# Setup GeoServer

GeoServer includes several styles designed for IMIS. To set up these styles in IMIS, create a workspace and corresponding datastore connecting to the IMIS Database after installing GeoServer. The corresponding information related to the names and other attribute information related to the workspace, stores, layer and styles information including CSS, SLDs, and SQL queries are included and regularly maintained in the technical document section with the filename `geoserver_document.md` within the documentation sub-directory in the GitHub repository named web-app source code. To create a new workspace, store, layer, and styles in a geoserver, follow the steps as mentioned in Section Creating New Workspace, Store, Layer & Style in Geoserver. Each workspace, store, layer and style must be created to ensure the proper functioning of the map interface. Currently, there is one workspace, nine stores, thirty layers and one hundred and seventeen styles. Each of these must be created as per the geoserver_document.md values in the newly set up geoserver. Alternatively, the present workspace is also provided in the GitHub repository under the deployment_documentation in the sub-directory geoserver_workspace_setup, which can also be imported into the geoserver through minor modifications to the XML file that is mentioned below.

To launch GeoServer, start the GeoServer service and navigate to http://localhost:8080/geoserver in your web browser. This URL directs you to the GeoServer homepage. Log in using the username and password configured during the installation process.
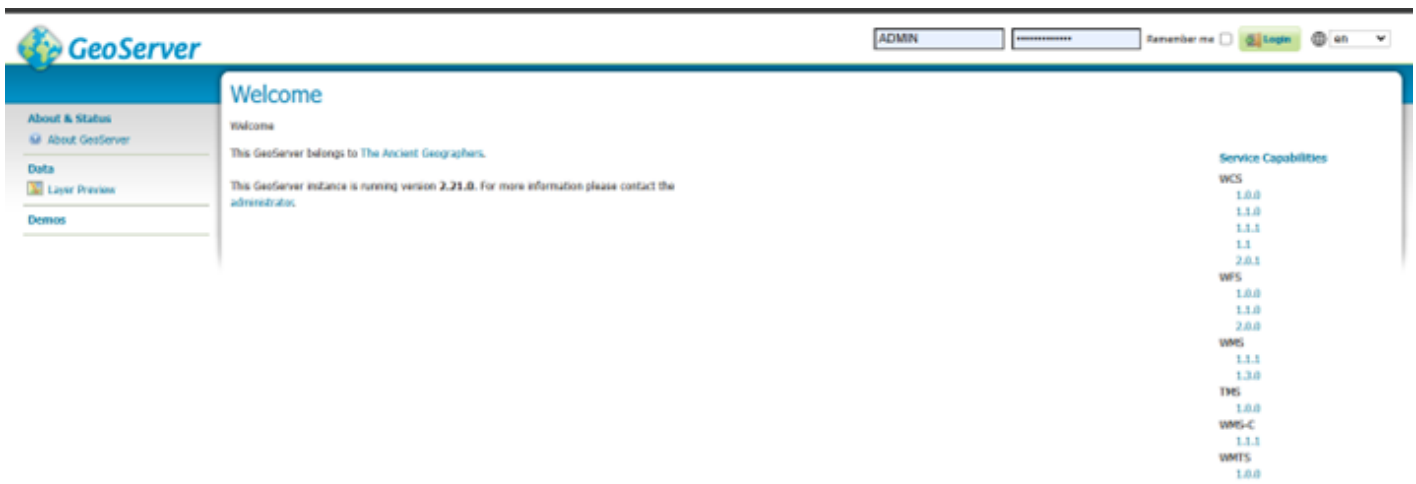
5.1   Add New Workspace

After logging into GeoServer, create a new workspace by navigating to the **Workspace** in the left-side menu and selecting **Add new workspace**.

Add a new workspace, for the workspace add the name of the current database that we are working on. For NameSpace URI add the URI that we are currently working on. In our local deployment Namespace URI may be localhost:8000

5.2 Add New Store

To add a new store in GeoServer, follow the steps mentioned below
• On the left sidebar, select Stores.
• This will open the main Stores page where existing stores are listed.
• On the Stores page, look for the Add New Store link or button.
• Click Add New Store to open the form for entering the details of the new store.

It will take you to new data source page where we are working with POSTGIS so we will select that:

New data source

Choose the type of data source you wish to configure

Vector Data Sources

- Directory of spatial files (shapefiles) - Takes a directory of shapefiles and exposes it as a data store
- GeoPackage - GeoPackage
- PostGIS - PostGIS Database
- PostGIS (JNDI) - PostGIS Database (JNDI)
- Properties - Allows access to Java Property files containing Feature information
- Shapefile - ESRI(tm) Shapefiles (*.shp)
- Web Feature Server (NG) - Provides access to the Features published a Web Feature Service, and the ability to perform transactions on the server (when supported / allowed).

Raster Data Sources

- ArcGrid - ARC/INFO ASCII GRID Coverage Format
- GeoPackage (mosaic) - GeoPackage mosaic plugin
- GeoTIFF - Tagged Image File Format with Geographic information
- ImageMosaic - Image mosaicking plugin
- WorldImage - A raster file accompanied by a spatial data file

Other Data Sources

- WMS - Cascades a remote Web Map Service
- WMTS - Cascades a remote Web Map Tile Service

After this will direct to the page where we will add details about new vector data source, On this page, you'll first encounter the Basic Store Info section.

Basic Store Info

Workspace *

iims_base

Data Source Name *

Description

☑ Enabled

☐ Auto disable on connection failure

- On the Workspace, dropdown select workspace you have created.
- In Data Source Name you need to give the name as given in the GeoServer Documentation in Git Hub Organization.

[Note: While adding store for "layer_info" make sure you tick "Expose Primary Key" option.]
After this on the connection parameters section:

Connection Parameters

host *
```
localhost
```

port *
```
5432
```

database
```
imis_base
```

schema
```
public
```

user *
```
postgres
```

passwd
```
•••
```

Namespace *
http://192.168.1.79:8000

- On database label, put the database name which you are working on.
- On the schema put the schema name you are working on.
- For the user and password, you will put the credentials used for the pgAdmin.

Once finished, click on the apply and then save.

5.3   Add New Layers

To add a new layer in GeoServer, follow the steps below
- On the left sidebar, select Stores.
- This will open the main Layers page where existing Layers are listed.
- On the Layers page, look for the Add New Layer link or button.
- Click Add New Layer to open the form for entering the details of the new Layer.

Layers

Manage the layers being published by GeoServer
⊕ Add a new layer   ⊖ Remove selected layers

| << | < | 1 | 2 | 3 | 4 | 5 | > | >> | Results 1 to 25 (out of 182 items) | | 🔍 Search |

	Type	Title	Name	Store	Enabled	Native SRS
☐	▤	buildings	imis_africa:buildings_layer	imis_africa_building_info	✔	EPSG:4326
☐	▤	citypolys	imis_africa:citypolys_layer	imis_atrica_layer_info	✔	EPSG:4326
☐	•	containments	imis_africa:containments_layer	imis_base_fsm	✔	EPSG:4326

To add the new layer select the table you want to add layer to.

Figure 6 Adding new layer

Once you have chosen the table you can either publish or Configure new SQL view, which you can see in the figure 5.

5.6   Resolving CORS Errors in GeoServer

If you encounter a CORS error in the web interface during deployment, follow these steps to resolve it, Locate the following path
C:\Program Files\GeoServer\webapps\geoserver\WEB-INF\web

Open web.xml as administrator, and uncomment the following section

```
 <!-- Uncomment following filter to enable CORS in Jetty. Do not forget the second config block further down. -->
   <filter>
    <filter-name>cross-origin</filter-name>
    <filter-class>org.eclipse.jetty.servlets.CrossOriginFilter</filter-class>
    <init-param>
     <param-name>chainPreflight</param-name>
     <param-value>false</param-value>
    </init-param>
    <init-param>
     <param-name>allowedOrigins</param-name>
     <param-value>*</param-value>
```

```
    </init-param>
    <init-param>
     <param-name>allowedMethods</param-name>
     <param-value>GET,POST,PUT,DELETE,HEAD,OPTIONS</param-value>
    </init-param>
    <init-param>
     <param-name>allowedHeaders</param-name>
     <param-value>*</param-value>
    </init-param>
   </filter>

<!-- Uncomment following filter to enable CORS    -->
   <filter-mapping>
     <filter-name>cross-origin</filter-name>
     <url-pattern>/*</url-pattern>
   </filter-mapping>
```

5.7   Importing New Workspace, Store, Layer & Style in Geoserver

To streamline the process of importing manually creating the individual workspaces, stores, layers and styles in geoserver, an alternative approach is to import the entire workspace into geoserver directly. The workspace is provided in the GitHub repository.

 Download the zip file and follow the instructions provided below to import the existing workspace into geoserver:

Step 1 :   After the geoserver has been installed, create a new workspace named "sample_workspace" with URL 127.0.0.1.

Step 2 :   Create a store named "sample_datastore" with the following details:

a.   Select the PostGIS (PostGIS Database) option
b.   Set the the Workspace as "sample_worspace"
c.   Set the Data Source Name as "imis_base_building_info"
d.   Set the Host with the IP of the database that has been setup
e.   Set the port that has been used.
f.   Set the database name (base_imis)
g.   Set the Schema as building_info.
h.   Set the database username and password that has been used.
i.   Save the Store

Step 3 :   Open to the file:
 C:\ProgramData\GeoServer\workspaces\sample_workspace\datastore.xml

Step 4 :   Copy the following lines of code:

```
<entry key="passwd">crypt1:dummyCrypt</entry>
```

Step 5 :   Download the provided zip file (Repository: Deployment_Manual, Sub Directory: Geoserver Workspace Setup) and unzip to your local machine.

Step 6 :   Navigate to the datastore.xml of each data store and replace the line of code with the

line of code copied in Step 4.

Step 7 :   Ensure the following lines of code match the values as well:

<entry key="database">{{DATABASE_NAME}}</entry>

<entry key="host">localhost</entry>

<entry key="namespace">">127.0.0.1</entry>

Step 8 :   Once each datastore's passwd key and other values have been updated, copy the entire workspace folder and paste it into the location: C:\ProgramData\GeoServer\workspaces\

Step 9 :   Ensure that the folder name matches the workspace you are importing (base_imis) and that the sub-folder within the folder contains the data stores.

Step 10 :   Restart the tomcat service.

Step 11 :   Check if the workspace, store, layer and style is present in the geoserver and remove the dummy workspace that was created.