

Open Source Projects
- GeoServer, GeoTools, JTS, ImageN, uDig

Open Source Geospatial Foundation
- Incubation Committee Chair
- GeoTools Project Officer
- Marketing Committee Co-chair

Eclipse Foundation
- Technology Project Chair
Popular GeoServer Talks

- **State of GeoServer**
  - 2.16
  - Annual team update and tour of what is new!

- **GeoServer Feature FRENZY**
  - Beloved features, and impressive tricks, showing what is possible with GeoServer

- **GeoServer Ecosystem**
  - GeoServer use in organizations, and technologies, around the worlds.
Popular GeoServer Talks: Updated!

- State of GeoServer
  - Annual team update and tour of what is new!

- GeoServer Feature FRENZY
  - Beloved features, and impressive tricks, showing what is possible with GeoServer

- GeoServer Ecosystem
  - GeoServer use in organizations, and technologies, around the worlds.

- GeoServer Orientation
  - Downloaded GeoServer, now what does it do again?
What is it for?
GeoServer Vision

gesion.org:

“GeoServer is an open source server for sharing geospatial data.”

GeoServer is passionate about sharing data. GeoServer was founded out of a community project to map trees in New York city.

- If you are here to share a map.. GeoServer can do that.
- But don’t stop at a map ... GeoServer will share the data
- And don’t stop at the data … GeoServer encourages data editing
GeoServer approach

Approach:

*Designed for interoperability, it publishes data from any major spatial data source using open standards.*

GeoServer really does not want to “lock-in” your data, from the ground up:

- Share your data with a wide audience
- Access your data where it is
- Start with “industry standards”
- Add protocols over time

Opposite of a “not-invented-here” attitude.

If GeoServer has to invent something in isolation chances are we are doing it wrong.
Maps Publishing

GeoServer can publish maps:

- Each dataset is a “layer” of content
- Gather into a “layer group” to arrange layers in order
- And there is a small layer preview for local testing

GeoServer is the rendering engine producing maps, for access from web clients and desktop clients.
Map Publishing Continued

GeoServer map publishing is about publishing a visualization.

More than one approach:

- Publish “Map” supporting, ad-hoc requests to draw imagery
- Publish entire “Tilesets” of content

The result does not always look like an image:

- SVG and KML output combine vector output with styling
- Extensions provide PDF output for printing
- Vector-tile output pre-processed for client side drawing
In the spirit of open, GeoServer is setup to share the data being used to draw the map.

To reach a wide audience:

- Supports queries
- Range of output formats available (and more available to install)
Sharing does not stop at providing access, the ability to *edit information together* is key to collaboration.

GeoServer is not used to interactively edit data, instead it **provides a protocol** for QGIS Desktop and OpenLayers library to edit information. These clients are responsible for providing an interactive editing experience.

The same protocols can be used in your own scripts for batch editing and processing.

Hint: We made sure editing is off by default!
Raster data access

Direct access to raster data is available.

- Imagery
- Digital Elevation Model
- NetCDF for scientific modeling

Hint: The language “grid coverage” is used to emphasis that measurements are being provided (rather than just a visualization)
GeoServer at a Glance

Java Web Application to share and edit geospatial data.

Publish data from any major spatial data source using open standards.

Core Protocols

- **WMS** – maps
- **WFS** – vector
- **WFS-T** – editing
- **WCS** – coverage
- **WMTS** – tiles
- **TMS** – tiles
- **WMS-C** – tiles

Extension/community protocols

- **WPS** – process
- **CSW** – search

**OpenSearch for Earth Observation** – search

**OGC API** - prototype json + rest standards
How do I use it?
GeoServer Configuration

GeoServer provides a web application for configuration and setup.

A REST API is also available for automation.

Notes:

- “Web admin” is not really geoserver
  - Only used for setup geoserver
  - In production hide this!
- The links to “service capabilities” are the actual web services
Natural Earth

A great public domain dataset used for web illustrations.

Our example uses the “Quickstart” download containing a range of vector and raster data

See: [www.naturalearthdata.com](http://www.naturalearthdata.com)
Workspace
Workspace

Workspace is used to:

- Folder to organize content
- “Namespace” for XML content publication

Notes:

- Workspace name used as a prefix for each layer
- Namespace should be a unique URI
- One workspace can be the “default”
- Advanced: Each workspace can be used as its own “virtual web service”
New Workspace

1. Data → Workspaces
2. Create new workspace
3. Fill in:
   a. Name: ne
   b. Namespace: http://ne
4. Mark as default workspace
Data Source
Vector Data Source

Data store:

- Used to connect to your data
  - Shapefile needs a file location
  - Database needs connection parameters
- Managed in a workspace

Notes:

- DataStore name is “internal” to GeoServer
  - So you can switch from Oracle to PostGIS
- Called a “DataStore” as we were focused on editing (and thus storing information)
New Vector Data Source

1. Data → Stores
2. Add new store
3. Select Directory of spatial files (shapefiles)
4. Name: Culture
5. Connection Parameters
   a. Use browse button to locate Natural Earth 4.1.1/10m_cultural
6. Save
Add Raster Data Source

1. Data → Stores
2. Add new store
3. Select Directory of spatial files (shapefiles)
4. Name: NE
5. Connection Parameters
   a. Browser to locate
      NE1_50M_SR_W/NE1_50M_SR_W.tif
6. Save
Layer
Vector Layer

Layer:

- GeoServer publishes information as distinct layers
- To publish you need to check
  - Name and title for the layer
  - Spatial Reference System
  - Bounds

Notes:

- Each protocol uses a different word for its data product (feature type, coverage, tileset)
New Vector Layer

1. Data → Layers
2. Use add new layer, and select ne:Culture
3. Ne_10m_admin_1_states_provinces_shp
New Vector Layer: Description

1. Data → Layers
2. Use add new layer, and select *ne:*Culture
3. **Ne_10m_admin_1_states_provinces_shp**
4. Data
   a. Name: *states_provinces_shp*
   b. Title: *States and Provinces*
New Vector Layer: Bounds

1. Data → Layers
2. Use **add new layer**, and select ne:Culture
3. **Ne_10m_admin_1_states_provinces_shp**
4. Data
   a. Name: **states_provinces_shp**
   b. Title: **States and Provinces**
5. Spatial Reference System: EPSG:4326
6. Native bounding box:
   Click compute from data
7. Lat/Lon Bounds:
8. Click compute from native bounds
9. Save
1. Publish “NE1_50M_SR_W”
2. Name: “ne”
3. Title: Natural Earth I
4. Double check:
   a. Spatial Reference System
   b. Bounds
5. Save
Layer Group
Layer Group

Layer groups are used:

- Can be used as a basemap
- Order of layers is in **draw order**

Notes:

- Can also use layer groups to define a table-of-contents structure for published content
Create Layer Group

1. Data → Layer Group
2. Click Add new layer group
3. Details:
   a. Name: basemap
   b. Title: Basemap
Create Layer Group

1. Data → Layer Group
2. Click Add new layer group
3. Details:
   a. Name: basemap
   b. Title: Basemap
4. Scroll down to Layers heading
5. Use add layer to add ne:ne1
6. Use add layer to add ne:state_province_shp
7. Scroll up to bounds
8. Click Generate Bounds
9. Save
Style

Used to configure the rendering process:

- We use an XML format called SLD
  - Intended for machine-to-machine use!
- We recommend people use CSS or YSLD

Notes:

- Built-in styles provided
- Styles folder used for icons and fonts
- Each workspace has a styles folder also
New Style

1. Data → Styles
2. Click **Add new style**
   a. Workspace: ne
   b. Name: mapcolor9
   c. Format: YSLD
3. Select **Polygon**, and **Generate**
New Style

1. Data → Styles
2. Click **Add new style**
   a. Workspace: ne
   b. Name: mapcolor9
   c. Format: YSLD
3. Select **Polygon**, and **Generate**
4. Apply, and then Layer Preview tab
New Style

1. Data → Styles
2. Click Add new style
   a. Workspace: ne
   b. Name: mapcolor9
   c. Format: YSLD
3. Select Polygon, and Generate
4. Apply, and then Layer Preview tab
5. Fill in style shown, and Apply
   (Theme on mapcolor9 attribute)

symbolizers:
- polygon:
  stroke-color: 'gray'
  stroke-width: 0.5
  fill-color: ${Recode(mapcolor9, '1', '#8dd3c7', '2', '#ffffb3', '3', '#bebada', '4', '#fb8072', '5', '#80b1d3', '6', '#fdb462', '7', '#b3de69', '8', '#fccde5', '9', '#d9d9d9')}

Theme from user guide style workshop
New Style

1. Data → Styles
2. Click **Add new style**
   a. Workspace: ne
   b. Name: mapcolor9
   c. Format: YSLD
3. Select **Polygon**, and **Generate**
4. Apply, and then Layer Preview tab
5. Fill in style shown, and Apply
   (Theme on mapcolor9 attribute)
New Style

1. Data → Styles
2. Click Add new style
   a. Workspace: ne
   b. Name: mapcolor9
   c. Format: YSLD
3. Select Polygon, and Generate
4. Apply, and then Layer Preview tab
5. Fill in style shown, and Apply
   (Theme on mapcolor9 attribute)
6. Use Publishing tab, set default for ne:state_provinces_shp
Layer Preview

1. Data → Layer Preview
2. Locate `ne:basemap`
How does it work?
GeoServer Internals

- **Java Web Application**
  - Spring Framework - powerful framework for wiring together application out of components
  - Apache Wicket User Interface - framework only a java developer could love
- **Modular Architecture**
  - Enables the community to build extensions
    (a key success factor for open source projects)
- **GeoTools Library**
  - Handles the data access, reprojection, and rendering
  - Try and keep GeoServer “light” and focused on sharing by moving heavy lifting here
- **Java Advanced Imaging (and JAI-ExT)**
  - Image processing library offering on-demand processing model
  - JAI-EXT teaching the image processing engine new tricks (“no-data”, “foot-prints”, …)
Doing more
Extensions

We saw a lot of effort devoted to keeping GeoServer modular.

- Official GeoServer plugins are called extensions
- Extensions are formally “part of GeoServer”
  - Must meet the same quality assurance and documentation requirements as core
  - Included in the release process, they version numbers match!
Check out some extensions

- **Oracle DataStore**
  - Is Oracle required for your data management policies? You can still use open standards.

- **WPS Extension**
  - Providing an **entirely new services**, in this case one used for processing and analysis.

- **GeoMesa**
  - Cloud database are supported also!
  - GeoMesa offers both a DataStore for data access, and also WPS Processes to better summarize and work with massive cloud data volumes.

- **GeoFence**
  - Integrates “GeoFence” security model, for “edit permission” with a specific operating area.

- **Vector Tiles**
  - Generate pre-processed tiles of vector content ready for client-side rendering.
Community Modules (for RnD)

Community modules started as a way to help developers share RnD ideas.

- Very helpful when working with grad students
- Also used by consulting teams take on specific challenges for their customers
- Ideally we want these to attract funding and become extensions

These are very much “use at your own risk”:

- These are not released as part of GeoServer
- You are expected to compile these yourself
- Not subject to any quality assurance or IP check
Check out some community modules

- **JDBC Config / JDBC Store**
  - Replace the entire catalog (config) and data directory (store)
  - Use with AWS to “carefully” manage a cloud cluster

- **Backup and Restore**
  - Transport configuration settings from test to production

- **SOLR datastore**

- **Dynamic colormap generation**

- **MapML**

- **SAP Hana**

- **OGC API Features**
Brought to you by
GeoServer is setup as an “Open Source Geospatial Foundation” project. With a core “project steering committee” and notes on how to keep everything running in a fair and consistent manner.

Steering committee members vote on change proposals, that come from a wide range of organizations participating in RND.

We try and have a mix of skills, with developers, users, managers, and designer joining the team as required to help out.

PSC Members:

- Alessio Fabiani
- Andrea Aime
- Ian Turton
- Jody Garnett
- Jukka Rahkonen
- Kevin Smith
- Nuno Oliveira
- Simone Giannecchini
- Torben Barsballe
Strong history of collaboration

Collaboration happens at the personal level, and also across organizations.

The OpenPlannng Project started GeoServer to support community mapping activities. Early collaborations with GeoSolutions, Refractions Research helped establish procedures to make this an open project.

GeoServer supports an active RnD scene with research labs, consulting companies and others joining to add new features.

Being “vendor-neutral” has allowed GeoServer to weather the success and failures of participating organizations.

In addition to an open source license we make use of the OSGeo Contributor License agreement to help in this respect.

This also has allowed the PSC to contribute code “up-stream” to more permissive projects.
Try it out!
Running GeoServer

Options:

- “binary” download used to try things out on the command line, often for testing or evaluation
- “Web Archive” for use with your application server (we recommend Tomcat)
- Sorry no Windows or macOS installers at present, due to vandalism

Alternatives:

- Docker? Nothing official, everyone is making their own …
- Hosted? A few companies offer hosting
Running GeoServer

```
02 Feb 06:53:42 INFO [gwc.config] - Initializing GeoServer specific GWC configuration from gwc-gs.xml.
02 Feb 06:53:43 INFO [diskquota.DiskQuotaMonitor] - 0 layers configured with their own quotas.
02 Feb 06:53:43 INFO [diskquota.DiskQuotaMonitor] - 25 layers attached to global quota 500 MB.
02 Feb 06:53:43 INFO [diskquota.DiskQuotaMonitor] - Disk quota periodic enforcement task set up every 10 SECONDS.
02 Feb 06:53:43 INFO [georss.GeoRSSPoller] - Initializing GeoRSS poller in a background job...
02 Feb 06:53:43 INFO [georss.GeoRSSPoller] - No enabled GeoRSS feeds found, poller will not run.
02 Feb 06:53:43 INFO [wms.WMSService] - Will proxy requests to backend that are not getmap or getcapabilities.
02 Feb 06:53:45 WARN [gce.imag mosaic] - Unable to set ordering between tiff readers, SPI.
02 Feb 06:53:49 INFO [geoserver.security] - Start reloading user/groups for service named default.
02 Feb 06:53:49 INFO [geoserver.security] - AuthenticationCache Initialized with 1000 Max Entries, 300 seconds idle time, 600 seconds time to live and 3 concurrency level.
02 Feb 06:53:49 INFO [geoserver.security] - AuthenticationCache Eviction Task created to run every 600 seconds.
2020-02-02 06:53:49.771 INFO:oejs.ContextHandler:main: Started o.e.j.w.WebAppContext@52a26356(GeoServer,geoserver,fire://Users/jgarnett/Desktop/geoserver-2.16.2/webapps/geoserver/,AVAILABLE)[/geoserver]
2020-02-02 06:53:49.793 INFO:oejs.AbstractConnector:main: Started ServerConnector@df2c697(HTTP/1.1,[http/1.1])@0.0.0.0:8080
2020-02-02 06:53:49.794 INFO:oejs.Server:main: Started @13514ms
```
Thanks

Questions welcome!
Mini production check-list

1. Please change the default **user name** and **password**
2. Also change the **master password** (used to encrypt on disk)
3. Update contact information (See “Claudius Ptolomaeus”)

GeoServer 1.0

![GeoServer 1.0](image1.png)

GeoServer 2.16.2

![GeoServer 2.16.2](image2.png)