Table of Contents

Introduction to GeoServer

Welcome to the **Introduction to GeoServer - Workshop**.

This workshop is developed to be used within the OSGeo-Live 16.0 environment and is intended to give you a comprehensive overview of GeoServer as a web mapping solution.

> **INFO** The workshop can be downloaded as PDF here.

Please make sure that you have completed the steps of the preparatory work and general information-chapter to ensure a smooth and frictionless process.

The workshop is based on a number of modules. In each module you will solve a series of tasks to achieve a specific goal. Each module will expand and strengthen your knowledge about GeoServer.

The following modules will be part of the workshop:

- Preparatory work and general information: Basic knowledge about the workshop environment (OSGeoLive, paths, URLs, Credentials)
- Basic knowledge about GeoServer: General information about GeoServer
- Administration interface: Configuration of GeoServer via the web interface
- Data publishing: Publish geodata from different data sources
- Layer groups: Publish a layer group
- Styling: Style a layer with SLD (Styled Layer Descriptor)

Authors

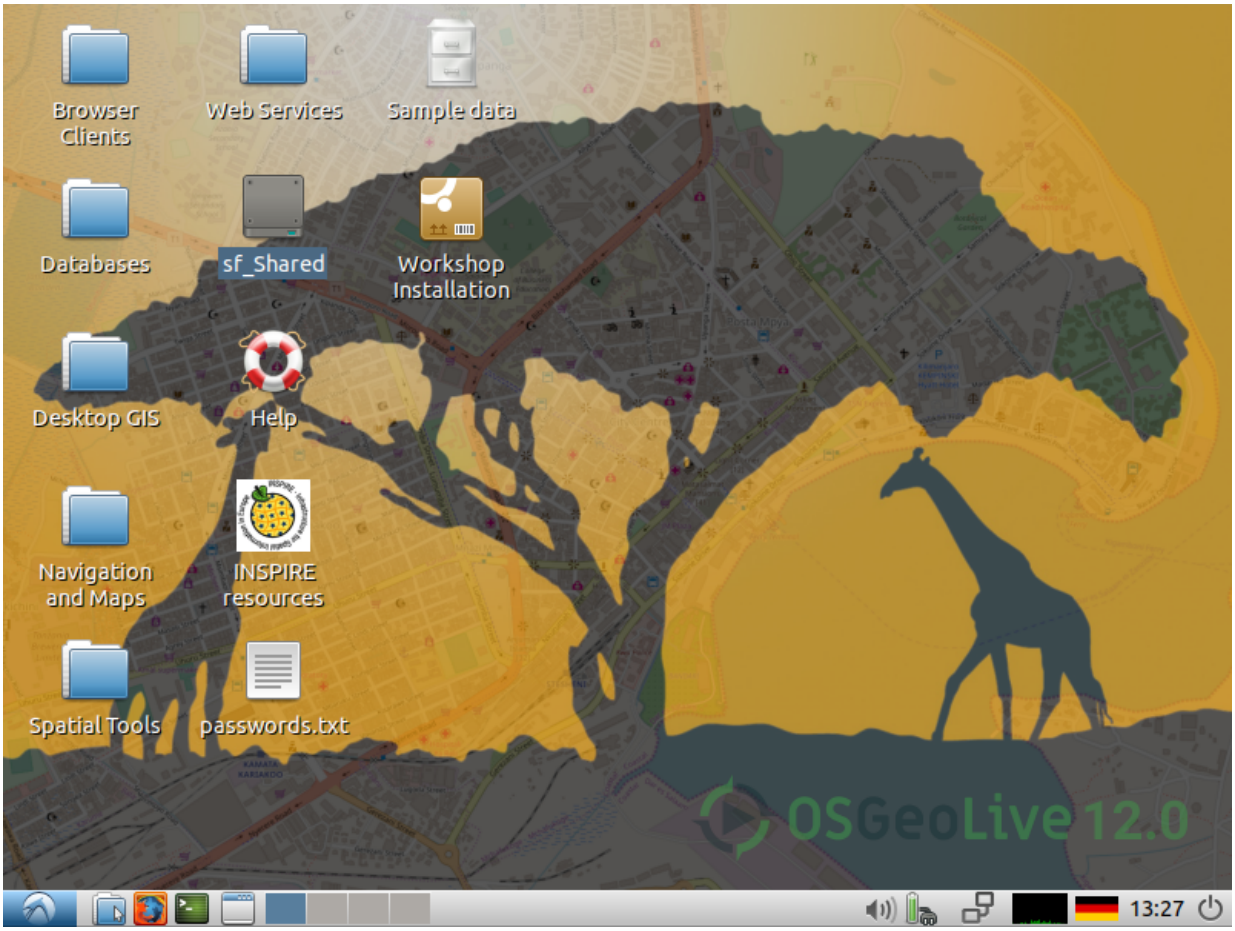
- Nils Bühner (buehner@terrestris.de)
- André Henn (henn@terrestris.de)
- Daniel Koch (koch@terrestris.de)
- Fabian Schmidt (fschmidt@terrestris.de)
- Fritz Höing (hoeing@terrestris.de)

(Authors are sorted alphabetically by their last name.)

Preparatory work and general information

Before you can start with the workshop, please complete the following steps:

- Boot your computer including OSGeoLive-Medium
- Choose your language
- Choose *try ubuntu without installation*
- User: user; Password: user (probably not necessary)



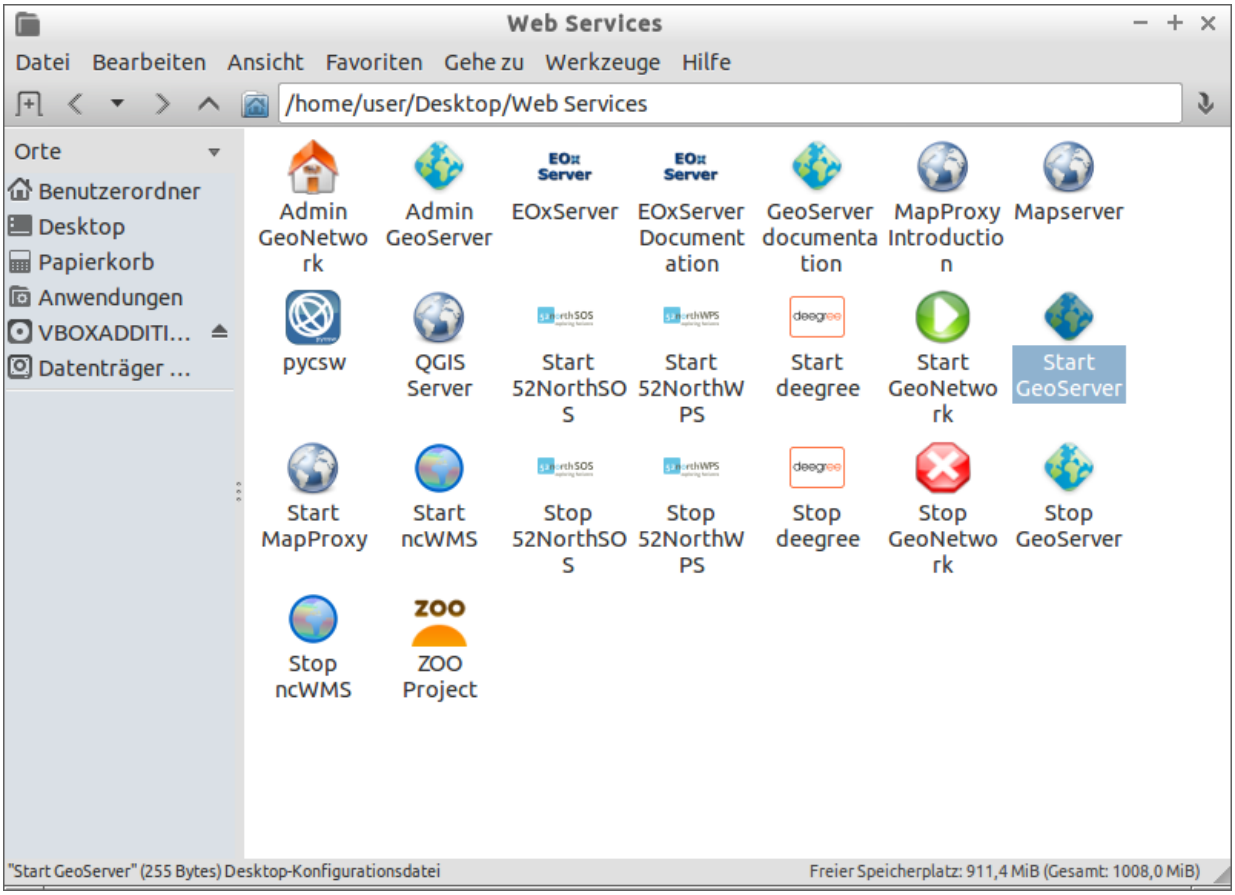
Paths, URLs, Credentials

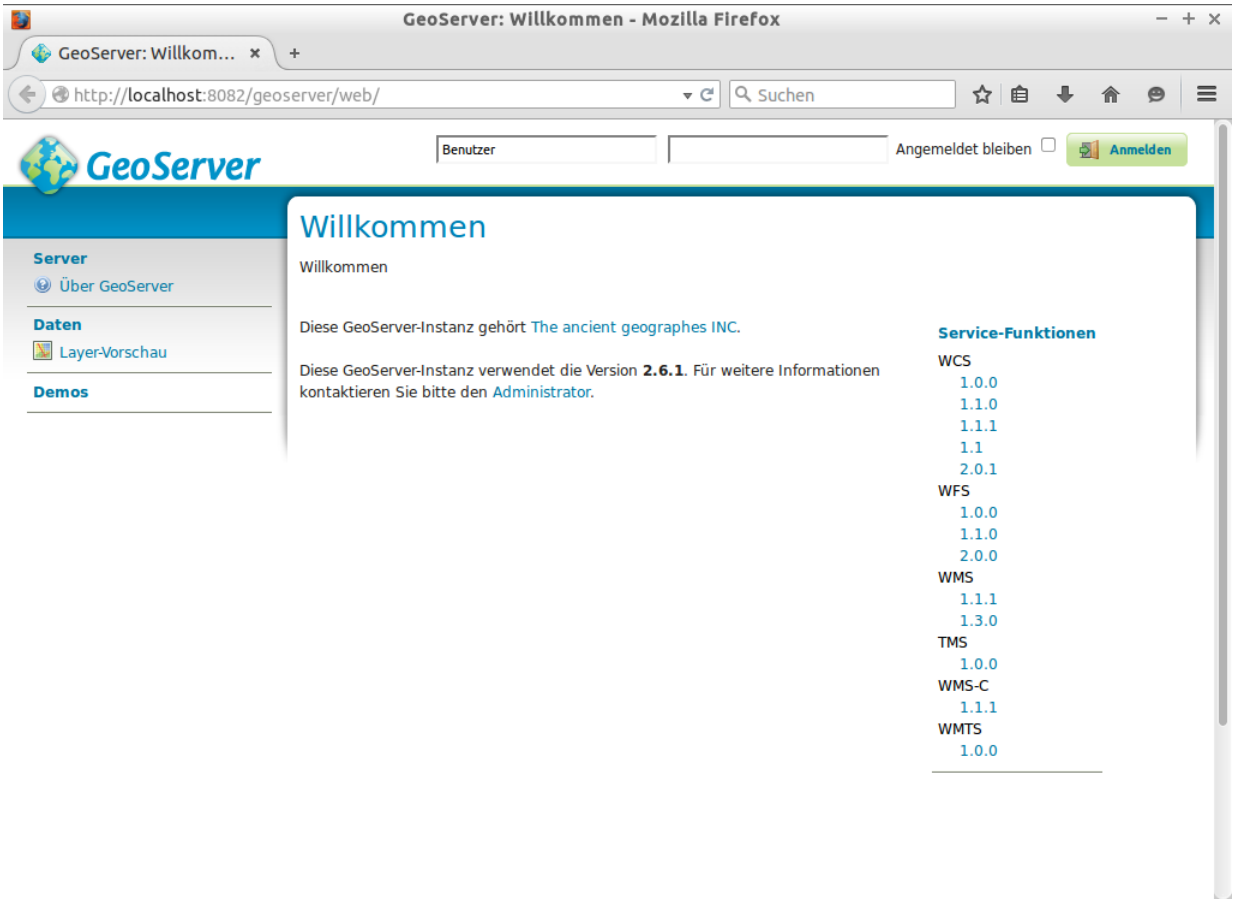
- GeoServer: http://localhost:8082/geoserver (first has to be started, see below)
- Credentials GeoServer:
  - User: `admin`
  - Password: `geoserver`
- GeoServer (file system): `/usr/local/lib/geoserver-2.22.2/`

Start GeoServer

GeoServer is launched by double-clicking **Start GeoServer** in the folder **Web Services** on OSGeoLive desktop:

> **INFO**
>
> If GeoServer can **not** be launched this way, you can try following command in the terminal:
>
> ```
> sudo /usr/local/lib/geoserver/bin/startup.sh
> ```
>
> The terminal/process has to remain open during the workshop!





In the following chapter we will continue with basic knowledge about GeoServer.

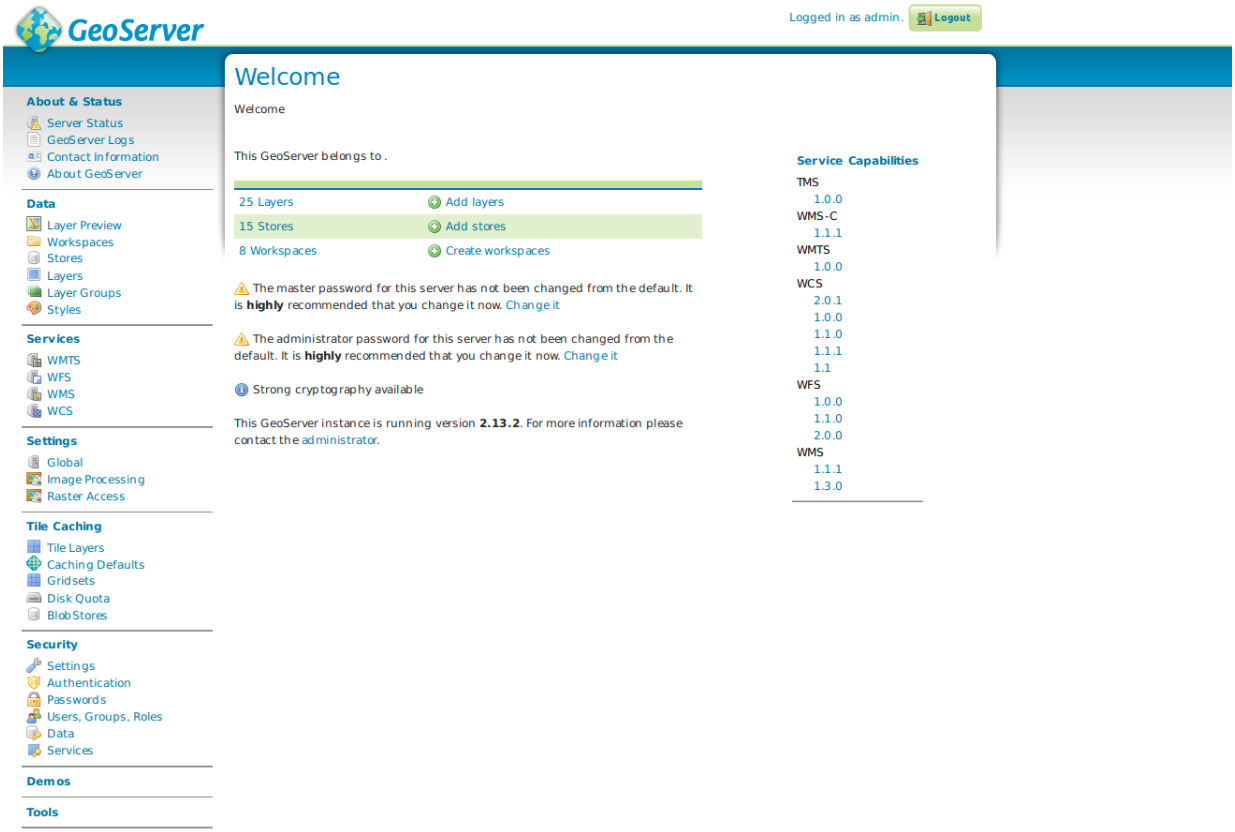
Basic knowledge about GeoServer

GeoServer is an open, Java-based server, which allows to display and edit geodata based on the standards of the Open Geospatial Consortium (OGC) (in particular WMS and WFS). A strength of GeoServer is its flexibility with which it can be extended by additional functionality.

GeoServer is well documented. The documentation is divided into a user-documentation and a developer-documentation:

- User-documentation: https://docs.geoserver.org/stable/en/user/
- Developer-documentation: https://docs.geoserver.org/stable/en/developer/

The two links refer to the documentation of the last stable version. The *stable* in the respective URL can also be replaced by a version number, if you want to access the documentation of a specific GeoServer version. Within this workshop **Version 2.22.2** is used, so the resulting URL for the user-documentation is https://docs.geoserver.org/stable/en/user/.



Usually, GeoServer is provided for productive operation as a (Java-)standalone servlet in the form of a `.war` - file, which can be downloaded from http://geoserver.org/download/. The `.war` - file must then be published (often called *deploy*) to a servlet container (for example Tomcat or Jetty) afterwards. The web interface of GeoServer can then be called from the browser.

Further information and details about a classic WAR-installation you can find here.

> **INFO**
>
> GeoServer is already pre-installed on the OSGeoLive system and can be accessed during the workshop at http://localhost:8082/geoserver (see here). This variant differs from the classical *deployment* as a .war file, since a Java program (start.jar) is executed, which programmatically starts a Jetty-server with Geoserver. This is not important for the contents of the workshop.

In the following section we will first get an overview about GeoServer's administration interface. This includes general information about the server settings, logging files as well as security issues. We will also take a closer look at the menu item *data*.

Administration interface

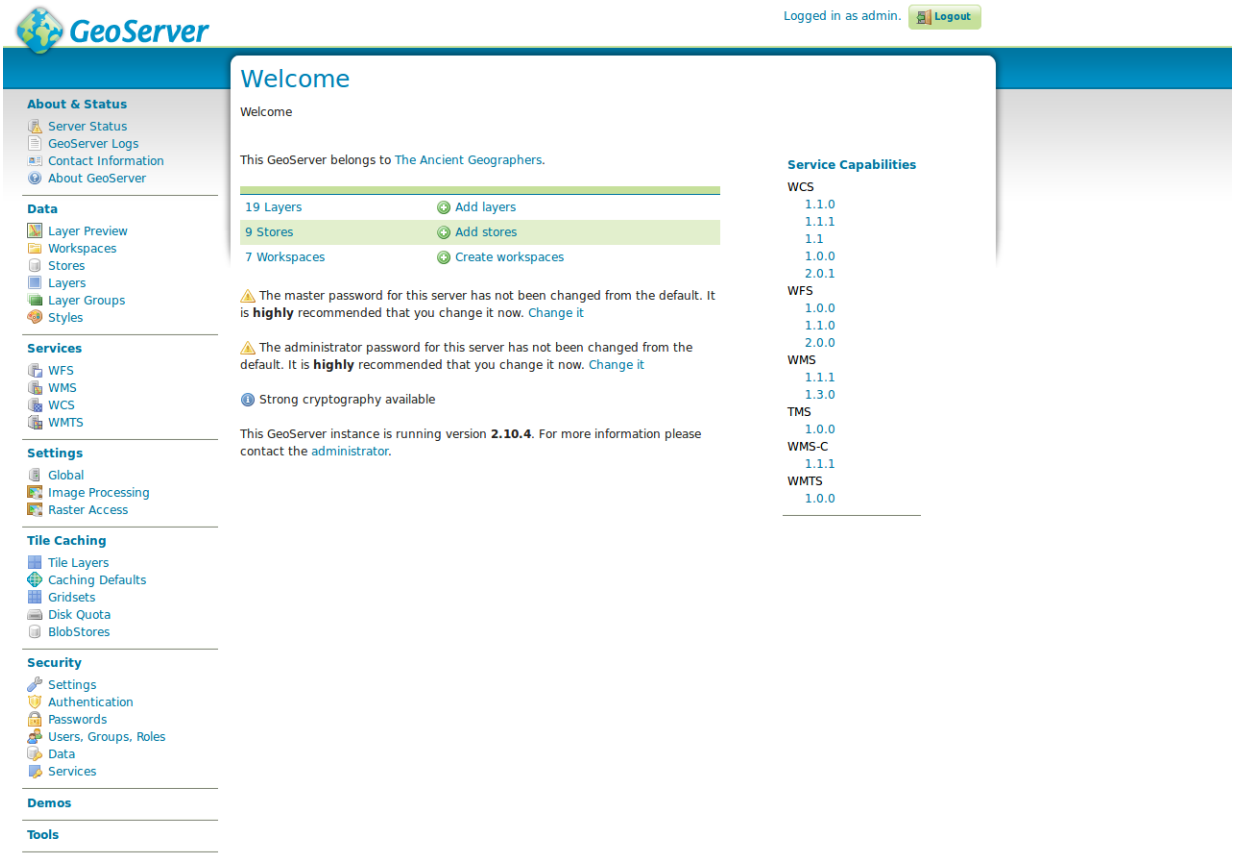
If not already done, please open the GeoServer web interface with a browser of your choice via the following URL:

http://localhost:8082/geoserver

Log in with the following credentials:

- User: `admin`
- Password: `geoserver`

After successful login (as administrator) the access to all the functions of the map server are activated and the start screen changes. The view should be similar to the following figure.

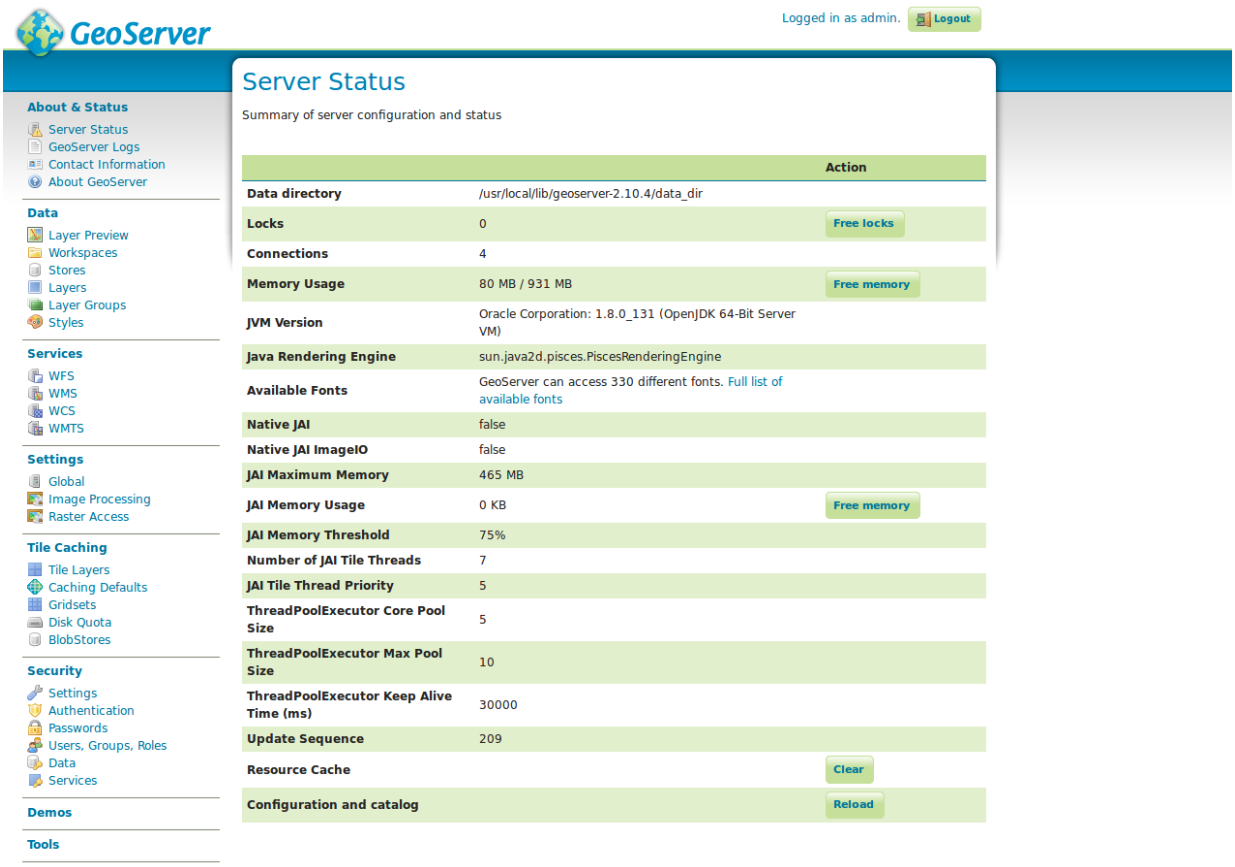


Overview of function elements

The interface of the GeoServer is divided into two sections: On the left, the navigation menu, in which all setting options and lists appear, sorted by subcategory. In the central section the corresponding form fields are displayed, depending on the category selected. Depending on the selection, these forms can be divided into several tabs. In the following, the most important subcategories and use cases are listed and explained.

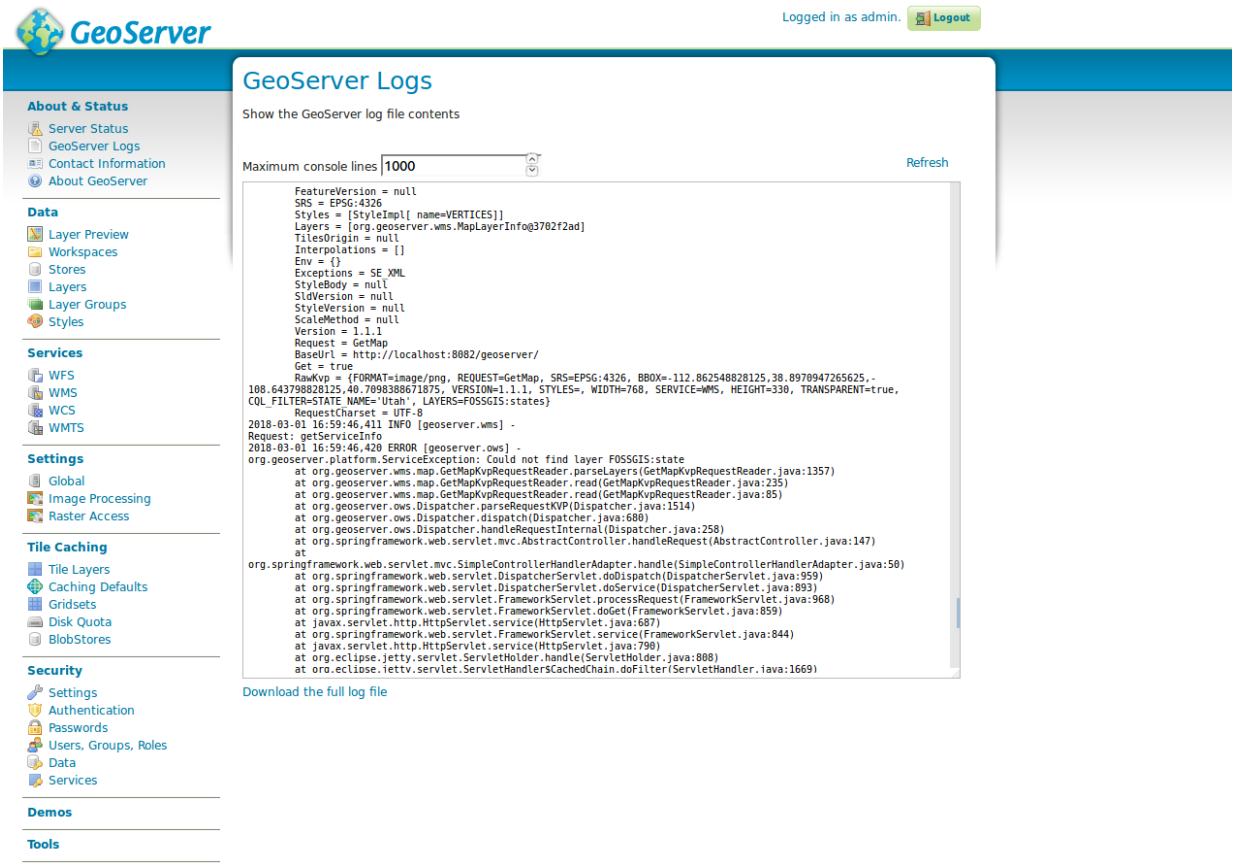
Server

Server Status



Under **Server Status** you can find information about the current state of the map server. In addition to various system specifications (e.g. available fonts, Java version used or memory currently in use), this includes installation specifics such as the currently used data directory. The status page is the first port of call if the performance of a layer does not match the usual speed or changes were made at the file level of GeoServer (e.g. installation of new fonts, changes in configuration files). Changes at file level are not recommended, but they cannot be prevented if, for example, the configuration of another GeoServer is adopted. In this case, the configuration must be set with the button **Load new** under **Configuration and Catalog**. If the user interface responds very sluggish or layers appear only very slowly in the application it can often be helpful to clear the resource cache with the button **Clear** and/or release the working memory via **Free memory**.

Logging



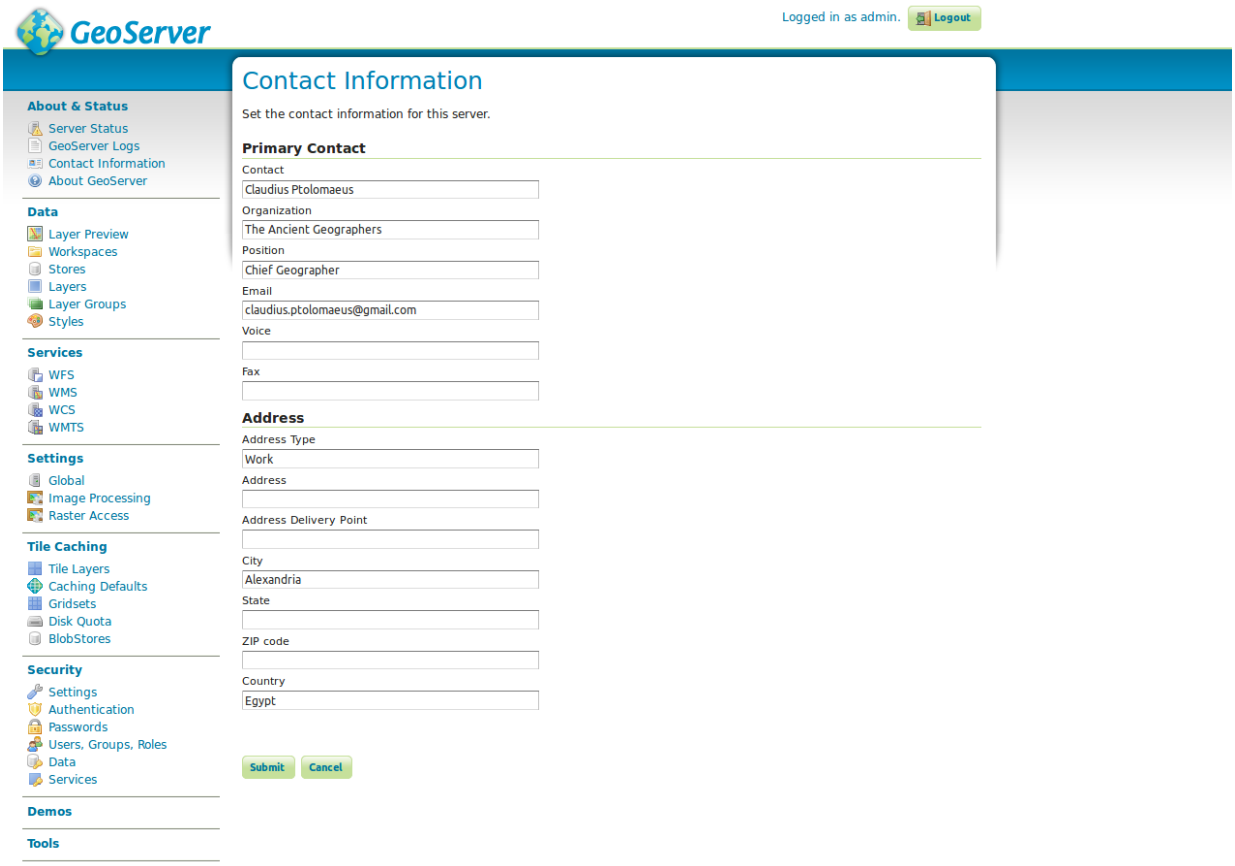
For any errors that can be clearly assigned to GeoServer (e.g. no or false response of a map server) the protocol is the first point of contact. The GeoServer protocol can either be directly accessed via the GUI (see illustration above) or directly from the file system (/usr/local/lib/geoserver-2.22.2//logs/geoserver.log). If the error output is too short, you can change the length of the output using the field *Maximum number of rows* via the console. The error output of Java/GeoServer is usually very detailed, so that the error-causing component can be quickly located. Serious errors are usually designated with the abbreviation `ERROR`. For example, the error output

```
2014-10-06 09:16:33,492 ERROR [geoserver.wms] - Getting feature source: featureType: GEOSERVER:EXAMPLE_LAYER doe
```

when retrieving the data source for a layer is caused by an uncleanly configured data store. This can either be caused by a false configuration or a failure of the source database, for example.

Hint: There are several protocol modes, each with a different sensitivity generating output. Further information in the section Settings.

Contact



The figure above shows the options for entering contact data in GeoServer, which are primarily relevant for the map services of the server, because they appear in the GetCapabilities document.

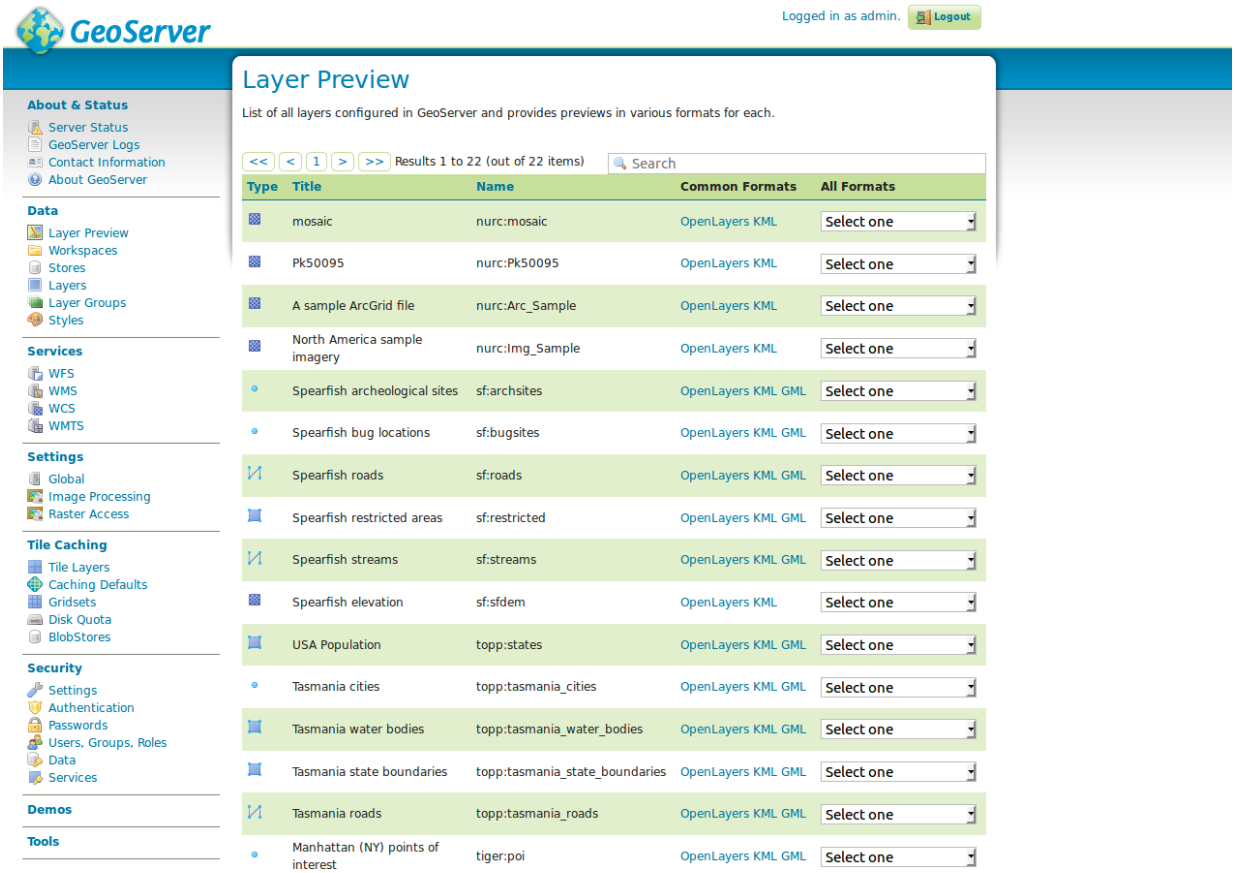
Task:

1. Call the `GetCapabilities` document of GeoServer. To do this, click WMS -> 1.3.0 on the start page on the right under `Service Capabilities`. Please enter your `Contact information` under contact information. Call again the `GetCapabilities` document. What do you notice?

Data

The following sections describe the most comprehensive configuration area of GeoServer, the steps for publishing a service.

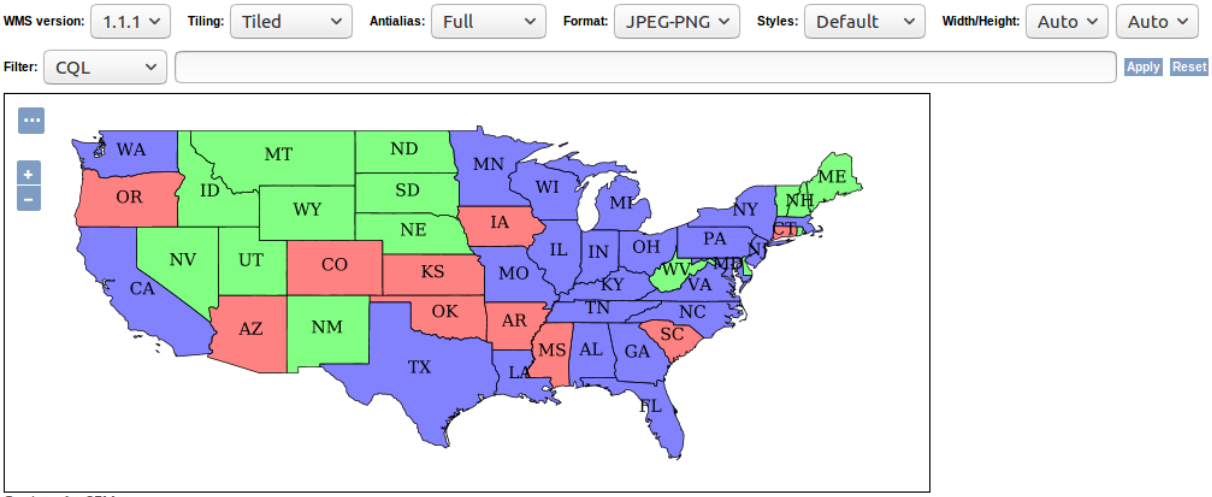
Layer preview



The **layer preview** provides an overview of all layers published on this GeoServer. For a layer to appear in this overview (and also in the GetCapabilities document of the instance), it must be marked as `prefigured`. Each layer has a specification of the type (see table below), a internal layer name (including name of the work space), a layer title (short description) and a selection box of possible preview formats.

type	description
	vector layer (type unknown)
	vector layer ((Multi-)Point)
	vector layer ((Multi-)Line)
	vector layer ((Multi-)Polygon)
	raster layer
	layer group
	cascaded WMS

The layer list can be sorted ascending and descending by left-clicking on the column names `type`, `name` or `title`. In addition to an overview of all available layers, a preview of a layer can be displayed in various formats. This is recommended in particular for a quick and easy comfortable checking of a new created layer. The fastest way to get a preview is to use the "Format" OpenLayers, which creates a new window with a preview map of the selected layer.



Scale = 1 : 35M

states

fid	STATE_NAME	STATE_FIPS	SUB_REGION	STATE_ABBR	LAND_KM	WATER_KM	PERSONS	FAM
states.15	Texas	48	W S Cen	TX	688219.07	17337.549	1.712202E7	4377

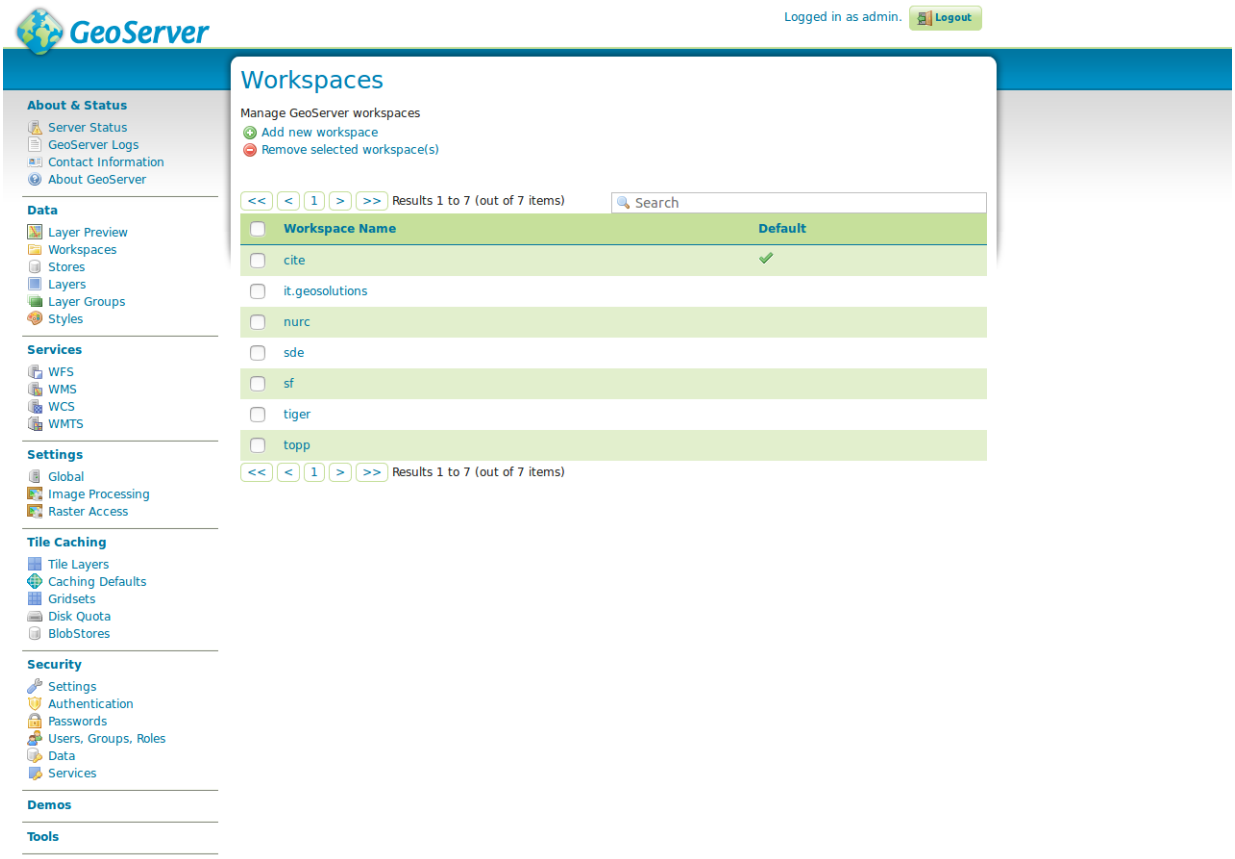
The OpenLayers map allows a free navigation within the layer and a GetFeatureInfo query by a left click inside the layer. The button **toggle options toolbar** above the navigation allows you to display a toolbar with the options to manipulate the GetMap call (format and size, Antialiasing etc.). Further formats are listed below the selection box **All formats**. They are structured according to WMS and WFS. For the preview of a WMS „OpenLayers“ is recommended, while for a preview of a WFS you should choose the "GML2" format. **Hint:** Via the selection box the export as Shapefile is also possible. Shapefiles can be integrated into any standard desktop GIS for further processing.

Task:

1. Call up the OpenLayers layer preview of any layer and change the parameters `Tiling` und `Format` to a parameter of your choice. Afterwards call GetFeatureInfo.

Workspaces

The menu item **Workspaces** provides an overview of all available workspaces of GeoServer. The GUI can be used to create new workspaces or to edit existing ones.



The GeoServer stores layers using the following hierarchy:

```
Workspace
 └ store
  └ Layer
 └ Layer(groups)
 └ Style
```

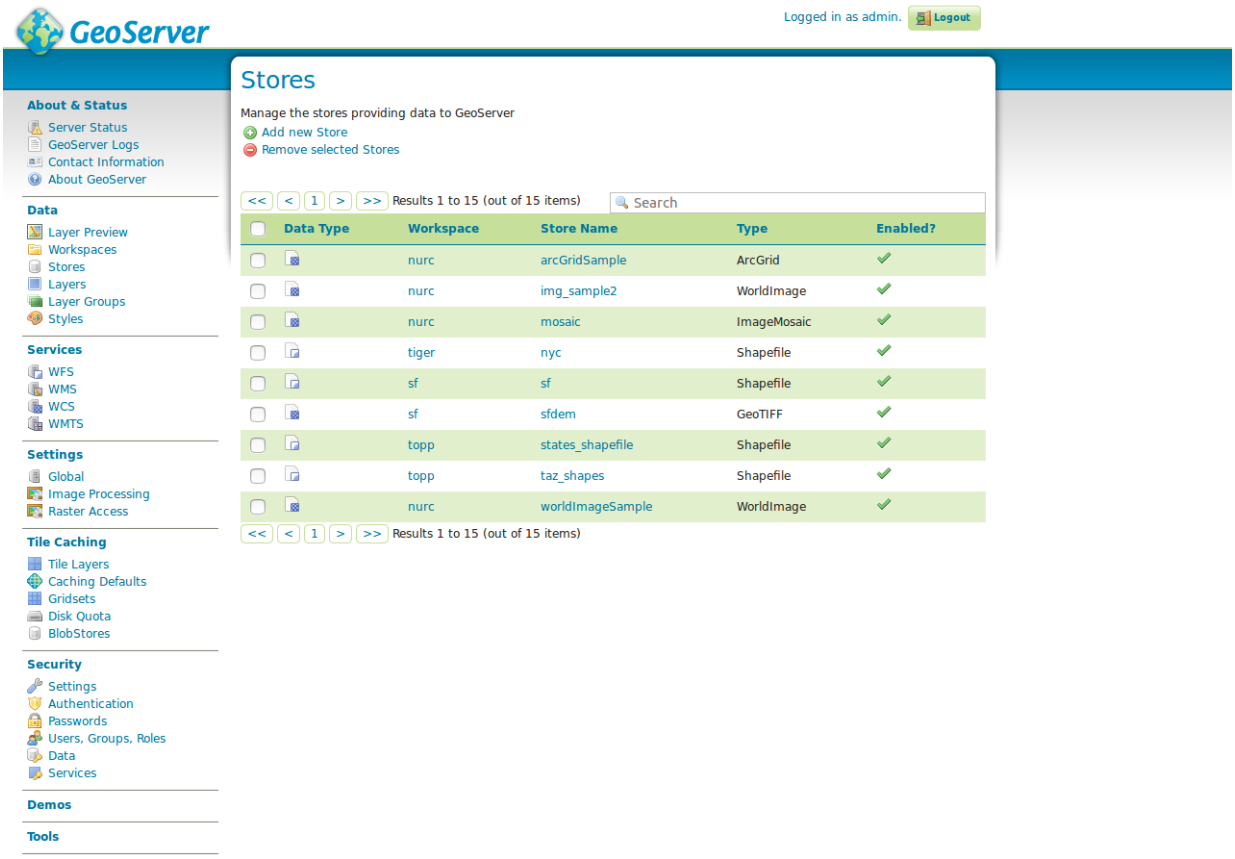
The central element is the so-called workspace, which can initially be understood as a collection object for layers. Similar to a namespace, the workspace organizes objects of a common theme, e.g. the layers of a specific department or topic. Further configuration elements can now be assigned to each workspace. This includes among others the store, the layer (-group), styles and higher-level settings of GeoServer like contact information or global WMS-settings. **Important:** If a GeoServer is initially put into operation, the sequence outlined above must be strictly adhered when creating a layer, that means, first a workspace is created, then a store and then a layer (with styles).

Hint: The GeoServer has several example workspaces in the delivery state (*cite, it.geosolutions, nurc, sde, sf, tiger, topp*). These can be deleted without hesitation in productive operation.

Task:

1. Create a new workspace with the name `FOSSGIS`. For Namespace URI set `http://geoserver.org/fossgis`. Select this workspace as the default workspace.

store

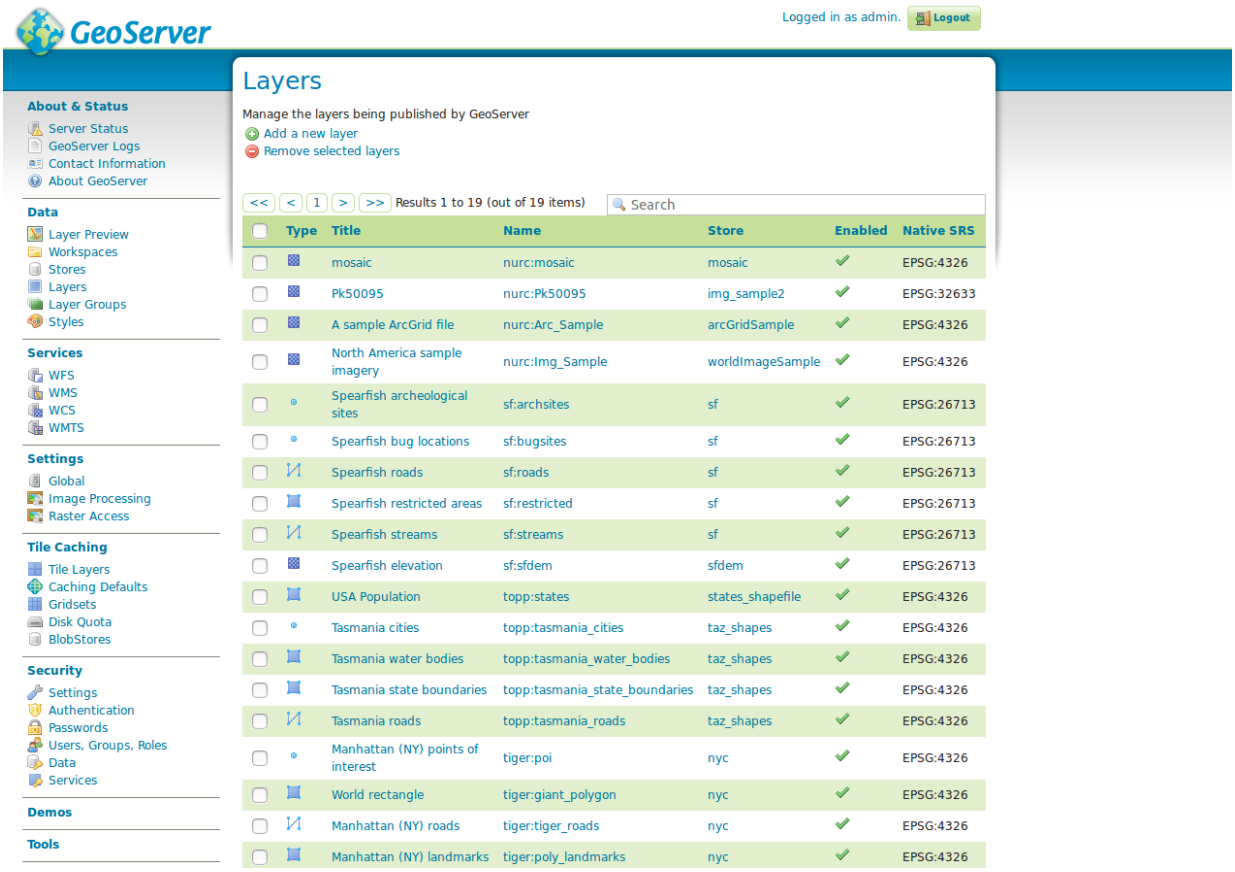


The **(data) store** is a reference to a data source, which contains vector or raster data for publication. Each store is assigned to exactly one workspace. A store includes connection parameters to a database or the path to a shapefile for example. The figure above shows an overview of all available stores, which can be accessed via the store field in the left-hand navigation menu. The overview contains the columns *data type, workspace, store name, type* and *enabled*. The *data type* describes the type of the store, which can be one of the data types listed in the following table. *Workspace* contains the superior name of the workspace, *store name* the name of the store, *type* the concrete storage type (for example, the database) and *enabled* the status of the store.

type	description
	single file/directory (vector data)
	single file/directory (raster data)
	database (vector data)
	WMS
	WFS

The form can be either used to create a new store or to edit an existing one.

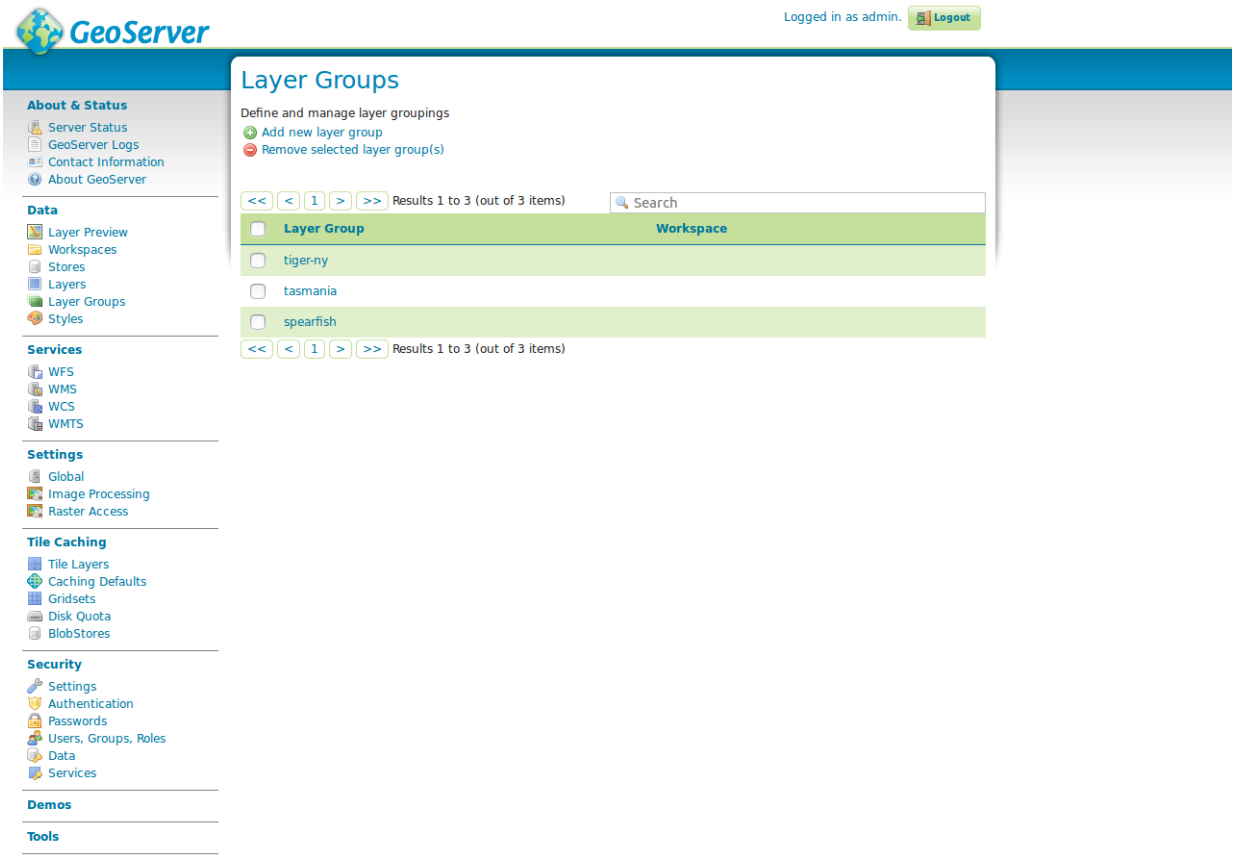
Layers



Layers are the representations of geodata (vector or raster). Each layer contains several map elements (Features), which can be retrieved as rendered raster data (WMS) or as raw data (WFS or WCS). Each layer in GeoServer has in common that it has exactly one workspace and exactly one store. **Important Hint:** GeoServer automatically creates a WMS and WFS for each layer, a seperate creation is not possible. You can only define in the workspace, if all layers in the current workspace can be received as WMS and WFS, only as WMS or only as WFS.

The overview has columns for specifying the respective type, name of the workspace, store and the layer's name, status and the coordinate reference system as EPSG Code. Like all types of lists, you can sort the list ascending and descending by the corresponding columns by left-clicking each columns title. You can also call the parameters of the workspace and of the store directly from the list by left-clicking the corresponding title.

Layer groups



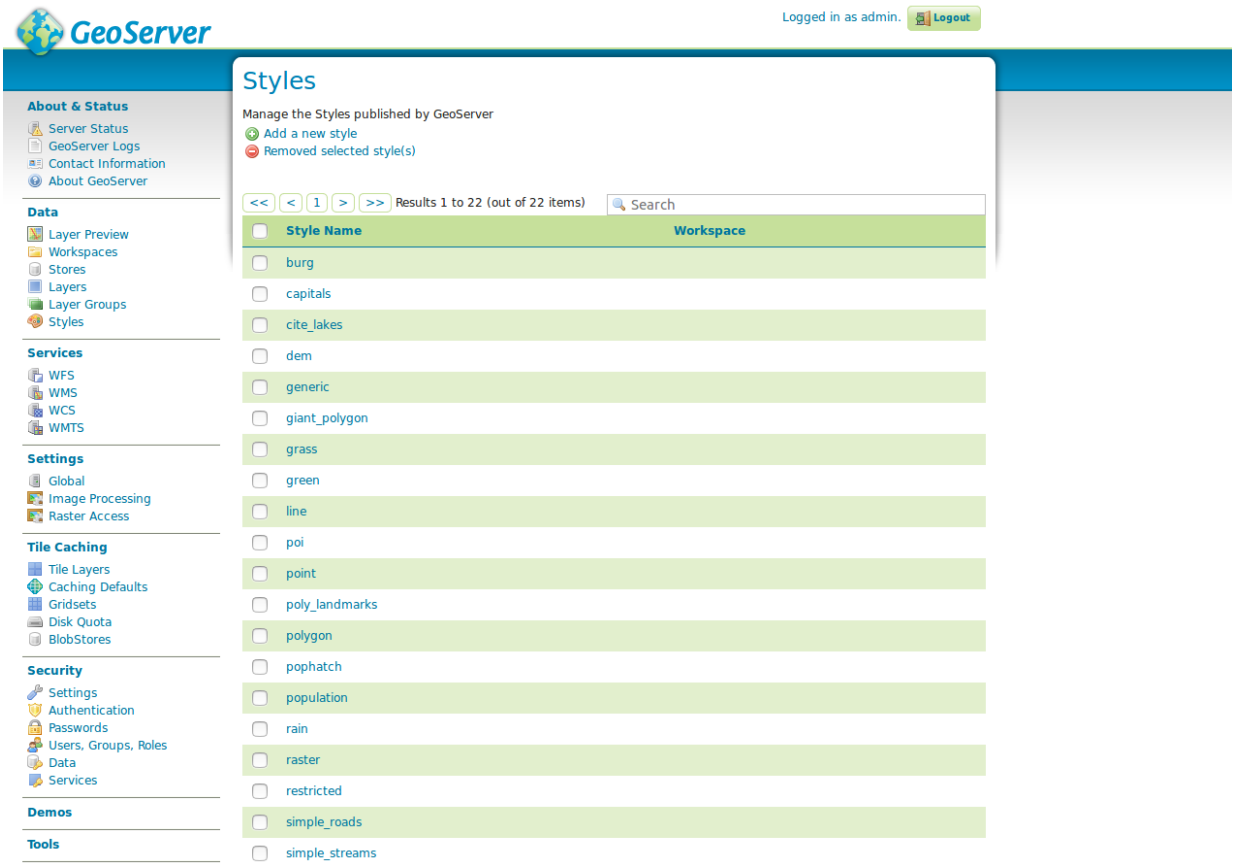
Layer groups are a collection of layers already published in the GeoServer, which are requested together via only one layer.

For creating a layer group using the form. **Add new layer group** at least the following settings are necessary:

- `Name` : Name of the layer.
- `Title` : Title of the layer.
- `Workspace` : The name of the workspace in which the group layer should be created.
- `BoundingBox` : The four fields (Min X, Min Y, Max X, Max Y) contain the BoundingBox of the data of this layer group in the native coordinate reference system. The input can be done manually or automatically with the help of **Generate Bounds** (recommended). **Important:** Without a BoundingBox the layer cannot be created successfully. The values should always contain the complete dataset, because these are queried using the GetCapabilities request when loading a WMS (e.g. in QGIS). There it is relevant for the initial map section.
- `Coordinate Reference System` : Coordinate Reference System as EPSG-Code.
- `Layer` : The buttons **Add Layer** or **Add Layer Group** can be used to add individual layers or other grouped layers to the layer group. A click on the corresponding button opens the *Select Layer* window, in which all available layers of the GeoServer instance appear. A layer can be selected from the window by clicking on the layer's name and is then passed to the layer group. Each selected layer then appears in the table below the layer form element.

In the overview table the order of the layers can be determined by the arrow symbols, where the uppermost layer is drawn as the lowest layer in this layer group. You can also assign a different style to the layer or remove the layer from the group.

Layer style



The drawing rule determines the appearance of a layer depending on attribute properties. GeoServer displays the symbology of a layer with the help of SLD (Styled Layer Descriptor, a XML based dialect), which can be created or changed in the styles dialog.

The list contains all available styles of this GeoServer and allows you to either create a new style by clicking the *Add new Style* button, delete an existing style (Select the style using the checkbox and click *Delete selected styles*) or changing an existing style by clicking on the style's name.

When creating a new style or editing an existing style, the form *New style* respective *Style Editor*. The form offers the following options:

- `Name` : Name of the style. If the style is used by exactly one layer, the style's name should reflect this (e.g. same name as layer).
- `Workspace` : Workspace of the style.
- `Copy from an existing style` : If elements of the styles are used by other styles, an existing style can be chosen as a template.
- `SLD input field` : The layer style is entered in the input field. A full explanation of the SLD syntax would go beyond the scope of this workshop, therefore the GeoServer documentation will be referred to here, structured according to purposes:
  - Table of contents: https://docs.geoserver.org/stable/en/user/styling/sld/index.html
  - points: https://docs.geoserver.org/stable/en/user/styling/sld/cookbook/points.html
  - lines: https://docs.geoserver.org/stable/en/user/styling/sld/cookbook/lines.html
  - polygons: https://docs.geoserver.org/stable/en/user/styling/sld/cookbook/polygons.html
- `Select file` : If the SLD was created in an external editor, this file can be loaded directly into the form.

Important hint: Before the style is saved via *Save*, it should be checked for correctness via *Validate*. If there is an error (e.g. syntax error), an error appears at the top of the form with a hint to the cause of the error.

Settings

Global

The **Global Settings** dialog contains the following relevant configuration options, which may need to be adapted for development or production purposes:

- `Detailed messages` : If the checkbox is set, the XML responses of GeoServer are as readable as possible (by blanks, line breaks etc.). Since this causes larger files, this setting is only advisable for test purposes.
- `Detailed error output` : If the checkbox is set, the full Java stack trace is written to the log file. This causes huge log files, so this setting is only recommended for debugging.
- `Charset` : Which character encoding should GeoServer use? The default value *UTF-8* should not be changed to avoid encoding errors in the GeoServer's responses.
- `Proxy URL` : If the GeoServer is connected via a reverse-proxy (https://httpd.apache.org/docs/2.2/mod/mod_proxy.html), the address of the proxy can be entered in this field. Normally this field is set to an empty string.
- `Logging profile` : By default, GeoServer has five protocol profiles. They differ regarding their sensitivity of the log output. The choice always depends on the environment (development or production):
  - `DEFAULT_LOGGING` : Medium protocol level on almost all module levels of GeoServer.
  - `GEOSERVER_DEVELOPER_LOGGING` : Detailed logging at the GeoServer module level. Only useful if the GeoServer is debugged.
  - `GEOTOOLS_DEVELOPER_LOGGING` : Detailed logging at the GeoTools module level. This selection can be useful if you want to check which SQL statements (e.g. in a GetFeature query) are sent to the database.
  - `PRODUCTION_LOGGING` : Minimal logging, only errors are logged. This setting should be selected for productive use.
  - `VERBOSE_LOGGING` : Detailed logging on all levels of GeoServer. Only useful if GeoServer is debugged.
- `Storage location for log file` : Specifies the storage location for the logging files relative to GeoServer data directory (normally /usr/share/tomcat7/webapps/geoserver/data/). The path is usually to be left at the default of logs/geoserver.log.

Security

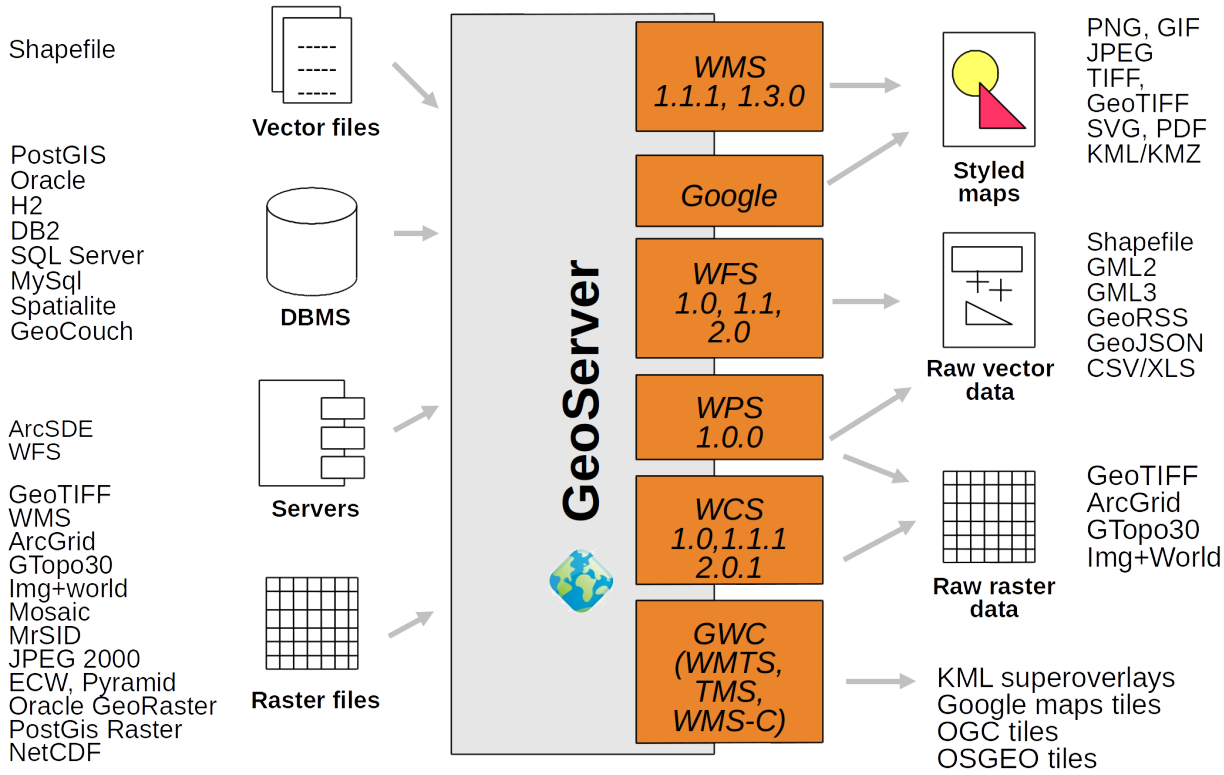
This menu item contains all possible settings for the security of GeoServer. In addition to the usual settings like User- and password management, there is the possibility to restrict access to data storage for certain users.

Task:

1. Please change the default password for user `admin` from `geoserver` to a password of your choice.

Data publishing

Geodata can be available in raster and vector format. These two formats differ in the way spatial information is stored. While vector data displays the information via mathematical vertices and paths, for example a sequence of x,y coordinates for a line feature, raster data stores its information in cells. Depending on the scale and application, it is recommended to use one or the other format for the visualization of the geodata. The following graphic shows an overview of the possible input and output formats of GeoServer.



In the following, the publication of layers from a vector data source is explained first.

Vector data sources

Shapefile

The Shapefile Format, developed by the company *ESRI*, is the most widely used vector format in the GIS world. Shapefiles (also called Shapes) are supported by both proprietary and OpenSource desktop GIS and are thus industry standard.

Task:

1. Create a new data store, which can be used to publish a Shapefile via GeoServer. Choose `us_states` as name and use *states.shp* from the directory `data_dir/data/shapefiles`.
2. Publish your shapefile via GeoServer. This requires information on the spatial reference system (SRS) in which GeoServer should publish the geodata. Enter `EPSG:4326` into the field *Declared SRS*, so the layer is published in the WGS 84 reference system. The BoundingBox must also be specified. It can be calculated automatically by clicking on **Compute from data** and **Compute from native bounds**.
3. Open the newly created layer. Choose OpenLayers as Format for the preview and see your result.


Scale = 1 : 35M
Click on the map to get feature info

PostGIS

PostGIS is a spatial extension for the popular object-relational open source database system PostgreSQL. The extension makes it possible to query spatial data via SQL and moreover offers a variety of options for processing geodata. On OSGeoLive the database system is already installed and set up with a test database.

Task:

1. Create a new data store, which can be used to publish a PostgreSQL/PostGIS table. Choose the following connection settings:
   - host: `localhost`
   - port: `5432`
   - database: `natural_earth2`

- schema: `public`
- user: `user`
- passwd: `user`

Now you have successfully established a connection to a PostGIS database which contains geodata from the *Natural Earth* dataset.

2. Publish a table of your choice via GeoServer! Please enter SRS `EPSG:54009` as Standard.
3. Open the newly created layer in the layer preview!

WFS

GeoServer also offers the possibility to use remote web services as a source for data storage. A WFS (Web Feature Service) offers internet-based access to vector data. In addition to *Basis WFS*, which provides read-only access, *Transaction WFS* also provides written access.

Task:

1. Add a new store, which can be used to publish an already existing WFS cascaded via GeoServer. Use for example the following service: https://www.wfs.nrw.de/geobasis/wfs_nw_dvg?SERVICE=WFS&VERSION=1.1.0&REQUEST=GetCapabilities
2. Publish a service of your choice via GeoServer!
3. Open the newly created layer in the layer preview.
4. If the display does not work with OpenLayers, check the settings of the layer in the tab `Publishing`. GeoServer is not always able to recognize the appropriate style for the geodata automatically. If the *Default Style* is set to *generic*, change it to *Polygon*, *Point* or *Line* (depending on the type of geodata of the WFS).

Raster data sources

GeoTIFF

Task:

GeoTIFF is a common raster data format, as it is well suited for processing geodata due to its lossless storage. In addition to the TIFF image format, GeoTIFF contains coordinates for georeferencing and information about the used map projection.

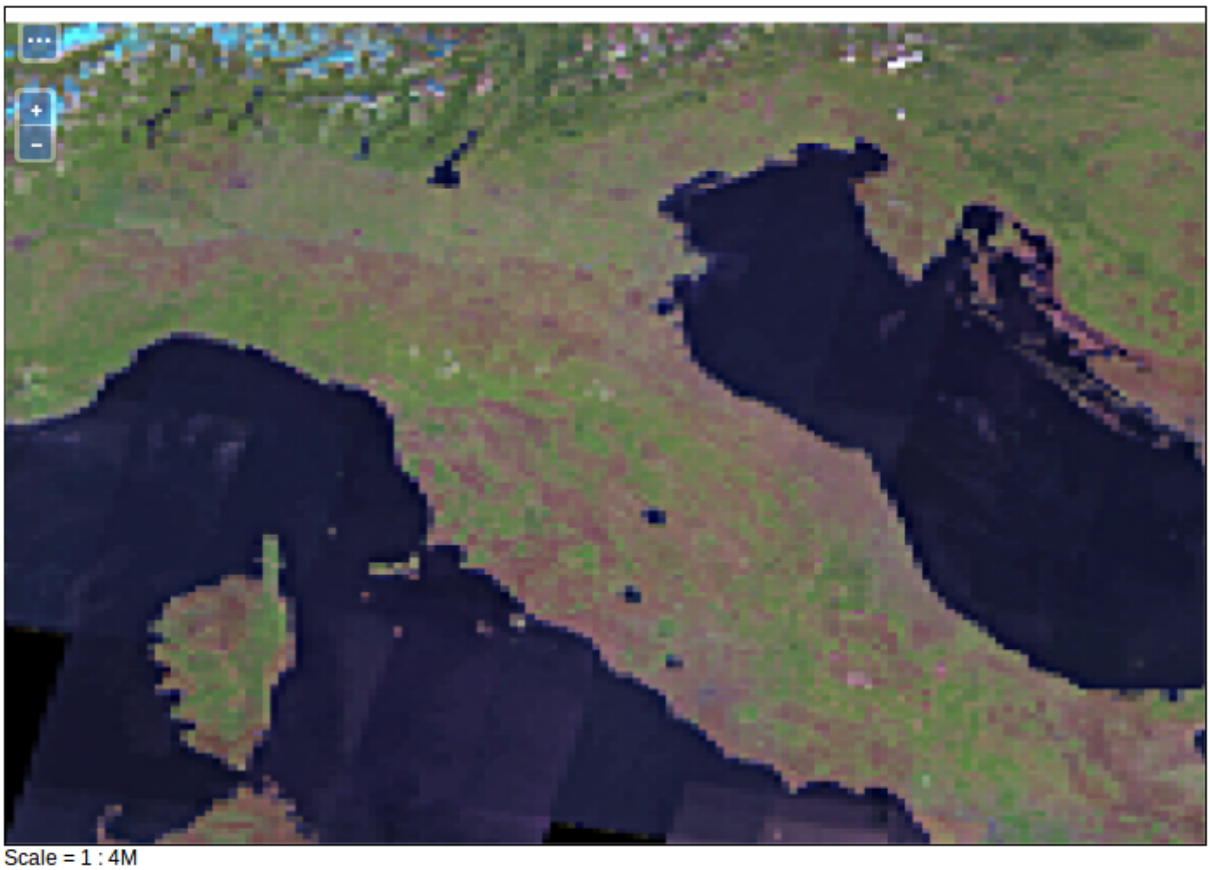
1. Create a new data store, which can be used to publish a GeoTIFF via GeoServer. Use `data/sf/sfdem.tiff` for this.
2. Publish your GeoTIFF with GeoServer! Under the tab `Publishing` and further `WMS Settings` you can select predefined layer styles. Your GeoTIFF is a digital elevation model (DEM) from South Dakota. Choose `dem` as style for this layer.
3. Open the preview for the newly created layer.

ImageMosaic

Task:

A data store of the type ImageMosaic offers the possibility to combine several georeferenced raster data to a mosaic. Thus, for example, two or more overlapping satellite images can be published as a coherent grid.

1. Create a new data store, which can be used to publish a ImageMosaic via GeoServer. Use `coverages/mosaic_sample` for it.
2. Publish your ImageMosaic via GeoServer!
3. Open the preview for the newly created layer.



Scale = 1 : 4M

mosaic

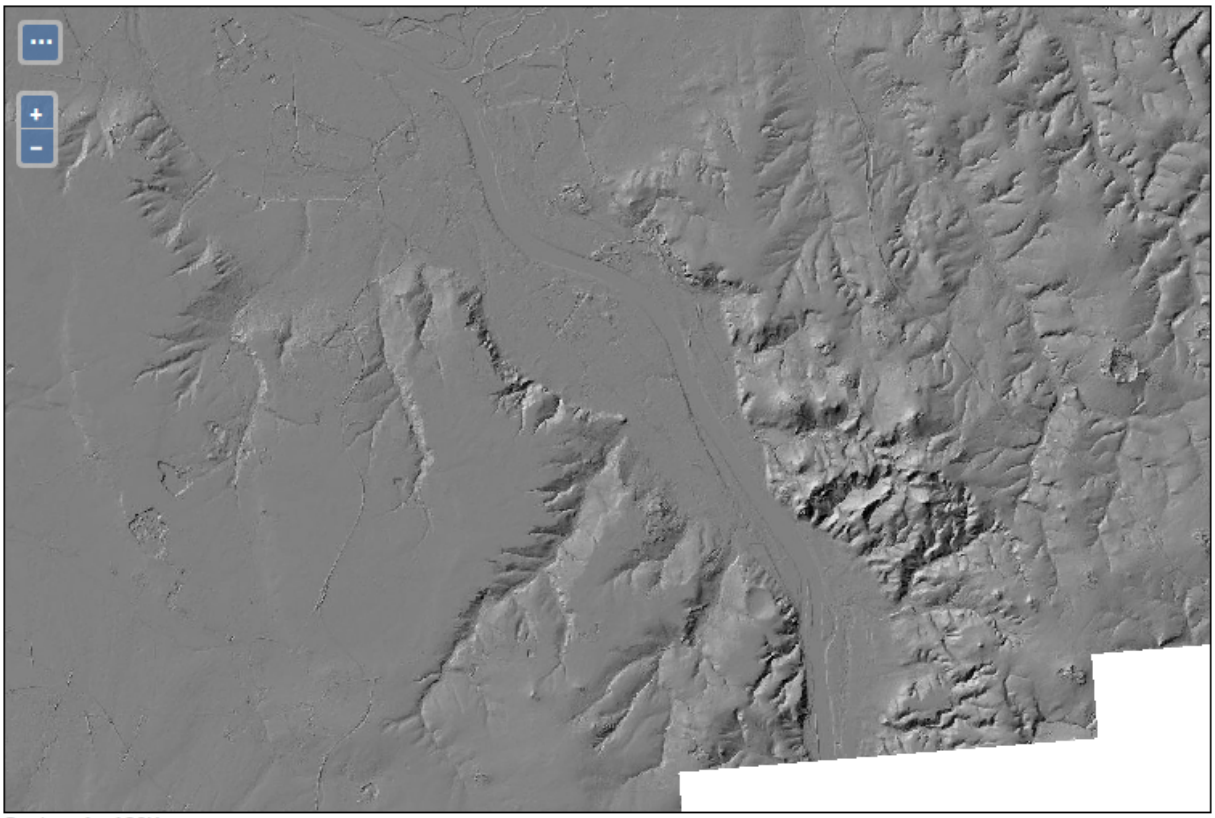
fid	RED_BAND	GREEN_BAND	BLUE_BAND
	122.0	101.0	92.0

WMS

Task:

Cascading an external WMS service to GeoServer can be extremely useful in many situations, because now you have full access to specify all parameters of the service. Even though the remote WMS service was not published with GeoServer, you have the option of applying the full scope of GeoServer to the WMS service locally.

1. Add a new data store, which can be used to publish an already existing WMS cascaded via GeoServer. For example, use the following service: https://www.wms.nrw.de/geobasis/wms_nw_dgm-schummerung
2. Publish your service of choice via GeoServer!
3. Open the newly created layer in the layer preview!



Scale = 1 : 136K

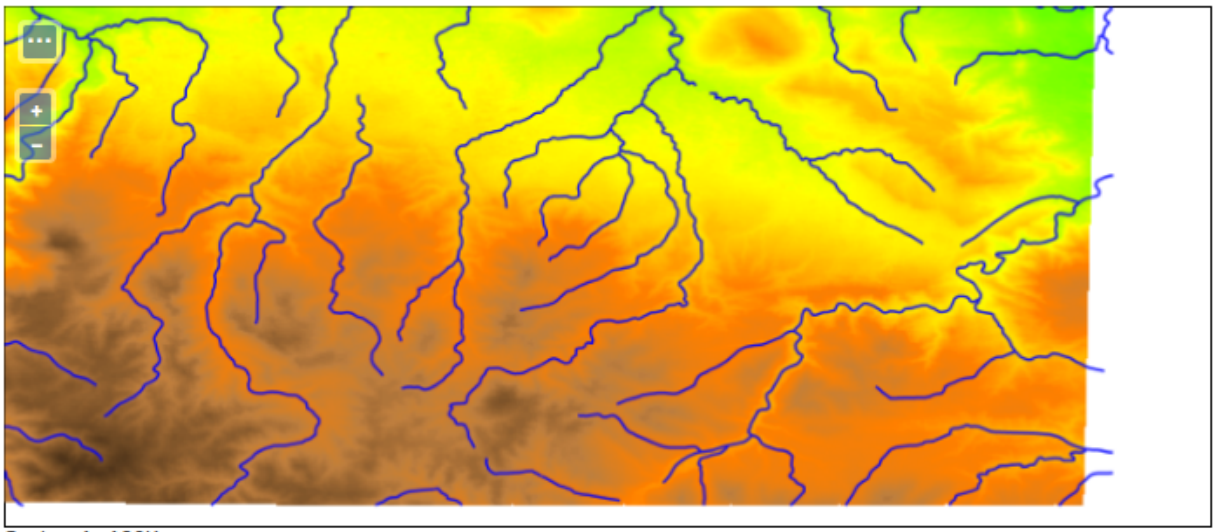
Click on the map to get feature info

Layer groups

Layer groups are a collection of layers already published in GeoServer, which are requested together for only one layer. (see chapter 3.2)

Task:

1. Create a new layer group with different individual layers of your choice.
2. Edit the layer hierarchy. Surface layers (e.g. administrative areas or satellite data) should be displayed in the background and smaller units (streets, towns or rivers) in the foreground.
3. Publish the layer group.



Scale = 1 : 136K
Click on the map to get feature info

Styling

Introduction to SLD

The previous chapters have already made it clear that you can add your own styles to the geodata. GeoServer offers various style formats for this purpose: SLD, CSS, YSLD and MBStyle.

SLD stands for "Styled Layer Descriptor" and is a standard developed by the Open Geospatial Consortium (OGC). It is an XML-based schema that is used to describe and standardize the visual representation of geodata in geographic information systems (GIS).

SLD allows users to specify how geographic data should be displayed on maps. The schema provides options for defining styles for different types of geospatial data, including **points**, **lines**, **polygons** and **raster data**. SLD allows you to specify **colors**, **transparency**, **line widths**, **fill patterns** and other visual properties.

GeoServer uses SLD as one of the main formats for defining styles. By using SLD, the representation of geodata can be standardized and made interoperable, regardless of the GIS software or platform used.

SLD Key Elements

The use of XML-based SLD documents enables fine-grained control over the styling of geometries such as points, lines and polygons. The layer styles are defined using various key components:

- **XML-based style definition:** SLD uses XML (eXtensible Markup Language) as a syntax to specify style information. This enables a clear and structured representation of the desired presentation of geodata.
- **Layer style mapping:** The SLD concept allows style definitions to be mapped to specific GeoServer layers (also known as feature types). Each layer can have an individual appearance, which is defined by a corresponding SLD document.
- **Layer style hierarchy:** Different styles can be defined for the same layer within an SLD document. This enables a hierarchical structure of styles, allowing different styles to be applied based on conditions or scales.
- **Symbolizers:** At the core of the SLD concept are symbolizers that define how different types of geodata (points, lines, polygons, rasters) should be represented. Symbolizers include properties such as color, line thickness (stroke) and fill to shape the visual appearance.
- **Rules and filters:** SLD allows the definition of rules that determine under which conditions certain styles should be applied. Filters can be used to control the selection of rules to be applied based on the properties of the geodata.
- **Zoom-dependent styling:** GeoServer supports zoom-dependent styling, where different styles can be applied depending on the zoom level. This allows detailed control of the appearance at different scales.

Overall, the SLD concept in GeoServer provides a flexible and powerful method for defining and customizing the visual appearance of geospatial data, allowing users to precisely control the appearance of their maps and layers.

Supported Data Formats

GeoServer can style various data types using the Styled Layer Descriptor (SLD) format. A selection of the supported data types is listed here:

- **Points:** Allows the styling of individual points on the map. This can be useful to emphasize locations of cities, points of interest or other individual objects.
- **Lines:** Allows you to define styles for lines, such as roads, rivers or other linear features.
- **Polygons:** Styles surfaces, such as country borders, administrative areas or other polygonal geodata.
- **Raster data:** Stylization of raster data, such as elevation maps or aerial images, where different colors, transparency and other visual properties can be defined.
- **Text labels:** Allows the addition of text labels to specific points, lines or polygons on the map.
- **Complex data:** SLD can also be used for complex geodata structures that combine multiple geometry types or require specialized geometry operations.

The possibilities also depend on the capabilities and extensions of the GeoServer. It is important to check the GeoServer documentation and the specific versions of the supported SLD functions to ensure that the desired data types and styles are supported.

Tasks:

A detailed description of all functionalities as well as some example SLDs can be found in the GeoServer User Manual. Try to solve the following tasks with the help that manual.

1. Define your own style for the States Layer (see task Vector -> Shape), which:
    - All surfaces in a color of your choice (and with a transparency of 50%).
    - All outlines in a color of your choice.
    - Displays all areas with the name of the state.
2. Customize the labeling style so that the labeling is only drawn at a scale of > 10,000,000.
3. Bonus: Have all states colored according to their population. Select your own class boundaries for this.

Style Editor - FOSSGIS:us_states

Edit the current style. The editor can provide syntax highlighting and automatic formatting. Click on the "validate" button to verify the style is a valid SLD document.

Data | **Publishing** | **Layer Preview** | **Layer Attributes**

Previewing on layer: FOSSGIS:states

Style Editor

12pt

```
            <PolygonSymbolizer>
              <Fill>
                <CssParameter name="fill">#CC9E56
                </CssParameter>
                <CssParameter name="fill-opacity">0.5</CssParameter>
              </Fill>
              <Stroke>
                <CssParameter name="stroke">#ffffff</CssParameter>
                <CssParameter name="stroke-width">0.5</CssParameter>
              </Stroke>
            </PolygonSymbolizer>

          </Rule>
        <Rule>
          <TextSymbolizer>
            <Label>
              <ogc:PropertyName>STATE_NAME</ogc:PropertyName>
            </Label>
          </TextSymbolizer>
        </Rule>
      </FeatureTypeStyle>
     </UserStyle>
    </NamedLayer>
</StyledLayerDescriptor>
```

Results:

- Task 1
- Task 2
- Task 3 (Option A)
- Task 3 (Option B)