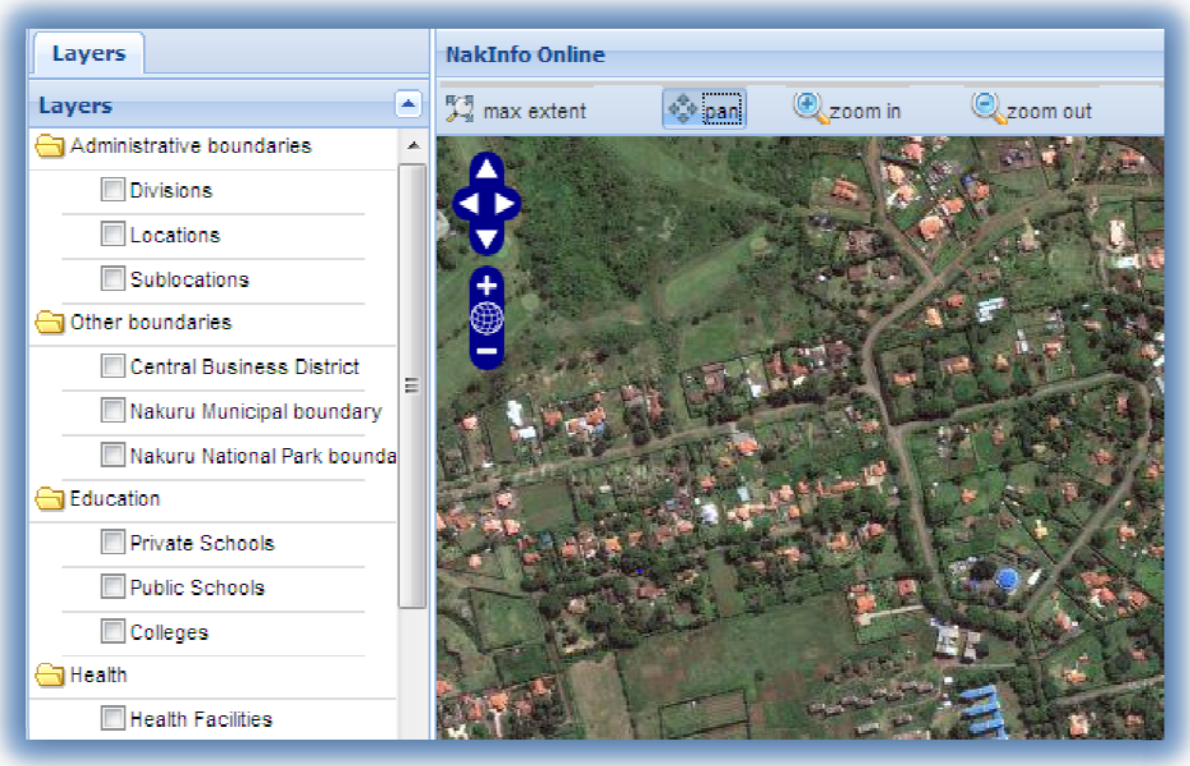
FLAMETREE SYSTEMS ENGINEERING LTD

NAKINFO ONLINE



DEVELOPER'S MANUAL

Version 1

source: https://doi.org/10.7892/boris.68367 | downloaded: 19.10.2024

NakInfo Online Developer's Manual version 1, 2011
By Flametree Systems Engineering Ltd
www.flametree-systems.co.ke

Table of Contents

1. Introduction

The Local Urban Observatory in Nakuru (LUO, Kenya 2003) has developed a progressive and to date unique electronic information service called NakInfo. The objective of LUO is to make residents aware of public services delivery by their Local Authority, in this case the Municipal Council of Nakuru, and give them a voice in achieving improved quality of life. NakInfo facilitates community participation in local government business and demonstrates how to implement such participation in a developing country. The LUO project was formally initiated by the Municipal Council of Nakuru in January 2003, in collaboration with the Centre for Development and Environment (CDE) of the University of Berne (Switzerland) with funding from the Swiss Agency for Development and Cooperation (SDC).

NakInfo Online is an interactive web map viewer developed to provide the public with geographic information for the areas of Nakuru town under the jurisdiction of the Municipal Council of Nakuru, Kenya. NakInfo Online is developed under the Local Urban Observatory (LUO) project, sponsored by Centre of Development and Environment (CDE), University of Berne.

NakInfo Online is a web version of NakInfo 2.1, a desktop Geographic Information System (GIS) application available at the Nakuru Information Centre, Municipal Council of Nakuru offices.

1.1. Technology behind NakInfo Online

NakInfo Online is developed on an open source platform. The open source technologies used in the development of NakInfo include:

- GeoServer (http://GeoServer.org)
- OpenLayers (http://OpenLayers.org)
- GeoExt (http://www.GeoExt.org/)

GeoServer is an open source software server written in Java that allows users to share and edit geospatial data. GeoServer publishes the NakInfo data.

OpenLayers is a pure JavaScript library for displaying map data in most modern web browsers. OpenLayers serves the layers in the form of a map on the web browsers.

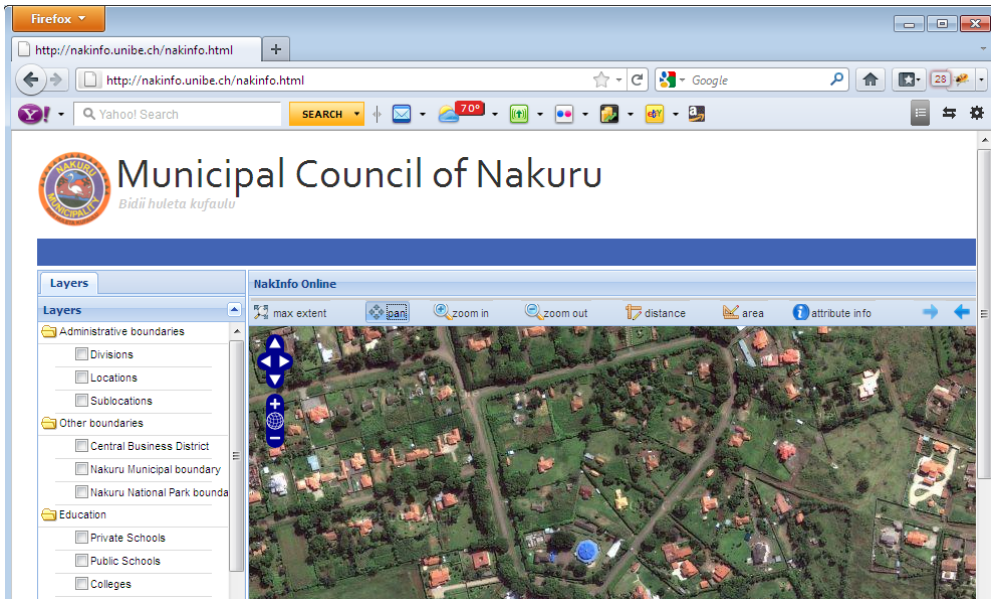
GeoExt brings together the geospatial know-how of OpenLayers with the user interface savvy of Ext JS to help you build powerful desktop style GIS applications on the web with JavaScript. GeoExt enhances the graphical interface of NakInfo online.

1.2. The application: NakInfo Online

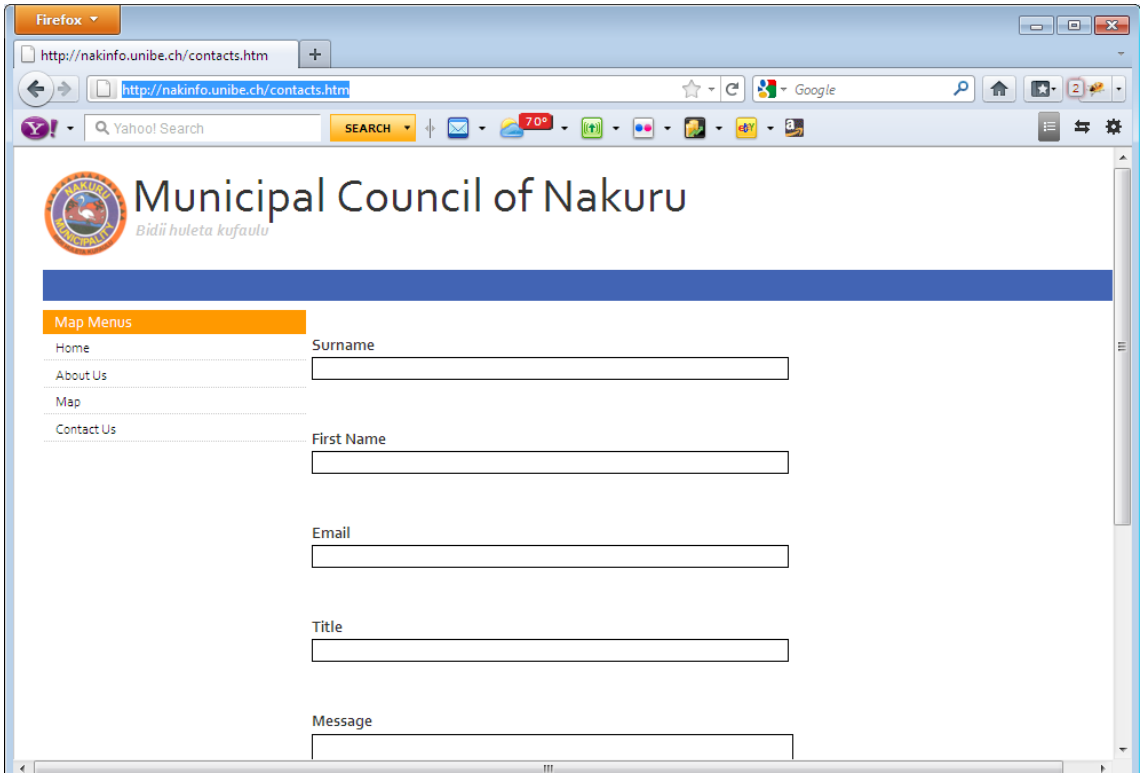
NakInfo Online is presently accessible at http://NakInfo.unibe.ch.

The NakInfo Online website contains a simple html home page shown above and two important links:

- Interactive map (http://NakInfo.unibe.ch/NakInfo.html)



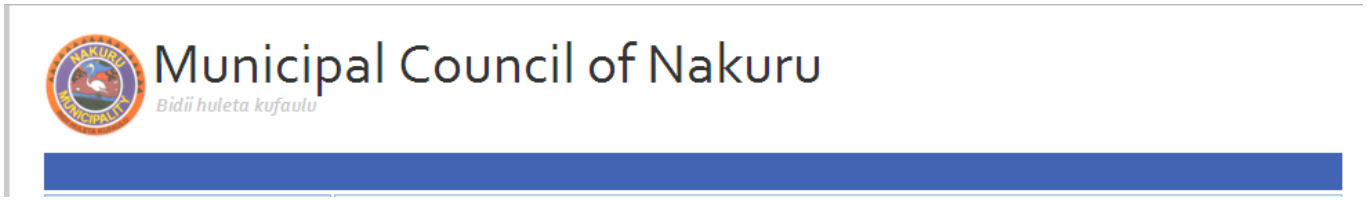
- Contact us http://NakInfo.unibe.ch/contacts.htm



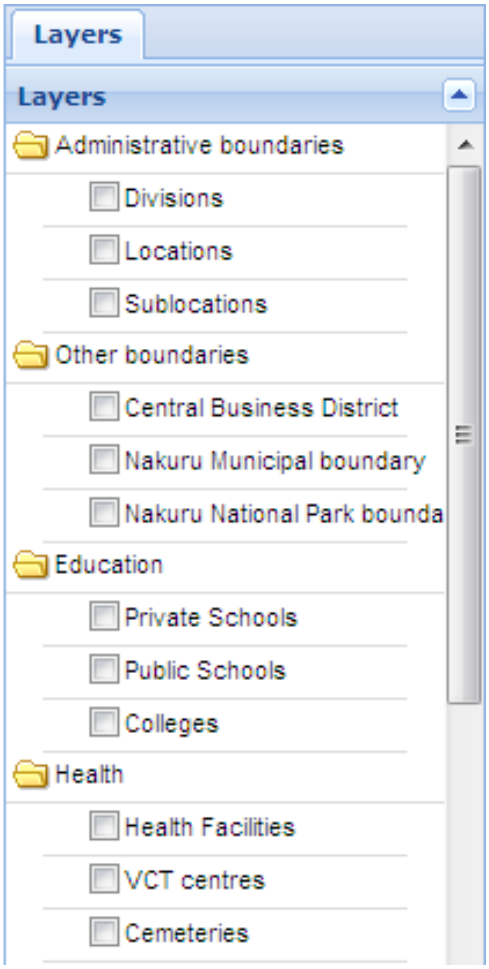
1.3. *Interactive map*

The interactive map of NakInfo Online has three sections:

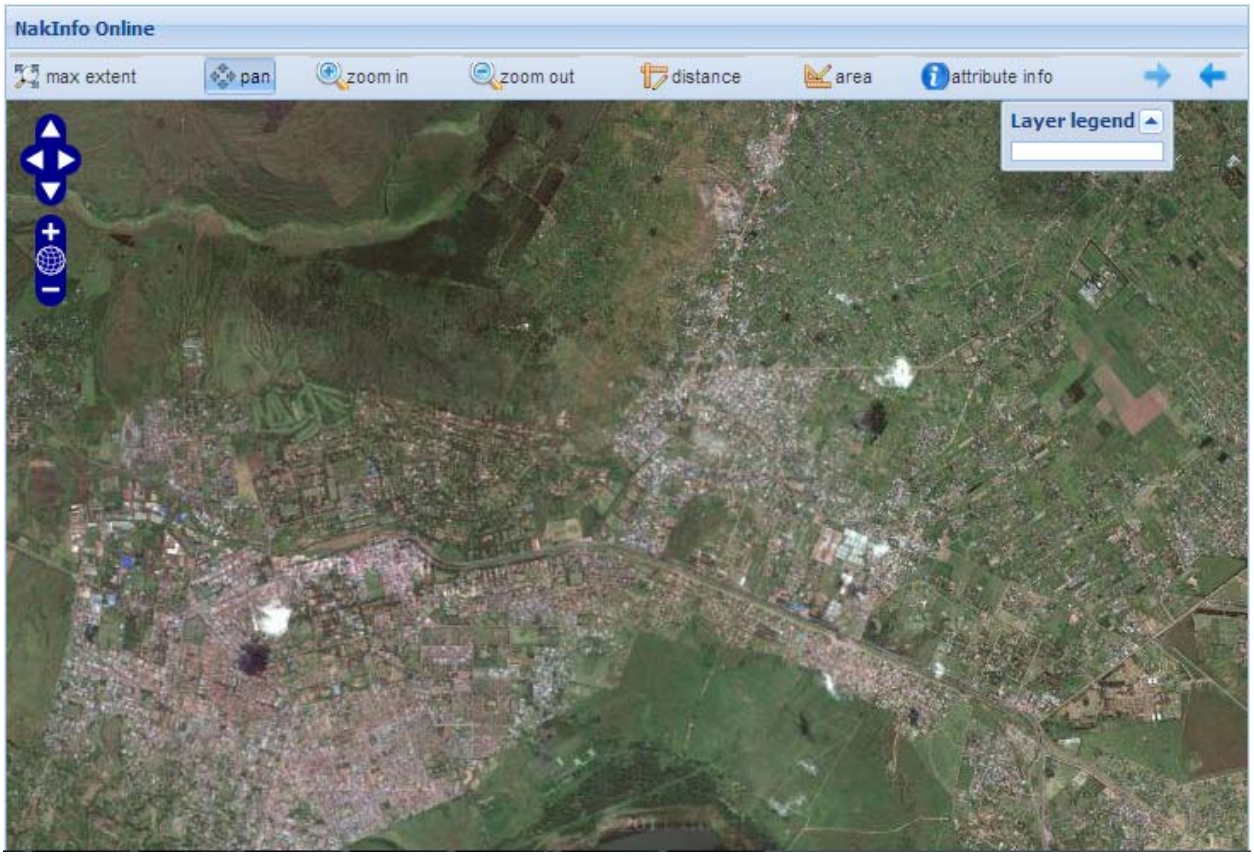
1 Header - which is standard for all the html pages



2 Layer Panel – gives a list of information layers available for website visitors

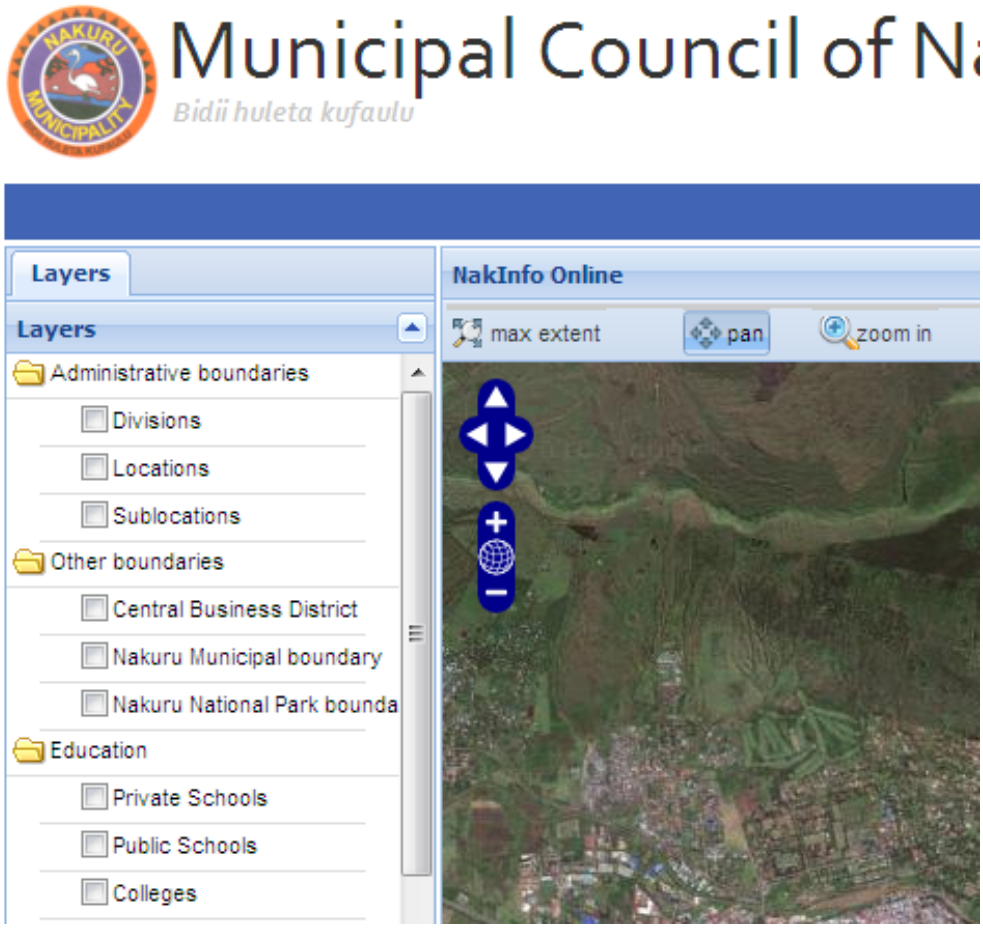


3 Map panel – shows the location of features once the information layer is activated from the Layer Panel

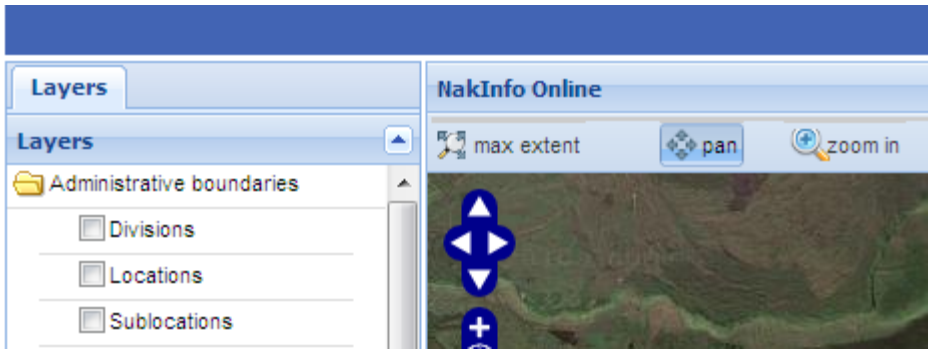


2. Managing the NakInfo Online application

NakInfo Online in the version used for the present manual has 77 shapefiles published on the GeoServer; 36 of these shapefiles are served in the NakInfo Online application. These 36 shapefiles are available online to the public and appear on the Layer Panel as shown below.



The layers are grouped into thematic groups/category. For example, the first thematic group/category is A*dministrative Boundaries* which contain the layers Division, Location, and Sub-location as shown below.



The layers in the layers panel can be removed from the layers panel to hide these from website visitors and at the same time other layers are introduced into the layers panel making these available to visitors.

2.1. Add / replace a layer onto Layer Panel

The process of adding a layer into the layers panel can be invoked by two scenarios:

1. Introduce new layer in the layer tree: You need to introduce a layer that does not already exist in the layer tree
2. Modified shapefile: The shapefile of one of the layers in the Layer Panel has been modified, therefore the existing layer must be updated by replacing its existing shapefile with the modified shapefile.

The process of achieving the above listed task is the same, since each task involves uploading the relevant shapefile onto the GeoServer, publishing this shapefile and serving the resulting layers onto the application via OpenLayers. The steps to follow are:

1. Prepare the shapefile of interest:

   Load the shapefile of interest onto a GIS software such as ArcEditor or open source software such as QGIS and prepare it for publishing. The preparation includes data cleaning, labeling, setting the symbology, spell checking and verification of the projection system.

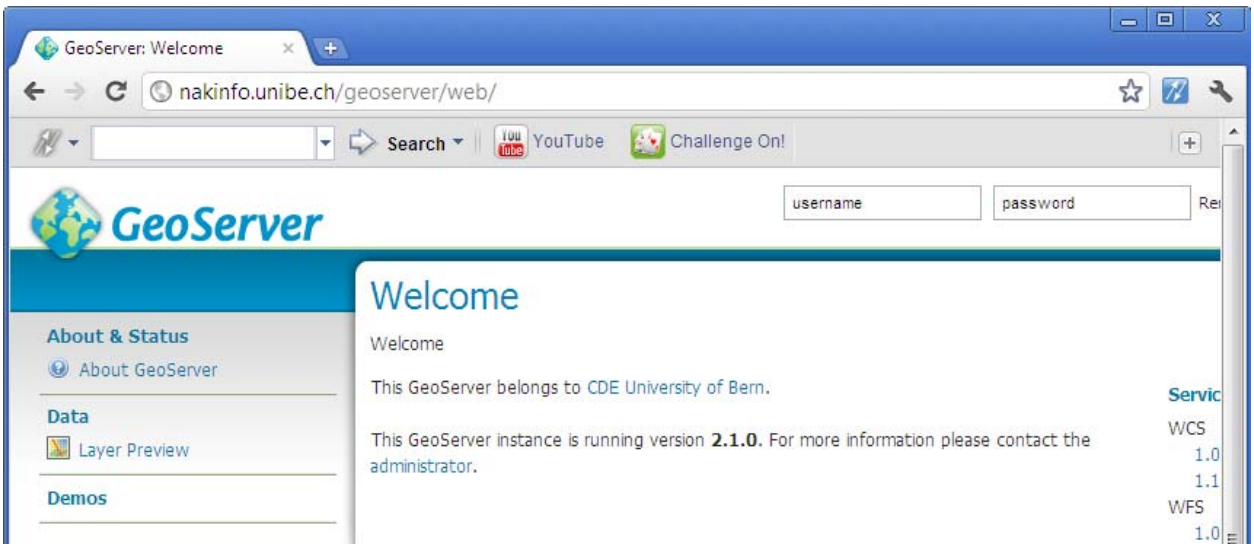
   Note: NakInfo uses projection system WGS 1984 UTM Zone 37 S.
2. Upload the shapefile onto the GeoServer:

Using FTP file transfer client applications such as BitVise Tunnelier and coreFTP, upload the shapefile from your local computer to the remote server. Depending on the type of shapefile, whether vector or raster, the shapefile should be uploaded to the remote folder /GeoServer_data/data/NakInfo/vector or /GeoServer_data/data/NakInfo/raster respectively.

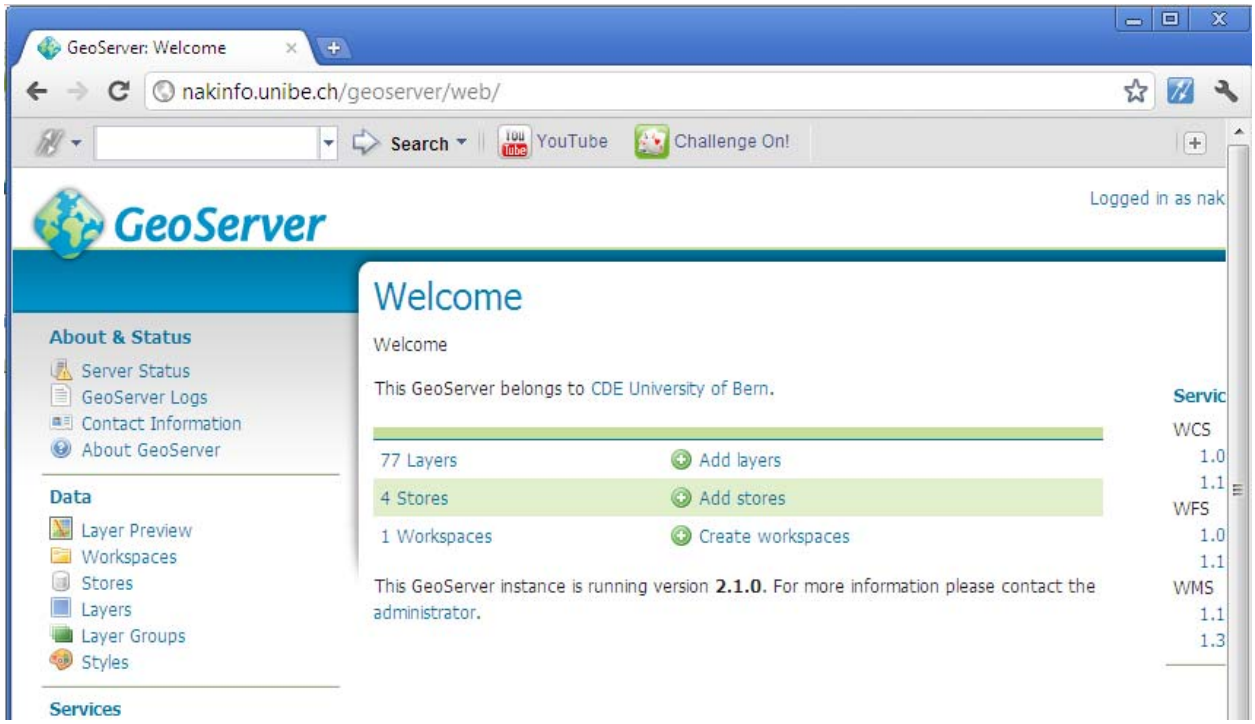
3. Publish the shapefile in GeoServer:

   At the writing of this manual the GeoServer is hosted by Centre of Development and Environment (CDE), University of Berne.

   3.1. Access the GeoServer at http://NakInfo.unibe.ch/GeoServer/web/. Below is a snapshot of the GeoServer welcome screen.

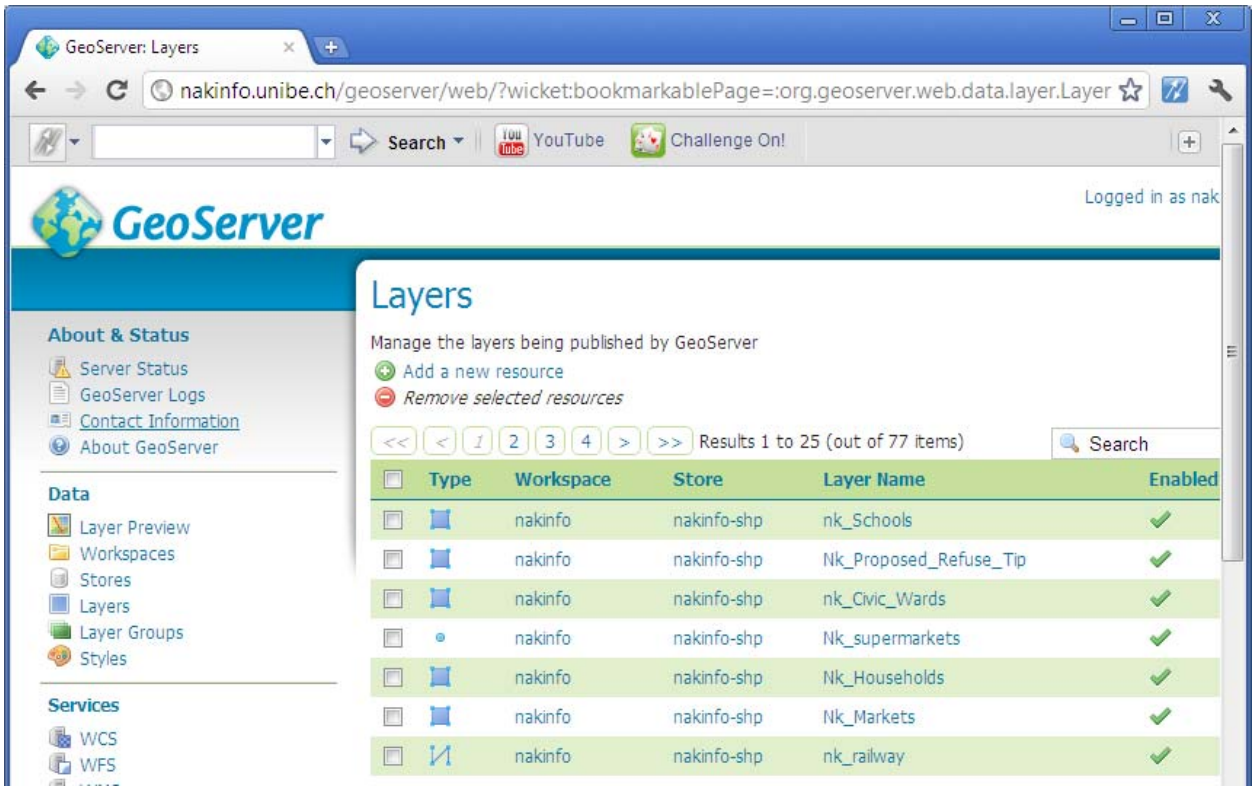


   3.2. Login in with the credentials provided for your username and password. The screen after login displays the contents of the GeoServer and is as shown below.

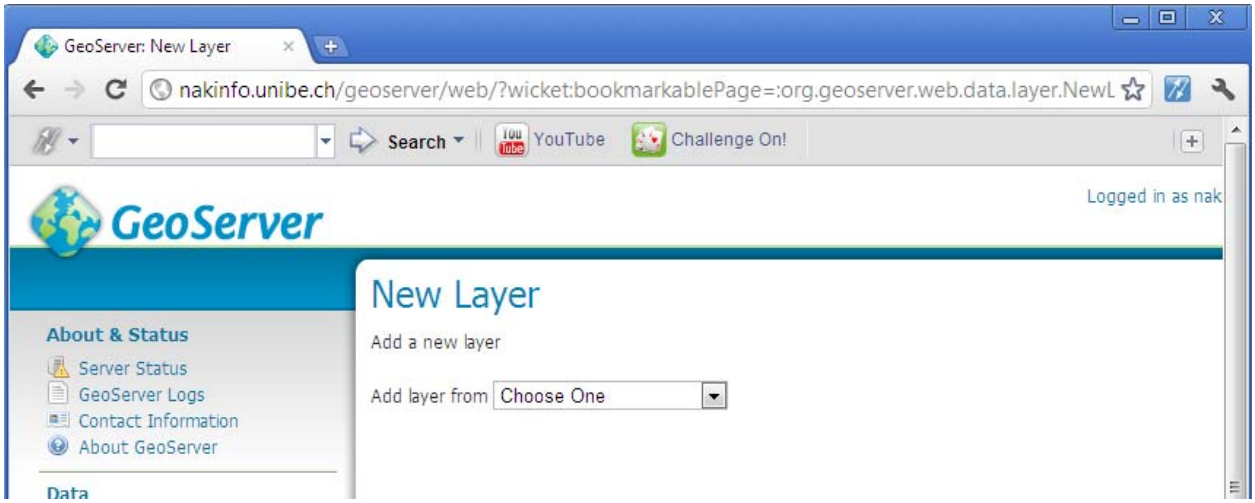


3.3. On the right side of the window, under the *Data* section, Click on *Layers* Layers

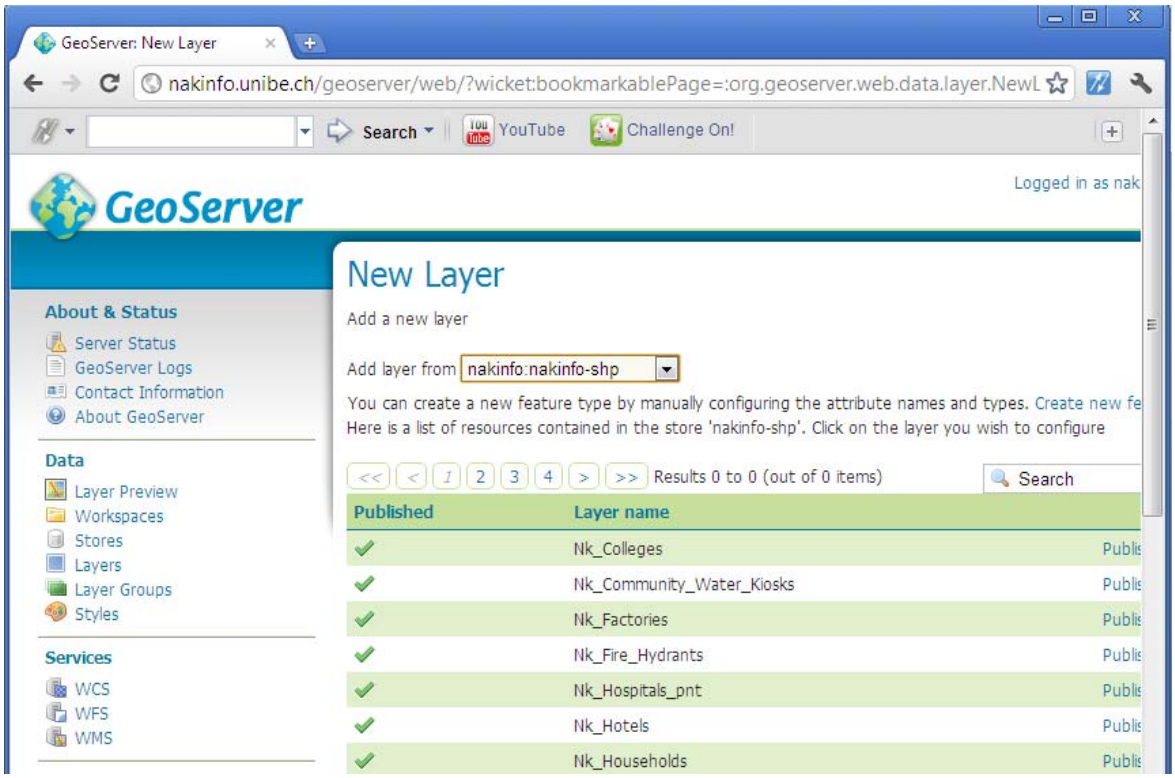
This action opens a window showing all the layers in the GeoServer that have been published as shown below.



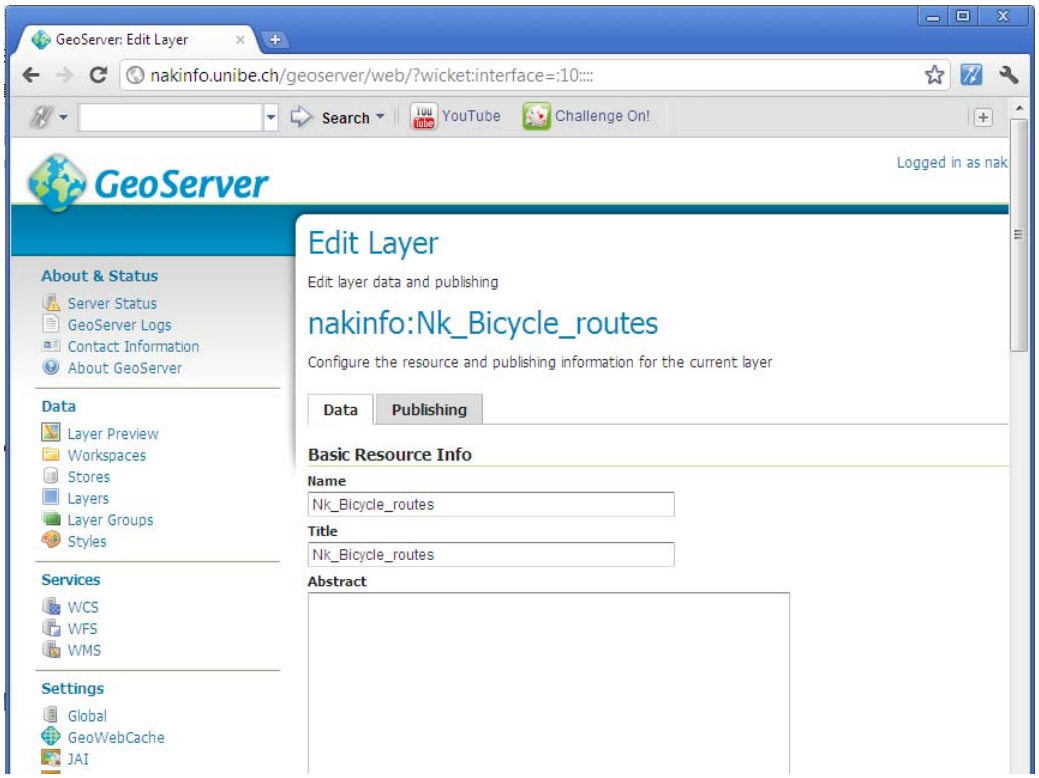
3.4. Click on *Add a new resource,* the window below appears and allows you to choose the workspace containing the shapefiles. The NakInfo online GeoServer has one workspace already defined *NakInfo-shp* and when following step 2 above, the shapefiles are automatically placed into this workspace.



3.5. Choose NakInfo:*NakInfo-shp* from the dropdown menu to open the window shown below.



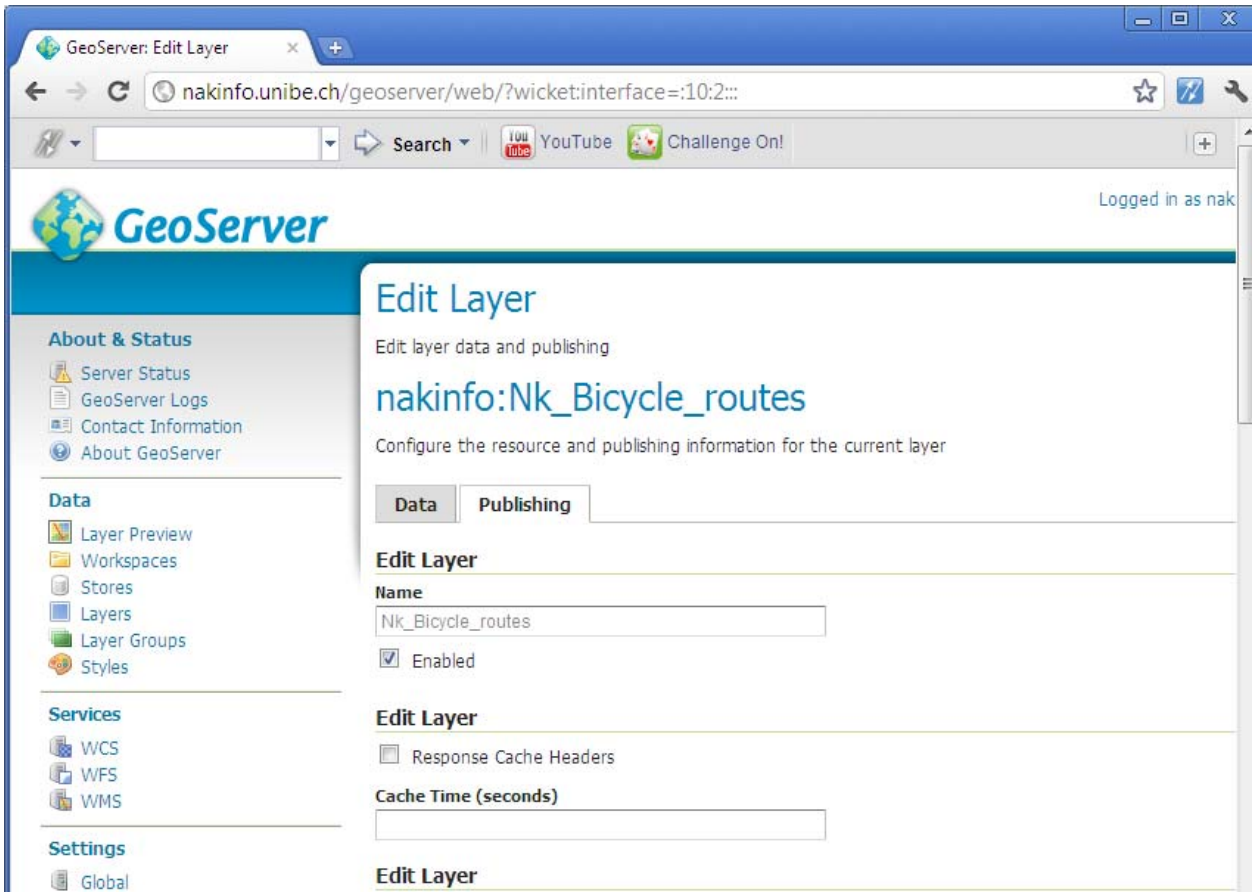
3.6. Find the shapefile that you want to publish and Click on *Publish*.



3.7. Under the *Data* tab, ensure that the following fields are specified as follows.

- Name: use the default name that appears in the text box
- Title: you can also use the default name
- Coordinate Reference Systems section
  - Declared SRS: EPSG 4326
  - SRS Handling: Select '*Reproject native to declared*'
- Bounding Boxes section
  - Native Bounding Box: Click '*Compute from data*'
  - Lat/Lon Bounding Box: Click '*Compute from data*'

3.8. Before clicking on *Save* button, specify the symbology for the layer by clicking on the Publishing tab. This opens the window below.



3.9. Under the WMS Setting section, set the *Default Style* by choosing the SLD file that defines the symbology of the layer being published (SLD files will be discussed in a later paragraph).

3.10. Click the *Save* button.

Detailed explanations on how to publish shapefiles on GeoServer can be found here: http://docs.GeoServer.org/stable/en/user/gettingstarted/shapefile-quickstart/index.html

The shapefile is now published and the layer can be served to NakInfo Online application OpenLayers.

4. Serve the published layer making it available to the public on NakInfo Online.

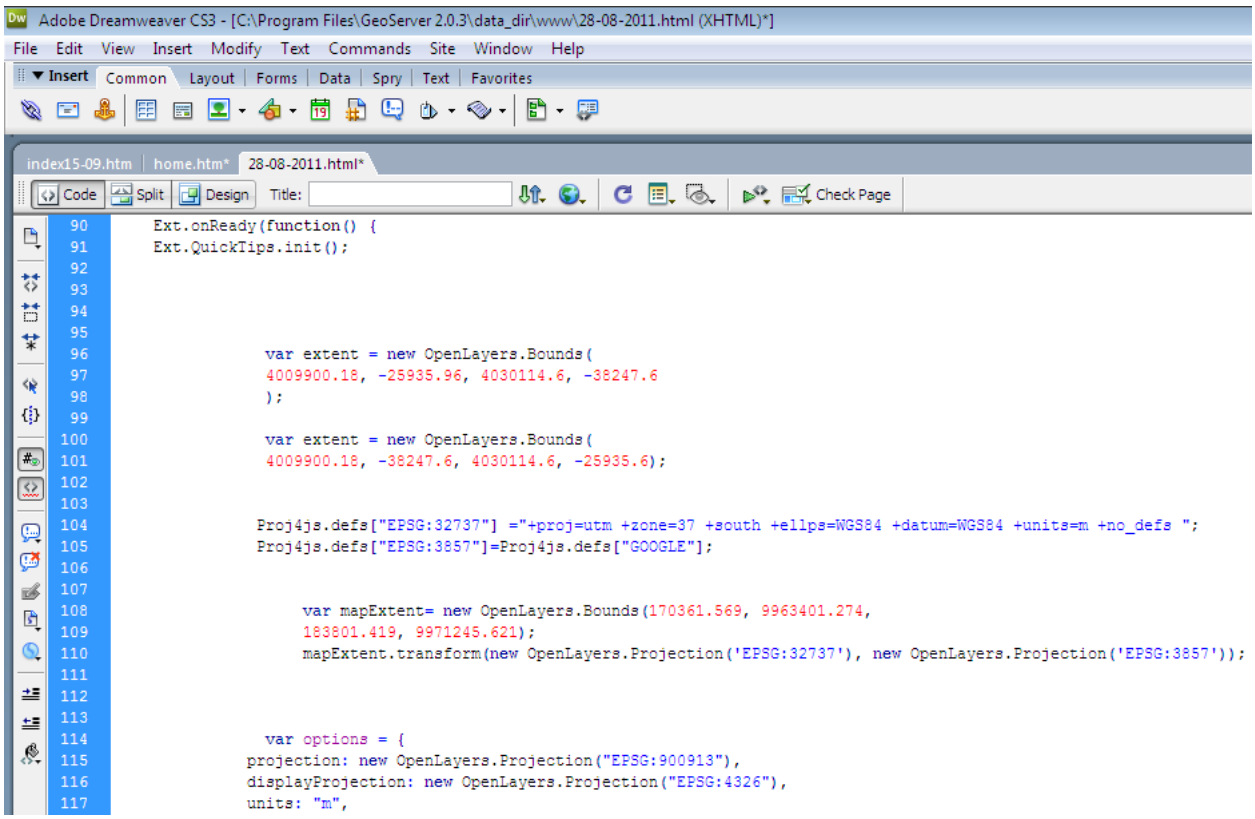
   Making the published layer visible to the public involves writing code using OpenLayers Java script libraries.

   4.1. Download the NakInfo.html (the file that contains NakInfo Online javascript code) from the remote server.

   Note: create a back-up of this file before making any changes to it.

   4.2. Open the file using an html editor.

   The screenshot below shows the open file when using Adobe Dreamweaver CS3.



   The code is divided into sections using comment such as

   //**********Define Layers **********

   //**********Define Toolbar Controls**********

4.3. Define the layer of interest: Find the section *Define Layers* in the code indicated as below.

```
137
138
139  var map = new OpenLayers.Map('map',options);
140
141
142 //*******Define layers*****************************************************
143
144
145         //Administrative boundaries
146         var divisions = new OpenLayers.Layer.WMS(
147         "Divisions",
148         "http://nakinfo.unibe.ch:80/geoserver/nakinfo/wms",
149         {layers:"nakinfo:nk_Division",transparent: true, format: "image/gif"},
150         {visibility: false, opacity:0.4, singleTile: true }
151         );
152
153         var locations = new OpenLayers.Layer.WMS(
154         "Locations",
155         "http://nakinfo.unibe.ch:80/geoserver/nakinfo/wms",
156         {layers:"nakinfo:nk_Locations",transparent: true, format: "image/gif"},
157         //{buffer: 0}
158         {visibility: false, opacity:0.4, singleTile: true}
159         );
160
161         var sublocations = new OpenLayers.Layer.WMS(
162         "Sublocations",
163         "http://nakinfo.unibe.ch:80/geoserver/nakinfo/wms",
164         {layers:"nakinfo:nk_Sublocations",transparent: true, format: "image/gif"},
```

4.4. Copy and paste the code below, replacing the values shown in red with the relevant words.

```
var layername = new OpenLayers.Layer.WMS(
        "Name",
        "http://NakInfo.unibe.ch:80/GeoServer/NakInfo/wms",
        {
                layers:"shapefileName",
                transparent: true,
                format: "image/gif"
        },
        {
                visibility: false,
                opacity:0.4,
                singleTile: true
         }
                );
map.addLayer(layername);
```

Example - defining the layer to display health facilities:

Layername = hospitals,(refers to the name of the object of the type OpenLayers.Layer.WMS, this name will be used to refer to this object within the OpenLayers code)

Name = Health Facilities(this is the name that will be displayed on the Layer Panel

Layers: NakInfo:Nk_Hospitals (this is the name of the shapefile you uploaded and published in GeoServer, it takes the form of 'workspace-name: shapefile-name')

	Type	Workspace	Store	Layer Name	Enabled?
		nakinfo	nakinfo-shp	nk_Schools	✓
		nakinfo	nakinfo-shp	Nk_Households	✓
		nakinfo	nakinfo-shp	Nk_Hospitals	✓

4.5. Define Layer store for the layer.

After defining the layer, the next thing is to include the newly defined layer into a layer store. The layer store can be an already existing layer store or a newly defined layer store. The layer stores are used to categorize the layers into thematic groups which are then used to categorize the layers in the *Layer Panel* on the NakInfo Online application

Find the section Define Layerstore, indicated as

//********Define LayerStore ********** as shown below

```
//******** Define LayerStore ***************************
var waterLayerStore = new GeoExt.data.LayerStore({
                                                    map:map,
                                                    initDir:0,
                                                    layers:[scarcityAreas,waterTreatmentPlant, w_kiosks,boreholes, waterPipes]});
var waterBodiesLayerStore = new GeoExt.data.LayerStore({
                                                    map:map,
                                                    initDir:0,
                                                    layers:[rivers]});

var adminLayerStore = new GeoExt.data.LayerStore({
                                                    map:map,
                                                    initDir:0,
                                                    layers:[sublocations, locations, divisions]});

var boundLayerStore = new GeoExt.data.LayerStore({
                                                    map:map,
                                                    initDir:0,
                                                    layers:[centralBusinessDistrict,municipalBoundary]});
```

As at the writing of this manual, NakInfo Online has 17 layer stores defined. These include: waterLayerStore, waterBodiesLayerStore, adminLayerStore, demographicLayerStore, etc.

4.6. Determine the thematic group to place the newly defined layer that you are adding to the application.

4.7. Find the LayerStore for this thematic group.

4.8. ADD the layername defined in step 4.4 in the layers:[] list of the layer store.

Example - if you are adding a layer to display supermarkets:

Set the layername to supermkts as in step 4.4.

Supermarkets belong to the thematic group Commerce which has a layer store already defined as

```
var commerceLayerStore = new GeoExt.data.LayerStore({
                                                    map:map,
                                                    initDir:0,
                                                    layers:[hotels, factories]});
```

ADD the layername supermkts into the layers:[] list which currently contains two layers; hotels and factories

```
layers:[hotels, factories]});
```

The above layer store code now becomes:

```
var commerceLayerStore = new GeoExt.data.LayerStore({
                                                    map:map,
                                                    initDir:0,
                                                    layers:[hotels, factories, supermkts]});
```

NB: If the newly defined layer fits into any of the existing layer store then, the above steps will allow you to add a new layer to the Layer Panel on NakInfo Online. However, if the layer does not fit in any of the already existing thematic groups, you will have to create a new layer store and add this new layer store into the tree node.

To do this:

4.9. Copy and paste the following code replacing the terms in red color with relevant terms depending on the layer. Example: set LayerStoreName = securityLayerStore and layers:[policeStation, crimeHotspots]

```
var LayerStoreName = new GeoExt.data.LayerStore({
                                                map:map,
                                                initDir:0,
                                                layers:[layername or list of layernames)]});
```

4.10. Define a tree-node for the layer store just created

Find the section Define TreePanel indicated as //********Define TreePanel******* shown below

```
//************Define TreePanel*************************************************
    var layername;
    var tree = new Ext.tree.TreePanel({
        //renderTo:"layerlist",
       // region: "west",
        title: "Layers",
        width: 200,
        autoScroll: true,
        collapsible: true,
        autoscroll: true,
        enableDD: true,
        // apply the tree node component plugin to layer nodes
        plugins: [{
            ptype: "gx_treenodecomponent"
        }],
        loader: {
            applyLoader: false,
```

Within this section find the code; children:[]

```
1152        root: {
1153            nodeType: "async",
1154            children:[
1155                        //{nodeType: "gx_baselayercontainer"},
1156                        {nodeType:"gx_overlaylayercontainer",
1157                         text:"Administrative boundaries",
1158                         layerStore: adminLayerStore ,
1159                         singleClickExpand: true,
1160                         qtip: "click to expand/collapse theme",
1161                         //expandable: true, ** expands on double click
1162                         expanded: true,
1163                         //expandable: true,
1164                         leaf:false
1165                         },
1166                         {nodeType:"gx_overlaylayercontainer",
1167                         text:"Other boundaries",
1168                         layerStore: boundLayerStore ,
1169                         singleClickExpand: true,
1170                         qtip: "click to expand/collapse theme",
1171                         //expandable: true, ** expands on double click
1172                         expanded: true,
1173                         //expandable: true,
```

4.11. Copy and paste the following code once again replacing the words in red with relevant terms depending on the layer of interest

```
//Code

        {
        nodeType:"gx_overlaylayercontainer",
        text:"Administrative boundaries"
                layerStore: adminLayerStore,
                singleClickExpand: true,
                qtip: "click to expand/collapse theme",
                expanded: true,
       leaf:false
        },
```

The steps above adds a new layer onto NakInfo Online

2.2. Remove a layer from the Layer Panel

To remove a layer from the Layer Panel in the NakInfo Online, you have to remove it from the list of layers in the layer store, this will remove the layer from the Layer Panel.

Example - to remove the layer supermkts from the Layer Panel:

Find the relevant layerStore in this case it's the layer store commerceLayerStore

```
var commerceLayerStore = new GeoExt.data.LayerStore({
                                                        map:map,
                                                        initDir:0,
                                                        layers:[hotels, factories, supermkts]});
```

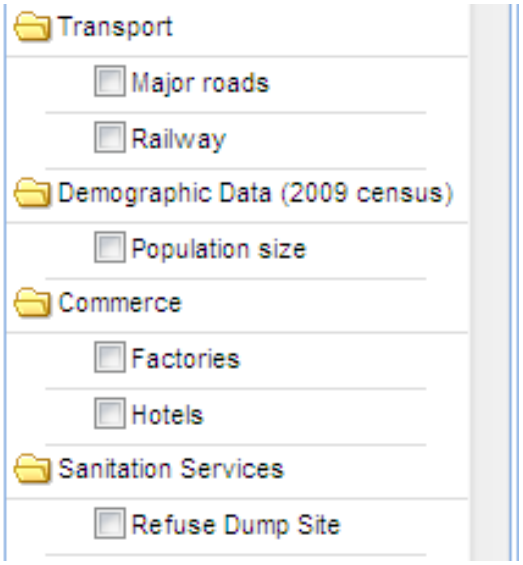
Delete the layer supermkts from the layer:[] list so that the code now becomes as shown below.

```
var commerceLayerStore = new GeoExt.data.LayerStore({
                                                        map:map,
                                                        initDir:0,
                                                        layers:[hotels, factories]});
```

2.3. Remove a Category from Layer Panel

To remove a category from the Layer Panel in NakInfo Online, you have to remove the category's definition from the Tree Panel code.

Example - to remove the category *Commerce* from the Tree Panel:



1. Find the relevant code in the tree containing the relevant layer store in this case it's the category *commerceLayerStore*

```
1216 {nodeType:"gx_overlaylayercontainer",
1217 text:"Demographic Data (2009 census)",
1218 layerStore: demographicLayerStore,
1219 qtip: "click to expand/collapse theme",
1220 singleClickExpand: true,
1221 expanded: true,
1222 leaf:false
1223 },
1224 {nodeType:"gx_overlaylayercontainer",
1225 text:"Commerce",
1226 layerStore: commerceLayerStore,
1227 qtip: "click to expand/collapse theme",
1228 singleClickExpand: true,
1229 expanded: true,
1230 leaf:false
1231 },
1232 {nodeType:"gx_overlaylayercontainer",
1233 text:"Sanitation Services",
1234 layerStore: sanitationLayerStore,
1235 qtip: "click to expand/collapse theme",
1236 singleClickExpand: true,
1237 expanded: true,
1238 leaf:false
1239 },
```

2. Delete the piece of code defining the category *commerceLayerStore* from the Tree Panel code. The relevant code for this example is encircled in red in the screenshot shown above. The above code now becomes:

```
1216 {nodeType:"gx_overlaylayercontainer",
1217 text:"Demographic Data (2009 census)",
1218 layerStore: demographicLayerStore,
1219 qtip: "click to expand/collapse theme",
1220 singleClickExpand: true,
1221 expanded: true,
1222 leaf:false
1223 },
1224 {nodeType:"gx_overlaylayercontainer",
1225 text:"Sanitation Services",
1226 layerStore: sanitationLayerStore,
1227 qtip: "click to expand/collapse theme",
1228 singleClickExpand: true,
1229 expanded: true,
1230 leaf:false
1231 },
```

The two actions above change the categories on the Layer Panel and the Layer Panel now looks like this:

2.4. Change the layer name appearing on the Layer Panel

The layer name appearing on the Layer Panel is defined in the code under *Define Layer* section. To change the layer name that appears on the Layer Panel you need to change the name provided as a parameter in the ***text:"",*** property definition in the code that defines the specific layer.

Example:

To change the layer name *Major roads* to *Class A roads.*

1. Find the layers definition code for the layer for which you want to change the name, in this case we find the *roads* layer definition shown below.

```
334        var road = new OpenLayers.Layer.WMS(
335        "Major roads",
336        "http://nakinfo.unibe.ch:80/geoserver/nakinfo/wms",
337        {
338        layers:"nakinfo:nk_Major_rds",
339        format:"image/gif",
340        transparent:true
341        },
342        {visibility: false}
343        );
```

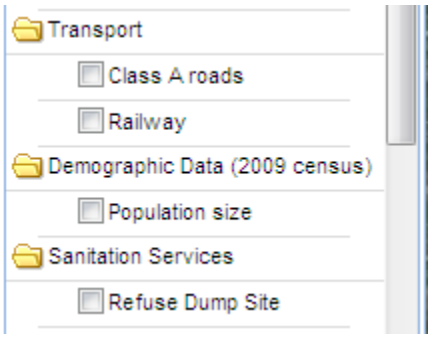
The syntax for a layer's definition is:

```
Var layername = new OpenLayers.Layer.WMS(
                "NameOnLayerPanel",
                "url",
                {layerProperties}
                );
```

2. Change the NameOnLayerPanel which in this case is *Major roads* to *Class A roads* so that the code now becomes:

```
334        var road = new OpenLayers.Layer.WMS(
335        "Class A roads",
336        "http://nakinfo.unibe.ch:80/geoserver/nakinfo/wms",
337        {
338        layers:"nakinfo:nk_Major_rds",
339        format:"image/gif",
340        transparent:true
341        },
342        {visibility: false}
343        );
```

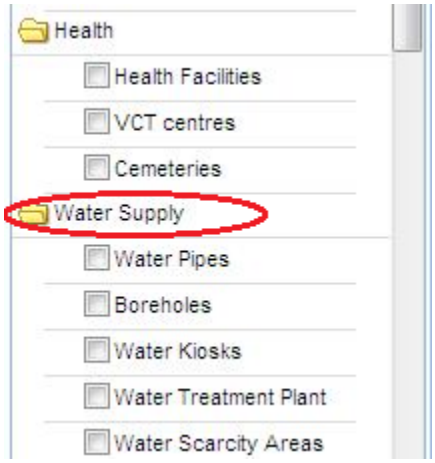
The Layer Panel now shows *Class A roads* instead of *Major roads* as shown below:



2.5. Change category name appearing on the Layer Panel

The category name is defined the *Define Tree Panel* code section. To change the category name, you change the *text:"",* definition provided in the Tree Panel code.

Example:



To change the category name ***Water Supply*** to ***Water Supply Infrastructure***,

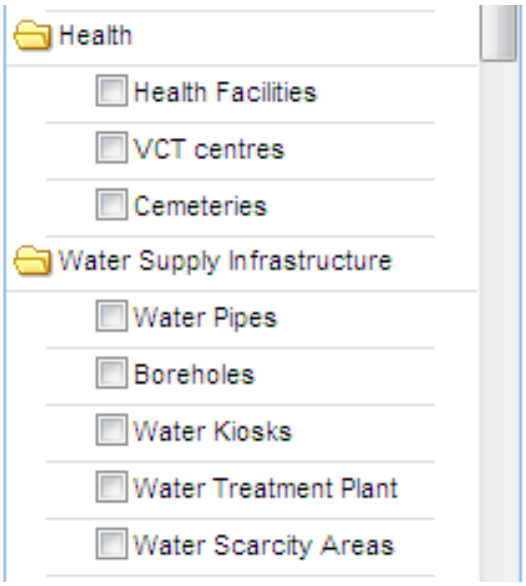
1. Find the Define Tree Panel section /******Define Tree Panel***********/
2. Find the code defining the Water Supply category as shown below:

```
1194    {nodeType:"gx_overlaylayercontainer",
1195    text:"Water Supply",
1196    layerStore: waterLayerStore,
1197    qtip: "click to expand/collapse theme",
1198    singleClickExpand: true,
1199    expanded: true,
1200    leaf:false
1201    },
```

3. Change the ***text:"",*** definition from ***Water supply*** to ***Water Supply Infrastructure,*** so that the code becomes:

```
1194    {nodeType:"gx_overlaylayercontainer",
1195    text:"Water Supply Infrastructure",
1196    layerStore: waterLayerStore,
1197    qtip: "click to expand/collapse theme",
1198    singleClickExpand: true,
1199    expanded: true,
1200    leaf:false
1201    },
```

The result of the task above will change the category name on the Layer Panel from *Water Supply* to *Water Supply Infrastructure*.



2.6. Change a layer's symbology

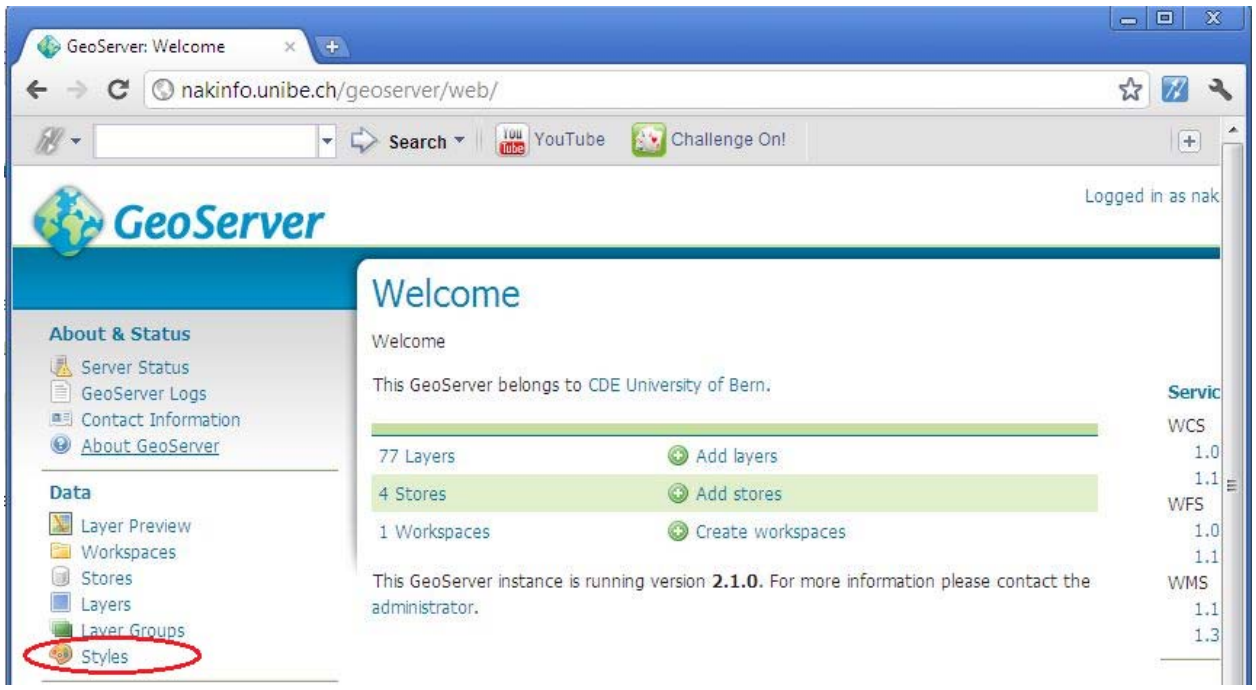
The symbology of a layer is defined in an SLD file, which is uploaded unto GeoServer and the style applied to the layer.

1. Create SLD file

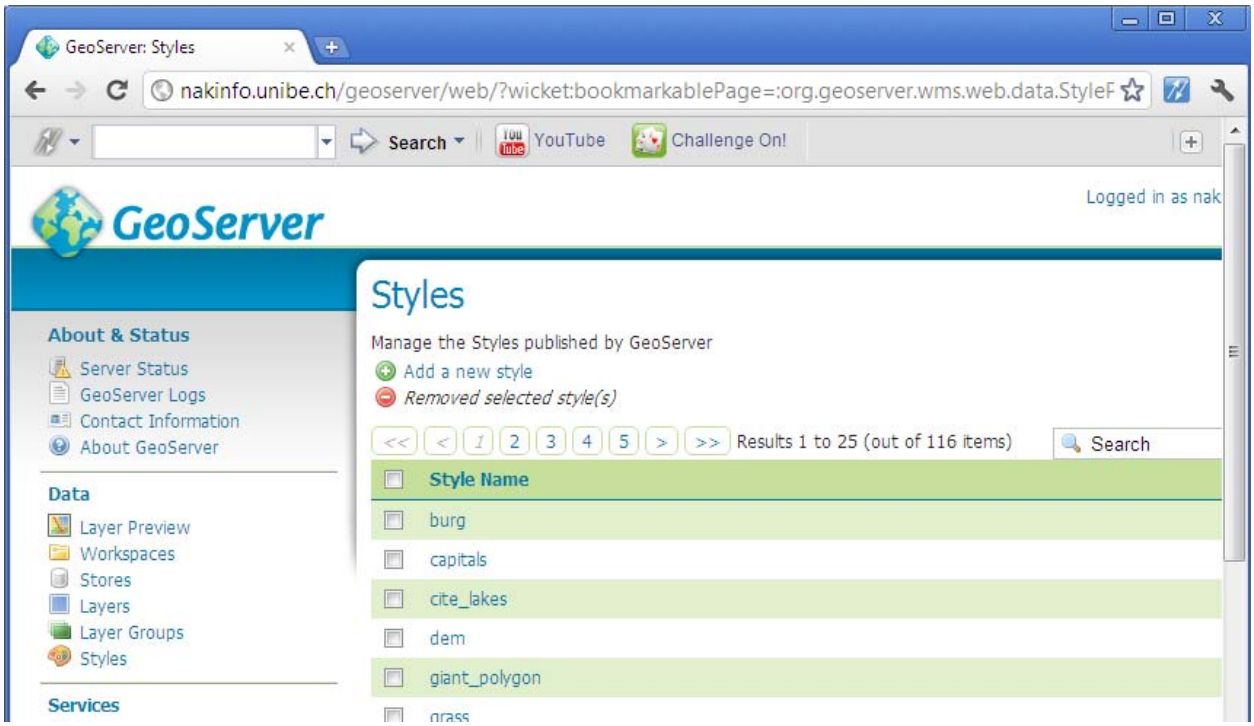
You can write the SLD code from scratch or use applications such as Arc2Earth (open source) to automatically generate the SLD file. Once you have generated the SLD, the next step is to upload it as a style on GeoServer.

2. Log in on the GeoServer

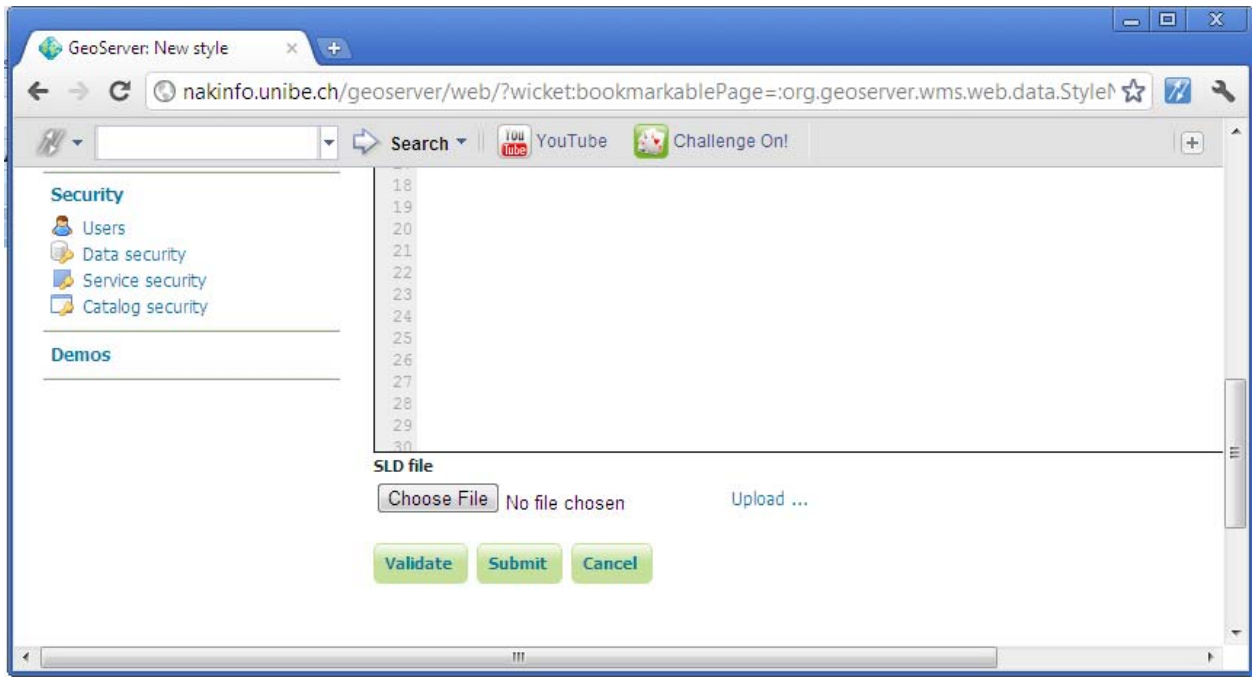
3. Click on *Styles* indicated on the snapshot below.



4. Click *Add New Style*

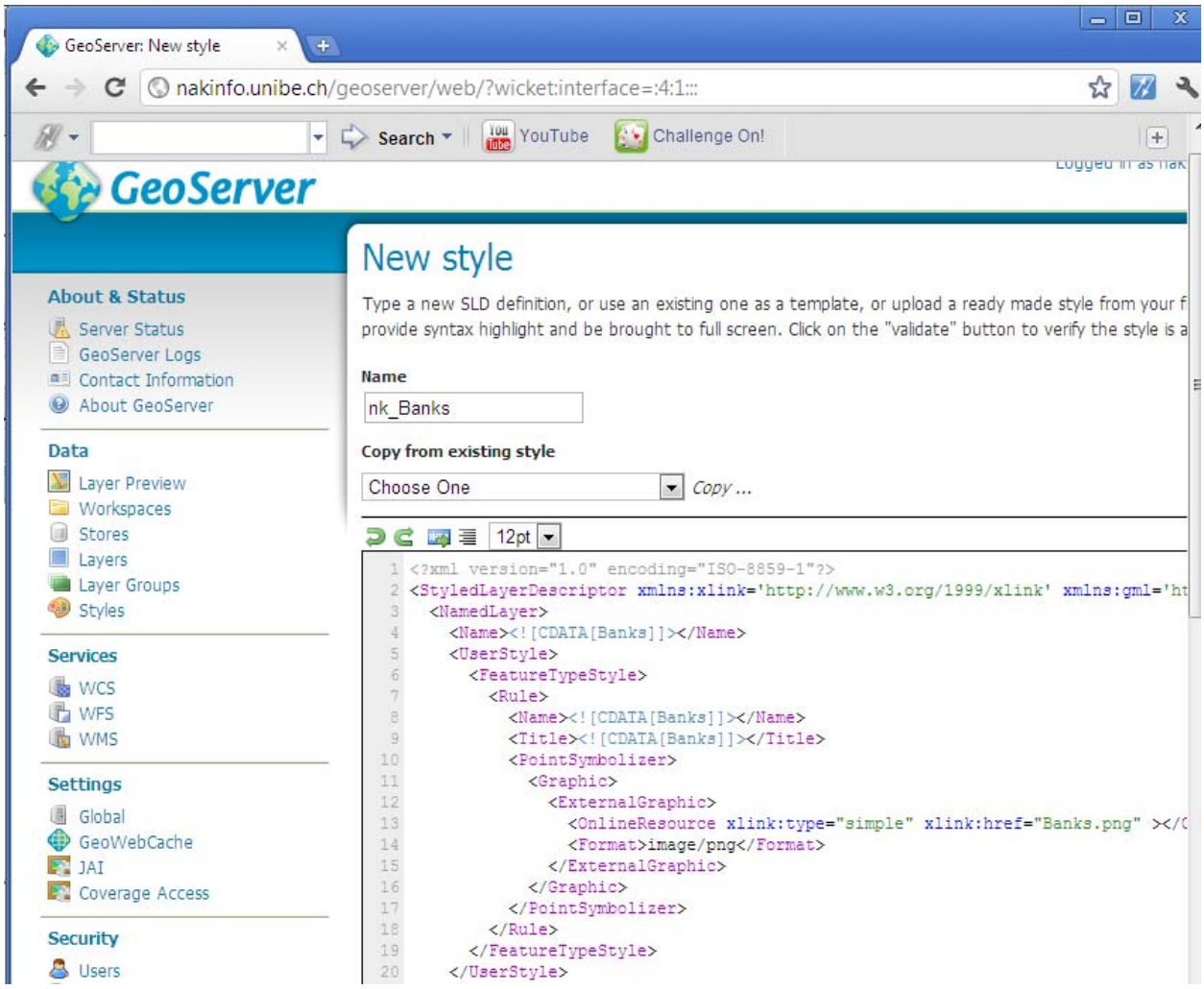


5. On the *New Style* page, click on *Choose File* button at the bottom:



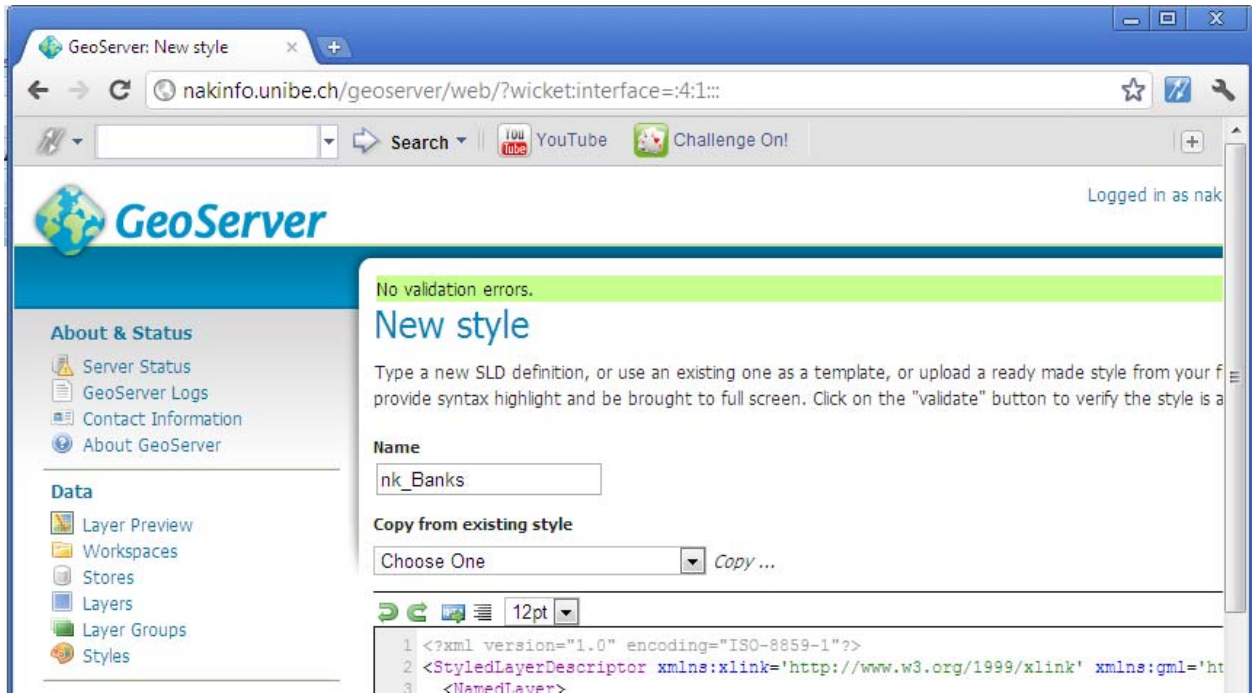
Browse to the SLD file that you have either developed by writing code or the SLD file automatically generated by relevant software.

6. Click *upload.* This will upload the contents of the SLD file into the text box, so that the page now looks like this:



7. Click *Validate*, to validate the code.

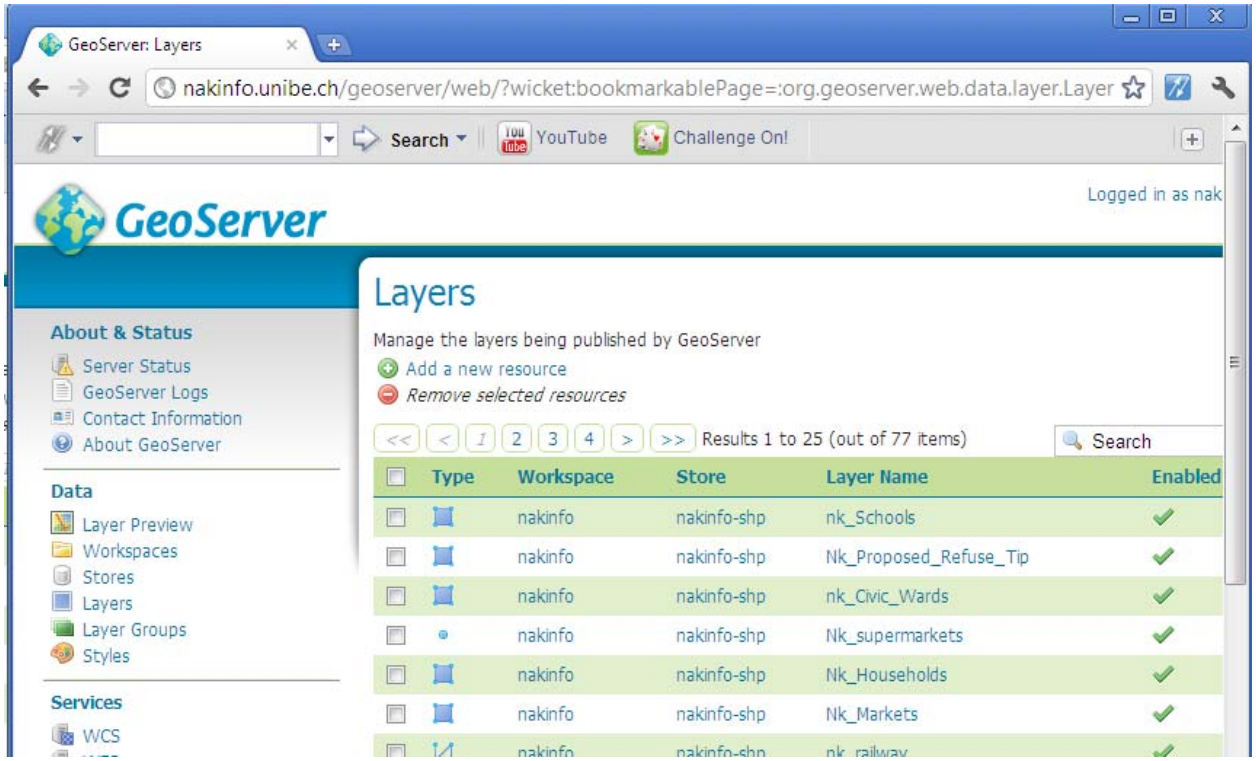
The Validate button may either return the message 'No validation errors" or highlights errors, as shown below:



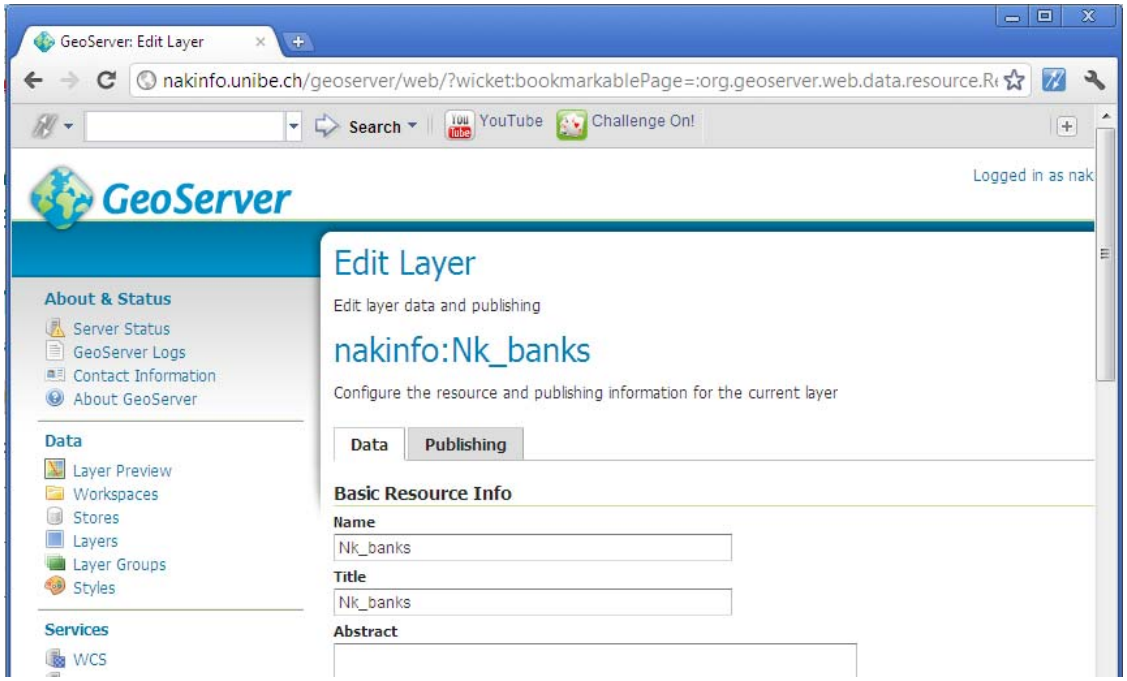
8. Click Submit. Otherwise read the errors returned on the screen and fix these.

Once the style is submitted, the next step is to apply the style to the relevant layer.

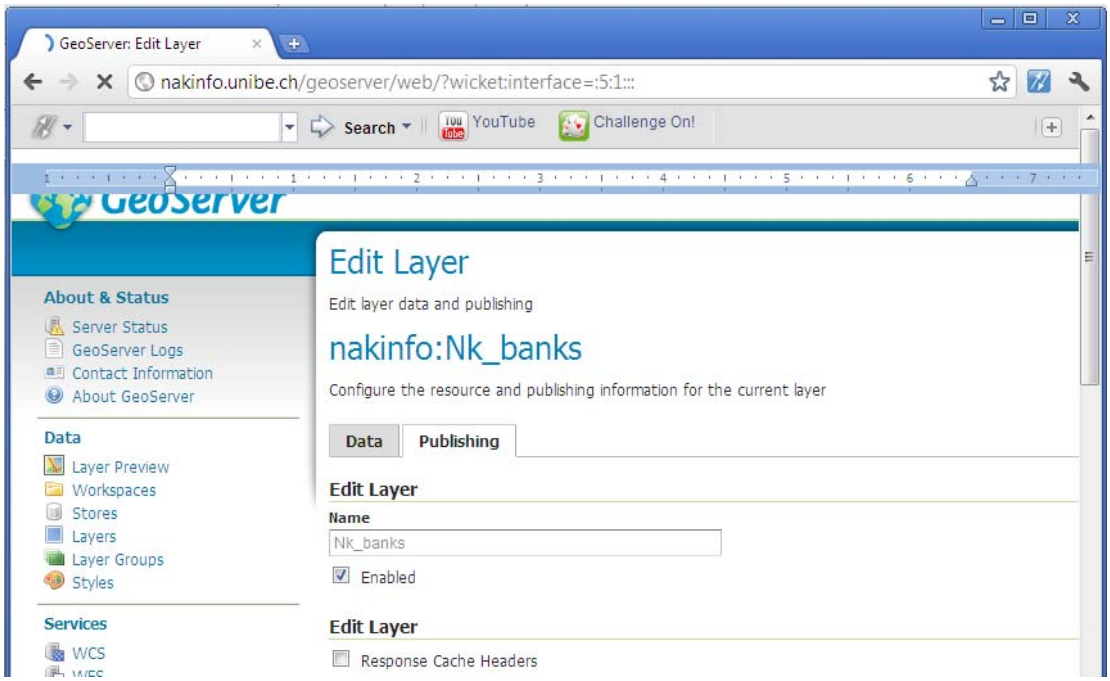
9. Open the Layers page by clicking on *Layers*, you should see the following page:



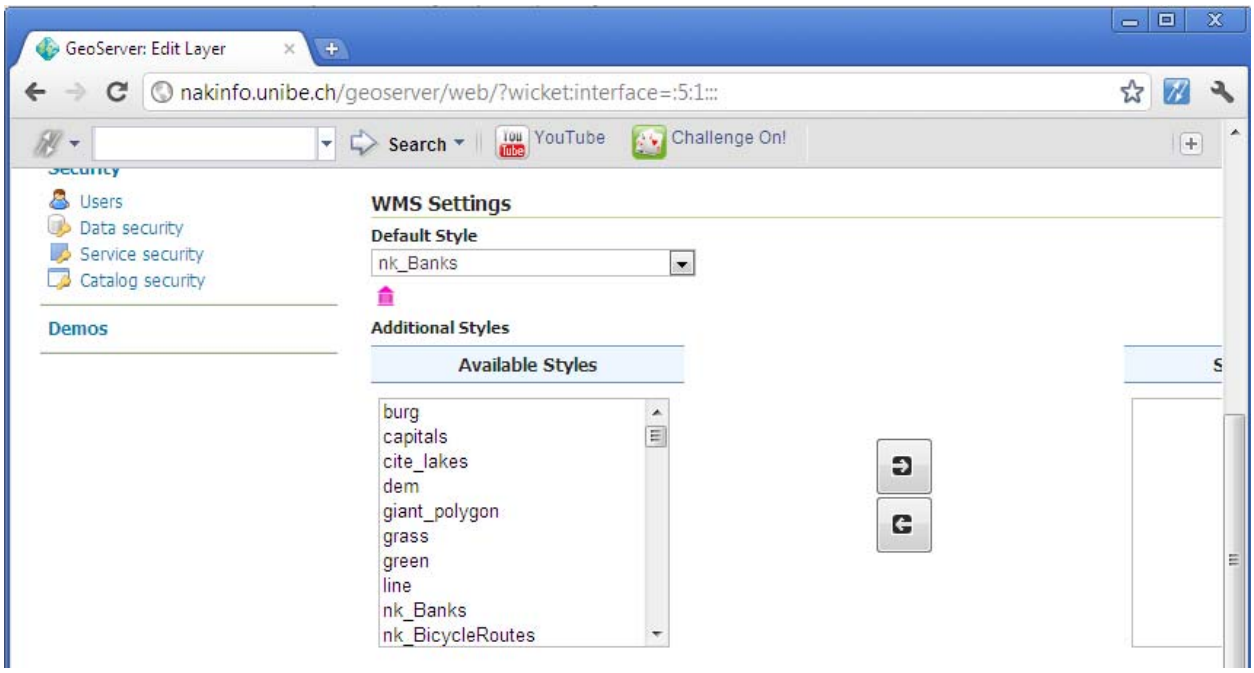
10. Select the layer you wish to apply the new style defined in the previous step. The following screen will appear:



11. Click on the *Publishing* tab to open the following screen:



12. Scroll to the section *WMS Settings*



13. Click on the *Default Style* dropdown and select the style to apply to the layer, select the style you created in the previous steps.
14. Click the *Save* button, to apply the changes to the layer.

+++