

DECIDE IDSS / TECHNICAL DOCUMENTATION

Table of Contents

[Installation of the system on a VM 4](#_Toc454278752)

[Install PostgreSQL/PostGIS on Ubuntu 14.04 LTS 4](#_Toc454278753)

[Install PostGIS 4](#_Toc454278754)

[Install Alfresco 4](#_Toc454278755)

[Database Migration 5](#_Toc454278756)

[Install GeoServer 7](#_Toc454278757)

[VM crontab scripts 10](#_Toc454278758)

[VM administrative tasks 11](#_Toc454278759)

[GeoServer: How to set the style of a layer 12](#_Toc454278760)

[Find a layer 12](#_Toc454278761)

[Maps & Data (category geodata) 12](#_Toc454278762)

[Critical Infrastructures & areas (category infrastructure) 14](#_Toc454278763)

[Edit layer style 14](#_Toc454278764)

[GeoServer: How to import new ShapeFiles/Layers 18](#_Toc454278765)

[GeoServer: CORS configuration 20](#_Toc454278766)

[Alfresco 21](#_Toc454278767)

[Alfresco: Configuration 21](#_Toc454278768)

[Alfresco: Manage Groups/Agencies 26](#_Toc454278769)

[Create group/agency 26](#_Toc454278770)

[Search group/agency 27](#_Toc454278771)

[Browse group/agency 27](#_Toc454278772)

[Edit/Delete group/agency 27](#_Toc454278773)

[Alfresco: Manage Users 27](#_Toc454278774)

[Create user 28](#_Toc454278775)

[Search user 28](#_Toc454278776)

[Alfresco: Manage documents 28](#_Toc454278777)

[Upload a new document 29](#_Toc454278778)

[Edit properties 29](#_Toc454278779)

[Alfresco: Manage Categories & Tags 30](#_Toc454278780)

[Category Manager 30](#_Toc454278781)

[Tag Manager 32](#_Toc454278782)

[Source Code 34](#_Toc454278783)

[Source Code: Change language messages 34](#_Toc454278784)

[Source Code: How to rebuild & publish IDEA project on the VM 35](#_Toc454278785)

# Installation of the system on a VM

## Install PostgreSQL/PostGIS on Ubuntu 14.04 LTS

During this documentation, the operating system of the used VM is Ubuntu 14.04 LTS with the codename **trusty**.

With a clean VM at hand, initially the PostgreSQL has to be installed. Simple instructions are given in the PostgreSQL wiki: <https://wiki.postgresql.org/wiki/Apt>.

We follow the instructions given in the above wiki page and select to run the [shell script](https://anonscm.debian.org/cgit/pkg-postgresql/postgresql-common.git/plain/pgdg/apt.postgresql.org.sh) which automates the process. The resulting Postgres installation is PostgreSQL version **9.5.3**. We need to change the default *postgres* user password so we execute the following commands:

$ sudo -u postgres psql postgres

postgres=# \password postgres

The above command changes the password to **password** for the default user **postgres**.

Then, in order to check if the PostgreSQL is running, we have to execute the following command:

sudo service postgresql status

This service runs by default on port **5432**.

For the purposes of the iDSS project we need to create an empty database with the name **gis**.

## Install PostGIS

Now, that we have successfully installed Postgres we need to enable the PostGIS extension in order to give our database (**gis** database only) geospatial properties. In the following link we can see a list of useful advices listed under the chapter *Enabling PostGIS*: <http://postgis.net/install/>.

Before executing the CREATE EXTENSION command we need to install the postgis scripts by executing the following:

sudo apt-get install postgresql-9.5-postgis-scripts

Then, we execute on pgadmin or in postgres console:

CREATE EXTENSION postgis;

This command is enabling PostGIS (including raster). The other features mentioned in this page can be skipped as they are not currently necessary in the scope of the project.

## Install Alfresco

Now that our DB has been installed we can proceed to the Alfresco Community System installation. Alfresco is the main system utilized for the user and document management and when installing it a tomcat is embedded in it. The following guidelines are pretty straightforward:

<http://docs.alfresco.com/community/tasks/simpleinstall-community-lin.html>

The main tasks are:

* Download the .bin file.
* Select the ***Advanced*** installation type; otherwise another postgres installation will be created inside the alfresco installation which complicates the installation process.
* On the *Select Components* (step 7) choose **NOT** to install *PostgreSQL, Solr1, Web Quick Start*

Next, you have to follow the default steps. You should follow the instructions presented on the above manual.

A very important task during the installation process is to fill in the correct DB connection data, which for our project are:

db.driver:org.postgresql.Driver

db.username=postgres

db.password=password

db.name=gis

db.url=jdbc.postgresql://localhost:5432/gis

Further help on setting up the DB connection can be found here: <http://docs.alfresco.com/5.1/tasks/postgresql-config.html>

At this point it is very important to mention that during the Alfresco installation the database **gis** should be empty (there will be only one postgis table created during the extension enabling).

The alfresco installation files are located in the path /opt/alfresco-community (in the current VM installation).

When the installation is finished, tomcat starts running automatically and we can test the alfresco community site in the following address: <http://160.40.63.119:8080/share/page/>.

We login to alfresco with username/password **admin/admin**.

Right now our database has ~100 tables. These have been created by Alfresco during the initialization process. At this point we need to import the rest of the tables (irrelevant to alfresco) related to the project (Play framework and GeoServer) manually.

### Database Migration

Unfortunately, Postgres does not have any easy tool to export specific tables, but only a total database dump which is reasonable to the fact that there are several indexes (sequences etc.) included in the schema. So, the following steps need to be followed:

1. Export the whole gis.backup from the old system’s database.
2. Import it to the new postgres installation in a **test** database.
3. In the test database, drop all **alfresco related** tables/sequences.
4. Import the **test** database to the **gis** database.

As a result, in the final database we have the new and clean alfresco tables created during the recent installation and the old custom tables created during the development phase.

At this point we need to truncate (with RESTART IDENTITY) the following DB tables when setting up the new system:

\_actions (Then we execute the script parseXLSfile (/var/www/public/scripts) to fill in the table with the default actions)

\_agency\_sharing (After emptying the table we have to insert one record for the Alfresco admin – see Figure 1).



Figure 1. Initial data to be added in \_agency\_sharing table

\_contacts

\_contacts\_alfresco

\_contacts\_groups

\_forecast\_precipitation, \_forecast\_capacitation, \_forecast\_temperature, \_forecast\_wind,

\_infrastructure\_\*\*\* except for table \_infrastructure\_mapping

\_notifications

\_machinery\_layer

\_machinery\_type (Here keep only rows inserted by the agency “ALFRESO\_ADMINISTRATORS”. These are the default machinery types.)

\_measures

\_meteo\_current\_year\_precipitation, \_meteo\_current\_year\_temperature, \_meteo\_current\_year\_wind, \_meteo\_currentmonth, \_meteo\_last2days

\_notifications

\_sms

Now we can continue to work on the new database only.

## Install GeoServer

The next step is to install the GeoServer system under Tomcat. As mentioned in the [GeoServer](http://docs.geoserver.org/latest/en/user/installation/index.html) page the first thing to do is to install Java 8. We do so by running the following commands:

sudo add-apt-repository ppa:webupd8team/java

$ sudo apt-get update

$ sudo apt-get install oracle-java8-installer

We verify the installed java with java –version. In the current installation the java version is **1.8.0\_91**.

After installing Java we are ready to install GeoServer. First, we need to download it through the page: <http://geoserver.org/download/>.

We select a Stable release and we download a Web Archive package (war). The only thing we need to do is copy the war file included in the zip to our tomcat webapps directory.

In our current installation we place the document in the path:

/opt/alfresco-community/tomcat/webapps/geoserver

Now we are ready to configure the GeoServer installation. We can open GeoServer in the following URL: <http://160.40.63.119:8080/geoserver/>

The default credentials are **u:admin** and **p:geoserver**.

Initially, we need to configure the **Stores** through the menu. We need to add three stores called **idss**, **geodata** and **infrastructure** respectively. These will contain the several layers used by the system (IDSS).

We select Add new **Store > PostGIS Database > workspace:sf > Data Source Name:idss** and we set the connection parameters to:

Host: localhost

Port: 5432

Database: gis

User: postgres

Password: password

We repeat the above store creation for the *geodata* and *infrastructure* store.

Then, we need to setup the Service access rules list through the menu **Security > Services**. We add the following rules (Figure 2):

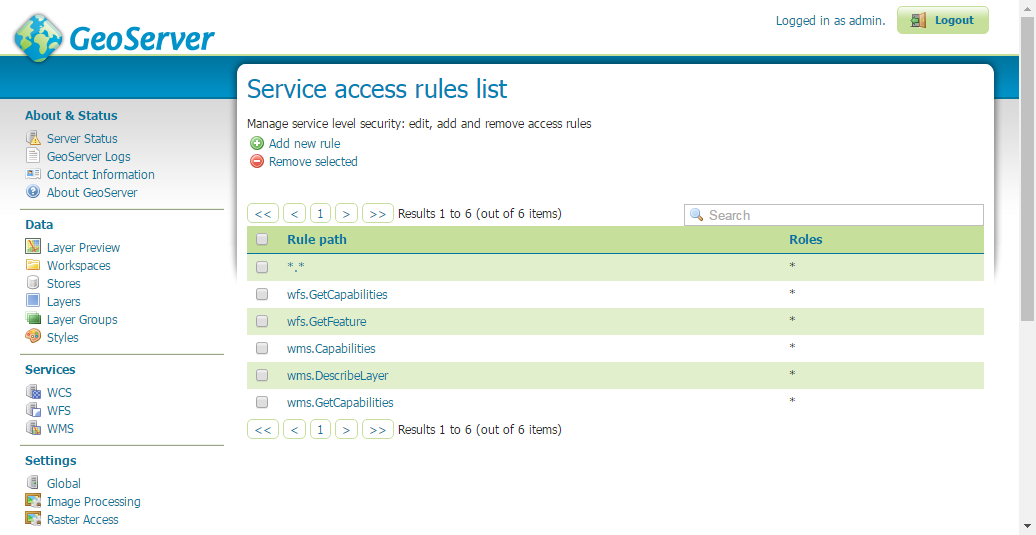


Figure 2. Service access rules

Finally, we are ready to publish the layers which are used by IDSS. In the menu we select **Layers > Add a new resource**.

For each of the 3 created Stores we add the following layers:

**idss**

\_machinery\_layer > Publish > Title: Θέσεις οχημάτων > Compute Bounding Boxes

**geodata**

In the same way for sf:geodata we publish the following 35 layers:

* \_gd\_dasarxeia > Δασαρχεία
* \_gd\_dimoi\_kallikratis > Δήμοι
* \_gd\_oria\_diamerismatwn > Όρια διαμερισμάτων
* \_gd\_oria\_foreon\_diaxeirisis > Φορείς Διαχείρισης Προστατευόμενων Περιοχών
* \_gd\_oria\_pyrosvestikwn\_yphresiwn > Όρια Επιχειρησιακής Ευθύνης Πυροσβεστικών Υπηρεσιών και Κλιμακίων
* \_gd\_perifereiakes\_enothtes > Περιφερειακές Ενότητες
* \_gd\_periphereies > Περιφέρειες Ελλάδας
* \_gd\_apokentrwmenes\_dioikhseis > Αποκεντρωμένες Διοικήσεις
* \_gd\_oikismoi > Οικισμοί
* \_gd\_wildlife\_refuges > Καταφύγια Άγριας Ζωής
* \_gd\_wildlife\_national\_forests > Εθνικοι Δρυμοί
* \_gd\_wildlife\_sensitive\_to\_fire > Περιοχές δασών και δασικών εκτάσεων ευαίσθητων σε πυρκαγιές
* \_gd\_wildfire\_aisthitika\_dash > Αισθητικά Δάση
* \_gd\_hydro\_departments > Υδατικά Διαμερίσματα
* \_gd\_hydro\_network > Υδρογραφικό Δίκτυο
* \_gd\_hydroscope\_rivers > Ποταμοί
* \_gd\_hydro\_rivers > Ποταμοί και Ρέματα
* \_gd\_greece\_lakes > Λίμνες
* \_gd\_drainage\_basins > Λεκάνες Απορροής
* \_gd\_flood\_dangers\_WestMacedonia > Ζώνες Δυνητικά Υψηλού Κινδύνου Πλημμύρας – Υ.Δ. Δυτικής Μακεδονίας
* \_gd\_flood\_dangers\_CentralMacedonia > Ζώνες Δυνητικά Υψηλού Κινδύνου Πλημμύρας – Υ.Δ. Κεντρικής Μακεδονίας
* \_gd\_floods\_important\_east\_macedonia > Σημαντικές Ιστορικές Πλημμύρες Υ.Δ. Ανατολικής Μακεδονίας
* \_gd\_floods\_important\_western\_macedonia > Σημαντικές Ιστορικές Πλημμύρες Υ.Δ. Δυτικής Μακεδονίας
* \_gd\_floods\_important\_central\_macedonia > Σημαντικές Ιστορικές Πλημμύρες Υ.Δ. Κεντρικής Μακεδονίας
* \_gd\_floods\_important\_thrace > Σημαντικές Ιστορικές Πλημμύρες Υ.Δ. Θράκης
* \_gd\_floods\_east\_macedonia > Ιστορικές Πλημμύρες Υ.Δ. Ανατολικής Μακεδονίας
* \_gd\_floods\_western\_macedonia > Ιστορικές Πλημμύρες Υ.Δ. Δυτικής Μακεδονίας
* \_gd\_floods\_central\_macedonia > Ιστορικές Πλημμύρες Υ.Δ. Κεντρικής Μακεδονίας
* \_gd\_floods\_thrace > Ιστορικές Πλημμύρες Υ.Δ. Θράκης
* \_gd\_technological\_accidents\_seveso > Εγκαταστάσεις της Οδηγίας SEVESO
* \_gd\_earthquake\_dangerous\_zones > Zώνες Σεισμικής Επικινδυνότητας
* \_gd\_earthquake\_dangerous\_municipalities > Σεισμική επικινδυνότητα ανά Δημοτική Ενότητα
* \_gd\_natura > Δίκτυο ΝΑΤURA 2000 και προστατευόμενες περιοχές
* \_gd\_florina\_pois > Σημεία Ενδιαφέροντος / Σημαντικές Υποδομές
* \_gd\_railnetwork > Σιδηροδρομικό Δίκτυο

**infrastructure**

And in the same way we publish for sf:infrastructure the following 24 layers:

* \_infrastructure\_refuge > Χώροι καταφυγής
* \_infrastructure\_domino > Eγκαταστάσεις που εκτίθενται σε κίνδυνο φαινομένου domino
* \_infrastructure\_salt > Χώροι αποθήκευσης άλατος
* \_infrastructure\_xada > Χ.Α.Δ.Α.
* \_infrastructure\_xyta > X.Y.T.A.
* \_infrastructure\_waste > Εγκαταστάσεις επεξεργασίας λυμάτων
* \_infrastructure\_seveso > Εγκαταστάσεις υψηλής όχλησης
* \_infrastructure\_building\_earthquake > Κτίρια της διοίκησης που υπόκεινται σε προσεισμικό έλεγχο
* \_infrastructure\_ireland\_crossings > Ιρλανδικές διαβάσεις (Επικίνδυνες διαβάσεις χειμάρρων)
* \_infrastructure\_dike\_damage > Ευάλωτα σημεία αναχωμάτων
* \_infrastructure\_artificial\_dike > Σημεία τεχνητής θραύσης αναχωμάτων
* \_infrastructure\_fire\_hydrant > Πυροσβεστικοί κρουνοί
* \_infrastructure\_water\_tank > Δεξαμενές νερού για πυρόσβεση
* \_infrastructure\_camp > Χώροι καταυλισμού
* \_infrastructure\_frequent\_floods > Περιοχές που συνήθως πλημμυρίζουν
* \_infrastructure\_river\_dikes > Αναχώματα Ποταμών
* \_infrastructure\_suburban\_forests > Περιαστικά δάση στα οποία επιβάλλεται απαγόρευση κυκλοφορίας
* \_infrastructure\_sliding\_areas > Περιοχές που κινδυνεύουν από κατολίσθηση
* \_infrastructure\_barriers > Φράγματα-Ταμιευτήρες
* \_infrastructure\_gate\_barriers > Θυροφράγματα
* \_infrastructure\_fire\_guardhouses > Πυροφυλάκια
* \_infrastructure\_firewalls > Αντιπυρικές ζώνες
* \_infrastructure\_tunnels > Σήραγγες
* \_infrastructure\_aqueducts > Υδραγωγεία

# VM crontab scripts

Cron jobs are necessary in order to run scripts required by the iDSS application.

We need to set up **crontab** in order to allow cron jobs. First we ensure that cron is up and running by executing:

sudo service cron status

In the path **/var/www/public/scripts** we upload all of our scripts (Figure 3).

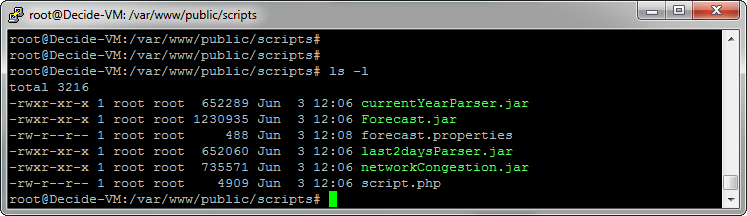


Figure 3. Scripts path

Figure 4 shows the permissions and data that can be found in the directory /var/www/public.

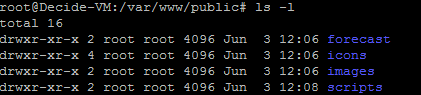


Figure 4. Data and permissions on public directory

In this directory, except for the **scripts** that will be executed by cron we have a **forecast** folder which stores data retrieved from the ftp server of the [Hellenic National Meteorological Service.](http://www.hnms.gr/hnms/greek/index_html) Also, in this directory we put the **icons** that are used from the iDSS application and the uploaded **images**.

Since one of the scripts is in php we have to install php at this point. To do so we run the command apt-get install php5-cli. We also need to install the postgres extension by running sudo apt-get install php5-pgsql.

Now, we are ready to write our crontab which should look like Figure 5 (crontab –l).

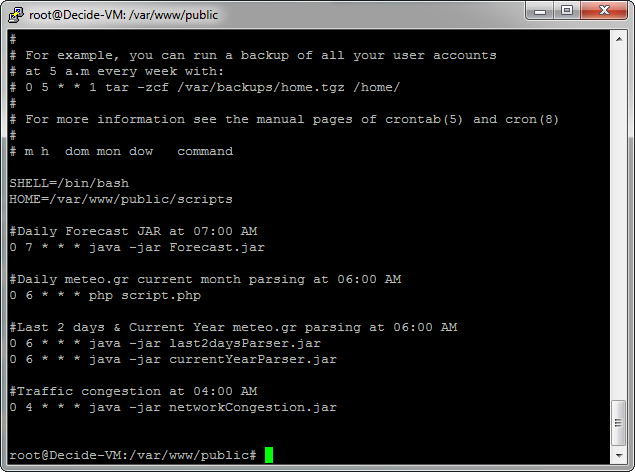


Figure 5. Cron-job configurations

# VM administrative tasks

After installing the above subsystems you may need to perform some administrative tasks. Some basic commands through ssh could be the followings:

* Restart alfresco as well as the embedded tomcat: sudo service alfresco restart.
* Restart postgres: sudo service postgresql restart.

# GeoServer: How to set the style of a layer

## Find a layer

In order to edit a layer and furthermore modify its style (*the style within which it appears on the browser*), we first need to find the name of the layer and then search for the layer in the installation of GeoServer (<http://160.40.63.119:8080/geoserver> login with credentials **u:**admin/**p:**geoserver).

In the scope of the developed platform DECIDE-IDSS, there are several layers imported into the system’s database. These are found in the following 2 categories:

### Maps & Data (category geodata)

The data displayed in this functionality are listed in the DB table **\_geodata\_layers**. The following list shows the table name (thus the layer name) and its title for all depicted layers.

Table 1. GeoServer layers (geodata)

|  |  |  |
| --- | --- | --- |
| 1 | \_gd\_dasarxeia | Δασαρχεία |
| 2 | \_gd\_dimoi\_kallikratis | Δήμοι |
| 3 | \_gd\_oria\_diamerismatwn | Όρια διαμερισμάτων |
| 4 | \_gd\_oria\_foreon\_diaxeirisis | Φορείς Διαχείρισης Προστατευόμενων Περιοχών |
| 5 | \_gd\_oria\_pyrosvestikwn\_yphresiwn | Όρια Επιχειρησιακής Ευθύνης Πυροσβεστικών Υπηρεσιών και Κλιμακίων |
| 6 | \_gd\_perifereiakes\_enothtes | Περιφερειακές Ενότητες |
| 7 | \_gd\_periphereies | Περιφέρειες Ελλάδας |
| 8 | \_gd\_apokentrwmenes\_dioikhseis | Αποκεντρωμένες Διοικήσεις |
| 9 | \_gd\_oikismoi | Οικισμοί |
| 10 | \_gd\_wildlife\_refuges | Καταφύγια Άγριας Ζωής |
| 11 | \_gd\_wildlife\_national\_forests | Εθνικοι Δρυμοί |
| 12 | \_gd\_wildlife\_sensitive\_to\_fire | Περιοχές δασών και δασικών εκτάσεων ευαίσθητων σε πυρκαγιές |
| 13 | \_gd\_wildfire\_aisthitika\_dash | Αισθητικά Δάση |
| 14 | \_gd\_hydro\_departments | Υδατικά Διαμερίσματα |
| 15 | \_gd\_hydro\_network | Υδρογραφικό Δίκτυο |
| 16 | \_gd\_hydroscope\_rivers | Ποταμοί |
| 17 | \_gd\_hydro\_rivers | Ποταμοί και Ρέματα |
| 18 | \_gd\_greece\_lakes | Λίμνες |
| 19 | \_gd\_drainage\_basins | Λεκάνες Απορροής |
| 21 | \_gd\_flood\_dangers\_WestMacedonia | Ζώνες Δυνητικά Υψηλού Κινδύνου Πλημμύρας – Υ.Δ. Δυτικής Μακεδονίας |
| 22 | \_gd\_flood\_dangers\_CentralMacedonia | Ζώνες Δυνητικά Υψηλού Κινδύνου Πλημμύρας – Υ.Δ. Κεντρικής Μακεδονίας |
| 23 | \_gd\_floods\_important\_east\_macedonia | Σημαντικές Ιστορικές Πλημμύρες Υ.Δ. Ανατολικής Μακεδονίας |
| 24 | \_gd\_floods\_important\_western\_macedonia | Σημαντικές Ιστορικές Πλημμύρες Υ.Δ. Δυτικής Μακεδονίας |
| 25 | \_gd\_floods\_important\_central\_macedonia | Σημαντικές Ιστορικές Πλημμύρες Υ.Δ. Κεντρικής Μακεδονίας |
| 26 | \_gd\_floods\_important\_thrace | Σημαντικές Ιστορικές Πλημμύρες Υ.Δ. Θράκης |
| 27 | \_gd\_floods\_east\_macedonia | Ιστορικές Πλημμύρες Υ.Δ. Ανατολικής Μακεδονίας |
| 28 | \_gd\_floods\_western\_macedonia | Ιστορικές Πλημμύρες Υ.Δ. Δυτικής Μακεδονίας |
| 29 | \_gd\_floods\_central\_macedonia | Ιστορικές Πλημμύρες Υ.Δ. Κεντρικής Μακεδονίας |
| 30 | \_gd\_floods\_thrace | Ιστορικές Πλημμύρες Υ.Δ. Θράκης |
| 31 | \_gd\_technological\_accidents\_seveso | Εγκαταστάσεις της Οδηγίας SEVESO |
| 32 | \_gd\_earthquake\_dangerous\_zones | Zώνες Σεισμικής Επικινδυνότητας |
| 33 | \_gd\_earthquake\_dangerous\_municipalities | Σεισμική επικινδυνότητα ανά Δημοτική Ενότητα |
| 34 | \_gd\_natura | Δίκτυο ΝΑΤURA 2000 και προστατευόμενες περιοχές |
| 35 | \_gd\_florina\_pois | Σημεία Ενδιαφέροντος / Σημαντικές Υποδομές |
| 36 | \_gd\_railnetwork | Σιδηροδρομικό Δίκτυο |
| 37 | \_infrastructure\_refuge | Χώροι καταφυγής |
| 38 | \_infrastructure\_camp | Χώροι καταυλισμού |
| 39 | \_infrastructure\_building\_earthquake | Κτίρια της διοίκησης που υπόκεινται σε προσεισμικό έλεγχο |
| 40 | \_infrastructure\_domino | Eγκαταστάσεις που εκτίθενται σε κίνδυνο φαινομένου domino |
| 41 | \_infrastructure\_salt | Χώροι αποθήκευσης άλατος |
| 42 | \_infrastructure\_xada | Χ.Α.Δ.Α. |
| 43 | \_infrastructure\_xyta | X.Y.T.A. |
| 44 | \_infrastructure\_waste | Εγκαταστάσεις επεξεργασίας λυμάτων |
| 45 | \_infrastructure\_seveso | Εγκαταστάσεις υψηλής όχλησης |
| 46 | \_infrastructure\_aqueducts | Υδραγωγεία |
| 47 | \_infrastructure\_ireland\_crossings | Ιρλανδικές διαβάσεις |
| 48 | \_infrastructure\_dike\_damage | Ευάλωτα σημεία αναχωμάτων |
| 49 | \_infrastructure\_artificial\_dike | Σημεία τεχνητής θραύσης αναχωμάτων |
| 50 | \_infrastructure\_fire\_hydrant | Πυροσβεστικοί κρουνοί |
| 51 | \_infrastructure\_water\_tank | Δεξαμενές νερού για πυρόσβεση |
| 52 | \_infrastructure\_frequent\_floods | Περιοχές που συνήθως πλημμυρίζουν |
| 53 | \_infrastructure\_river\_dikes | Αναχώματα Ποταμών |
| 54 | \_infrastructure\_suburban\_forests | Περιαστικά δάση στα οποία επιβάλλεται απαγόρευση κυκλοφορίας |
| 55 | \_infrastructure\_sliding\_areas | Περιοχές που κινδυνεύουν από κατολίσθηση |
| 56 | \_infrastructure\_barriers | Φράγματα-Ταμιευτήρες |
| 57 | \_infrastructure\_gate\_barriers | Θυροφράγματα |
| 58 | \_infrastructure\_fire\_guardhouses | Πυροφυλάκια |
| 59 | \_infrastructure\_firewalls | Αντιπυρικές ζώνες |
| 60 | \_infrastructure\_tunnels | Σήραγγες |

The layers 1 – 36 are layers imported into the system displaying various geospatial data.

The layers 37 – 60 are layers created for the purposes of the system and contain data that can be edited by the users of the system.

In sort, the layers of the format **\_gd\_xxx** are layers used for viewing only in the functionality *Maps & Data*, and the layers of the format **\_infrastructure\_xxx** are layers created for viewing in *Maps & Data* and editing in the menu *Critical Infrastructures & areas.*

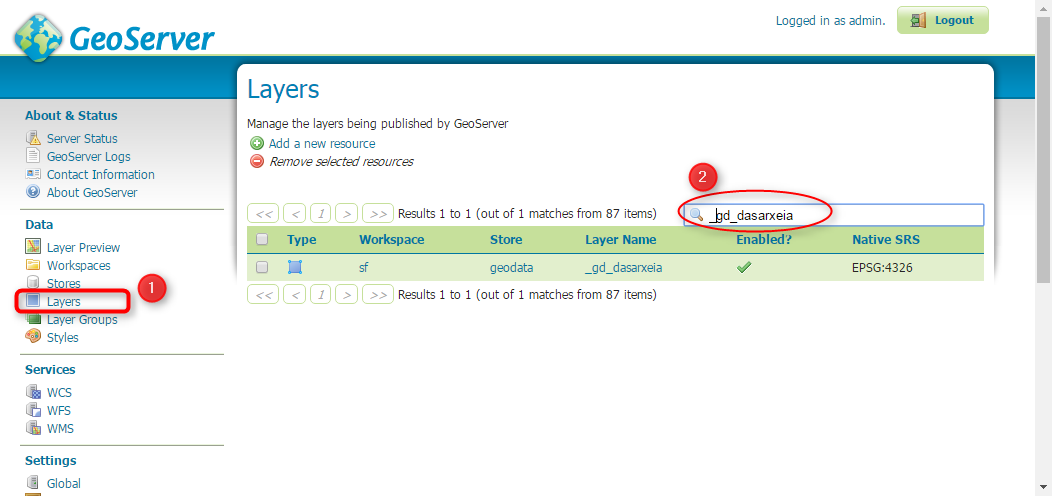


Figure 6. Geoserver - search for layer

In order to edit a layer in GeoServer we need to find its name from the above mapping and search for it in the menu: **GeoServer > Data > Layers > Search area enter layer name**.

The result in the table displayed in Figure 6, is a single row. In the example given in this image, we searched for the layer title «Δασαρχεία» which according to Table 1 corresponds to the layer name «\_gd\_dasarxeia». The result is a layer of the type *Polygon*.

### Critical Infrastructures & areas (category infrastructure)

Likewise to the category geodata mentioned above, the category infrastructure concerns the tables listed in Table 1 from index 37 to index 60. These are layers used for viewing in the iDSS menu *Maps & Data* and for editing in the menu *Critical Infrastructures & areas*.

## Edit layer style

A GeoServer layer comes by default with specific styles provided by the GeoServer installation. Each geometry type (Polygon, Point, and Polyline) has a default style which can be edited through GeoServer (Figure 7).

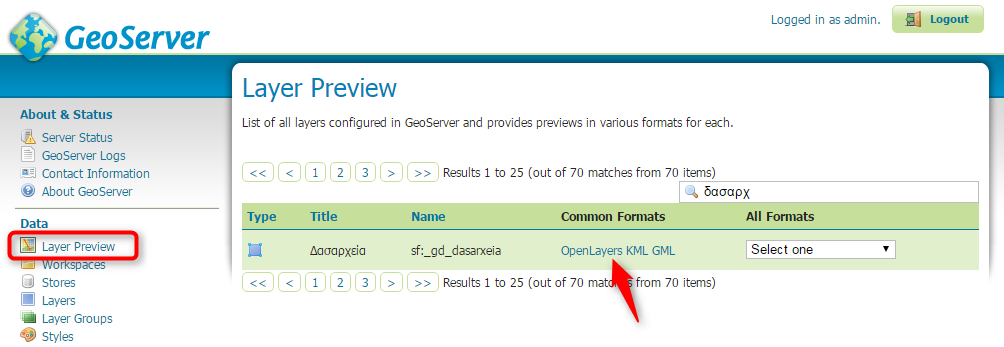


Figure 7. Geoserver – Layer Preview

For example, for the layer «\_gd\_dasarxeia» we can see in the *Layer Preview* menu of GeoServer that the polygons have a gray fill (Figure 8), which can be also seen via the Layers menu for the selected layer (Figure 9).

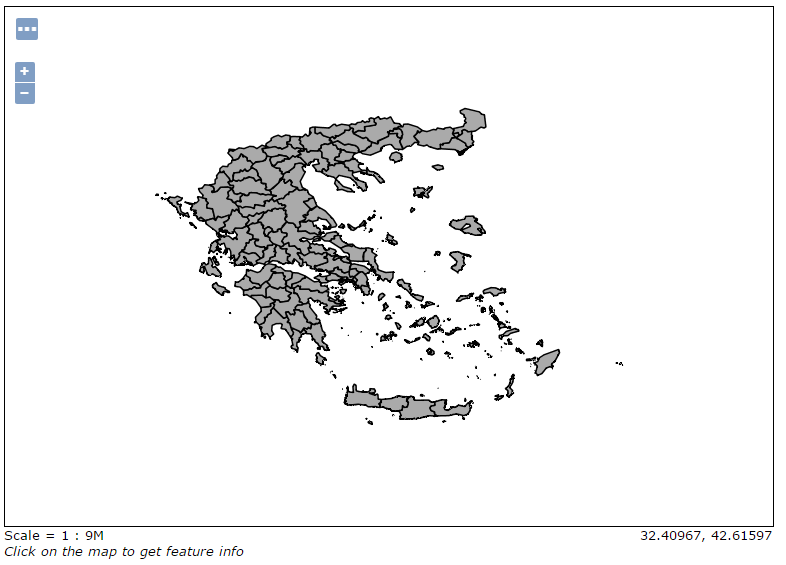


Figure 8. GeoServer – Default polygon style

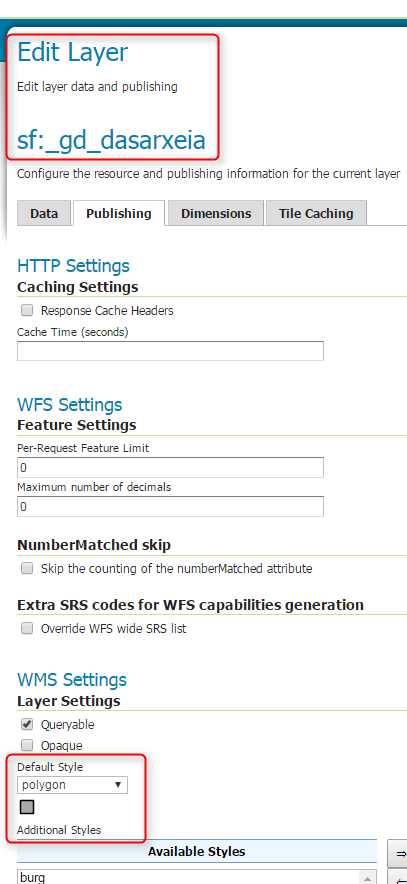


Figure 9. GeoServer – Default polygon style through Layers menu

In order to set another style for the current layer you should navigate to the *Styles* menu and *Add a new style* (Figure 10).



Figure 10. GeoServer - Styles

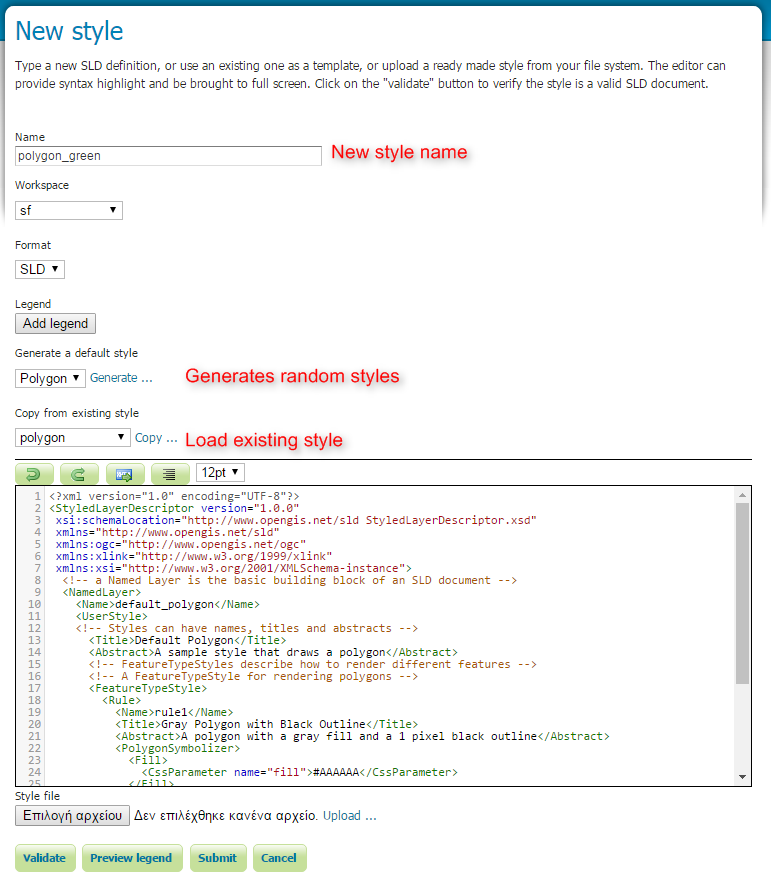


Figure 11. GeoServer – add new style

As displayed in the Figure 11, we can define a new style either from a random style generation mechanism or from scratch. In the editor we can load a style from an existing one and change the css parameters (hex colors etc.).

From the *Layers (Publishing option)* menu, we can set the new just added style (Figure 12). Figure 13 shows the final outcome after modifying the layer’s color.

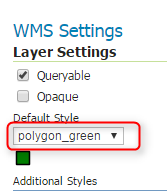


Figure 12. GeoServer – new style



Figure 13. GeoServer – new style applied

# GeoServer: How to import new ShapeFiles/Layers

For the needs of the iDSS system, we have imported several ShapeFiles with the assistance of the [QGIS](http://www.qgis.org/en/site/) desktop program.

After setting up the connection to our PostgreSQL database, we chose to import a new Layer through the **DB Manager** (Figure 14)**.**

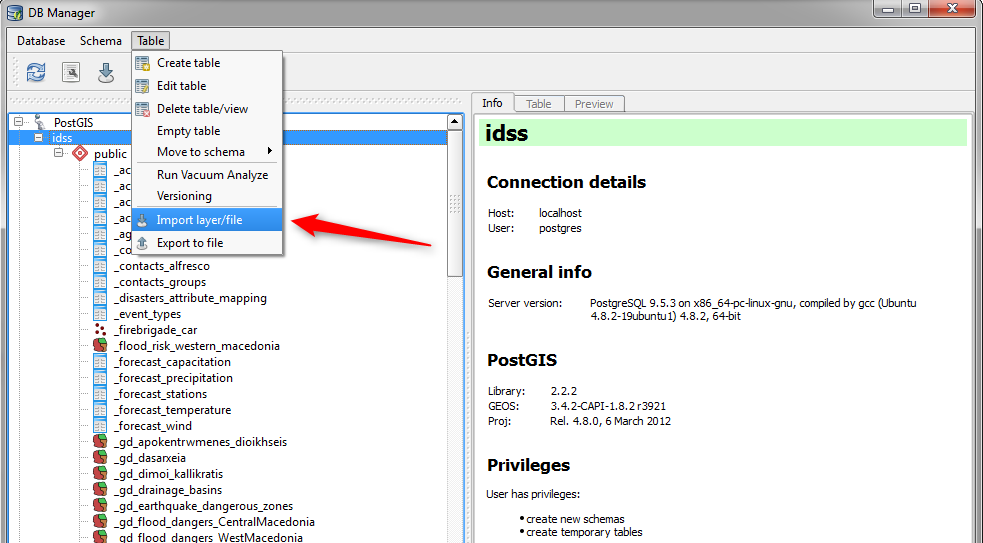


Figure 14. QGIS – DB Manager

It is important to set the proper SRID when importing a ShapeFile. The file will be eventually imported in the defined table (Figure 15).

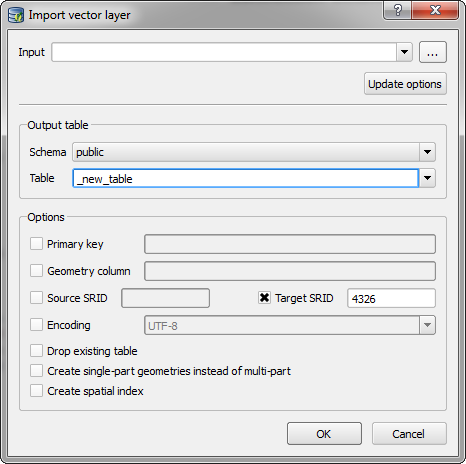


Figure 15. QGIS - import vector layer

For the grouped display of layers in the application we have created the DB table **\_geodata\_layers** (Figure 16)**.**

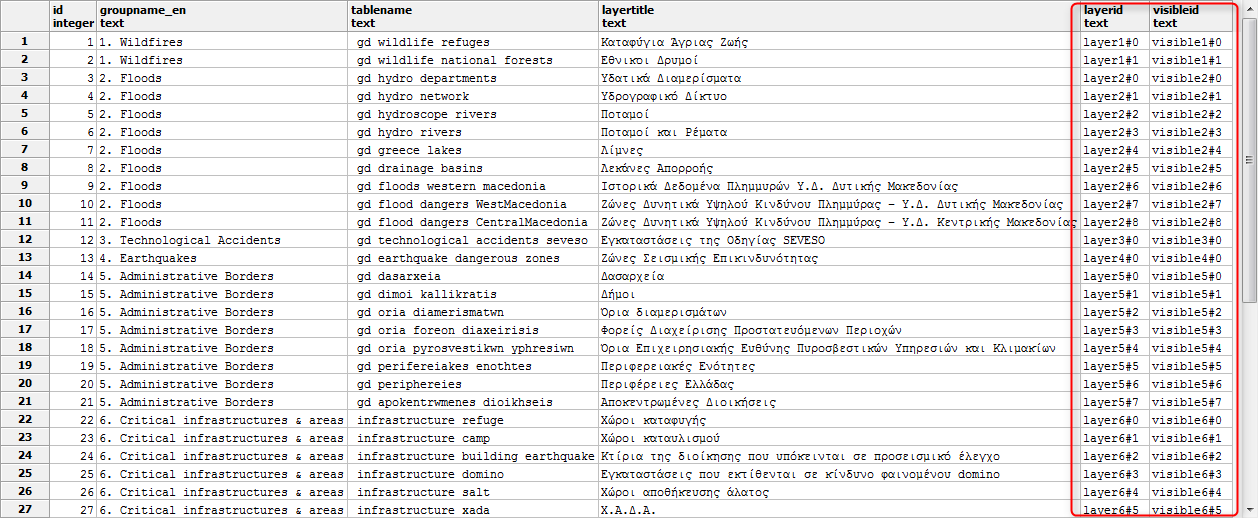
****

Figure 16. DB table \_geodata\_layers

The above tree grouping is connected to the layer tree (Figure 17) in the menu *Maps & Data* of IDSS.

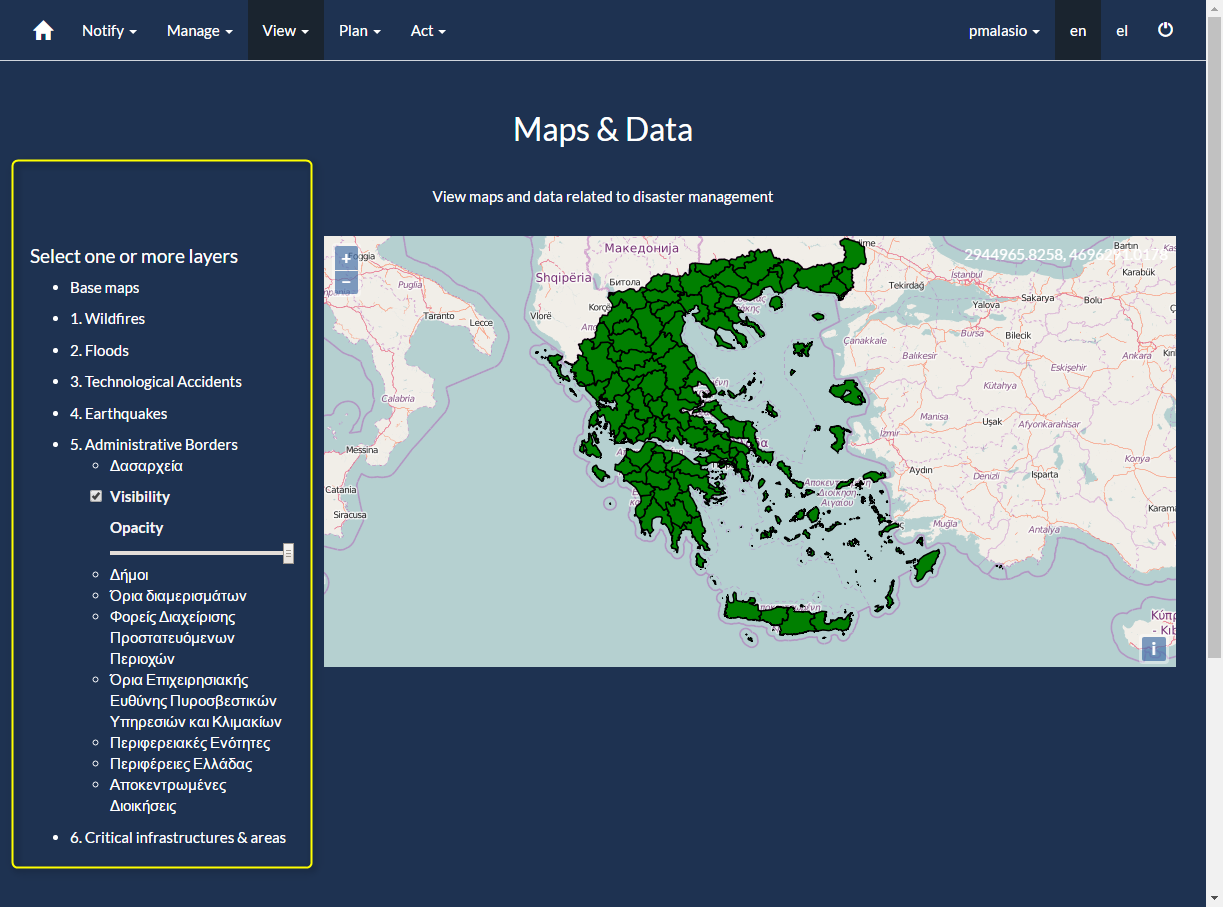


Figure 17. IDSS - Layer tree connected to DB table \_geodata\_layers

It is very important to note that the table columns **layerid, visibleid** should be of the appropriate format. For example, for the group of layers Admininstrative Borders the layerid column should be of the form **layer5#x**.Also, the column **id** defines the order of the layers in the tree.

# GeoServer: CORS configuration

The CORS settings (Cross-origin resource sharing) are essential for the iDSS application, as it allows the system to post AJAX XML data to GeoServer. The following settings are necessary.

In the apache directory **/etc/apache2/sites-available/** we add to the file **000-default.conf** the following header:

Header set Access-Control-Allow-Origin "\*"

Then we execute a2enmod headers and then sudo service apache2 restart.

After setting up the apache CORS header we need to modify the GeoServer config file **web.xml** in the path **/opt/alfresco-community/tomcat/webapps/geoserver/WEB-INF/**. In this xml we need to add the next filter.

<filter>

<filter-name>CorsFilter</filter-name>

<filter-class>org.apache.catalina.filters.CorsFilter</filter-class>

</filter>

<filter-mapping>

<filter-name>CorsFilter</filter-name>

<url-pattern>/\*</url-pattern>

</filter-mapping>

# Alfresco

To proceed with any action throughout the Alfresco system you need to login to <http://160.40.63.119:8080/share/page> with credentials **u:**admin/**p:**admin.

**iDSS alfresco site**

From the top menu navigate to the **Sites** option and select **Create Site**. Complete the field **Name**: iDSS and specify the site **Visibility** as: Public (Figure 18).

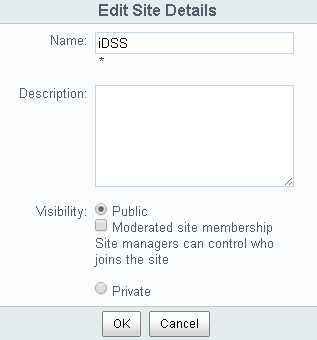


Figure 18. Create new site

## Alfresco: Configuration

When first entering the Alfresco platform we need to make some configurations in order to prepare it according to iDSS specifications.

First from the menu we need to create a site: **Sites > Create Site > idss > public**.

Then, we need to create some rules that apply on the **Document Library**.

According to iDSS each Agency that registers in the system should have its own folder in the document library where it will upload the agency documents. Also, there should be a central folder which would be managed by the Alfresco Administrator.

First, we create in Alfresco a folder called **CommonDocuments** (Figure 19)**.**

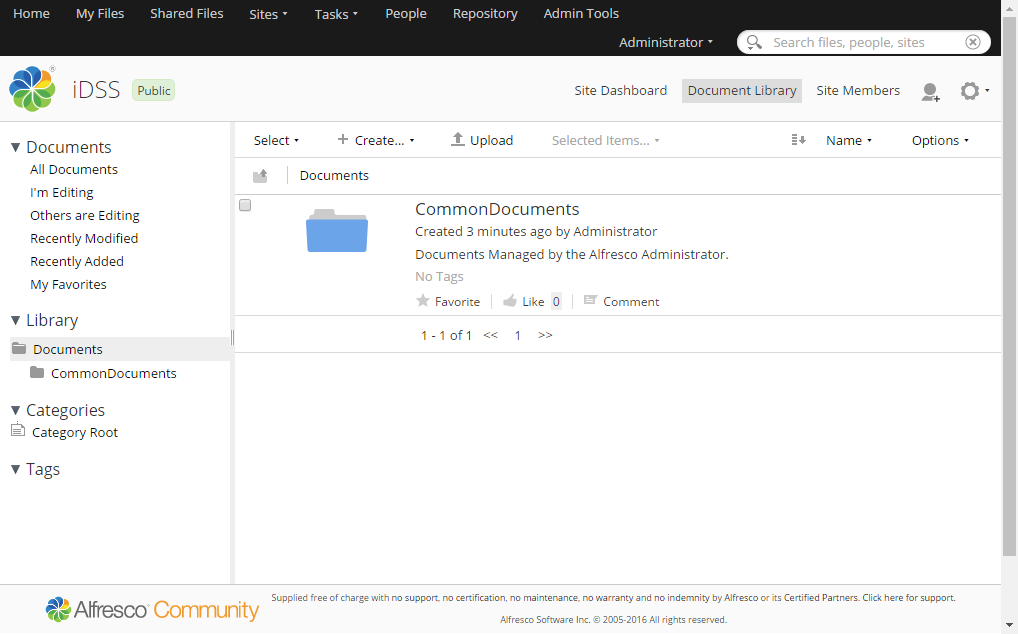


Figure 19. Create CommonDocuments directory

For the folder **CommonDocuments**, we set the permissions for the group “**EVERYONE**” (Figure 20).

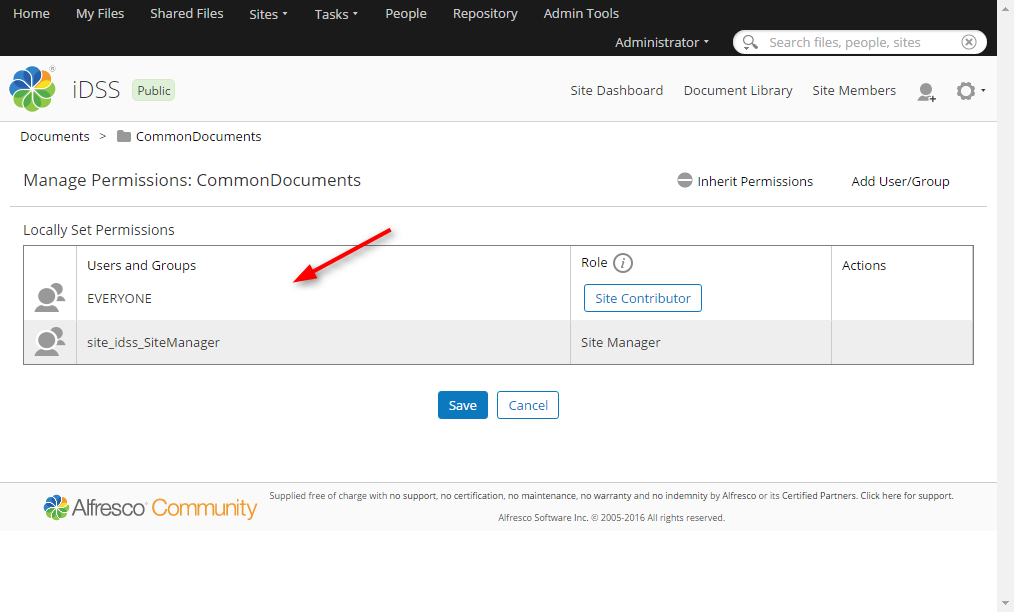


Figure 20. CommonDocuments permissions

In this Library there will be created folders (with trimmed spaces) when a new agency is created. For that to happen we need to create a rule.

We create a javascript file named Persmissions.js with the following content:

var folder = document;

folder.setInheritsPermissions(false);

folder.setPermission("SiteManager" , "GROUP\_"+folder.name);

Then, we upload the .js file to **Repository > Data Dictionary > Scripts** and in the **documentLibrary** folder (Repository > Sites > idss) we apply the following rule (Manage rules option):

- When: Items are created or enter this folder

- If all criteria are met: Is of type (or sub type) 'Folder'

- Perform Action: Execute 'Permission.js' script

- Run in background

- Rule applied to subfolders

Another rule that has to be created is the **Dublin core aspect**. In order to have additional property fields for every document the Dublin core aspect will be added to all documents.

Select the documentLibrary folder (Repository> Sites> idss) and apply the following rule (Manage rules option)

- When: Items are created or enter this folder

- If all criteria are met: Is of type (or sub type) 'Content'

- Perform Action: Add 'Dublin Core' aspect

- Run in background

- Rule applied to subfolders

After the definition of the above rule all documents will have the following properties of the Dublin core aspect.

Publisher

Contributor

Type

Identifier

Source

Coverage

Rights

Subject

Finally a **Rendition rule** needs to be created in order to allow the creation of a cmis rendition on file uploading.

We create a javascript file named Rendition.js with the following content:

var doc;

if (typeof document === 'undefined') { // in Javascript Console

doc = space.children[0]; //0 = first file/subfolder in the 'space' defined above

} else {

doc = document; // in Folder Rule

}

logger.log(doc.name);

// Create a small thumbnail if the file is a JPEG, if the thumbail doesn't exist already.

//if (doc.mimetype=='image/jpeg') {

// Does a doclib rendition (small thumbnail) exist for the file?

var thumbnail = doc.getThumbnail('doclib');

logger.log(thumbnail);

if (!thumbnail) {

var async=true;

// do not run asynchronously, else MANY .convert.bin processes are created and crash tomcat

doc.createThumbnail("doclib", !async);

}

//}

Then, we upload the .js file to **Repository > Data Dictionary > Scripts** and in the **documentLibrary** folder (Repository > Sites > idss) we apply the following rule (Manage rules option):

- When: Items are created or enter this folder

- If all criteria are met: All Items

- Perform Action: Execute Rendition.js' script

- Run in background

- Rule applied to subfolders

After the above steps, there should be 3 rules created (Figure 21).

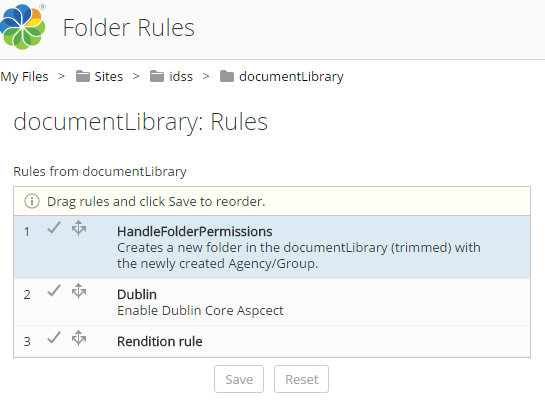


Figure 21. Document library rules

At this point we need to add some **aspects** to the documentLibrary (Figure 22).

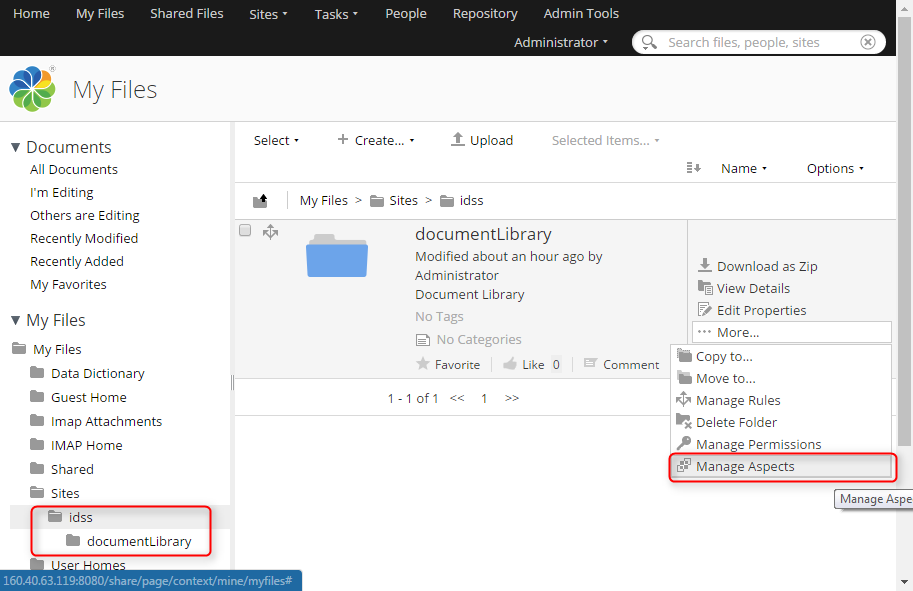


Figure 22. Manage documentLibrary Aspects

We add the following aspects to our document library (Figure 23):

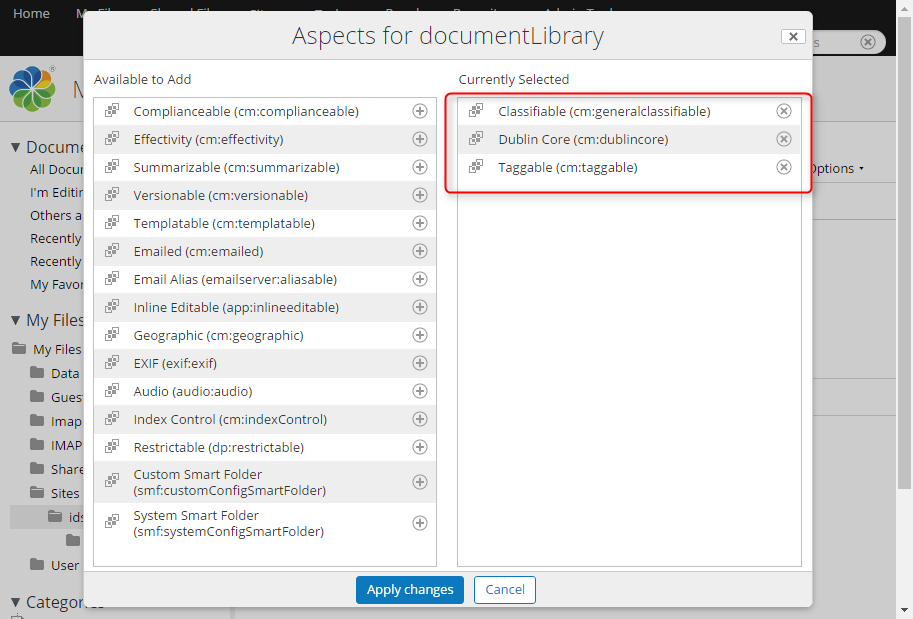


Figure 23. Added CMIS Aspects

Then, we need to create 3 basic groups managed by iDSS. From the menu **Admin Tools> Groups> Browse> (+)** we create the groups **idss\_admin, idss\_edit, idss\_view** which are necessary for the proper functioning of the system (Figure 24).

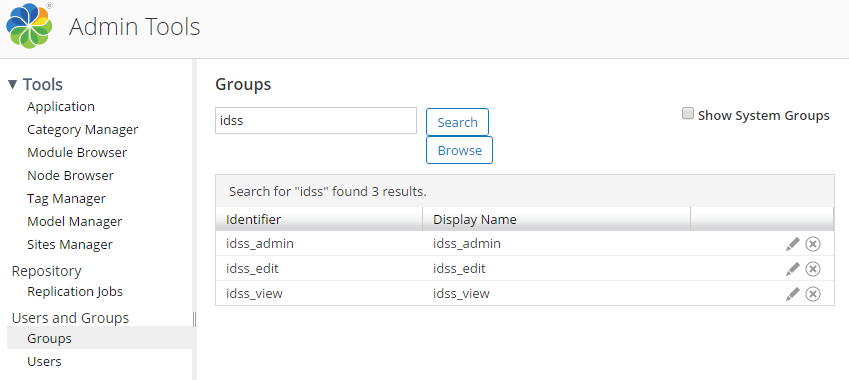


Figure 24. Necessary initial groups

Finally, we need to add the Categories (see Category Manager Section) and Tags (see Tag Manager Section) necessary for the Documents function of IDSS.

## Alfresco: Manage Groups/Agencies

For actions that should be executed via the Alfresco interface, initially you have to navigate at the **Admin Tools** option from the top menu.

### Create group/agency

If you want to create a new group via the alfresco interface, first you have to select the **Groups** option under the **Users and Groups** menu. In order to create a top level group, select the **Browse** option and then click the **plus** button (Figure 25).

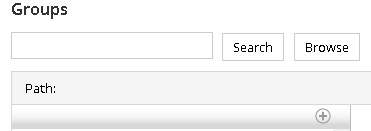


Figure 25. Create new group

Figure 26 shows the fields to be completed for creating a new group. The *Identifier* is the name that is used from the system to identify the group (you should not leave spaces between words), while the Display Name is the group name that appears to members of this group. Having completed the required fields, you should click the **Create Group** option.

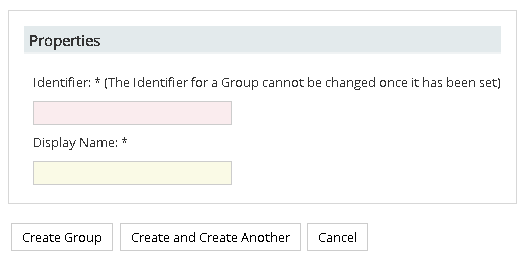


Figure 26. Fields to be completed for creating new group

### Search group/agency

In the search box appeared in Figure 25 insert the Identifier (full or partially) of the desired group. Then click **Search**.

### Browse group/agency

Click the **Browse** option (see Figure 25) and then a list with all the existing groups will appear to you. In order to also see the system groups, you have to select the **Show System Groups** checkbox and then click **Browse**.

### Edit/Delete group/agency

Having spotted the desired group (either via searching or browsing options), you should hover over it in order the Edit and Delete options to be available to you (Figure 27).



Figure 27. Edit/Delete group/agency

With the **Edit Group** option, you can modify the group’s Display Name, while with the **Delete Group** button you can remove the group/agency.

In case you delete a group/agency it is important to delete the folder that is related to such group. So, you have to navigate to **Sites -> iDSS** and select the **Document Library** option. Then hover over the folder to be deleted and click **More -> Delete Folder**.

Finally, the group/agency to be deleted may be associated with sharing permissions (i.e. its content visibility to other groups/agencies). The \_agency\_sharing table in the **gis** database is responsible for maintaining such information. Next, we overview the scheme of such table.

|  |
| --- |
| **id integer NOT NULL DEFAULT nextval('\_agency\_sharing\_id\_seq'::regclass)**  **agency text**  **share boolean**  **insert\_date timestamp without time zone**  **agency\_displayname text** |

The query to be executed for removing the necessary record from the \_agency\_sharing table is the following:

**DELETE FROM \_agency\_sharing WHERE agency=’Identifier-of-the-group’**

## Alfresco: Manage Users

Under the **Users and Group** menu select the **Users** option. From the new screen (Figure 28) you can either search for an existing user or add a new one.

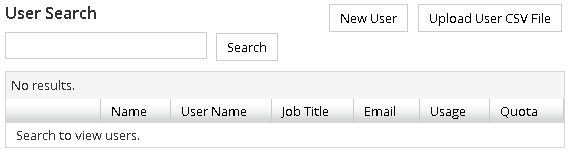


Figure 28. User menu

### Create user

From the **New User** option (Figure 28) you can create a new user. You have to fill in some general information about the user and the groups that he/she belongs to.

### Search user

Having successfully searched for an existing user, then some general information is presented to you (Figure 29).

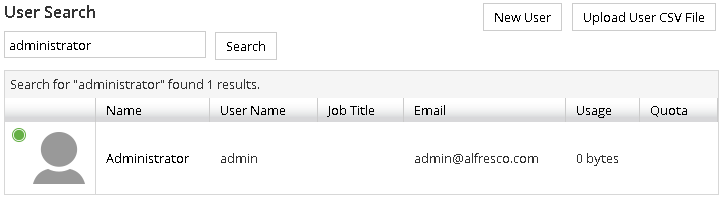


Figure 29. General information about an existing user

If you click on the user’s name, then you will be redirected to a new screen, where more details will be presented to you. From there you can Edit or Delete the selected user.

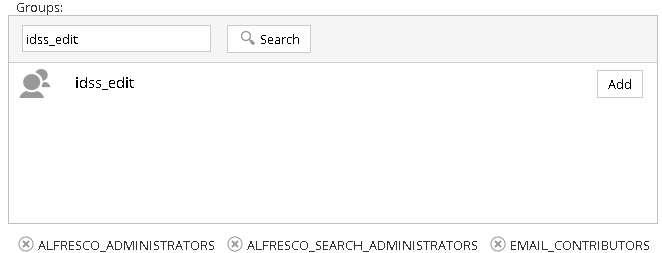


Figure 30. Edit user groups

Apart from a user’s general information (i.e. first name, last name, email, photo, password) you can also modify the groups that he/she belongs to (Figure 30). First you have to search for the group for which you are interested in (e.g. idss\_edit) and then you have to click on the **Add** button in order to complete the action. On the lower level you can overview/delete the group(s) that a user already belongs to.

## Alfresco: Manage documents

Having logged in to the Alfresco system first you need to select the **iDSS site** option(Figure 31).



Figure 31. Select iDSS site

Then, from the right upper corner of the displayed page you have to select the **Document Library** option. The next screen shows all the available agencies of the system, as well as the *CommonDocuments* folder which contains the common documents to all agencies.

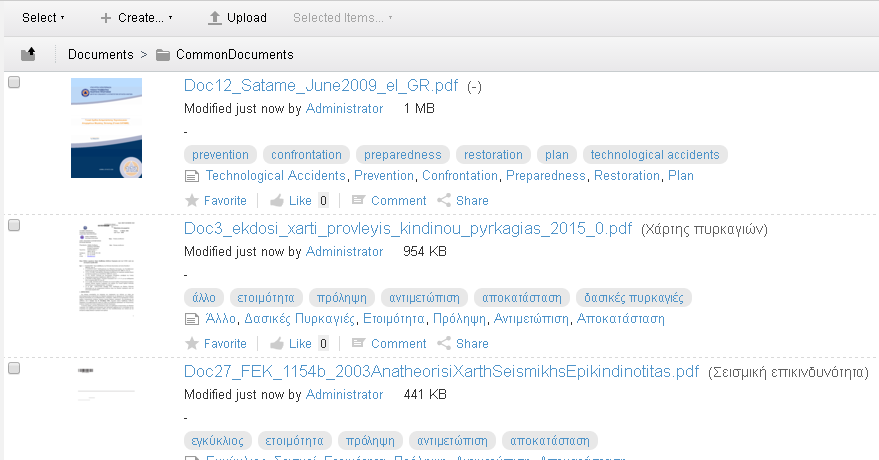


Figure 32. Indicative list of available documents in the CommonDocuments folder

Selecting a specific folder, you can view/edit/delete the contained documents. For instance, if we select the CommonDocuments folder (Figure 32) we can overview all the currently existing documents.

### Upload a new document

As it is presented in Figure 32, you can upload a new document by selecting the relevant option (i.e. Upload), where a popup appears for selecting the new file to be uploaded (Figure 33).

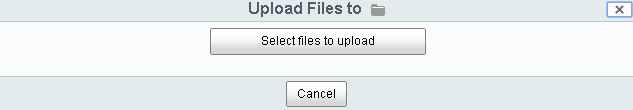


Figure 33. Select files to be uploaded

### Edit properties

Hovering upon a document a set of options is presented from where you can for instance Download, View on browser, Delete or Edit the document’s properties (Figure 34).

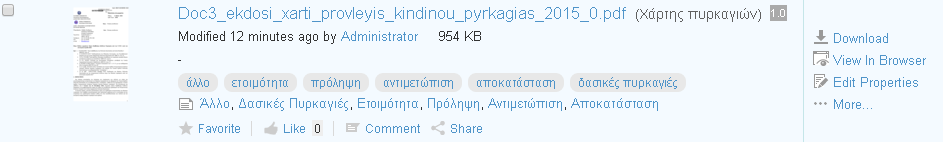


Figure 34. Available options after hovering on a document

When you select the **Edit Properties** option, a popup is presented (Figure 35). From here you can edit all the available properties, namely Name, Title, Description, Tags, and Categories.

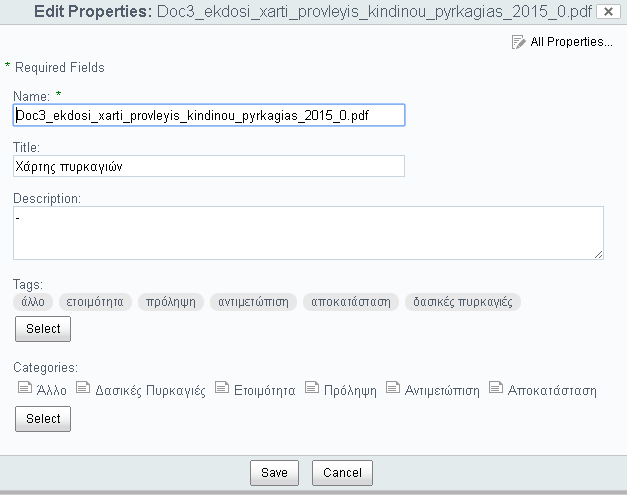


Figure 35. Edit a document's properties

Finally, if you select the **All Properties** option, you will be redirected to a new screen where additional properties will be presented (Figure 36).

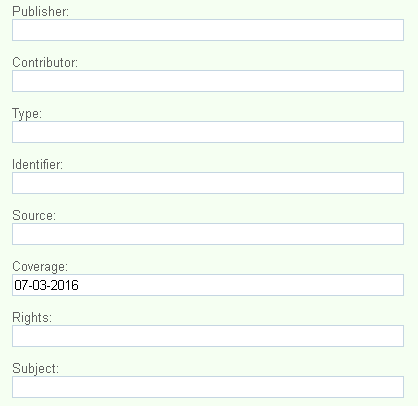


Figure 36. Additional available properties

## Alfresco: Manage Categories & Tags

First select the **Admin Tools** option from the top menu. Then under the **Tools** menu you can select either the **Category Manager** or the **Tag Manager** options.

### Category Manager

In the Category Manager menu (Figure 37) you can see the hierarchy path of the categories inserted in the system. Click on a category in order to overview the nested sub-categories (if any). From here you can modify the categories that a user can add for characterizing a document.

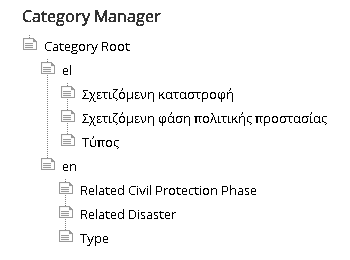


Figure 37. Category Manager overview

Figure 38 and Figure 39 overview all the categories and sub-categories that should be added into the system.

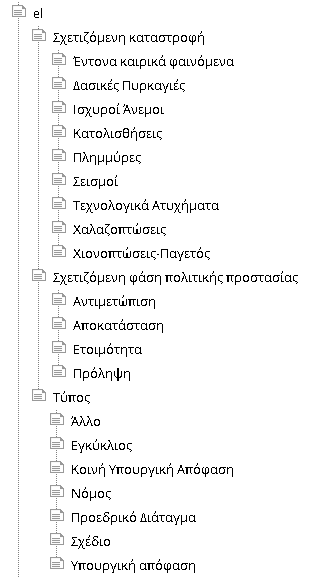


Figure 38. Greek categories

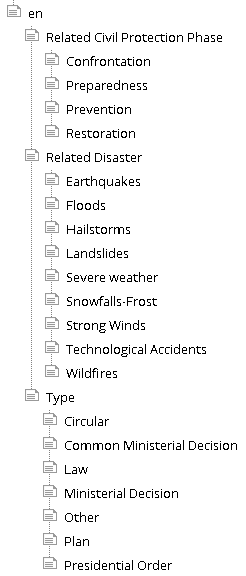


Figure 39. English categories

If you hover over a category (Figure 38) you can Edit (i.e. rename category) and Delete a category, or add a new one.



Figure 40. Modify a category

### Tag Manager

In the Tag Manager menu (Figure 41) you can overview all the existing tags, while also you can search for a specific tag.

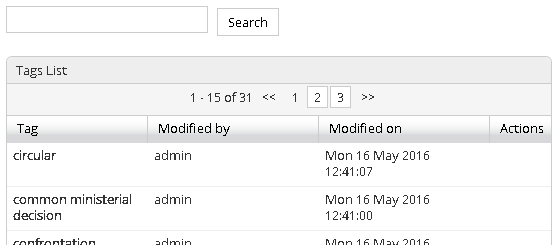


Figure 41. Tag Manager overview

If you hover over a tag, the Edit (i.e. rename tag) and Delete options will be presented to you (Figure 42).



Figure 42. Edit/Delete tag

Table 2 presents the tag list that should be added in the system when a new site is created.

Table 2. Tags to be added in the system

|  |  |
| --- | --- |
| **English** | **Greek** |
| confrontation | αντιμετώπιση |
| preparedness | ετοιμότητα |
| prevention | πρόληψη |
| restoration | αποκατάσταση |
| earthquakes | σεισμοί |
| floods | πλημμύρες |
| hailstorms | χαλαζοπτώσεις |
| landslides | κατολισθήσεις |
| severe weather | έντονα καιρικά φαινόμενα |
| snowfalls-frost | χιονοπτώσεις-παγετός |
| strong winds | ισχυροί άνεμοι |
| technological accidents | τεχνολογικά ατυχήματα |
| wildfires | δασικές πυρκαγιές |
| circular | εγκύκλιος |
| common ministerial decision | κοινή υπουργική απόφαση |
| law | νόμος |
| ministerial decision | υπουργική απόφαση |
| other | άλλο |
| plan | σχέδιο |
| presidential order | προεδρικό διάταγμα |

# Source Code

The iDSS system has been implemented with the Java [Play Framework](https://www.playframework.com/). Also for the development, the Community [IntelliJ IDEA](https://www.jetbrains.com/idea/) IDE has been used.

The main project properties are tracked in two files of the source code (Figure 43), the **application.conf** and **messages** files. The first one contains basic settings for the application languages and the Database configuration. The second includes several parameters gathered in this file and used by other parts of the project. These two files need to be carefully customized if any (server, db, etc.) change occurs in order to assure the proper functioning of the system.

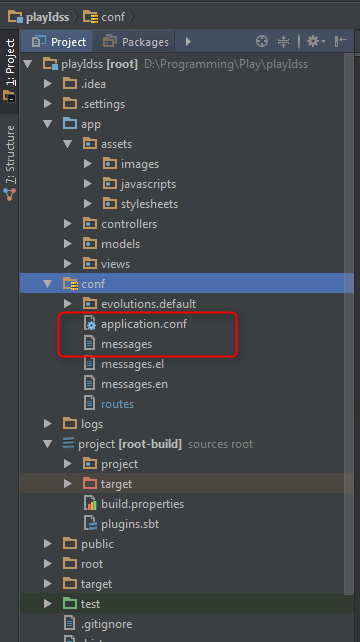


Figure 43. IDEA project configuration

## Source Code: Change language messages

In order to modify the titles and messages that appear in the DECIDE-IDSS platform you need to edit two separate files from the source code of the project, namely the ***messages.el*** (Figure 44) and ***messages.en*** (Figure 45) files (located under the **conf** directory of the project), for editing the Greek and English texts, respectively.

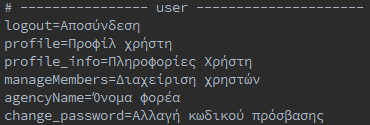


Figure 44. Indicative example of messages.el file

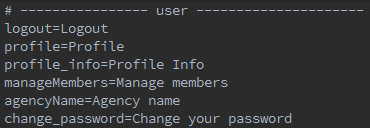


Figure 45. Indicative example of messages.en file

Each message is correlated with a unique index which is the same for the associated English and Greek texts. For instance, in case of “**profile\_info**=Profile Info” the first part (the bold one) corresponds to the index, while the second part is the actual text that appears in the DECIDE-IDSS platform. Thus, for changing the display name you have to modify the second part of each message.

## Source Code: How to rebuild & publish IDEA project on the VM

For the deployment of the application on the server we use the **dist** command provided by the Play Framework (Figure 46).

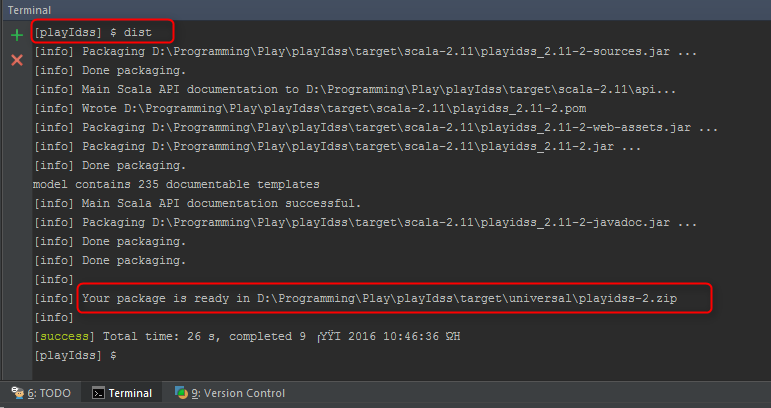


Figure 46. Play Framework “dist” task

This task produces a ZIP file containing all files needed to run the application. We upload the unzipped package on our server ($ chmod +x /path/to/bin/<project-name>) and we are ready to start the application (Figure 47).

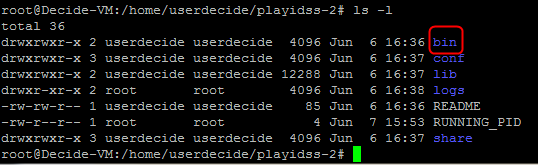


Figure 47. Contents of ZIP file after dist

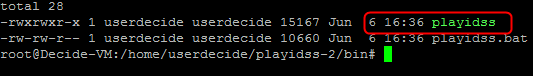


Figure 48. playidss executable

In the bin directory (Figure 48) you can find the executable “playidss”. We select the **Screen** manager in order to set the application running. We execute the following commands:

screen

./playidss

The iDSS application is up and running (Figure 49). After closing the terminal, the application will continue running on port 9000.

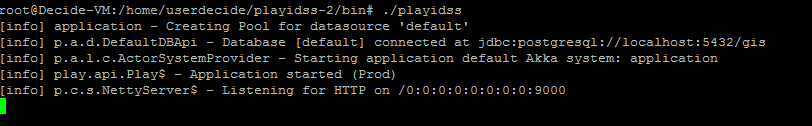


Figure 49. playidss running on Screen