



QGIS Server 3.16 User Guide

QGIS Project

02 apr 2022

1	Introduzione	1
2	Come Iniziare	3
2.1	INstallazione su sistemi Debian	3
2.1.1	Apache HTTP Server	4
2.1.2	NGINX HTTP Server	6
2.1.3	Xvfb	11
2.2	Installazione su Windows	12
2.3	Serve a project	14
2.4	Configura il tuo progetto	15
2.4.1	WMS capabilities	17
2.4.2	WFS capabilities	18
2.4.3	WCS capabilities	18
2.4.4	Fine tuning your OWS	18
2.5	Integrazione con terze parti	18
2.5.1	Integrazione con QGIS Desktop	18
2.5.2	Integrazione con MapProxy	18
2.5.3	Integrazione con QWC2	19
3	Servizi	21
3.1	Web Map Service (WMS)	21
3.1.1	GetMap	22
3.1.2	GetFeatureInfo	29
3.1.3	GetPrint	32
3.1.4	GetLegendGraphics	35
3.1.5	GetProjectSettings	36
3.2	Web Feature Service (WFS)	36
3.2.1	GetFeature	37
3.3	Web Map Tile Service (WMTS)	39
3.3.1	GetCapabilities	39
3.3.2	GetTile	40
3.3.3	GetFeatureInfo	42
3.4	WFS3 (funzioni API OGC)	44
3.4.1	Rappresentazione risorse	45
3.4.2	Endpoint	45
3.4.3	Paginazione	49
3.4.4	Filtro delle geometrie	49
3.4.5	Ordinamento elemento	50
3.4.6	Selezione attributo	50
3.4.7	Personalizzare le pagine HTML	51
3.5	Parametri extra supportati da tutti i tipi di richiesta	52

3.6	REDLINING	52
3.7	Layer WMS esterni	54
3.8	QGIS Server catalog	55
4	Plugin	59
4.1	Installazione	59
4.2	Configura il server HTTP	59
4.2.1	Apache	59
4.3	Come usare un plugin	60
5	Configurazione avanzata	61
5.1	Registrazioni di controllo	61
5.2	Variabili d'ambiente	61
5.3	Settings summary	63
5.4	Short name for layers, groups and project	64
5.5	Connection to service file	64
5.6	Add fonts to your linux server	64
6	Server di sviluppo	67
7	Sviluppo ed esecuzione in container	69
7.1	Simple docker images	69
7.1.1	First run	71
7.1.2	Usable sample	71
7.1.3	Cleanup	72
7.2	Docker stacks	72
7.2.1	Swarm/docker-compose	72
7.2.2	Kubernetes	73
7.3	Cloud deployment	77
7.3.1	AWS usecase	77
8	Frequently Asked Question	79

QGIS Server è un'implementazione open source di WMS, WFS, OGC API for Features 1.0 (WFS3) e WCS che, inoltre, implementa funzionalità cartografiche avanzate per la mappatura tematica. QGIS Server è un'applicazione FastCGI/CGI (Common Gateway Interface) scritta in C++ che lavora insieme a un server web (ad esempio, Apache, Nginx). Ha un supporto per i plugin Python che permette uno sviluppo veloce ed efficiente e l'implementazione di nuove funzionalità.

QGIS Server utilizza QGIS come backend per la restituzione delle mappe e per le operazioni gis. Inoltre, la libreria Qt viene utilizzata per la grafica e per la programmazione C++ indipendente dalla piattaforma. A differenza di altri server WMS, QGIS Server usa regole cartografiche come linguaggio di configurazione, sia per la configurazione a livello server che per le configurazioni utente.

Come QGIS desktop, QGIS Server utilizza le stesse librerie di visualizzazione e le mappe che vengono pubblicate sulla rete hanno lo stesso aspetto che in desktop GIS.

Nelle sezioni seguenti, forniremo una configurazione di esempio per impostare un server QGIS su Linux (Debian, Ubuntu e derivati) e su Windows. Per maggiori informazioni sullo sviluppo di plugin per server, leggi `server_plugins`.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section `gnu_fdl`.

2.1 INstallazione su sistemi Debian

Daremo un breve e semplice how-to di installazione per una configurazione minima funzionante su sistemi basati su Debian (incluso Ubuntu e derivati). Tuttavia, molte altre distribuzioni e sistemi operativi forniscono package per QGIS Server.

Nota: In Ubuntu puoi usare il tuo normale utente, aggiungendo `sudo` ai comandi che richiedono permessi di admin. In Debian puoi lavorare come admin (`root`), senza usare `sudo`.

Requirements and steps to add official QGIS repositories to install QGIS Server on a Debian based system are provided in [QGIS installers page](#). You may want to install at least the latest Long Term Release.

Once the target version repository is configured and QGIS Server installed, you can test the installation with:

```
/usr/lib/cgi-bin/qgis_mapserv.fcgi
```

If you get the following output, the server is correctly installed.

Nota: Depending on the version of QGIS, you might see slightly different output reported when you run `qgis_mapserv.fcgi`.

```
QFSFileEngine::open: No file name specified
Warning 1: Unable to find driver ECW to unload from GDAL_SKIP environment variable.
Warning 1: Unable to find driver ECW to unload from GDAL_SKIP environment variable.
Warning 1: Unable to find driver JP2ECW to unload from GDAL_SKIP environment_
↳variable.
Warning 1: Unable to find driver ECW to unload from GDAL_SKIP environment variable.
Warning 1: Unable to find driver JP2ECW to unload from GDAL_SKIP environment_
↳variable.
Content-Length: 206
Content-Type: text/xml; charset=utf-8

<ServiceExceptionReport version="1.3.0" xmlns="https://www.opengis.net/ogc">
```

(continues on next page)

(continua dalla pagina precedente)

```
<ServiceException code="Service configuration error">Service unknown or  
↳ unsupported</ServiceException>  
</ServiceExceptionReport>
```

Nota: As seen below, QGIS reports a Status 400 code, which correctly identifies the request has failed because there is no active http session. This is not a bug and indicates the server is functioning properly.

```
Application path not initialized  
Application path not initialized  
Warning 1: Unable to find driver ECW to unload from GDAL_SKIP environment variable.  
Warning 1: Unable to find driver ECW to unload from GDAL_SKIP environment variable.  
Warning 1: Unable to find driver JP2ECW to unload from GDAL_SKIP environment  
↳ variable.  
"Loading native module /usr/lib/qgis/server/libdummy.so"  
"Loading native module /usr/lib/qgis/server/liblandingpage.so"  
"Loading native module /usr/lib/qgis/server/libwcs.so"  
"Loading native module /usr/lib/qgis/server/libwfs.so"  
"Loading native module /usr/lib/qgis/server/libwfs3.so"  
"Loading native module /usr/lib/qgis/server/libwms.so"  
"Loading native module /usr/lib/qgis/server/libwmts.so"  
QFSFileEngine::open: No file name specified  
Content-Length: 102  
Content-Type: application/json  
Server: QGIS FCGI server - QGIS version 3.16.6-Hannover  
Status: 400  
[{"code": "Bad request error", "description": "Requested URI does not match any  
↳ registered API handler"}]
```

Let's add a sample project. You can use your own, or one from [Training demo data](#):

```
mkdir /home/qgis/projects/  
cd /home/qgis/projects/  
wget https://github.com/qgis/QGIS-Training-Data/archive/release_3.16.zip  
unzip release_3.16.zip  
mv QGIS-Training-Data-release_3.16/exercise_data/qgis-server-tutorial-data/world.  
↳ qgs .  
mv QGIS-Training-Data-release_3.16/exercise_data/qgis-server-tutorial-data/  
↳ naturalearth.sqlite .
```

Of course, you can use your favorite GIS software to open this file and take a look at the configuration and available layers.

To properly deploy QGIS server you need a HTTP server. Recommended choices are **Apache** or **NGINX**.

2.1.1 Apache HTTP Server

Nota: In the following, please replace `qgis.demo` with the name or IP address of your server.

Install Apache and `mod_fcgid`:

```
apt install apache2 libapache2-mod-fcgid
```

You can run QGIS Server on your default website, or configure a virtualhost specifically for this, as follows.

In the `/etc/apache2/sites-available` directory let's create a file called `qgis.demo.conf`, with this content:


```

<VirtualHost *:80>
  ServerAdmin webmaster@localhost
  ServerName qgis.demo

  DocumentRoot /var/www/html

  # Apache logs (different than QGIS Server log)
  ErrorLog ${APACHE_LOG_DIR}/qgis.demo.error.log
  CustomLog ${APACHE_LOG_DIR}/qgis.demo.access.log combined

  # Longer timeout for WPS... default = 40
  FcgidIOTimeout 120

  FcgidInitialEnv LC_ALL "en_US.UTF-8"
  FcgidInitialEnv PYTHONIOENCODING UTF-8
  FcgidInitialEnv LANG "en_US.UTF-8"

  # QGIS log
  FcgidInitialEnv QGIS_SERVER_LOG_STDERR 1
  FcgidInitialEnv QGIS_SERVER_LOG_LEVEL 0

  # default QGIS project
  SetEnv QGIS_PROJECT_FILE /home/qgis/projects/world.qgs

  # QGIS_AUTH_DB_DIR_PATH must lead to a directory writeable by the Server's FCGI_
  ↪process user
  FcgidInitialEnv QGIS_AUTH_DB_DIR_PATH "/home/qgis/qgisserverdb/"
  FcgidInitialEnv QGIS_AUTH_PASSWORD_FILE "/home/qgis/qgisserverdb/qgis-auth.db"

  # Set pg access via pg_service file
  SetEnv PGSERVICEFILE /home/qgis/.pg_service.conf
  FcgidInitialEnv PGPASSFILE "/home/qgis/.pgpass"

  # if qgis-server is installed from packages in debian based distros this is_
  ↪usually /usr/lib/cgi-bin/
  # run "locate qgis_mapserv.fcgi" if you don't know where qgis_mapserv.fcgi is
  ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
  <Directory "/usr/lib/cgi-bin/">
    AllowOverride None
    Options +ExecCGI -MultiViews -SymLinksIfOwnerMatch
    Order allow,deny
    Allow from all
    Require all granted
  </Directory>

  <IfModule mod_fcgid.c>
    FcgidMaxRequestLen 26214400
    FcgidConnectTimeout 60
  </IfModule>
</VirtualHost>

```

Ulteriori lettura:

- [QGIS Server logging](#)
- [pg-service-file in QGIS Server](#)

You can do the above in a linux Desktop system by pasting and saving the above configuration after doing:

```
nano /etc/apache2/sites-available/qgis.demo.conf
```

Nota: Some of the configuration options are explained in the Server *environment variables* section.

Let's now create the directories that will store the QGIS Server logs and the authentication database:

```
mkdir -p /var/log/qgis/  
chown www-data:www-data /var/log/qgis  
mkdir -p /home/qgis/qgisserverdb  
chown www-data:www-data /home/qgis/qgisserverdb
```

Nota: `www-data` is the Apache user on Debian based systems and we need Apache to have access to those locations or files. The `chown www-data . . .` commands change the owner of the respective directories and files to `www-data`.

We can now enable the *virtual host*, enable the `fcgid` mod if it's not already enabled:

```
a2enmod fcgid  
a2ensite qgis.demo
```

Ora riavvia Apache per usare la nuova configurazione:

```
systemctl restart apache2
```

Now that Apache knows that he should answer requests to <http://qgis.demo> we also need to setup the client system so that it knows who `qgis.demo` is. We do that by adding `127.0.0.1 qgis.demo` in the `hosts` file. We can do it with `sh -c "echo '127.0.0.1 qgis.demo' >> /etc/hosts"`. Replace `127.0.0.1` with the IP of your server.

Nota: Remember that both the `qgis.demo.conf` and `/etc/hosts` files should be configured for your setup to work. You can also test the access to your QGIS Server from other clients on the network (e.g. Windows or macOS machines) by going to their `/etc/hosts` file and point the `myhost` name to whatever IP the server machine has on the network (not `127.0.0.1` as it is the local IP, only accessible from the local machine). On **nix* machines the `hosts` file is located in `/etc`, while on Windows it's under the `C:\Windows\System32\drivers\etc` directory. Under Windows you need to start your text editor with administrator privileges before opening the `hosts` file.

QGIS Server is now available at <http://qgis.demo>. To check, type in a browser, as in the simple case:

```
http://qgis.demo/cgi-bin/qgis_mapserv.fcgi?SERVICE=WMS&VERSION=1.3.0&  
→REQUEST=GetCapabilities
```

2.1.2 NGINX HTTP Server

Nota: In the following, please replace `qgis.demo` with the name or IP address of your server.

Puoi utilizzare QGIS Server con **NGINX**. A differenza di Apache, NGINX non genera automaticamente processi FastCGI. I processi FastCGI devono essere attivati.

Installa NGINX:

```
apt install nginx
```

- As a first option, you can use **spawn-fcgi** or **fcgiwrap** to start and manage the QGIS Server processes. Official Debian packages exist for both. When you have no X server running and you need, for example, printing, you can use *xvfb*.

- Another option is to rely on **Systemd**, the init system for GNU/Linux that most Linux distributions use today. One of the advantages of this method is that it requires no other components or processes. It's meant to be simple, yet robust and efficient for production deployments.

Configurazione NGINX

The `include fastcgi_params`; used in the previous configuration is important, as it adds the parameters from `/etc/nginx/fastcgi_params`:

```
fastcgi_param QUERY_STRING      $query_string;
fastcgi_param REQUEST_METHOD    $request_method;
fastcgi_param CONTENT_TYPE      $content_type;
fastcgi_param CONTENT_LENGTH    $content_length;

fastcgi_param SCRIPT_NAME        $fastcgi_script_name;
fastcgi_param REQUEST_URI        $request_uri;
fastcgi_param DOCUMENT_URI      $document_uri;
fastcgi_param DOCUMENT_ROOT     $document_root;
fastcgi_param SERVER_PROTOCOL   $server_protocol;
fastcgi_param REQUEST_SCHEME    $scheme;
fastcgi_param HTTPS              $https if_not_empty;

fastcgi_param GATEWAY_INTERFACE  CGI/1.1;
fastcgi_param SERVER_SOFTWARE    nginx/$nginx_version;

fastcgi_param REMOTE_ADDR        $remote_addr;
fastcgi_param REMOTE_PORT        $remote_port;
fastcgi_param SERVER_ADDR        $server_addr;
fastcgi_param SERVER_PORT        $server_port;
fastcgi_param SERVER_NAME        $server_name;

# PHP only, required if PHP was built with --enable-force-cgi-redirect
fastcgi_param REDIRECT_STATUS    200;
```

Moreover, you can use some *Variabili d'ambiente* to configure QGIS Server. In the NGINX configuration file, `/etc/nginx/nginx.conf`, you have to use `fastcgi_param` instruction to define these variables as shown below:

```
location /qgisserver {
    gzip            off;
    include         fastcgi_params;
    fastcgi_param  QGIS_SERVER_LOG_STDERR  1;
    fastcgi_param  QGIS_SERVER_LOG_LEVEL  0;
    fastcgi_pass   unix:/var/run/qgisserver.socket;
}
```

FastCGI wrappers

Avvertimento: `fcgiwrap` is easier to set up than `spawn-fcgi`, because it's already wrapped in a Systemd service. But it also leads to a solution that is much slower than using `spawn-fcgi`. With `fcgiwrap`, a new QGIS Server process is created on each request, meaning that the QGIS Server initialization process, which includes reading and parsing the QGIS project file, is done on each request. With `spawn-fcgi`, the QGIS Server process remains alive between requests, resulting in much better performance. For that reason, `spawn-fcgi` is recommended for production use.

spawn-fcgi

If you want to use `spawn-fcgi`, the first step is to install the package:

```
apt install spawn-fcgi
```

Then, introduce the following block in your NGINX server configuration:

```
location /qgisserver {
    gzip            off;
    include        fastcgi_params;
    fastcgi_pass   unix:/var/run/qgisserver.socket;
}
```

And restart NGINX to take into account the new configuration:

```
systemctl restart nginx
```

Finally, considering that there is no default service file for `spawn-fcgi`, you have to manually start QGIS Server in your terminal:

```
spawn-fcgi -s /var/run/qgisserver.socket \
           -U www-data -G www-data -n \
           /usr/lib/cgi-bin/qgis_mapserv.fcgi
```

QGIS Server is now available at <http://qgis.demo/qgisserver>.

Nota: When using `spawn-fcgi`, you may directly define environment variables before running the server. For example:

```
export QGIS_SERVER_LOG_STDERR=1
```

Of course, you can add an init script to start QGIS Server at boot time or whenever you want. For example with **systemd**, edit the file `/etc/systemd/system/qgis-server.service` with this content:

```
[Unit]
Description=QGIS server
After=network.target

[Service]
;; set env var as needed
;Environment="LANG=en_EN.UTF-8"
;Environment="QGIS_SERVER_PARALLEL_RENDERING=1"
;Environment="QGIS_SERVER_MAX_THREADS=12"
;Environment="QGIS_SERVER_LOG_LEVEL=0"
;Environment="QGIS_SERVER_LOG_STDERR=1"
;; or use a file:
;EnvironmentFile=/etc/qgis-server/env

ExecStart=spawn-fcgi -s /var/run/qgisserver.socket -U www-data -G www-data -n /usr/
↪lib/cgi-bin/qgis_mapserv.fcgi

[Install]
WantedBy=multi-user.target
```

Then enable and start the service:

```
systemctl enable --now qgis-server
```

Avvertimento: With the above commands `spawn-fcgi` spawns only one QGIS Server process.

fcgiwrap

Using `fcgiwrap` is much easier to setup than `spawn-fcgi` but it's much slower. You first have to install the corresponding package:

```
apt install fcgiwrap
```

Then, introduce the following block in your NGINX server configuration:

```
1  location /qgisserver {
2      gzip            off;
3      include        fastcgi_params;
4      fastcgi_pass    unix:/var/run/fcgiwrap.socket;
5      fastcgi_param   SCRIPT_FILENAME /usr/lib/cgi-bin/qgis_mapserv.fcgi;
6  }
```

Finally, restart NGINX and `fcgiwrap` to take into account the new configuration:

```
systemctl restart nginx
systemctl restart fcgiwrap
```

QGIS Server is now available at <http://qgis.demo.qgisserver>.

Systemd

QGIS Server needs a running X Server to be fully usable, in particular for printing. In the case you already have a running X Server, you can use systemd services.

This method, to deploy QGIS Server, relies on two Systemd units:

- un Socket unit
- e un Service unit.

The **QGIS Server Socket unit** defines and creates a file system socket, used by NGINX to start and communicate with QGIS Server. The Socket unit has to be configured with `Accept=false`, meaning that the calls to the `accept()` system call are delegated to the process created by the Service unit. It is located in `/etc/systemd/system/qgis-server@.socket`, which is actually a template:

```
[Unit]
Description=QGIS Server Listen Socket (instance %i)

[Socket]
Accept=false
ListenStream=/var/run/qgis-server-%i.sock
SocketUser=www-data
SocketGroup=www-data
SocketMode=0600

[Install]
WantedBy=sockets.target
```

Now enable and start sockets:

```
for i in 1 2 3 4; do systemctl enable --now qgis-server@$i.socket; done
```

The **QGIS Server Service unit** defines and starts the QGIS Server process. The important part is that the Service process' standard input is connected to the socket defined by the Socket unit. This has to be configured using `StandardInput=socket` in the Service unit configuration located in `/etc/systemd/system/qgis-server@.service`:

```
[Unit]
Description=QGIS Server Service (instance %i)

[Service]
User=www-data
Group=www-data
StandardOutput=null
StandardError=journal
StandardInput=socket
ExecStart=/usr/lib/cgi-bin/qgis_mapserv.fcgi
EnvironmentFile=/etc/qgis-server/env

[Install]
WantedBy=multi-user.target
```

Nota: The QGIS Server *environment variables* are defined in a separate file, `/etc/qgis-server/env`. It could look like this:

```
QGIS_PROJECT_FILE=/etc/qgis/myproject.qgs
QGIS_SERVER_LOG_STDERR=1
QGIS_SERVER_LOG_LEVEL=3
```

Now start socket service:

```
for i in 1 2 3 4; do systemctl enable --now qgis-server@$i.service; done
```

Finally, for the NGINX HTTP server, lets introduce the configuration for this setup:

```
upstream qgis-server_backend {
    server unix:/var/run/qgis-server-1.sock;
    server unix:/var/run/qgis-server-2.sock;
    server unix:/var/run/qgis-server-3.sock;
    server unix:/var/run/qgis-server-4.sock;
}

server {
    ...

    location /qgis-server {
        gzip off;
        include fastcgi_params;
        fastcgi_pass qgis-server_backend;
    }
}
```

Now restart NGINX for the new configuration to be taken into account:

```
systemctl restart nginx
```

Thanks to Oslandia for sharing their tutorial.

2.1.3 Xvfb

QGIS Server needs a running X Server to be fully usable, in particular for printing. On servers it is usually recommended not to install it, so you may use `xvfb` to have a virtual X environment.

If you're running the Server in graphic/X11 environment then there is no need to install `xvfb`. More info at <https://www.itopen.it/qgis-server-setup-notes/>.

To install the package:

```
apt install xvfb
```

Create the service file, `/etc/systemd/system/xvfb.service`, with this content:

```
[Unit]
Description=X Virtual Frame Buffer Service
After=network.target

[Service]
ExecStart=/usr/bin/Xvfb :99 -screen 0 1024x768x24 -ac +extension GLX +render -
↳noreset

[Install]
WantedBy=multi-user.target
```

Abilita, avvia e verifica lo stato di `xvfb.service`:

```
systemctl enable --now xvfb.service
systemctl status xvfb.service
```

Then, according to your HTTP server, you should configure the **DISPLAY** parameter or directly use **xvfb-run**.

Con Apache

Then you can configure the **DISPLAY** parameter.

With Apache you just add to your *FastCGI* configuration (see above):

```
FcgidInitialEnv DISPLAY      ":99"
```

Ora riavvia Apache per usare la nuova configurazione:

```
systemctl restart apache2
```

Con NGINX

Then you can directly use **xvfb-run** or configure the **DISPLAY** parameter.

- With `spawn-fcgi` using `xvfb-run`:

```
xvfb-run /usr/bin/spawn-fcgi -f /usr/lib/cgi-bin/qgis_mapserv.fcgi \
-s /tmp/qgisserver.socket \
-G www-data -U www-data -n
```

- With the **DISPLAY** environment variable in the HTTP server configuration.

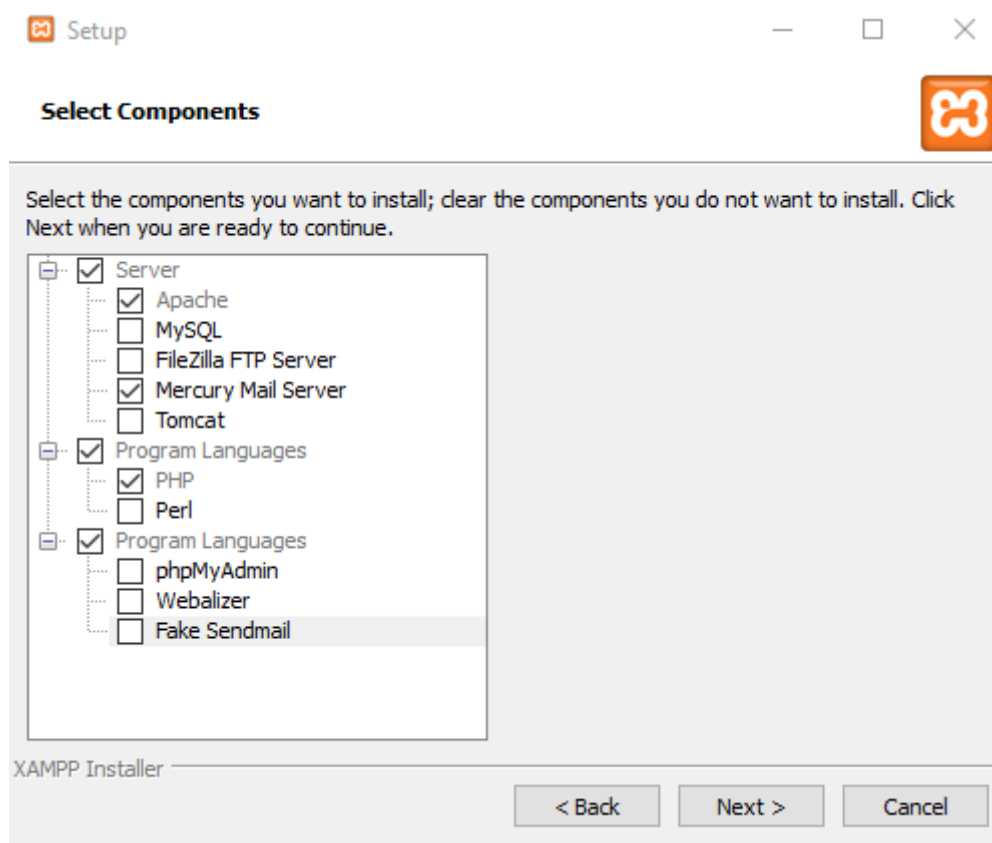
```
fastcgi_param DISPLAY      ":99";
```

2.2 Installazione su Windows

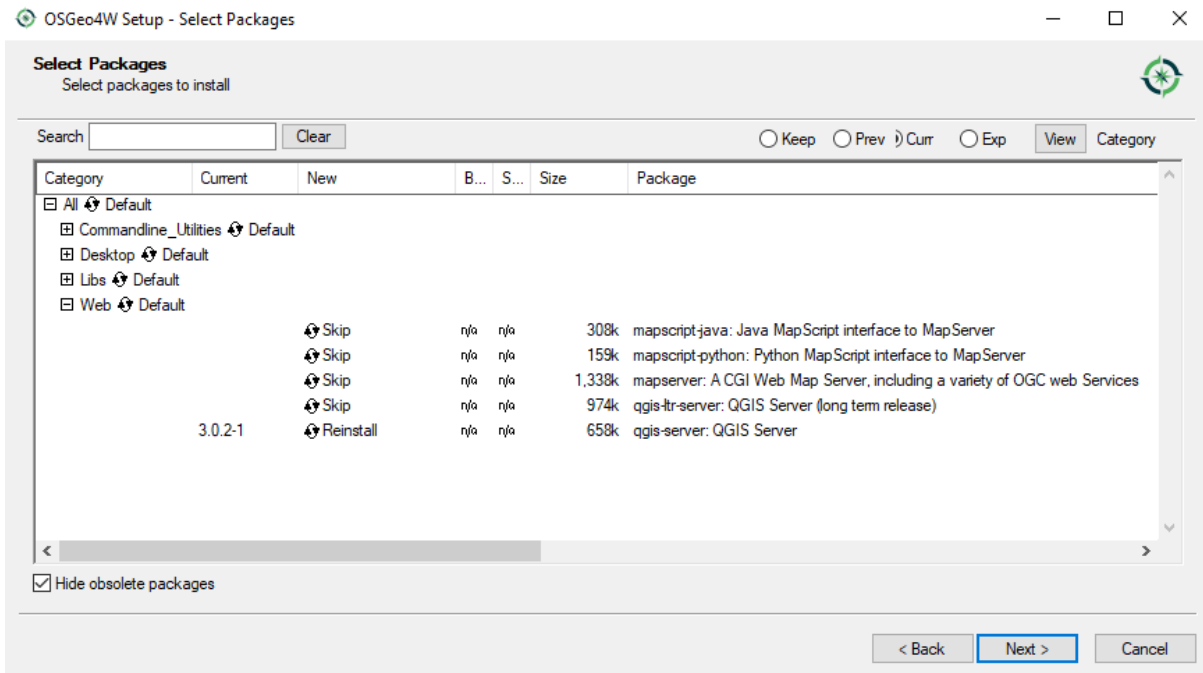
QGIS Server can also be installed on Windows systems. While the QGIS Server package is available in the 64 bit version of the OSGeo4W network installer (<https://qgis.org/en/site/forusers/download.html>) there is no Apache (or other web server) package available, so this must be installed by other means.

Una procedura semplice è la seguente:

1. Scarica il programma di installazione di XAMPP (<https://www.apachefriends.org/download.html>) per Windows e per installare Apache



2. Download the OSGeo4W installer, follow the «Advanced Install» and install both the QGIS Desktop and QGIS Server packages



3. Edit the httpd.conf file (C:\xampp\apache\conf\httpd.conf if the default installation paths have been used) and make the following changes:

Da:

```
ScriptAlias /cgi-bin/ "C:/xampp/cgi-bin/"
```

A:

```
ScriptAlias /cgi-bin/ "C:/OSGeo4W64/apps/qgis/bin/"
```

Da:

```
<Directory "C:/xampp/cgi-bin">
  AllowOverride None
  Options None
  Require all granted
</Directory>
```

A:

```
<Directory "C:/OSGeo4W64/apps/qgis/bin">
  SetHandler cgi-script
  AllowOverride None
  Options ExecCGI
  Order allow,deny
  Allow from all
  Require all granted
</Directory>
```

Da:

```
AddHandler cgi-script .cgi .pl .asp
```

A:

```
AddHandler cgi-script .cgi .pl .asp .exe
```

4. Then at the bottom of httpd.conf add:

```
SetEnv GDAL_DATA "C:\OSGeo4W64\share\gdal"  
SetEnv QGIS_AUTH_DB_DIR_PATH "C:\OSGeo4W64\apps\qgis\resources"  
SetEnv PYTHONHOME "C:\OSGeo4W64\apps\Python37"  
SetEnv PATH "C:\OSGeo4W64\bin;C:\OSGeo4W64\apps\qgis\bin;C:\OSGeo4W64\apps\Qt5\  
↪bin;C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem"  
SetEnv QGIS_PREFIX_PATH "C:\OSGeo4W64\apps\qgis"  
SetEnv QT_PLUGIN_PATH "C:\OSGeo4W64\apps\qgis\qtplugins;C:\OSGeo4W64\apps\Qt5\  
↪plugins"
```

5. Riavvia il server web Apache dal pannello di controllo XAMPP e apri la finestra del browser per testare una richiesta GetCapabilities al server QGIS

```
http://qgis.demo/cgi-bin/qgis_mapserv.fcgi.exe?SERVICE=WMS&VERSION=1.3.0&  
↪REQUEST=GetCapabilities
```

2.3 Serve a project

Now that QGIS Server is installed and running, we just have to use it.

Obviously, we need a QGIS project to work on. Of course, you can fully customize your project by defining contact information, precise some restrictions on CRS or even exclude some layers. Everything you need to know about that is described later in *Configura il tuo progetto*.

But for now, we are going to use a simple project already configured and previously downloaded in `/home/qgis/projects/world.qgs`, as described above.

By opening the project and taking a quick look on layers, we know that 4 layers are currently available:

- airports
- places
- countries
- countries_shapeburst

You don't have to understand the full request for now but you may retrieve a map with some of the previous layers thanks to QGIS Server by doing something like this in your web browser to retrieve the *countries* layer:

```
http://qgis.demo/qgisserver?  
MAP=/home/qgis/projects/world.qgs&  
LAYERS=countries&  
SERVICE=WMS&  
VERSION=1.3.0&  
REQUEST=GetMap&  
CRS=EPSG:4326&  
WIDTH=400&  
HEIGHT=200&  
BBOX=-90,-180,90,180
```

If you obtain the next image, then QGIS Server is running correctly:

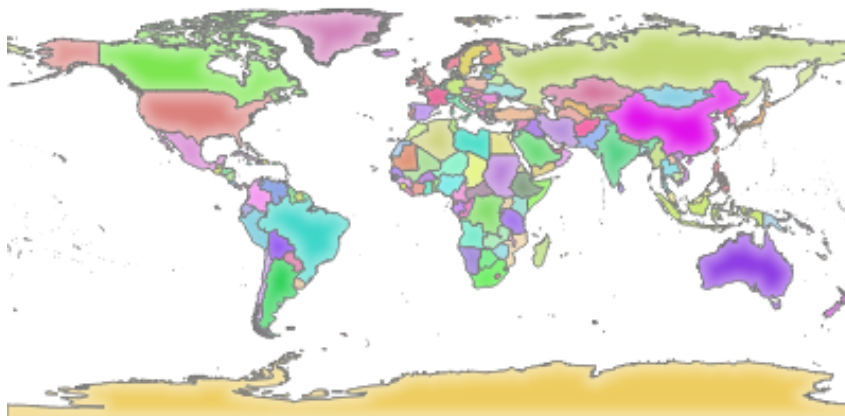


Fig. 2.1: Server response to a basic GetMap request

Note that you may define **QGIS_PROJECT_FILE** environment variable to use a project by default instead of giving a **MAP** parameter (see *Variabili d'ambiente*).

For example with spawn-fcgi:

```
export QGIS_PROJECT_FILE=/home/qgis/projects/world.qgs
spawn-fcgi -f /usr/lib/bin/cgi-bin/qgis_mapserv.fcgi \  
-s /var/run/qgisserver.socket \  
-U www-data -G www-data -n
```

2.4 Configura il tuo progetto

To provide a new QGIS Server WMS, WFS or WCS, you have to create a QGIS project file with some data or use one of your current project. Define the colors and styles of the layers in QGIS and the project CRS, if not already defined.

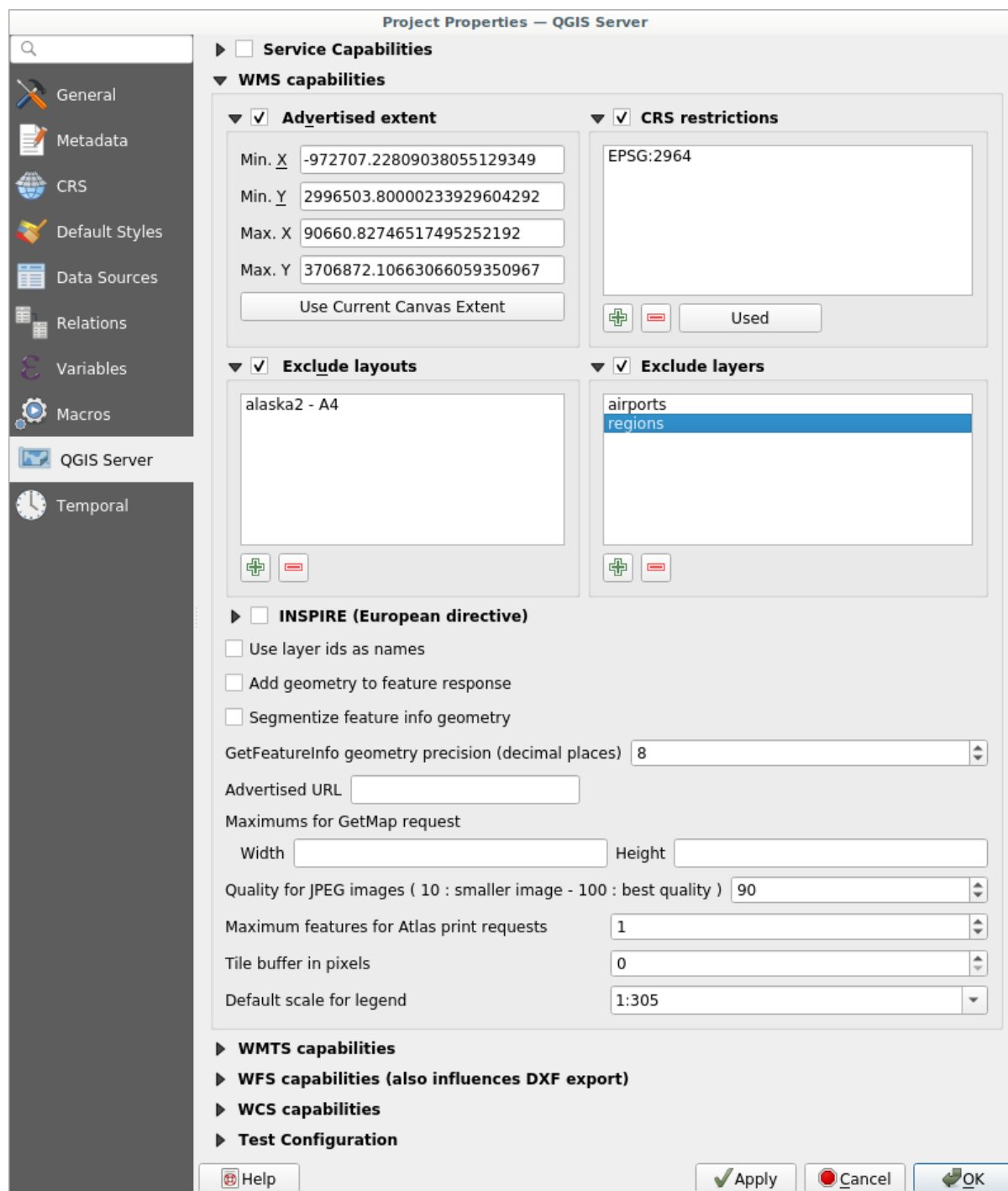





Fig. 2.2: Definizioni per un progetto QGIS Server WMS/WFS/WCS

Then, go to the *QGIS Server* menu of the *Project ► Properties...* dialog and provide some information about the OWS in the fields under *Service Capabilities*. This will appear in the *GetCapabilities* response of the WMS, WFS or WCS. If you don't check *Service capabilities*, QGIS Server will use the information given in the `wms_metadata.xml` file located in the `cgi-bin` folder.

2.4.1 WMS capabilities

In the *WMS capabilities* section, you can define the extent advertised in the WMS GetCapabilities response by entering the minimum and maximum X and Y values in the fields under *Advertised extent*. Clicking *Use Current Canvas Extent* sets these values to the extent currently displayed in the QGIS map canvas. By checking *CRS restrictions*, you can restrict in which coordinate reference systems (CRS) QGIS Server will offer to render maps. It is recommended that you restrict the offered CRS as this reduces the size of the WMS GetCapabilities response. Use the  button below to select those CRSs from the Coordinate Reference System Selector, or click *Used* to add the CRSs used in the QGIS project to the list.

If you have print layouts defined in your project, they will be listed in the *GetProjectSettings* response, and they can be used by the *GetPrint* request to create prints, using one of the print layouts as a template. This is a QGIS-specific extension to the WMS 1.3.0 specification. If you want to exclude any print layout from being published by the WMS, check *Exclude layouts* and click the  button below. Then, select a print layout from the *Select print layout* dialog in order to add it to the excluded layouts list.

If you want to exclude any layer or layer group from being published by the WMS, check *Exclude Layers* and click the  button below. This opens the *Select restricted layers and groups* dialog, which allows you to choose the layers and groups that you don't want to be published. Use the *Shift* or *Ctrl* key if you want to select multiple entries. It is recommended that you exclude from publishing the layers that you don't need as this reduces the size of the WMS GetCapabilities response which leads to faster loading times on the client side.

If you check *Use layer ids as name*, layer ids will be used to reference layers in the *GetCapabilities* response or *GetMap* `LAYERS` parameter. If not, layer name or short name if defined (see `vectorservermenu`) is used.

You can receive requested *GetFeatureInfo* as plain text, XML and GML. The default is XML.

If you wish, you can check *Add geometry to feature response*. This will include the bounding box for each feature in the *GetFeatureInfo* response. See also the `WITH_GEOMETRY` parameter.

As many web clients can't display circular arcs in geometries you have the option to segmentize the geometry before sending it to the client in a *GetFeatureInfo* response. This allows such clients to still display a feature's geometry (e.g. for highlighting the feature). You need to check the *Segmentize feature info geometry* to activate the option.

You can also use the *GetFeatureInfo geometry precision* option to set the precision of the *GetFeatureInfo* geometry. This enables you to save bandwidth when you don't need the full precision.

If you want QGIS Server to advertise specific request URLs in the WMS GetCapabilities response, enter the corresponding URL in the *Advertised URL* field.

Furthermore, you can restrict the maximum size of the maps returned by the *GetMap* request by entering the maximum width and height into the respective fields under *Maximums for GetMap request*.

You can change the *Quality for JPEG images* factor. The quality factor must be in the range 0 to 100. Specify 0 for maximum compression, 100 for no compression.

You can change the limit for atlas features to be printed in one request by setting the *Maximum features for Atlas print requests* field.

When QGIS Server is used in tiled mode (see *TILED parameter*), you can set the *Tile buffer in pixels*. The recommended value is the size of the largest symbol or line width in your QGIS project.

If one of your layers uses the Map Tip display (i.e. to show text using expressions) this will be listed inside the *GetFeatureInfo* output. If the layer uses a Value Map for one of its attributes, this information will also be shown in the *GetFeatureInfo* output.

2.4.2 WFS capabilities

In the *WFS capabilities* area you can select the layers you want to publish as WFS, and specify if they will allow update, insert and delete operations. If you enter a URL in the *Advertised URL* field of the *WFS capabilities* section, QGIS Server will advertise this specific URL in the WFS GetCapabilities response.

2.4.3 WCS capabilities

In the *WCS capabilities* area, you can select the layers that you want to publish as WCS. If you enter a URL in the *Advertised URL* field of the *WCS capabilities* section, QGIS Server will advertise this specific URL in the WCS GetCapabilities response.

2.4.4 Fine tuning your OWS

For vector layers, the *Fields* menu of the *Layer ► Layer Properties* dialog allows you to define for each attribute if it will be published or not. By default, all the attributes are published by your WMS and WFS. If you don't want a specific attribute to be published, uncheck the corresponding checkbox in the *WMS* or *WFS* column.

You can overlay watermarks over the maps produced by your WMS by adding text annotations or SVG annotations to the project file. See the *sec_annotations* section for instructions on creating annotations. For annotations to be displayed as watermarks on the WMS output, the *Fixed map position* checkbox in the *Annotation text* dialog must be unchecked. This can be accessed by double clicking the annotation while one of the annotation tools is active. For SVG annotations, you will need either to set the project to save absolute paths (in the *General* menu of the *Project ► Properties...* dialog) or to manually modify the path to the SVG image so that it represents a valid relative path.

2.5 Integrazione con terze parti

QGIS Server provides standard OGC web services like [WMS](#), [WFS](#), etc. thus it can be used by a wide variety of end user tools.

2.5.1 Integrazione con QGIS Desktop

QGIS Desktop is the map designer where QGIS Server is the map server. The maps or QGIS projects will be served by the QGIS Server to provide OGC standards. These QGIS projects can either be files or entries in a database (by using *Project ► Save to ► PostgreSQL* in QGIS Desktop).

Furthermore, dedicated update workflow must be established to refresh a project used by a QGIS Server (ie. copy project files into server location and restart QGIS Server). For now, automated processes (as server reloading over message queue service) are not implemented yet.

2.5.2 Integrazione con MapProxy

[MapProxy](#) is a tile cache server and as it can read and serve any WMS/WMTS map server, it can be directly connected to QGIS server web services and improve end user experience.

2.5.3 Integrazione con QWC2

QWC2 is a responsive web application dedicated to QGIS Server. It helps you to build a highly customized map viewer with layer selection, feature info, etc.. Also many plugins are available like authentication or print service, the full list is available in this [repository](#).

QGIS è in grado di gestire i dati secondo i protocolli standard come descritto da **Open Geospatial Consortium (OGC)**:

- WMS 1.1.1 e 1.3.0
- WFS 1.0.0 e 1.1.0
- OGC API - Features (WFS3)
- WCS 1.1.1
- WMTS 1.0.0

Ulteriori parametri e richieste sono supportati in aggiunta allo standard originale che migliora notevolmente le possibilità di personalizzazione grazie al motore di visualizzazione QGIS.

3.1 Web Map Service (WMS)

Gli standard **1.1.1** e **1.3.0** WMS implementati in QGIS Server forniscono un'interfaccia HTTP per richiedere immagini di mappe o legende generate da un progetto QGIS. Una tipica richiesta WMS definisce il progetto QGIS da usare, i layer da visualizzare e il formato dell'immagine da generare. Il supporto di base è disponibile anche per **Styled Layer Descriptor (SLD)**.

Specifiche:

- WMS 1.1.1
- WMS 1.3.0
- SLD 1.1.0 WMS profile

Richieste standard fornite da QGIS Server:

Request	Descrizione
<i>GetCapabilities</i>	Restituisce metadati XML con informazioni sul server
<i>GetMap</i>	Restituisce una mappa
<i>GetFeatureInfo</i>	Recupera i dati (geometria e valori) per una posizione di pixel
<i>GetLegendGraphics</i>	Restituisce i simboli della legenda

Richieste del fornitore date da QGIS Server:

Request	Descrizione
<i>GetPrint</i>	Restituisce una composizione QGIS
<i>GetProjectSettings</i>	Restituisce informazioni specifiche su QGIS Server

3.1.1 GetMap

Parametri standard per la richiesta **GetMap** secondo le specifiche OGC WMS 1.1.1 e 1.3.0:

Parametro	Richiesto	Descrizione
<i>SERVICE</i>	Si	Nome del servizio (WMS)
<i>VERSION</i>	Si	Versione del servizio
<i>REQUEST</i>	Si	Nome della richiesta (GetMap)
<i>LAYERS</i>	No	Layer da visualizzare
<i>STYLES</i>	No	Stile del layers
<i>SRS / CRS</i>	Si	Sistema di riferimento delle Coordinate
<i>BBOX</i>	Si	Estensione Mappa
<i>WIDTH</i>	Si	Larghezza dell'immagine in pixel
<i>HEIGHT</i>	Si	Altezza dell'immagine in pixel
<i>FORMAT</i>	No	Formato immagine
<i>TRANSPARENT</i>	No	Sfondo trasparente
<i>SLD</i>	No	URL di un SLD da usare per lo stile
<i>SLD_BODY</i>	No	SLD in linea (XML) da usare per lo stile

Oltre a quelli standard, QGIS Server supporta i seguenti parametri extra:

Parametro	Richiesto	Descrizione
<i>MAP</i>	Si	Specifica del file di progetto QGIS
<i>BGCOLOR</i>	No	Specifica del colore di sfondo
<i>DPI</i>	No	Specifica della risoluzione in uscita
<i>IMAGE_QUALITY</i>	No	Compressione JPEG
<i>OPACITIES</i>	No	Opacità per il layer o il gruppo
<i>FILTER</i>	No	Sottoinsieme di elementi
<i>SELECTION</i>	No	Elementi principali
<i>FILE_NAME</i>	No	Nome del file scaricato Solo per <i>FORMAT=application/dxf</i>
<i>FORMAT_OPTIONS</i>	No	Opzioni del formato di file specificato Solo per <i>FORMAT=application/dxf</i>
<i>TILED</i>	No	Lavorare in <i>modalità tassellata</i>

Esempio di URL:

```
http://localhost/qgis_server?
SERVICE=WMS
&VERSION=1.3.0
&REQUEST=GetMap
&MAP=/home/qgis/projects/world.qgs
&LAYERS=mylayer1,mylayer2,mylayer3
&STYLES=style1,default,style3
&OPACITIES=125,200,125
&CRS=EPSG:4326
&WIDTH=400
&HEIGHT=400
&FORMAT=image/png
```

(continues on next page)

(continua dalla pagina precedente)

```
&TRANSPARENT=TRUE
&DPI=300
&TILED=TRUE
```

SERVICE

Questo parametro deve essere WMS.

Per esempio:

```
http://localhost/qgisserver?
SERVICE=WMS
&...
```

VERSION

Questo parametro permette di specificare la versione del servizio da utilizzare. I valori disponibili per il parametro VERSION sono:

- 1.1.1
- 1.3.0

A seconda del numero di versione, ci si devono aspettare leggere differenze come spiegato più avanti per i prossimi parametri:

- CRS / SRS
- BBOX

REQUEST

Questo parametro è GetMap nel caso della richiesta **GetMap**.

LAYERS

Questo parametro permette di specificare i layer da visualizzare sulla mappa. I nomi devono essere separati da una virgola.

Inoltre, QGIS Server ha introdotto alcune opzioni per selezionare i layer:

- un nome breve
- il layer id

Il nome breve di un layer può essere configurato attraverso :menuselection: *Proprietà* → *Metadati* nel menu del layer. Se il nome breve è definito, allora viene usato come predefinito al posto del nome del layer:

```
http://localhost/qgisserver?
SERVICE=WMS
&REQUEST=GetMap
&LAYERS=mynickname1,mynickname2
&...
```

Inoltre, c'è un'opzione del progetto che permette di selezionare i layer in base al loro id nel menu *OWS Server* ► *WMS capabilities* della finestra di dialogo :menuselection: *Progetto* → *Proprietà*.... Per attivare questa opzione, la casella di controllo *Usa gli id del layer come nomi* deve essere selezionata.

```
http://localhost/qgisserver?  
SERVICE=WMS  
&REQUEST=GetMap  
&LAYERS=mylayerid1,mylayerid2  
&...
```

STYLES

Questo parametro può essere usato per specificare lo stile di un layer per il processo di visualizzazione. Gli stili devono essere separati da una virgola. Il nome dello stile predefinito è `default`.

SRS / CRS

Questo parametro permette di indicare il sistema di riferimento spaziale in uscita dalla mappa in WMS **1.1.1** e deve essere formulato come `EPSG:XXXX`. Nota che `SR` è supportato anche se la versione corrente è **1.1.1**.

Per WMS **1.3.0**, il parametro `SR` è preferibile ma è supportato anche `SRS`.

Nota che se entrambi i parametri `SR` e `SRS` sono indicati nella richiesta, allora è la versione corrente indicata nel parametro `VERSION` che è decisiva.

Nel prossimo caso, il parametro `SRS` è mantenuto qualunque sia il parametro `VERSION` perché `SR` non è indicato:

```
http://localhost/qgisserver?  
SERVICE=WMS  
&REQUEST=GetMap  
&VERSION=1.3.0  
&SRS=EPSG:2854  
&...
```

Nel prossimo caso, il parametro `SRS` viene tenuto al posto di `SR` a causa del parametro `VERSION`:

```
http://localhost/qgisserver?  
SERVICE=WMS  
&REQUEST=GetMap  
&VERSION=1.1.1  
&CRS=EPSG:4326  
&SRS=EPSG:2854  
&...
```

Nel prossimo caso, il parametro `SR` viene tenuto al posto di `SRS` a causa del parametro `VERSION`:

```
http://localhost/qgisserver?  
SERVICE=WMS  
&REQUEST=GetMap  
&VERSION=1.3.0  
&CRS=EPSG:4326  
&SRS=EPSG:2854  
&...
```

BBOX

Questo parametro permette di specificare l'estensione della mappa con unità secondo il SR corrente. Le coordinate devono essere separate da una virgola.

Il parametro BBOX è formato come `min_a, min_b, max_a, max_b` ma la definizione degli assi `a` e `b` è diversa secondo il parametro corrente `VERSION`:

- in **WMS 1.1.1**, la disposizione degli assi è sempre est/nord
- in **WMS 1.3.0**, la disposizione degli assi dipende dall'autorità del SR

Per esempio nel caso di `EPSG:4326` e **WMS 1.1.1**, `a` è la longitudine (est) e `b`` la latitudine (nord), portando a una richiesta come:

```
http://localhost/qgisserver?
SERVICE=WMS
&REQUEST=GetMap
&VERSION=1.1.1
&SRS=epsg:4326
&BBOX=-180,-90,180,90
&...
```

Ma nel caso di **WMS 1.3.0**, l'ordine degli assi definito nel database `EPSG` è nord/est quindi `a` è la latitudine e `b`` la longitudine:

```
http://localhost/qgisserver?
SERVICE=WMS
&REQUEST=GetMap
&VERSION=1.3.0
&CRS=epsg:4326
&BBOX=-90,-180,90,180
&...
```

WIDTH

Questo parametro permette di specificare la larghezza in pixel dell'immagine in uscita.

HEIGHT

Questo parametro permette di specificare l'altezza in pixel dell'immagine in uscita.

FORMAT

Questo parametro può essere usato per specificare il formato dell'immagine della mappa. I valori disponibili sono:

- `jpg`
- `jpeg`
- `image/jpeg`
- `image/png`
- `image/png; mode=1bit`
- `image/png; mode=8bit`
- `image/png; mode=16bit`

- `application/dxf`` Solo i layer che hanno accesso in lettura nel servizio WFS sono esportati nel formato DXF.

Esempio di URL:

```
http://localhost/qgisserver?
SERVICE=WMS&VERSION=1.3.0
&REQUEST=GetMap
&FORMAT=application/dxf
&LAYERS=Haltungen,Normschacht,Spezialbauwerke
&STYLES=
&CRS=EPSG%3A21781&BBOX=696136.28844801,245797.12108743,696318.91114315,245939.
↪25832905
&WIDTH=1042
&HEIGHT=811
&FORMAT_OPTIONS=MODE:SYMBOLLAYERSYMBOLGY;SCALE:250&FILE_NAME=plan.dxf
```

TRANSPARENT

Questo parametro booleano può essere usato per specificare la trasparenza dello sfondo. I valori disponibili sono (non sensibili alle maiuscole):

- TRUE
- FALSE

Tuttavia, questo parametro viene ignorato se il formato dell'immagine della mappa indicato con `FORMAT` è diverso da PNG.

MAP

Questo parametro permette di definire il file di progetto QGIS da utilizzare.

Come menzionato in *GetMap parameters table*, `MAP` è obbligatorio perché una richiesta ha bisogno di un progetto QGIS per funzionare effettivamente. Tuttavia, la variabile d'ambiente `QGIS_PROJECT_FILE` può essere usata per definire un progetto QGIS predefinito. In questo caso specifico, `MAP` non è più un parametro richiesto. Per ulteriori informazioni puoi fare riferimento a *Configurazione avanzata*.

BGCOLOR

Questo parametro permette di indicare un colore di sfondo per l'immagine della mappa. Tuttavia non può essere combinato con il parametro `TRANSPARENT` in caso di immagini PNG (la trasparenza ha la priorità). Il colore può essere letterale o in notazione esadecimale.

Esempio di URL con la notazione letterale:

```
http://localhost/qgisserver?
SERVICE=WMS
&REQUEST=GetMap
&VERSION=1.3.0
&BGCOLOR=green
&...
```

Esempio di URL con la notazione esadecimale:

```
http://localhost/qgisserver?
SERVICE=WMS
&REQUEST=GetMap
&VERSION=1.3.0
&BGCOLOR=0x00FF00
&...
```

DPI

Questo parametro può essere usato per specificare la risoluzione in uscita richiesta.

IMAGE_QUALITY

Questo parametro è usato solo per le immagini JPEG. Per impostazione predefinita, la compressione JPEG è -1.

Puoi cambiare il default per ogni progetto QGIS nel menu *OWS Server ► WMS capabilities* della finestra di dialogo *Progetto ► Proprietà...* Se vuoi sovrascriverlo in una richiesta *GetMap* puoi farlo usando il parametro *IMAGE_QUALITY*.

OPACITIES

Lista separata da virgole di valori di opacità. L'opacità può essere impostata a livello di layer o di gruppo. I valori consentiti vanno da 0 (completamente trasparente) a 255 (completamente opaco).

FILTER

Un sottoinsieme di layer può essere selezionato con il parametro *FILTER*. La sintassi è fondamentalmente la stessa della stringa del sottoinsieme di QGIS. Tuttavia, ci sono alcune restrizioni per evitare introduzioni SQL nei database tramite QGIS Server. Se viene trovata una stringa pericolosa nel parametro, QGIS Server restituirà l'errore successivo:

```
<ServiceExceptionReport>
  <ServiceException code="Security">The filter string XXXXXXXXX has been rejected.
  ↳because of security reasons.
  Note: Text strings have to be enclosed in single or double quotes. A space.
  ↳between each word / special character is mandatory.
  Allowed Keywords and special characters are IS,NOT,NULL,AND,OR,IN,=,<,<,>,>=,!=,
  ↳',',',(',') ,DMETAPHONE,SOUNDEX.
  Not allowed are semicolons in the filter expression.</ServiceException>
</ServiceExceptionReport>
```

Esempio di URL:

```
http://localhost/qgisserver?
SERVICE=WMS
&REQUEST=GetMap
&LAYERS=mylayer1,mylayer2,mylayer3
&FILTER=mylayer1:"col1";mylayer1,mylayer2:"col2" = 'blabla'
&...
```

In questo esempio, lo stesso filtro (il campo *col2* è uguale alla stringa *blabla*) è applicato ai layer *mylayer1* e *mylayer2*, mentre il filtro su *col1* è applicato solo a *mylayer1*.

Nota: È possibile fare ricerche di attributi tramite *GetFeatureInfo* e omettere il parametro *X/Y* se c'è un *FILTER*. QGIS Server restituisce quindi le informazioni sugli elementi corrispondenti e genera un perimetro di delimitazione combinato nell'XML in uscita.

SELECTION

Il parametro `SELECTION` può evidenziare elementi da uno o più layer. Gli elementi vettoriali possono essere selezionati passandogli liste separate da virgole con gli id degli elementi.

```
http://localhost/qgisserver?  
SERVICE=WMS  
&REQUEST=GetMap  
&LAYERS=mylayer1,mylayer2  
&SELECTION=mylayer1:3,6,9;mylayer2:1,5,6  
&...
```

L'immagine seguente presenta la risposta di una richiesta `GetMap` utilizzando l'opzione `SELECTION`, ad esempio `http://myserver.com/...&SELECTION=countries:171,65`.

Poiché questi id degli elementi corrispondono nell'insieme dei dati di origine a **France** e **Romania** sono evidenziati in giallo.



Fig. 3.1: Risposta del server a una richiesta `GetMap` con il parametro `SELECTION`

FORMAT-OPTIONS

Questo parametro può essere usato per specificare le opzioni per il formato selezionato. Solo per `FORMAT=application/dxf`. Una lista di coppie chiave:valore separate da punto e virgola:

- `SCALE`: da usare per le regole della simbologia, i filtri e gli stili (non la scalatura effettiva dei dati - i dati rimangono nella scala originale).
- `MODE`: corrisponde alle opzioni di esportazione offerte nella finestra di dialogo di esportazione DXF di QGIS Desktop. I valori possibili sono `NOSYMBOLLOGY`, `FEATURESYMBOLLOGY` e `SYMBOLLAYERSYMBOLLOGY`.
- `LAYERSATTRIBUTES`: specifica un campo che contiene valori per i nomi dei layer DXF - se non specificato, vengono usati i nomi originali dei layer QGIS.
- `USE_TITLE_AS_LAYERNAME`: se abilitato, il titolo del layer sarà usato come nome del layer.
- `CODEC`: specifica un codec da usare per la codifica. Il valore predefinito è `ISO-8859-1`, verificare la finestra di dialogo di esportazione DXF di QGIS desktop per i valori validi.
- `NO_MTEXT`: usa `TEXT` invece di `MTEXT` per le etichette.
- `FORCE_2D`: Forza il risultato in 2D. Questo è richiesto per la larghezza della polilinea.

TILED

Per ragioni di performance, QGIS Server può essere usato in modalità tiled. In questa modalità, il client richiede diverse piccoli tasselli di dimensioni fisse, e le assembla per formare l'intera mappa. Facendo questo, i simboli al o vicino al confine tra due tasselli possono apparire tagliati, perché sono presenti solo in uno dei tasselli.

Imposta il parametro `TILED` a `TRUE` per dire a QGIS Server di lavorare in modalità *tiled*, e di applicare il *Tassello buffer* configurato nel progetto QGIS (vedi *Configura il tuo progetto*).

Quando `TILED` è `TRUE` e quando un Tassello buffer non nullo è configurato nel progetto QGIS, gli elementi al di fuori dell'estensione del tassello sono disegnati per evitare simboli di taglio ai confini del tassello.

Per default, `TILED` è impostato su `FALSE`.

3.1.2 GetFeatureInfo

Parametri standard per la richiesta **GetFeatureInfo** secondo le specifiche OGC WMS 1.1.1 e 1.3.0:

Parametro	Richiesto	Descrizione
<i>SERVICE</i>	Sì	Nome del servizio (WMS)
<i>VERSION</i>	No	Versione del servizio
<i>REQUEST</i>	Sì	Nome della richiesta (GetFeatureInfo)
<i>QUERY_LAYERS</i>	Sì	Layer su cui fare la query
<i>LAYERS</i>	Sì	Layer da visualizzare (identico a <i>QUERY_LAYERS</i>)
<i>STYLES</i>	No	Stile del layers
<i>SRS</i> / <i>CRS</i>	Sì	Sistema di riferimento delle Coordinate
<i>BBOX</i>	No	Estensione Mappa
<i>WIDTH</i>	Sì	Larghezza dell'immagine in pixel
<i>HEIGHT</i>	Sì	Altezza dell'immagine in pixel
<i>TRANSPARENT</i>	No	Sfondo trasparente
<i>INFO_FORMAT</i>	No	Formato in uscita
<i>FEATURE_COUNT</i>	No	Numero massimo di elementi da restituire
<i>I</i>	No	Colonna pixel del punto da interrogare
<i>X</i>	No	Come il parametro <i>I</i> , ma in WMS 1.1.1
<i>J</i>	No	Riga pixel del punto da interrogare
<i>Y</i>	No	Come il parametro <i>J</i> , ma in WMS 1.1.1
<i>WMS_PRECISION</i>	No	La precisione (numero di cifre) da usare quando si restituisce la geometria (vedi <i>how to add geometry to feature response</i>). Il valore predefinito è -1 che significa che viene usata la precisione definita nel progetto.

Oltre a quelli standard, QGIS Server supporta i seguenti parametri extra:

Parametro	Richiesto	Descrizione
<i>MAP</i>	Sì	Specifica del file di progetto QGIS
<i>FILTER</i>	No	Sottoinsieme di elementi
<i>FL_POINT_TOLERANCE</i>	No	Tolleranza in pixel per i layer punto
<i>FL_LINE_TOLERANCE</i>	No	Tolleranza in pixel per i layer linea
<i>FL_POLYGON_TOLERANCE</i>	No	Tolleranza in pixel per i layer poligono
<i>FILTER_GEOM</i>	No	Filtrare geometria
<i>WITH_MAPTIP</i>	No	Aggiungere suggerimenti mappa al risultato
<i>WITH_GEOMETRY</i>	No	Aggiungere la geometria al risultato

Esempio di URL:

```
http://localhost/qgisserver?
SERVICE=WMS
&VERSION=1.3.0
&REQUEST=GetMap
&MAP=/home/qgis/projects/world.qgs
&LAYERS=mylayer1,mylayer2,mylayer3
&CRS=EPSG:4326
&WIDTH=400
&HEIGHT=400
&INFO_FORMAT=text/xml
&TRANSPARENT=TRUE
&QUERY_LAYERS=mylayer1
&FEATURE_COUNT=3
&I=250
&J=250
```

REQUEST

Questo parametro è `GetFeatureInfo` nel caso della richiesta **GetFeatureInfo**.

INFO_FORMAT

Questo parametro può essere usato per specificare il formato del risultato. I valori disponibili sono:

- `text/xml`
- `text/html`
- `text/plain`
- `application/vnd.ogc.gml`
- `application/json`

QUERY_LAYERS

Questo parametro specifica i layer da visualizzare sulla mappa. I nomi sono separati da una virgola.

Inoltre, QGIS Server introduce opzioni per selezionare i layer in base a:

- nome breve
- layer id

Vedi il parametro `LAYERS` definito in *GetMap* per maggiori informazioni.

FEATURE_COUNT

Questo parametro specifica il numero massimo di elementi per layer da restituire. Per esempio se `QUERY_LAYERS` è impostato su `layer1, layer2` e `FEATURE_COUNT` è impostato su 3 allora saranno restituite un massimo di 3 elementi dal layer1. Allo stesso modo verrà restituito un massimo di 3 elementi dal layer2.

Per impostazione predefinita, viene restituito solo 1 elemento per layer.

I

Questo parametro, definito in WMS 1.3.0, permette di specificare la colonna pixel del punto di interrogazione.

X

Stesso parametro di `I`, ma definito in WMS 1.1.1.

J

Questo parametro, definito in WMS 1.3.0, permette di specificare la riga pixel del punto di interrogazione.

Y

Stesso parametro di J, ma definito in WMS 1.1.1.

FI_POINT_TOLERANCE

Questo parametro specifica la tolleranza in pixel per i layer punto.

FI_LINE_TOLERANCE

Questo parametro specifica la tolleranza in pixel per i layer linea.

FI_POLYGON_TOLERANCE

Questo parametro specifica la tolleranza in pixel per i layer poligono.

FILTER_GEOM

Questo parametro specifica una geometria WKT con cui gli elementi devono intersecarsi.

WITH_MAPTIP

Questo parametro specifica se aggiungere suggerimenti mappa al risultato.

I valori disponibili sono (non sensibili alle maiuscole):

- TRUE
- FALSE

WITH_GEOMETRY

Questo parametro specifica se aggiungere geometrie al risultato. Per usare questa funzione devi prima abilitare l'opzione *Aggiungi la geometria alla risposta dell'oggetto* nel progetto QGIS. Vedi *Configure your project*.

I valori disponibili sono (non sensibili alle maiuscole):

- TRUE
- FALSE

3.1.3 GetPrint

QGIS Server ha la capacità di creare output di layout di stampa in formato pdf o pixel. Le finestre di layout di stampa nel progetto pubblicato sono usate come modelli. Nella richiesta **GetPrint**, il client ha la possibilità di specificare i parametri delle mappe e delle etichette del layout contenuto.

Parametri per la richiesta **GetPrint**:

Parametro	Richiesto	Descrizione
<i>MAP</i>	Si	Specifica del file di progetto QGIS
<i>SERVICE</i>	Si	Nome del servizio (WMS)
<i>VERSION</i>	No	Versione del servizio
<i>REQUEST</i>	Si	Nome della richiesta (GetPrint)
<i>LAYERS</i>	No	Layer da visualizzare
<i>TEMPLATE</i>	Si	Modello di layout da utilizzare
<i>SRS / CRS</i>	Si	Sistema di riferimento delle Coordinate
<i>FORMAT</i>	No	Formato in uscita
<i>ATLAS_PK</i>	No	Caratteristiche dell'Atlante
<i>STYLES</i>	No	Stile del layers
<i>TRANSPARENT</i>	No	Sfondo trasparente
<i>OPACITIES</i>	No	Opacità per il layer o il gruppo
<i>SELECTION</i>	No	Elementi principali
<i>mapX:EXTENT</i>	No	Estensione della mappa "X"
<i>mapX:LAYERS</i>	No	Layer della mappa "X"
<i>mapX:STYLES</i>	No	Stile dei layer della mappa "X"
<i>mapX:SCALE</i>	No	Scala dei layer della mappa "X"
<i>mapX:ROTATION</i>	No	Rotazione della mappa "X"
<i>mapX:GRID_INTERVAL_X</i>	No	Intervallo del reticolo sull'asse x della mappa "X"
<i>mapX:GRID_INTERVAL_Y</i>	No	Intervallo del reticolo sull'asse y della mappa "Y"

Esempio di URL:

```
http://localhost/qgisserver?
SERVICE=WMS
&VERSION=1.3.0
&REQUEST=GetPrint
&MAP=/home/qgis/projects/world.qgs
&CRS=EPSG:4326
&FORMAT=png
&TEMPLATE=Layout%201
&map0:EXTENT=-180,-90,180,90
&map0:LAYERS=mylayer1,mylayer2,mylayer3
&map0:OPACITIES=125,200,125
&map0:ROTATION=45
```

Nota che il modello di layout può contenere più di una mappa. In questo modo, se vuoi configurare una mappa specifica, devi usare i parametri mappaX: dove X è un numero positivo che puoi recuperare grazie alla richiesta **GetProjectSettings**.

Per esempio:

```
<WMS_Capabilities>
...
<ComposerTemplates xsi:type="wms:_ExtendedCapabilities">
<ComposerTemplate width="297" height="210" name="Druckzusammenstellung 1">
<ComposerMap width="171" height="133" name="map0"/>
<ComposerMap width="49" height="46" name="map1"/></ComposerTemplate>
</ComposerTemplates>
...
</WMS_Capabilities>
```

REQUEST

Questo parametro deve essere `GetPrint` per la richiesta **GetPrint**.

TEMPLATE

Questo parametro può essere utilizzato per specificare il nome di un modello di layout da utilizzare per la stampa.

FORMAT

Questo parametro specifica il formato dell'immagine della mappa. I valori disponibili sono:

- `png` (default value)
- `image/png`
- `jpg`
- `jpeg`
- `image/jpeg`
- `svg`
- `image/svg`
- `image/svg+xml`
- `pdf`
- `application/pdf`

Se il parametro `FORMAT` è diverso da uno di questi valori, allora viene restituita una eccezione.

ATLAS_PK

Questo parametro permette di attivare la visualizzazione dell'atlante indicando quali caratteristiche vogliamo stampare. Per ottenere un atlante con tutte gli elementi, si può usare il simbolo `*` (secondo il numero massimo di elementi consentito nella configurazione del progetto).

Quando il `FORMAT` è `pdf`, viene restituito un singolo documento PDF che combina le pagine degli elementi. Per tutti gli altri formati, viene restituita una singola pagina.

mapX:EXTENT

Questo parametro specifica l'estensione per un oggetto della mappa di layout come `xmin,ymin,xmax,ymax`.

mapX:ROTATION

Questo parametro specifica la rotazione della mappa in gradi.

mapX:GRID_INTERVAL_X

Questo parametro specifica la densità delle linee del reticolo nella direzione X.

mapX:GRID_INTERVAL_Y

Questo parametro specifica la densità delle linee del reticolo nella direzione Y.

mapX:SCALE

Questo parametro specifica la scala della mappa per un oggetto mappa del layout. Questo è utile per assicurare la visibilità dei layer e delle etichette in base alla scala, anche se client e server possono avere algoritmi diversi per calcolare il denominatore della scala.

mapX:LAYERS

Questo parametro specifica i layer per un elemento della mappa del layout. Vedi *GetMap Layers* per maggiori informazioni su questo parametro.

mapX:STYLES

Questo parametro specifica gli stili dei layer definiti in uno specifico oggetto della mappa di layout. Vedi *GetMap Styles* per maggiori informazioni su questo parametro.

3.1.4 GetLegendGraphics

Sono disponibili diversi parametri aggiuntivi per cambiare la dimensione degli elementi della legenda:

- **BOXSPACE** spazio tra la cornice della legenda e il contenuto (mm)
- **FORMAT**, *image/jpeg*, *image/png* or *application/json*. For JSON, symbols are encoded with Base64 and most other options related to layout or fonts are not taken into account because the legend must be built on the client side.
- **LAYERSPACE** spazio verticale tra i layer (mm)
- **LAYERTITLESPACE** spazio verticale tra il titolo del layer e le voci seguenti (mm)
- **SYMBOLSPACE** spazio verticale tra il simbolo e la voce seguente (mm)
- **ICONLABELSPACE** Spazio orizzontale tra il simbolo e il testo dell'etichetta (mm)
- **SYMBOLWIDTH** larghezza dell'anteprima del simbolo (mm)
- **SYMBOLHEIGHT** altezza dell'anteprima del simbolo (mm)

Questi parametri cambiano le proprietà del carattere per i titoli dei layer e le etichette degli oggetti:

- **LAYERFONTFAMILY / ITEMFONTFAMILY** famiglia di caratteri per il titolo del layer / testo dell'articolo
- **LAYERFONTBOLD / ITEMFONTBOLD** TRUE per usare un carattere in grassetto
- **LAYERFONTSIZE / ITEMFONTSIZE** Dimensione carattere in punti
- **LAYERFONTITALIC / ITEMFONTITALIC** TRUE per usare il carattere corsivo
- **LAYERFONTCOLOR / ITEMFONTCOLOR** Codice colore esadecimale (ad esempio #FF0000 per il rosso)
- **LAYERTITLE** FALSE per ottenere solo la grafica della legenda senza il titolo del layer.

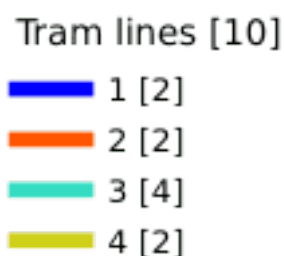
- **RULELABEL:**
 - FALSE grafica della legenda senza etichette degli elementi
 - AUTO nasconde l'etichetta dell'elemento per i layer con rappresentazione *Simbolo singolo*

Legenda basata sul contenuto. Questi parametri permettono al client di richiedere una legenda che mostri solo i simboli degli elementi che rientrano nell'area richiesta:

- **BBOX** l'area geografica per la quale la legenda dovrebbe essere costruita
- **CRS / SRS** il sistema di riferimento delle coordinate adottato per definire le coordinate della BBOX
- **SRCWIDTH / SRCHEIGHT** se impostati dovrebbero corrispondere ai parametri WIDTH e HEIGHT della richiesta GetMap, per permettere a QGIS Server di scalare i simboli secondo le dimensioni dell'immagine della mappa.

Gli elementi delle legende basate sul contenuto sono basate sull'implementazione di [UMN MapServer](#):

- **SHOWFEATURECOUNT** se impostato a TRUE aggiunge nella legenda il conteggio degli elementi come nell'immagine seguente:



- **RULE** impostarlo su un dato nome di regola per ottenere solo il simbolo della regola nominata
- **WIDTH/HEIGHT** la dimensione dell'immagine della legenda generata se il parametro **RULE** è impostato

3.1.5 GetProjectSettings

Questo tipo di richiesta funziona in modo simile a **GetCapabilities**, ma è più specifico per QGIS Server e permette ad un client di leggere informazioni aggiuntive che non sono disponibili nell'output di **GetCapabilities**:

- visibilità iniziale dei layer
- informazione sugli attributi vettoriali e i loro tipi di edit
- informazione sull'ordine dei layer e l'ordine di rappresentazione
- elenco dei layer pubblicati in WFS
- mostra se un gruppo nell'albero dei layer è mutuamente esclusiva

3.2 Web Feature Service (WFS)

Gli standard **1.0.0** e **1.1.0** WFS implementati in QGIS Server forniscono un'interfaccia HTTP per interrogare gli elementi geografici da un progetto QGIS. Una tipica richiesta WFS definisce il progetto QGIS da usare e il layer da interrogare.

Documento delle specifiche secondo il numero di versione del servizio:

- [WFS 1.0.0](#)
- [WFS 1.1.0](#)

Richieste standard fornite da QGIS Server:

Request	Descrizione
GetCapabilities	Restituisce metadati XML con informazioni sul server
GetFeature	Restituisce una selezione di elementi
DescribeFeatureType	Restituisce una descrizione dei tipi di elementi e delle proprietà
Transazione	Permette l'inserimento, l'aggiornamento o la cancellazione di elementi

3.2.1 GetFeature

Parametri standard per la richiesta **GetFeature** secondo le specifiche OGC WFS 1.0.0 e 1.1.0:

Parametro	Richiesto	Descrizione
SERVICE	Sì	Nome del servizio
VERSION	No	Versione del servizio
REQUEST	Sì	Nome della richiesta
TYPENAME	No	Nome dei layer
OUTPUTFORMAT	No	Formato risultato
RESULTTYPE	No	Tipo di risultato
PROPERTYNAME	No	Nome delle proprietà da restituire
MAXFEATURES	No	Numero massimo di elementi da restituire
SRSNAME	No	Sistema di riferimento delle Coordinate
FEATUREID	No	Filtrare gli elementi per id
FILTER	No	Codifica filtro OGC
BBOX	No	Estensione mappa
SORTBY	No	Ordinare i risultati

Oltre a quelli standard, QGIS Server supporta i seguenti parametri extra:

Parametro	Richiesto	Descrizione
MAP	Sì	Specifica del file di progetto QGIS
STARTINDEX	No	Paginazione
GEOMETRYNAME	No	Tipo di geometria da restituire
EXP_FILTER	No	Espressione di filtraggio

SERVICE

Questo parametro deve essere WFS nel caso della richiesta **GetFeature**.

Per esempio:

```
http://localhost/qgisserver?
SERVICE=WFS
&...
```

VERSION

Questo parametro permette di specificare la versione del servizio da utilizzare. I valori disponibili per il parametro VERSION sono:

- 1.0.0
- 1.1.0

Se nessuna versione è indicata nella richiesta, allora 1.1.0 è usato per default.

Esempio di URL:

```
http://localhost/qgisserver?  
SERVICE=WFS  
&VERSION=1.1.0  
&...
```

REQUEST

Questo parametro è `GetFeature` nel caso della richiesta **GetFeature**.

Esempio di URL:

```
http://localhost/qgisserver?  
SERVICE=WFS  
&VERSION=1.1.0  
&REQUEST=GetFeature  
&...
```

RESULTTYPE

Questo parametro può essere usato per specificare il tipo di risultato da restituire. I valori disponibili sono:

- `results`: il risultato predefinito
- `hits`: restituisce solo un conteggio degli elementi

Esempio di URL:

```
http://localhost/qgisserver?  
SERVICE=WFS  
&VERSION=1.1.0  
&REQUEST=GetFeature  
&RESULTTYPE=hits  
&...
```

GEOMETRYNAME

Questo parametro può essere usato per specificare il tipo di geometria da restituire per gli elementi. I valori disponibili sono:

- `extent`
- `centroid`
- `none`

Esempio di URL:

```
http://localhost/qgisserver?
SERVICE=WFS
&VERSION=1.1.0
&REQUEST=GetFeature
&GEOMETRYNAME=centroid
&...
```

STARTINDEX

Questo parametro è standard in WFS 2.0, ma è un'estensione per WFS 1.0.0. Attualmente, può essere usato per ignorare alcuni elementi nell'insieme dei risultati e in combinazione con MAXFEATURES, fornisce la possibilità di sfogliare i risultati.

Esempio di URL:

```
http://localhost/qgisserver?
SERVICE=WFS
&VERSION=1.1.0
&REQUEST=GetFeature
&STARTINDEX=2
&...
```

3.3 Web Map Tile Service (WMTS)

Lo standard **1.0.0** WMTS implementato in QGIS Server fornisce un'interfaccia HTTP per richiedere immagini di mappe tassellate generate da un progetto QGIS. Una tipica richiesta WMTS definisce il progetto QGIS da usare, alcuni parametri WMS come i layer da rappresentare, così come i parametri dei tasselli.

Documento di specifiche del servizio:

- [WMTS 1.0.0](#)

Richieste standard fornite da QGIS Server:

Request	Descrizione
GetCapabilities	Restituisce metadati XML con informazioni sul server
GetTile	Restituisce un tassello
GetFeatureInfo	Recupera i dati (geometria e valori) per una posizione di pixel

3.3.1 GetCapabilities

Parametri standard per la richiesta **GetCapabilities** secondo le specifiche OGC WMTS 1.0.0:

Parametro	Richiesto	Descrizione
SERVICE	Sì	Nome del servizio (WMTS)
REQUEST	Sì	Nome della richiesta (GetCapabilities)

Oltre a quelli standard, QGIS Server supporta i seguenti parametri extra:

Parametro	Richiesto	Descrizione
MAP	Sì	Specifica del file di progetto QGIS

Esempio di URL:

```
http://localhost/qgisserver?  
SERVICE=WMTS  
&REQUEST=GetCapabilities  
&MAP=/home/qgis/projects/world.qgs
```

SERVICE

Questo parametro deve essere `WMTS` nel caso della richiesta **GetCapabilities**.

REQUEST

Questo parametro è `GetCapabilities` nel caso della richiesta **GetCapabilities**.

MAP

Questo parametro permette di definire il file di progetto QGIS da utilizzare.

3.3.2 GetTile

Parametri standard per la richiesta **GetTile** secondo le specifiche OGC WMTS 1.0.0:

Parametro	Richiesto	Descrizione
SERVICE	Sì	Nome del servizio (WMTS)
REQUEST	Sì	Nome della richiesta (GetTile)
LAYER	Sì	Identificatore layer
FORMAT	Sì	Formato in uscita del tassello
TILEMATRIXSET	Sì	Nome della piramide
TILEMATRIX	Sì	Meshing
TILEROW	Sì	Coordinata riga nella mesh
TILECOL	Sì	Coordinata colonna nella mesh

Oltre a quelli standard, QGIS Server supporta i seguenti parametri extra:

Parametro	Richiesto	Descrizione
MAP	Sì	Specifica del file di progetto QGIS

Esempio di URL:

```
http://localhost/qgisserver?  
SERVICE=WMTS  
&REQUEST=GetTile  
&MAP=/home/qgis/projects/world.qgs  
&LAYER=mylayer  
&FORMAT=image/png  
&TILEMATRIXSET=EPSG:4326  
&TILEROW=0  
&TILECOL=0
```

SERVICE

Questo parametro deve essere WMTS nel caso della richiesta **GetTile**.

REQUEST

Questo parametro è `GetTile` nel caso della richiesta **GetTile**.

LAYER

Questo parametro permette di specificare il layer da visualizzare sul tassello

Inoltre, QGIS Server ha introdotto alcune opzioni per selezionare un layer:

- un nome breve
- il layer id

Il nome breve di un layer può essere configurato attraverso `:menuselection: Proprietà -> Metadati` nel menu del layer. Se il nome breve è definito, allora viene usato come predefinito al posto del nome del layer:

```
http://localhost/qgisserver?  
SERVICE=WMTS  
&REQUEST=GetTile  
&LAYER=mynickname  
&...
```

Inoltre, c'è un'opzione del progetto che permette di selezionare i layer in base al loro id nel menu *OWS Server ► WMS capabilities del servizio* della finestra di dialogo *Progetto ► Proprietà...* Per attivare questa opzione, la casella di controllo *Usa gli id del livello come nomi* deve essere selezionata.

```
http://localhost/qgisserver?  
SERVICE=WMTS  
&REQUEST=GetTile  
&LAYER=mylayerid1  
&...
```

FORMAT

Questo parametro può essere usato per specificare il formato dell'immagine del tassello. I valori disponibili sono:

- `jpg`
- `jpeg`
- `image/jpeg`
- `image/png`

Se il parametro `FORMAT` è diverso da uno di questi valori, allora viene usato il formato predefinito PNG.

TILEMATRIXSET

Questo parametro definisce il SR da utilizzare nel calcolo della piramide sottostante. Formato: EPSG:XXXX.

TILEMATRIX

Questo parametro permette di definire la matrice da utilizzare per il tassello in uscita.

TILEROW

Questo parametro permette di selezionare la riga del tassello da ottenere all'interno della matrice.

TILECOL

Questo parametro permette di selezionare la colonna del tassello da ottenere all'interno della matrice.

MAP

Questo parametro permette di definire il file di progetto QGIS da utilizzare.

Come menzionato in *GetMap parameters table*, MAP è obbligatorio perché una richiesta ha bisogno di un progetto QGIS per funzionare effettivamente. Tuttavia, la variabile d'ambiente QGIS_PROJECT_FILE può essere usata per definire un progetto QGIS predefinito. In questo caso specifico, MAP non è più un parametro richiesto. Per ulteriori informazioni puoi fare riferimento a *Configurazione avanzata*.

3.3.3 GetFeatureInfo

Parametri standard per la richiesta **GetFeatureInfo** secondo la specifica OGC WMTS 1.0.0:

Parametro	Richiesto	Descrizione
SERVICE	Sì	Nome del servizio (WMTS)
REQUEST	Sì	Nome della richiesta (GetFeatureInfo)
LAYER	Sì	Identificatore layer
INFOFORMAT	No	Formato in uscita
I	No	Coordinata X di un pixel
J	No	Coordinata Y di un pixel
TILEMATRIXSET	Sì	<i>See GetTile</i>
TILEMATRIX	Sì	<i>See GetTile</i>
TILEROW	Sì	<i>See GetTile</i>
TILECOL	Sì	<i>See GetTile</i>

Oltre a quelli standard, QGIS Server supporta i seguenti parametri extra:

Parametro	Richiesto	Descrizione
MAP	Sì	Specifica del file di progetto QGIS

Esempio di URL:

```
http://localhost/qgisserver?
SERVICE=WMTS
&REQUEST=GetFeatureInfo
&MAP=/home/qgis/projects/world.qgs
&LAYER=mylayer
```

(continues on next page)

(continua dalla pagina precedente)

```
&INFOFORMAT=image/html
&I=10
&J=5
```

SERVICE

Questo parametro deve essere `WMTS` nel caso della richiesta `GetFeatureInfo`.

REQUEST

Questo parametro è `GetFeatureInfo` nel caso della richiesta `GetFeatureInfo`.

MAP

Questo parametro permette di definire il file di progetto QGIS da utilizzare.

Come menzionato in *GetMap parameters table*, `MAP` è obbligatorio perché una richiesta ha bisogno di un progetto QGIS per funzionare effettivamente. Tuttavia, la variabile d'ambiente `QGIS_PROJECT_FILE` può essere usata per definire un progetto QGIS predefinito. In questo caso specifico, `MAP` non è più un parametro richiesto. Per ulteriori informazioni puoi fare riferimento a *Configurazione avanzata*.

LAYER

Questo parametro permette di specificare il layer da visualizzare sul tassello

Inoltre, QGIS Server ha introdotto alcune opzioni per selezionare un layer:

- un nome breve
- il layer id

Il nome breve di un layer può essere configurato attraverso `:menuselection: Proprietà -> Metadati` nel menu del layer. Se il nome breve è definito, allora viene usato come predefinito al posto del nome del layer:

```
http://localhost/qgisserver?
SERVICE=WMTS
&REQUEST=GetFeatureInfo
&LAYER=mynickname
&...
```

Inoltre, c'è un'opzione del progetto che permette di selezionare i layer in base al loro id nel menu *OWS Server ► WMS capabilities del servizio* della finestra di dialogo *Progetto ► Proprietà...* Per attivare questa opzione, la casella di controllo *Usa gli id del livello come nomi* deve essere selezionata.

```
http://localhost/qgisserver?
SERVICE=WMTS
&REQUEST=GetFeatureInfo
&LAYER=mylayerid1
&...
```

INFOFORMAT

Questo parametro permette di definire il formato in uscita del risultato. I valori disponibili sono:

- `text/xml`
- `text/html`
- `text/plain`
- `application/vnd.ogc.gml`

Il valore predefinito è `text/plain`.

I

Questo parametro permette di definire la coordinata X del pixel per il quale vogliamo recuperare l'informazione sottostante.

J

Questo parametro permette di definire la coordinata Y del pixel per il quale vogliamo recuperare l'informazione sottostante.

3.4 WFS3 (funzioni API OGC)

WFS3 è la prima implementazione della nuova generazione di protocolli OGC. È descritto dal documento [OGC API - Features - Part 1: Core](#)

Di seguito un rapido riassunto informale delle differenze più importanti tra il ben noto protocollo WFS e WFS3:

- WFS3 è basato su un'API REST
- WFS3 API deve seguire le specifiche OPENAPI
- WFS3 supporta più formati di output ma non ne impone nessuno (solo GeoJSON e HTML sono attualmente disponibili in QGIS WFS3) e usa la [negoiazione dei contenuti](#) per determinare quale formato deve essere servito al client
- JSON e HTML sono cittadini di prima classe in WFS3
- WFS3 è autodocumentato (attraverso l'endpoint `/api`)
- WFS3 è completamente navigabile (tramite link) e consultabile.

Importante: Mentre l'implementazione WFS3 in QGIS può fare uso del parametro `MAP` per specificare il file di progetto, la specifica OPENAPI non consente parametri di query aggiuntivi. Per questo motivo si raccomanda vivamente che `MAP` non sia esposto nell'URL e che il file di progetto sia specificato nell'ambiente con altri mezzi (ad esempio impostando `QGIS_PROJECT_FILE` nell'ambiente tramite una regola di riscrittura del server web).

Nota: L'endpoint API fornisce una documentazione completa di tutti i parametri supportati e dei formati in uscita del tuo servizio. I paragrafi seguenti descriveranno solo i più importanti.

3.4.1 Rappresentazione risorse

L'implementazione di QGIS Server WFS3 attualmente supporta i seguenti formati di rappresentazione delle risorse (output):

- HTML
- JSON

Il formato che viene effettivamente reso dipenderà dalla negoziazione del contenuto, ma un formato specifico può essere richiesto esplicitamente aggiungendo uno specifico formato agli endpoint.

Le estensioni di specifica del formato supportate sono:

- .json
- .html

Ulteriori alias di specifica di formato possono essere definiti da endpoint specifici:

- .openapi: alias per .json supportato dall'endpoint **API**
- .geojson: alias per .json supportato dagli endpoint **Features** e **Feature**

3.4.2 Endpoint

L'API fornisce una lista di endpoint che i client possono richiamare. Il sistema è progettato in modo tale che ogni risposta fornisce un insieme di link per navigare attraverso tutte le risorse fornite.

Punti endpoint forniti dall'implementazione di QGIS sono:

Nome	Percorso	Descrizione
Pagina di destinazione	/	Informazioni generali sul servizio e fornisce collegamenti a tutti gli endpoint disponibili
Conformità	/conformance	Informazioni sulla conformità del servizio agli standard
API	/api	Descrizione completa degli endpoint forniti dal servizio e della struttura dei documenti restituiti
Collezioni	/collections	Elenco di tutte le collezioni (cioè » layer vettoriali») fornite dal servizio
Collezione	/collections/{collectionId}	Informazioni su una collezione (nome, metadati, estensione ecc.)
Caratteristiche	/collections/{collectionId}/items	Elenco degli elementi forniti dalla collezione
Elemento	/collections/{collectionId}/items/{featureId}	Informazioni su un singolo elemento

Pagina di destinazione

L'endpoint principale è la **Landing Page**. Da questa pagina è possibile navigare verso tutti gli endpoint di servizio disponibili. La **Landing Page** deve fornire collegamenti a

- la definizione dell'API (percorso /api link delle relazioni `service-desc` e `service-doc`),
- la dichiarazione di Conformità (percorso /conformance, relazione di collegamento `conformance`), e
- le Collezioni (percorso /collections, relazione di collegamento `data`).

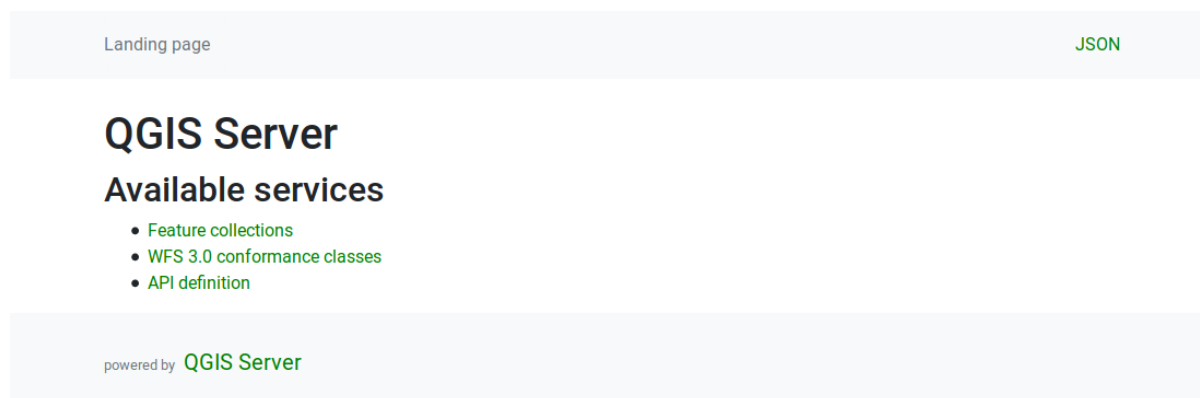


Fig. 3.2: Landing page del server WFS3

Definizione API

La **Definizione dell'API** è una descrizione conforme a OPENAPI dell'API fornita dal servizio. Nella sua rappresentazione HTML è una pagina navigabile dove tutti gli endpoint e i loro formati di risposta sono accuratamente elencati e documentati. Il percorso di questo endpoint è `/api`.

La definizione API fornisce una documentazione completa e autorevole del servizio, compresi tutti i parametri supportati e i formati restituiti.

Nota: Questo endpoint è analogo a `GetCapabilities` di WFS

Lista Collezioni

L'endpoint collezioni fornisce una lista di tutte le collezioni disponibili nel servizio. Poiché il servizio «serve» un singolo progetto QGIS, le collezioni sono i layer vettoriali del progetto corrente (se sono stati pubblicati come WFS nelle proprietà del progetto). Il percorso di questo endpoint è `/collections/`.

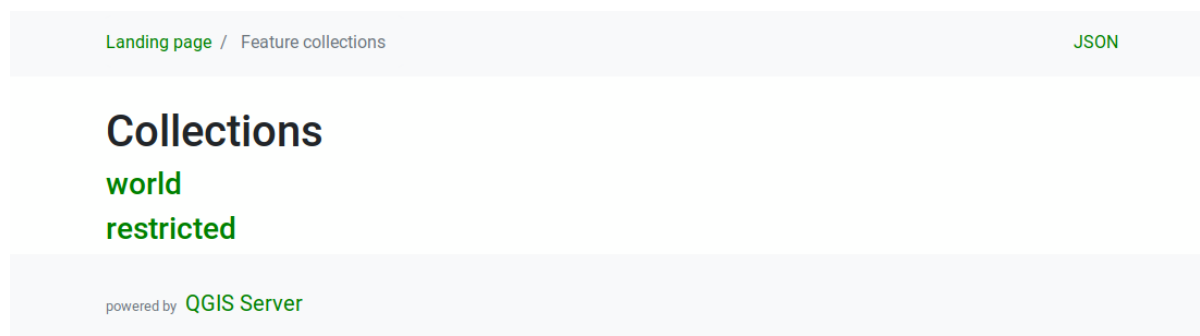


Fig. 3.3: Pagina dell'elenco delle collezioni del server WFS3

Dettaglio collezione

Mentre l'endpoint delle collezioni non fornisce informazioni dettagliate su ogni collezione disponibile, queste informazioni sono disponibili negli endpoint `/collections/{collectionId}`. Le informazioni tipiche includono l'estensione, una descrizione, i SR e altri metadati.

La rappresentazione HTML fornisce anche una mappa navigabile con gli elementi disponibili.

The screenshot shows a web page for a collection named 'world'. At the top, there is a breadcrumb trail: 'Landing page / Collections / world' and a 'JSON' link. The main heading is 'world' in green. Below it, the section 'Available CRSs' lists four URLs: <http://www.opengis.net/def/crs/OGC/1.3/CRS84>, <http://www.opengis.net/def/crs/EPSSG/9.6.2/4326>, <http://www.opengis.net/def/crs/EPSSG/9.6.2/3857>, and <http://www.opengis.net/def/crs/EPSSG/9.6.2/900913>. The 'Extent' section shows the following values: West: -179.99999999999997, South: -90.0, East: 180.0, North: 83.62359600000008. To the right of the text is a map of the world with a blue rectangular bounding box around the central region. The map includes a zoom control and a 'Leaflet | © OpenStreetMap contributors' footer. At the bottom of the page, it says 'powered by QGIS Server'.

Fig. 3.4: Pagina di dettaglio della collezione WFS3 del server

Lista elementi

Questo endpoint fornisce un elenco di tutti gli elementi in una collezione conoscendo l'ID della collezione. Il percorso di questo endpoint è `/collections/{collectionId}/items`.

La rappresentazione HTML fornisce anche una mappa navigabile con gli elementi disponibili.

Nota: Questo endpoint è analogo a `GetFeature` in WFS 1 e WFS 2.

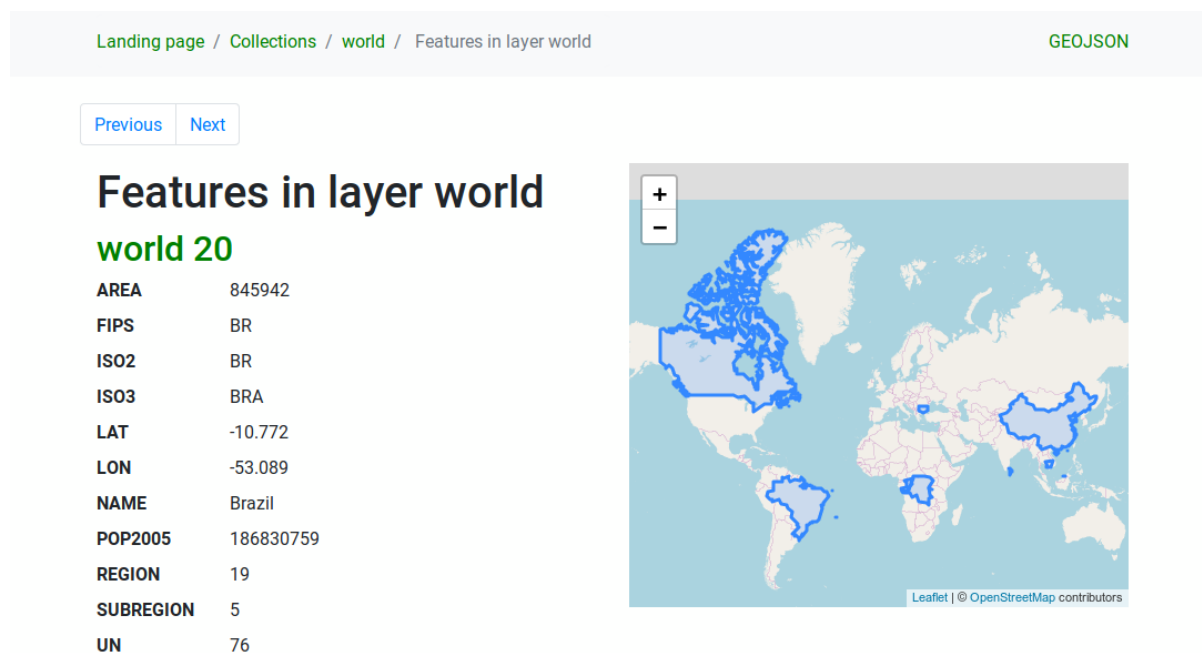


Fig. 3.5: Pagina dell'elenco degli elementi del server WFS3

Dettaglio elemento

Questo endpoint fornisce tutte le informazioni disponibili su un singolo elemento, compresi gli attributi dell'elemento e la sua geometria. Il percorso di questo endpoint è `/collections/{collectionId}/items/{itemId}`.

La rappresentazione HTML fornisce anche una mappa navigabile con la geometria degli elementi.



Fig. 3.6: Pagina di dettaglio elementi server WFS3

3.4.3 Paginazione

La paginazione di una lunga lista di elementi è implementata nell'API OGC attraverso i link `next`` e `prev`, il server QGIS costruisce questi link aggiungendo `limit` e `offset` come parametri della query string.

Esempio di URL:

```
http://localhost/qgisserver/wfs3/collection_one/items.json?offset=10&limit=10
```

Nota: Il valore massimo accettabile per `limit` può essere configurato con l'impostazione di configurazione del server `QGIS_SERVER_API_WFS3_MAX_LIMIT` (vedi: `qgis-server-envvar``).

3.4.4 Filtro delle geometrie

Gli elementi disponibili in una collezione possono essere filtrati/ricercati specificando uno o più filtri.

Filtro data e ora

Le collezioni con attributi `data` e/o `datetime` possono essere filtrate specificando un argomento `datetime` nella stringa della query. Per impostazione predefinita, il primo campo `data/ora` è usato per il filtraggio. Questo comportamento può essere configurato impostando una dimensione «Date» o «Time» nella sezione *QGIS Server* ► *Dimension* della finestra di dialogo delle proprietà del layer.

La sintassi del filtraggio di data e ora è completamente descritta nella *Definizione API* e supporta anche intervalli (i valori iniziali e finali sono inclusi) oltre a valori singoli.

Esempi URL:

Restituisce solo gli elementi con una dimensione di data corrispondente a 2019-01-01.

```
http://localhost/qgisserver/wfs3/collection_one/items.json?datetime=2019-01-01
```

Restituisce solo gli elementi con dimensione `datetime` che corrisponde a 2019-01-01T01:01:01.

```
http://localhost/qgisserver/wfs3/collection_one/items.json?datetime=2019-01-
↪01T01:01:01
```

Restituisce solo gli elementi con dimensione `datetime` nell'intervallo 2019-01-01T01:01:01 - 2019-01-01T12:00:00.

```
http://localhost/qgisserver/wfs3/collection_one/items.json?datetime=2019-01-
↪01T01:01:01/2019-01-01T12:00:00
```

Filtro rettangolo di delimitazione

Un filtro spaziale `bounding box` può essere specificato con il parametro `bbox`:

L'ordine degli elementi separati da virgola è:

- Angolo inferiore sinistro, longitudine WGS 84
- Angolo inferiore sinistro, latitudine WGS 84
- Angolo superiore destro, longitudine WGS 84
- Angolo superiore destro, latitudine WGS 84

Nota: Le specifiche OGC permettono anche uno specifico bbox a 6 elementi dove il terzo e il sesto elemento sono i componenti Z, questo non è ancora supportato dal server QGIS.

Esempio di URL:

```
http://localhost/qgisserver/wfs3/collection_one/items.json?bbox=-180,-90,180,90
```

Se il SR del riquadro di delimitazione non è WGS 84, un CRS diverso può essere specificato usando il parametro opzionale `bbox-crs`. L'identificatore del formato SR deve essere nel formato OGC URI:

Esempio di URL:

```
http://localhost/qgisserver/wfs3/collection_one/items.json?bbox=913191,5606014,↵913234,5606029&bbox-crs=http://www.opengis.net/def/crs/EPSG/9.6.2/3857
```

Filtro Attributo

I filtri degli attributi possono essere combinati con il filtro bounding box e sono nella forma generale: `<attribute name>=<attribute value>`. Filtri multipli possono essere combinati usando l'operatore AND `

Esempio di URL:

filtra tutti gli elementi in cui l'attributo `name` è uguale a «`my value`»

```
http://localhost/qgisserver/wfs3/collection_one/items.json?attribute_one=my%20value
```

Le corrispondenze parziali sono supportate anche usando un operatore `*` («star»):

Esempio di URL:

filtra tutti gli elementi in cui l'attributo `name` finisce con «`value`»

```
http://localhost/qgisserver/wfs3/collection_one/items.json?attribute_one=*value
```

3.4.5 Ordinamento elemento

È possibile ordinare l'insieme dei risultati per valore di campo usando il parametro della query `orderby`.

I risultati sono ordinati in ordine crescente per impostazione predefinita. Per ordinare i risultati in ordine decrescente, si può impostare un flag booleano (`sortdesc`):

```
http://localhost/qgisserver/wfs3/collection_one/items.json?orderby=name&sortdesc=1
```

3.4.6 Selezione attributo

Gli attributi elementari restituiti da una chiamata *Lista elementi* possono essere limitati aggiungendo una lista separata da virgole di nomi di attributi nell'argomento opzionale `properties` della query string.

Esempio di URL:

restituisce solo l'attributo `name`.

```
http://localhost/qgisserver/wfs3/collection_one/items.json?properties=name
```

3.4.7 Personalizzare le pagine HTML

La rappresentazione HTML usa un insieme di modelli HTML per generare la risposta. Il template viene analizzato da un motore di template chiamato *inja*. I template possono essere personalizzati sovrascrivendoli (vedi: *Sovrascrittura modello*). Il template ha accesso agli stessi dati che sono disponibili alla rappresentazione JSON e alcune funzioni aggiuntive sono disponibili al template:

Funzioni personalizzate modello

- `path_append(path)`: aggiunge un percorso di cartella all'url corrente
- `path_chomp(n)`: rimuove il numero specificato «n» di componenti di cartella dal percorso url corrente
- `json_dump()`: stampa i dati JSON passati al modello
- `static(path)`: restituisce l'URL completo del percorso statico specificato. Per esempio: «static(«/style/black.css»)» con un percorso radice «http://localhost/qgisserver/wfs3» restituirà «http://localhost/qgisserver/wfs3/static/style/black.css».
- `links_filter(links, key, value)`: Restituisce i link filtrati da un elenco di link
- `content_type_name(content_type)`: Restituisce un nome breve di un tipo di contenuto, per esempio «text/html» restituirà «HTML»

Sovrascrittura modello

I modelli e le risorse statiche sono memorizzati nelle sottocartelle della cartella delle risorse API di default del server QGIS (/usr/share/qgis/resources/server/api/ su un sistema Linux), la cartella di base può essere personalizzata cambiando la variabile di ambiente QGIS_SERVER_API_RESOURCES_DIRECTORY.

Una tipica installazione di Linux avrà il seguente albero di cartelle:

```

/usr/share/qgis/resources/server/api/
├── ogc
│   ├── schema.json
│   ├── static
│   │   ├── jsonFormatter.min.css
│   │   ├── jsonFormatter.min.js
│   │   └── style.css
│   └── templates
│       └── wfs3
│           ├── describeCollection.html
│           ├── describeCollections.html
│           ├── footer.html
│           ├── getApiDescription.html
│           ├── getFeature.html
│           ├── getFeatures.html
│           ├── getLandingPage.html
│           ├── getRequirementClasses.html
│           ├── header.html
│           ├── leaflet_map.html
│           └── links.html

```

Per sovrascrivere i modelli puoi copiare l'intero albero in un'altra posizione e assegnare QGIS_SERVER_API_RESOURCES_DIRECTORY alla nuova posizione.

3.5 Parametri extra supportati da tutti i tipi di richiesta

I seguenti parametri extra sono supportati da tutti i protocolli.

- **FILE_NAME**: se impostato, la risposta del server sarà inviata al client come un file allegato con il nome del file specificato.

Nota: Non disponibile per WFS3.

- **MAP**: Analogamente a MapServer, il parametro MAP può essere usato per specificare il percorso del file del progetto QGIS. Puoi specificare un percorso assoluto o un percorso relativo per la posizione dell'eseguibile del server (`qgis_mapserv.fcgi`). Se non specificato, QGIS Server cerca i file .qgs nella cartella dove si trova l'eseguibile del server.

Esempio:

```
http://localhost/cgi-bin/qgis_mapserv.fcgi?\
REQUEST=GetMap&MAP=/home/qgis/projects/world.qgs&...
```

Nota: Puoi definire un **QGIS_PROJECT_FILE** come variabile d'ambiente per dire all'eseguibile del server dove trovare il file di progetto QGIS. Questa variabile sarà la posizione in cui QGIS cercherà il file di progetto. Se non è definita, userà il parametro MAP nella richiesta e infine cercherà la cartella eseguibile del server.

3.6 REDLINING

Questa funzione è disponibile e può essere usata con le richieste `GetMap` e `GetPrint`.

La funzione `redlining` può essere usata per passare geometrie ed etichette nella richiesta che vengono sovrapposte dal server sull'immagine standard restituita (mappa). Questo permette all'utente di mettere l'accento o magari aggiungere dei commenti (etichette) ad alcune aree, luoghi ecc. che non sono nella mappa standard.

La richiesta `GetMap` è nel formato:

```
http://qgisplatform.demo/cgi-bin/qgis_mapserv.fcgi?map=/world.qgs&SERVICE=WMS&
↳VERSION=1.3.0&
REQUEST=GetMap
...
&HIGHLIGHT_GEOM=POLYGON((590000 5647000, 590000 6110620, 2500000 6110620, 2500000.
↳5647000, 590000 5647000))
&HIGHLIGHT_SYMBOL=<StyledLayerDescriptor><UserStyle><Name>Highlight</Name>
↳<FeatureTypeStyle><Rule><Name>Symbol</Name><LineSymbolizer><Stroke><SvgParameter.
↳name="stroke">%23ea1173</SvgParameter><SvgParameter name="stroke-opacity">1</
↳SvgParameter><SvgParameter name="stroke-width">1.6</SvgParameter></Stroke></
↳LineSymbolizer></Rule></FeatureTypeStyle></UserStyle></StyledLayerDescriptor>
&HIGHLIGHT_LABELSTRING=Write label here
&HIGHLIGHT_LABELSIZE=16
&HIGHLIGHT_LABELCOLOR=%23000000
&HIGHLIGHT_LABELBUFFERCOLOR=%23FFFFFF
&HIGHLIGHT_LABELBUFFERSIZE=1.5
```

L'equivalente di `GetPrint` è nel formato (si noti che il parametro `mappaX`: è aggiunto per dire quale mappa ha il `redlining`):

```
http://qgisplatform.demo/cgi-bin/qgis_mapserv.fcgi?map=/world.qgs&SERVICE=WMS&
↳VERSION=1.3.0&
REQUEST=GetPrint
```

(continues on next page)

(continua dalla pagina precedente)

```

...
&map0:HIGHLIGHT_GEOM=POLYGON((590000 5647000, 590000 6110620, 2500000 6110620, ↵
↵2500000 5647000, 590000 5647000))
&map0:HIGHLIGHT_SYMBOL=<StyledLayerDescriptor><UserStyle><Name>Highlight</Name>
↵<FeatureTypeStyle><Rule><Name>Symbol</Name><LineStyle><Stroke><SvgParameter ↵
↵name="stroke">%23ea1173</SvgParameter><SvgParameter name="stroke-opacity">1</
↵SvgParameter><SvgParameter name="stroke-width">1.6</SvgParameter></Stroke></
↵LineStyle></Rule></FeatureTypeStyle></UserStyle></StyledLayerDescriptor>
&map0:HIGHLIGHT_LABELSTRING=Write label here
&map0:HIGHLIGHT_LABELSIZE=16
&map0:HIGHLIGHT_LABELCOLOR=%23000000
&map0:HIGHLIGHT_LABELBUFFERCOLOR=%23FFFFFF
&map0:HIGHLIGHT_LABELBUFFERSIZE=1.5

```

Ecco l'immagine prodotta dalla richiesta di cui sopra in cui un poligono e un'etichetta sono disegnati sopra la mappa normale:



Fig. 3.7: Risposta del server a una richiesta GetMap con parametri di redlining

Puoi vedere che ci sono diversi parametri in questa richiesta:

- **HIGHLIGHT_GEOM:** Puoi aggiungere POINT, MULTILINESTRING, POLYGON ecc. Supporta le geometrie multiparte. Ecco un esempio: HIGHLIGHT_GEOM=MULTILINESTRING((0 0 0, 0 1, 1 1)). Le coordinate dovrebbero essere nel SR della richiesta GetMap/GetPrint.

- **HIGHLIGHT_SYMBOL**: Questo controlla come la geometria è evidenziata e si può cambiare la larghezza del tratto, il colore e l'opacità.
- **HIGHLIGHT_LABELSTRING**: Puoi passare il tuo testo di etichettatura a questo parametro.
- **HIGHLIGHT_LABELSIZE**: Questo parametro controlla la dimensione dell'etichetta.
- **HIGHLIGHT_LABELFONT**: Questo parametro controlla il carattere dell'etichetta (ad esempio Arial)
- **HIGHLIGHT_LABELCOLOR**: Questo parametro controlla il colore dell'etichetta.
- **HIGHLIGHT_LABELBUFFERCOLOR**: Questo parametro controlla il colore del buffer dell'etichetta.
- **HIGHLIGHT_LABELBUFFERSIZE**: Questo parametro controlla la dimensione del buffer dell'etichetta.

3.7 Layer WMS esterni

QGIS Server permette di includere layer da server WMS esterni nelle richieste WMS GetMap e WMS GetPrint. Questo è particolarmente utile se un client web usa un livello di sfondo esterno nella mappa web. Per ragioni di prestazioni, tali livelli dovrebbero essere richiesti direttamente dal client web (non in cascata attraverso il server QGIS). Per la stampa, invece, questi livelli dovrebbero essere collegati in cascata attraverso il server QGIS per apparire nella mappa stampata.

I layer esterni possono essere aggiunti al parametro LAYERS come EXTERNAL_WMS:1. I parametri per i layer WMS esterni (ad esempio url, format, dpiMode, crs, layers, styles) possono essere dati successivamente come parametri di servizio <layername>:<parameter>. In una richiesta GetMap, questo potrebbe apparire così:

```
http://localhost/qgisserver?  
SERVICE=WMS&REQUEST=GetMap  
...  
&LAYERS=EXTERNAL_WMS:basemap,layer1,layer2  
&STYLES=,,  
&basemap:url=http://externalserver.com/wms.fcgi  
&basemap:format=image/jpeg  
&basemap:dpiMode=7  
&basemap:crs=EPSG:2056  
&basemap:layers=orthofoto  
&basemap:styles=default
```

Analogamente, i layer esterni possono essere usati nelle richieste GetPrint:

```
http://localhost/qgisserver?  
SERVICE=WMS  
...  
&REQUEST=GetPrint&TEMPLATE=A4  
&map0:layers=EXTERNAL_WMS:basemap,layer1,layer2  
&map0:EXTENT=<minx,miny,maxx,maxy>  
&basemap:url=http://externalserver.com/wms.fcgi  
&basemap:format=image/jpeg  
&basemap:dpiMode=7  
&basemap:crs=EPSG:2056  
&basemap:layers=orthofoto  
&basemap:styles=default
```

3.8 QGIS Server catalog

Il QGIS Server Catalog è un semplice catalogo che mostra l'elenco dei progetti QGIS forniti dal QGIS Server. Fornisce un sito web user-friendly completamente navigabile con capacità di mappatura di base per sfogliare rapidamente i dataset presenti in quei progetti QGIS.

Il QGIS Server catalog usa le variabili `QGIS_SERVER_LANDING_PAGE_PROJECTS_DIRECTORIES` e `QGIS_SERVER_LANDING_PAGE_PROJECTS_PG_CONNECTIONS` (vedi *Variabili d'ambiente*)

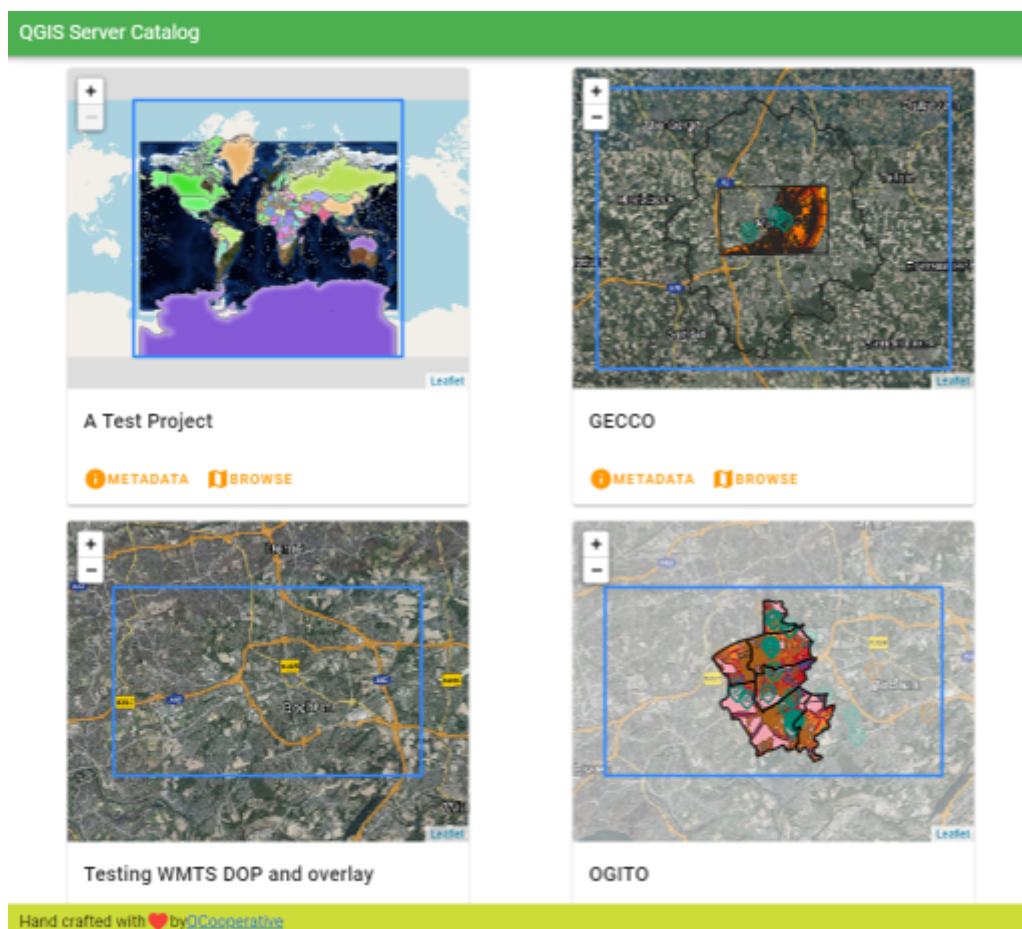


Fig. 3.8: Pagina dell'elenco dei progetti del Server Catalog

Puoi consultare i metadati associati a un progetto e i servizi che fornisce. Vengono forniti anche i link a questi servizi.

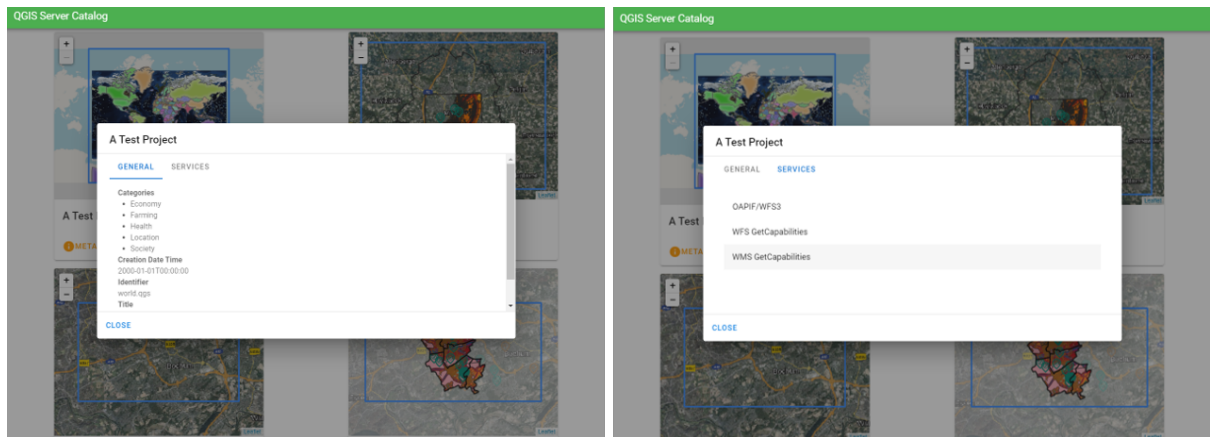


Fig. 3.9: Server Catalog, metadati associati a un progetto e servizi (link a) che fornisce.

Sfogliando un progetto, viene elencato il dataset che serve.

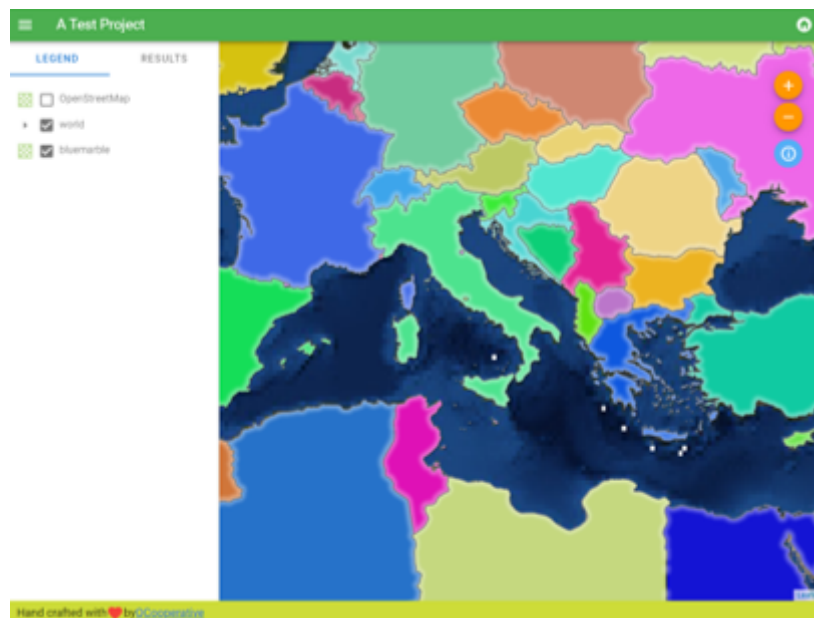


Fig. 3.10: Sfolciare un dataset fornito da un progetto nel catalogo del server

Usa il clic destro su un layer per visualizzare la tabella degli attributi ad esso associati.

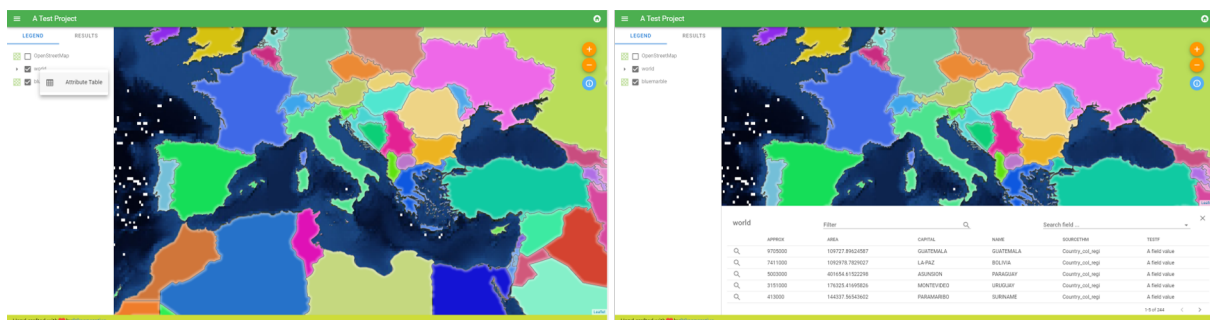


Fig. 3.11: Tabella degli attributi associati a un layer

È possibile consultare le informazioni degli elementi nella mappa come mostrato nell'immagine qui sotto:

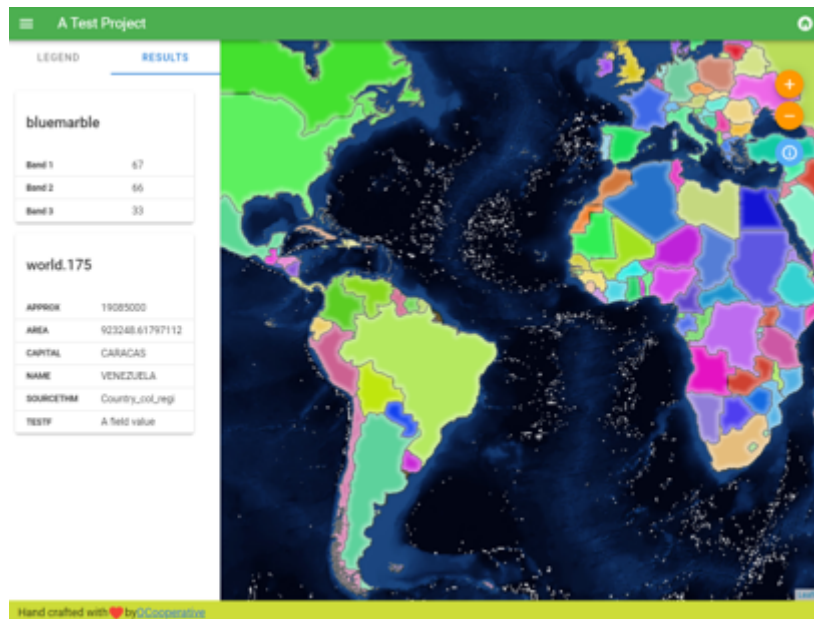


Fig. 3.12: Consultazione delle informazioni di un elemento della mappa

4.1 Installazione

Per installare il plugin di esempio HelloWorld per testare i server, bisogna innanzitutto creare una cartella per contenere i plugin del server. Questa sarà specificata nella configurazione dell'host virtuale e passata al server attraverso una variabile d'ambiente:

```
mkdir -p /var/www/qgis-server/plugins
cd /var/www/qgis-server/plugins
wget https://github.com/elpaso/qgis-helloserver/archive/master.zip
unzip master.zip
mv qgis-helloserver-master HelloServer
```

4.2 Configura il server HTTP

4.2.1 Apache

Per poter utilizzare i plugin del server, FastCGI ha bisogno di sapere dove cercare. Quindi, dobbiamo modificare il file di configurazione di Apache per indicare a FastCGI la variabile d'ambiente **QGIS_PLUGINPATH**:

```
FcgidInitialEnv QGIS_PLUGINPATH "/var/www/qgis-server/plugins"
```

Inoltre, un'autorizzazione HTTP di base è necessaria per lavorare con il plugin HelloWorld precedentemente introdotto. Quindi dobbiamo aggiornare il file di configurazione di Apache un'ultima volta:

```
# Needed for QGIS HelloServer plugin HTTP BASIC auth
<IfModule mod_fcgid.c>
  RewriteEngine on
  RewriteCond %{HTTP:Authorization} .
  RewriteRule .* - [E=HTTP_AUTHORIZATION:%{HTTP:Authorization}]
</IfModule>
```

Poi, riavvia Apache:

```
systemctl restart apache2
```

4.3 Come usare un plugin

Fai il test il server con il plugin HelloWorld:

```
wget -q -O - "http://localhost/cgi-bin/qgis_mapserv.fcgi?SERVICE=HELLO"  
HelloServer!
```

Puoi dare un'occhiata al GetCapabilities di default del server QGIS su:

```
http://localhost/cgi-bin/qgis_mapserv.fcgi?SERVICE=WMS&VERSION=1.3.0&  
↪REQUEST=GetCapabilities
```


5.1 Registrazioni di controllo

To log requests sent to the server, you have to set the following environment variable:

- *QGIS_SERVER_LOG_STDERR*

With the following variables the logging can be further customized:

- *QGIS_SERVER_LOG_LEVEL*
- *QGIS_SERVER_LOG_PROFILE*

5.2 Variabili d'ambiente

Puoi configurare alcuni aspetti di QGIS Server impostando le **environment variables**.

According to the HTTP server and how you run QGIS Server, there are several ways to define these variables. This is fully described in *Apache HTTP Server*.

Nome	Descrizione	Predefinito	Servizi
QGIS_OPTIONS_PATH	Specifies the path to the directory with settings. It works the same way as QGIS application <code>--optionspath</code> option. It is looking for settings file in <code><QGIS_OPTIONS_PATH>/QGIS/QGIS3.ini</code> .	""	All
QGIS_PLUGINPATH	Useful if you are using Python plugins for the server, this sets the folder that is searched for Python plugins.	""	All
QGIS_PROJECT_FILE	The <code>.qgs</code> or <code>.qgz</code> project file, normally passed as a parameter in the query string (with <i>MAP</i>), you can also set it as an environment variable (for example by using <code>mod_rewrite</code> Apache module). Note that you may also indicate a project stored in PostgreSQL, e.g. <code>postgresql://localhost:5432?sslmode=disable&dbname=mydb&schema=myschema&project=myproject</code> .	""	All
QGIS_SERVER_API_RESOURCE_DIRECTORY	Base directory for all OGC API (such as OAPIF/WFS3) static resources (HTML templates, CSS, JS, ...)	depends on packaging	WFS
QGIS_SERVER_API_WFS3_MAX_ITEMS	Valida il massimo per limit in una richiesta di elementi.	10000	WFS
QGIS_SERVER_CACHE_DIRECTORY	Specifies the network cache directory on the filesystem.	cache in profile directory	All
QGIS_SERVER_CACHE_SIZE	Sets the network cache size in MB.	50 MB	All
QGIS_SERVER_DISABLE_GETPRINT	PRINT an option at the project level to improve project read time by disabling loading of layouts. Activating this option disables the QGIS WMS GetPrint request. Set this QGIS project flag to not load layouts.	false	WMS
QGIS_SERVER_IGNORE_BADLAYERS	«Bad» layers are layers that cannot be loaded. The default behavior of QGIS Server is to consider the project as not available if it contains a bad layer. The default behavior can be overridden by setting this variable to <code>1</code> or <code>true</code> . In this case, «bad» layers will just be ignored, and the project will be considered valid and available.	false	All
QGIS_SERVER_LANDING_PAGE_PROJECTS_DIRECTORIES	Directories used by the landing page service to find <code>.qgs</code> and <code>.qgz</code> projects	<code>/qgis/server/projects_directories</code>	All
QGIS_SERVER_LANDING_PAGE_PROJECTS_PG_CONNECTIONS	PostgreSQL connection strings used by the landing page service to find projects	<code>/qgis/server/projects_pg_connections</code>	All
QGIS_SERVER_LOG_FILE	Specify path and filename. Make sure that server has proper permissions for writing to file. File should be created automatically, just send some requests to server. If it's not there, check permissions.	""	All
62	<p>Avvertimento: QGIS_SERVER_LOG_FILE is deprecated since QGIS 3.4, use QGIS_SERVER_LOG_STDERR instead.</p>	<p>Capitolo 5. Configurazione avanzata</p>	

5.3 Settings summary

When QGIS Server is starting, you have a summary of all configurable parameters thanks to environment variables. Moreover, the value currently used and the origin is also displayed.

For example with spawn-fcgi:

```
export QGIS_OPTIONS_PATH=/home/user/.local/share/QGIS/QGIS3/profiles/default/
export QGIS_SERVER_LOG_STDERR=1
export QGIS_SERVER_LOG_LEVEL=2
spawn-fcgi -f /usr/lib/cgi-bin/qgis_mapserv.fcgi -s /tmp/qgisserver.sock -U www-
↳data -G www-data -n

QGIS Server Settings:

- QGIS_OPTIONS_PATH / '' (Override the default path for user configuration): '/
↳home/user/.local/share/QGIS/QGIS3/profiles/default/' (read from ENVIRONMENT_
↳VARIABLE)

- QGIS_SERVER_PARALLEL_RENDERING / '/qgis/parallel_rendering' (Activate/
↳Deactivate parallel rendering for WMS getMap request): 'true' (read from INI_
↳FILE)

- QGIS_SERVER_MAX_THREADS / '/qgis/max_threads' (Number of threads to use when
↳parallel rendering is activated): '4' (read from INI_FILE)

- QGIS_SERVER_LOG_LEVEL / '' (Log level): '2' (read from ENVIRONMENT_VARIABLE)

- QGIS_SERVER_LOG_STDERR / '' (Activate/Deactivate logging to stderr): '1'
↳(read from ENVIRONMENT_VARIABLE)

- QGIS_PROJECT_FILE / '' (QGIS project file): '' (read from DEFAULT_VALUE)

- MAX_CACHE_LAYERS / '' (Specify the maximum number of cached layers): '100'
↳(read from DEFAULT_VALUE)

- QGIS_SERVER_CACHE_DIRECTORY / '/cache/directory' (Specify the cache
↳directory): '/root/.local/share/QGIS/QGIS3/profiles/default/cache' (read from
↳DEFAULT_VALUE)

- QGIS_SERVER_CACHE_SIZE / '/cache/size' (Specify the cache size): '52428800'
↳(read from INI_FILE)

Ini file used to initialize settings: /home/user/.local/share/QGIS/QGIS3/profiles/
↳default/QGIS/QGIS3.ini
```

In this particular case, we know that **QGIS_SERVER_MAX_THREADS** and **QGIS_SERVER_PARALLEL_RENDERING** values are read from the ini file found in **QGIS_OPTIONS_PATH** directory (which is defined through an environment variable). The corresponding entries in the ini file are **/qgis/max_threads** and **/qgis/parallel_rendering** and their values are **true** and **4** threads.

5.4 Short name for layers, groups and project

A number of elements have both a <Name> and a <Title>. The **Name** is a text string used for machine-to-machine communication while the **Title** is for the benefit of humans.

For example, a dataset might have the descriptive Title “Maximum Atmospheric Temperature” and be requested using the abbreviated **Name** “ATMAX”. The user can set the title for layers, groups and projects.

OWS name is based on the name used in the layer tree. This name is more a label for humans than a name for machine-to-machine communication. You can set a **Short name** for layers, groups or projects, to be used by QGIS Server as the layer identification name (in *LAYERS* parameter for instance).

You can set title, short name and abstract for:

- **Layers:** right-click on a layer and choose *Properties... ► QGIS Server ► Description*.
- **Groups:** right-click on a group and select *Set Group WMS data*

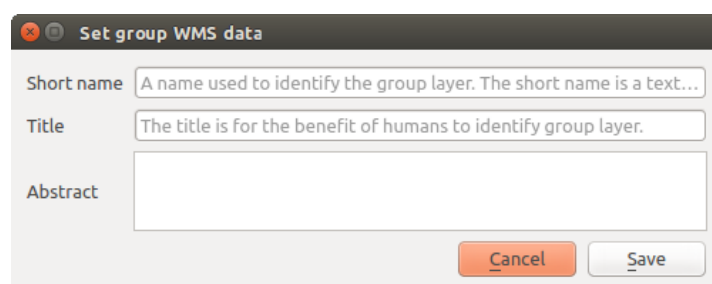


Fig. 5.1: Set group WMS data dialog

- **Project:** go to *Project ► Properties... ► QGIS Server ► Service Capabilities*.

5.5 Connection to service file

In order to make apache aware of the PostgreSQL service file (see the pg-service-file section) you need to make your *.conf file look like:

```
SetEnv PGSERVICEFILE /home/web/.pg_service.conf
<Directory "/home/web/apps2/bin/">
  AllowOverride None
.....
```

5.6 Add fonts to your linux server

Keep in mind that you may use QGIS projects that point to fonts that may not exist by default on other machines. This means that if you share the project, it may look different on other machines (if the fonts don't exist on the target machine).

In order to ensure this does not happen you just need to install the missing fonts on the target machine. Doing this on desktop systems is usually trivial (double clicking the fonts).

For linux, if you don't have a desktop environment installed (or you prefer the command line) you need to:

- Su sistemi basati su Debian:

```
sudo su
mkdir -p /usr/local/share/fonts/truetype/myfonts && cd /usr/local/share/fonts/
↳truetype/myfonts

# copy the fonts from their location
cp /fonts_location/* .

chown root *
cd .. && fc-cache -f -v
```

- Su sistemi basati su Fedora:

```
sudo su
mkdir /usr/share/fonts/myfonts && cd /usr/share/fonts/myfonts

# copy the fonts from their location
cp /fonts_location/* .

chown root *
cd .. && fc-cache -f -v
```

Server di sviluppo

Un'installazione e un deployment di produzione di QGIS Server di solito comporta la configurazione di un componente web server (ad esempio Apache o Nginx) che può inoltrare le richieste HTTP provenienti dai client all'applicazione binaria QGIS Server FastCGI.

Se vuoi testare rapidamente QGIS Server sulla tua macchina locale senza configurare e installare uno stack completo del server web, puoi usare il server QGIS Development Standalone.

Questa è un'applicazione indipendente che fornisce un server web molto semplice pronto a ospitare i file del tuo progetto.

Avvertimento: Il server di sviluppo standalone non è stato sviluppato con lo scopo di essere usato in produzione, non è stato controllato per le vulnerabilità di sicurezza o per altre condizioni di stress che normalmente si verificano su un server esposto pubblicamente.

Per avviare il server:

```
$ qgis_mapserver
```

La porta predefinita che il Development Server ascolta è 8000. Esempio in uscita:

```
QGIS Development Server listening on http://localhost:8000
CTRL+C to exit
127.0.0.1 [lun gen 20 15:16:41 2020] 5140 103ms "GET /wfs3/?MAP=/tests/testdata/
↪qgis_server/test_project.qgs HTTP/1.1" 200
127.0.0.1 [lun gen 20 15:16:41 2020] 3298 2ms "GET /wfs3/static/jsonFormatter.min.
↪js HTTP/1.1" 200
127.0.0.1 [lun gen 20 15:16:41 2020] 1678 3ms "GET /wfs3/static/jsonFormatter.min.
↪css HTTP/1.1" 200
127.0.0.1 [lun gen 20 15:16:41 2020] 1310 5ms "GET /wfs3/static/style.css HTTP/1.1
↪" 200
127.0.0.1 [lun gen 20 15:16:43 2020] 4285 13ms "GET /wfs3/collections?MAP=/tests/
↪testdata/qgis_server/test_project.qgs HTTP/1.1" 200
```

Il server ha alcune opzioni che possono essere passate come argomenti della linea di comando. Puoi vederle tutte invocando il server con `-h`.

```
Usage: qgis_mapserver [options] [address:port]
QGIS Development Server

Options:
-h, --help           Displays this help.
-v, --version        Displays version information.
-l <logLevel>       Sets log level (default: 0)
                    0: INFO
                    1: WARNING
                    2: CRITICAL
-p <projectPath>    Path to a QGIS project file (*.qgs or *.qgz),
                    if specified it will override the query string MAP argument
                    and the QGIS_PROJECT_FILE environment variable

Arguments:
addressAndPort       Listen to address and port (default: "localhost:8000")
                    address and port can also be specified with the environment
                    variables QGIS_SERVER_ADDRESS and QGIS_SERVER_PORT
```

Sviluppo ed esecuzione in container

Ci sono molti modi per utilizzare applicazioni containerizzate, dai più semplici (semplici immagini Docker) ai più sofisticati (Kubernetes e così via).

Nota: Questo tipo di sviluppo ha bisogno che l'applicazione `docker` sia installata e funzionante. Guarda questo [tutorial](#).

Suggerimento: Docker run pre packaged application (aka images) which can be retrieved as sources (Dockerfile and resources) to build or already built from registries (private or public).

Nota: QGIS Debian-Ubuntu package downloads need a valid gpg authentication key. Please refer to the [installation pages](#) to update the following Dockerfile with the latest key fingerprint

7.1 Simple docker images

As the docker image does not exist in a public registry. you will need to build it. To do so create a directory `qgis-server` and within its directory:

- crea un file `Dockerfile` con questo contenuto:

```
FROM debian:buster-slim

ENV LANG=en_EN.UTF-8

RUN apt-get update \
    && apt-get install --no-install-recommends --no-install-suggests --allow-
↪unauthenticated -y \
    gnupg \
    ca-certificates \
    wget \
    locales \
```

(continues on next page)

```

    && localedef -i en_US -f UTF-8 en_US.UTF-8 \
    # Add the current key for package downloading - As the key changes every year
↳at least
    # Please refer to QGIS install documentation and replace it with the latest one
    && wget -O - https://qgis.org/downloads/qgis-2020.gpg.key | gpg --import \
    && gpg --export --armor F7E06F06199EF2F2 | apt-key add - \
    && echo "deb http://qgis.org/debian buster main" >> /etc/apt/sources.list.d/
↳qgis.list \
    && apt-get update \
    && apt-get install --no-install-recommends --no-install-suggests --allow-
↳unauthenticated -y \
        qgis-server \
        spawn-fcgi \
        xauth \
        xvfb \
    && apt-get remove --purge -y \
        gnupg \
        wget \
    && rm -rf /var/lib/apt/lists/*

RUN useradd -m qgis

ENV TINI_VERSION v0.17.0
ADD https://github.com/krallin/tini/releases/download/${TINI_VERSION}/tini /tini
RUN chmod +x /tini

ENV QGIS_PREFIX_PATH /usr
ENV QGIS_SERVER_LOG_STDERR 1
ENV QGIS_SERVER_LOG_LEVEL 2

COPY cmd.sh /home/qgis/cmd.sh
RUN chmod -R 777 /home/qgis/cmd.sh
RUN chown qgis:qgis /home/qgis/cmd.sh

USER qgis
WORKDIR /home/qgis

ENTRYPOINT ["/tini", "--"]

CMD ["/home/qgis/cmd.sh"]

```

- crea un file cmd.sh con questo contenuto:

```

#!/bin/bash

[[ $DEBUG == "1" ]] && env

exec /usr/bin/xvfb-run --auto-servernum --server-num=1 /usr/bin/spawn-fcgi -p 5555
↳-n -d /home/qgis -- /usr/lib/cgi-bin/qgis_mapserv.fcgi

```

- costruisce l'immagine con:

```

docker build -f Dockerfile -t qgis-server ./

```

7.1.1 First run

To run the server you will need a QGIS project file. You can use one of yours or pick [this sample](#).

To do so, create a directory `data` within the directory `qgis-server` and copy your file in it. To comply with the following explanations, rename it to `osm.qgs`.

Nota: You may need to add advertised URLs under the *QGIS Server* tab of the *Project ► Properties* if the GetCapabilities are broken. For example if your server is exposed on port 8080, you will put this for advertised URL `http://localhost:8080/qgis-server/`. More information available in section *Configura il tuo progetto* and subsequent.

Ora, puoi avviare il server con:

```
docker network create qgis
docker run -d --rm --name qgis-server --net=qgis --hostname=qgis-server \
    -v $(pwd)/data:/data:ro -p 5555:5555 \
    -e "QGIS_PROJECT_FILE=/data/osm.qgs" \
    qgis-server
```

Opzioni utilizzate:

- **-d:** run in the background
- **--rm:** remove the container when it is stopped
- **--name:** name of the container to be created
- **--net:** (previously created) sub network
- **--hostname:** container hostname, for later referencing
- **-v:** local data directory to be mounted in the container
- **-p:** host/container port mapping
- **-e:** environment variable to be used in the container

To check, type `docker ps | grep qgis-server` and you should see a line with **qgis-server**:

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
↪ PORTS		NAMES		
4de8192da76e	qgis-server	"/tini -- /home/qgis..."	3 seconds ago	Up 2 seconds
↪ 0.0.0.0:5555->5555/tcp		qgis-server		

7.1.2 Usable sample

As the server is only accepting fastcgi connections, you need an HTTP server that handles this protocol. To do so we have to create a simple Nginx configuration file and start a Nginx image.

Create a file `nginx.conf` in the current directory with this content:

```
server {
    listen 80;
    server_name _;
    location / {
        root /usr/share/nginx/html;
        index index.html index.htm;
    }
    location /qgis-server {
        proxy_buffers 16 16k;
        proxy_buffer_size 16k;
    }
}
```

(continues on next page)

(continua dalla pagina precedente)

```
gzip off;
include fastcgi_params;
fastcgi_pass qgis-server:5555;
}
}
```

And type this command:

```
docker run -d --rm --name nginx --net=qgis --hostname=nginx \
-v $(pwd)/nginx.conf:/etc/nginx/conf.d/default.conf:ro -p 8080:80 \
nginx:1.13
```

To check capabilities availability, type in a browser <http://localhost:8080/qgis-server/?SERVICE=WMS&VERSION=1.3.0&REQUEST=GetCapabilities>

7.1.3 Cleanup

To cleanup the running images, type:

```
docker stop qgis-server nginx
```

7.2 Docker stacks

The previous method is scriptable, but not easily packageable nor standardized or easily manageable.

To work with a docker image set you could use a docker stack managed by an orchestrator. In a stack, the images are working in the same private network, and you can start / stop the whole stack or deploy the stack to other workers. There are many orchestrators, for example Swarm, Kubernetes and Mesos.

In the following, we will present simple configurations for testing purposes. They are not suitable for production.

7.2.1 Swarm/docker-compose

Docker now has its own orchestrator: Swarm (compatible with docker-compose files). You have to [enable it](#) (the Mac version will also work with Linux).

Stack description

Now that you have Swarm working, create the service file (see [Deploy to Swarm](#)) `qgis-stack.yaml`:

```
version: '3.7'

services:
  qgis-server:
    # Should use version with utf-8 locale support:
    image: qgis-server:latest
    volumes:
      - REPLACE_WITH_FULL_PATH/data:/data:ro
    environment:
      - LANG=en_EN.UTF-8
      - QGIS_PROJECT_FILE=/data/osm.qgs
      - QGIS_SERVER_LOG_LEVEL=0 # INFO (log all requests)
      - DEBUG=1 # display env before spawning QGIS Server

  nginx:
```

(continues on next page)

(continua dalla pagina precedente)

```

image: nginx:1.13
ports:
- 8080:80
volumes:
- REPLACE_WITH_FULL_PATH/nginx.conf:/etc/nginx/conf.d/default.conf:ro
depends_on:
- qgis-server
    
```

To deploy (or update) the stack, type:

```
docker stack deploy -c qgis-stack.yaml qgis-stack
```

Check the stack deployment status until you obtain **1/1** in the **replicas** column:

```
docker stack services qgis-stack
```

Something like:

ID	NAME	MODE	REPLICAS	
↪IMAGE	PORTS			
gmx7ewlvwsqt	qgis_nginx	replicated	1/1	↪
↪nginx:1.13	*:8080->80/tcp			
10v2e7c143u3	qgis_qgis-server	replicated	1/1	↪
↪qgis-server:latest				

To check WMS capabilities, type in a web browser <http://localhost:8080/qgis-server/?SERVICE=WMS&VERSION=1.3.0&REQUEST=GetCapabilities>

Cleanup

To cleanup, type:

```
docker stack rm qgis-stack
```

7.2.2 Kubernetes

Installazione

If you have a **Docker Desktop** installation, using Kubernetes (aka k8s) is pretty straight forward: [enable k8s](#).

If not, follow the [minikube tutorial](#) or [microk8s for Ubuntu](#).

As Kubernetes installation can be really complex, we will only focus on aspects used by this demo. For further / deeper information, check the [official documentation](#).

microk8s

microk8s needs extra steps: you have to enable the registry and tag the qgis-server image in order to have Kubernetes to find the created images.

Per prima cosa, abilita il registro:

```
microk8s enable dashboard dns registry
```

Then, tag and push the image to your newly created registry:

```
docker tag qgis-server 127.0.0.1:32000/qgis-server && docker push 127.0.0.1:32000/
↔qgis-server
```

Finally, add or complete the `/etc/docker/daemon.json` to have your registry **127.0.0.1:32000** listed in the **insecure-registries** field:

```
{
  "insecure-registries": ["127.0.0.1:32000"]
}
```

Creating manifests

Kubernetes describes the objects to deploy in yaml manifests. There are many different kinds, but we will only use deployments (handle pods, i.e. docker images) and services to expose the deployments to internal or external purposes.

Deployment manifests

Create a file `deployments.yaml` with this content:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: qgis-server
  namespace: default
spec:
  replicas: 1
  selector:
    matchLabels:
      myLabel: qgis-server
  template:
    metadata:
      labels:
        myLabel: qgis-server
    spec:
      containers:
        - name: qgis-server
          image: localhost:32000/qgis-server:latest
          imagePullPolicy: IfNotPresent
          env:
            - name: LANG
              value: en_EN.UTF-8
            - name: QGIS_PROJECT_FILE
              value: /data/osm.qgs
            - name: QGIS_SERVER_LOG_LEVEL
              value: "0"
            - name: DEBUG
              value: "1"
          ports:
            - containerPort: 5555
          volumeMounts:
            - name: qgis-data
              mountPath: /data/
      volumes:
        - name: qgis-data
          hostPath:
            path: REPLACE_WITH_FULL_PATH/data
---
apiVersion: apps/v1
```

(continues on next page)

(continua dalla pagina precedente)

```

kind: Deployment
metadata:
  name: qgis-nginx
  namespace: default
spec:
  replicas: 1
  selector:
    matchLabels:
      myLabel: qgis-nginx
  template:
    metadata:
      labels:
        myLabel: qgis-nginx
    spec:
      containers:
        - name: qgis-nginx
          image: nginx:1.13
          ports:
            - containerPort: 80
          volumeMounts:
            - name: nginx-conf
              mountPath: /etc/nginx/conf.d/default.conf
      volumes:
        - name: nginx-conf
          hostPath:
            path: REPLACE_WITH_FULL_PATH/nginx.conf

```

Service manifests

Create a file `services.yaml` with this content:

```

apiVersion: v1
kind: Service
metadata:
  name: qgis-server
  namespace: default
spec:
  type: ClusterIP
  selector:
    myLabel: qgis-server
  ports:
    - port: 5555
      targetPort: 5555
---
apiVersion: v1
kind: Service
metadata:
  name: qgis-nginx
  namespace: default
spec:
  type: NodePort
  selector:
    myLabel: qgis-nginx
  ports:
    - port: 80
      targetPort: 80
      nodePort: 30080

```

Deploying manifests

To deploy the images and services in Kubernetes, one can use the dashboard (click on the + on the upper right) or the command line.

Nota: When using the command line with `microk8s` you will have to prefix each command with `microk8s`.

To deploy or update your manifests:

```
kubectl apply -k ./
```

To check what is currently deployed:

```
kubectl get pods,services,deployment
```

You should obtain something like:

NAME	READY	STATUS	RESTARTS	AGE
pod/qgis-nginx-54845ff6f6-8skp9	1/1	Running	0	27m
pod/qgis-server-75df8ddd89-c7t7s	1/1	Running	0	27m

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	
↪ AGE					
service/Kubernetes	ClusterIP	10.152.183.1	<none>	443/TCP	↪
↪ 5h51m					
service/qgis-exec-server	ClusterIP	10.152.183.218	<none>	5555/TCP	↪
↪ 35m					
service/qgis-nginx	NodePort	10.152.183.234	<none>	80:30080/TCP	↪
↪ 27m					
service/qgis-server	ClusterIP	10.152.183.132	<none>	5555/TCP	↪
↪ 27m					

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/qgis-nginx	1/1	1	1	27m
deployment.apps/qgis-server	1/1	1	1	27m

To read nginx/qgis logs, type:

```
kubectl logs -f POD_NAME
```

To check WMS capabilities, type in a web browser <http://localhost:30080/qgis-server/?SERVICE=WMS&VERSION=1.3.0&REQUEST=GetCapabilities>

Cleanup

To clean up, type:

```
kubectl delete -n default service/qgis-server service/qgis-nginx deployment/qgis-  
↪nginx deployment/qgis-server
```


7.3 Cloud deployment

Managing your own cluster of servers to handle the deployment of containerized applications, is a complex job. You have to handle multiple issues, such as hardware, bandwidth and security at different levels.

Cloud deployment solutions can be a good alternative when you do not want to focus on infrastructure management.

A cloud deployment may use proprietary mechanisms, but they are also compatible with the stages explained previously (*docker images* and *stack management*).

7.3.1 AWS usecase

With Amazon AWS, through [ECS \(Elastic Container Service\)](#) functionalities, you can use `docker-compose` or Kubernetes compatible wrappers to manage your stack. You will have to create an [image registry](#) for your custom images to be accessible.

To use `docker-compose` alike functionalities, you need to install the **ecs-cli** client and have [proper permissions / roles](#). Then, with the help of the `ecs-cli compose` commands (see the [ecs-cli compose manual](#) and [ecs-cli tutorial](#)), you can reuse the [stack description](#).

To use Kubernetes, you can use the AWS web console or the command line tool `eksctl` and have the proper permissions / roles. Then with a well configured `kubectl` environment, you can reuse the [Kubernetes manifests](#).

Frequently Asked Question

- *What are the differences between QGIS Desktop and QGIS Server?*

QGIS Desktop has a graphical user interface and allows you to create and modify maps. QGIS Server is a server application serving your QGIS project files to end user applications via OGC web services like [WMS](#), [WFS](#), etc..

- *What is OGC?*

The [OGC \(Open Geospatial Consortium\)](#) is an international not for profit organization committed to making quality open standards for the global geospatial community.

- *Name some other web mapping servers?*

ArcGIS server, Geoserver, Mapserver, Mapnik etc.

- *How to compare QGIS server to other web mapping servers? (2021/01/01)*

Features	QGIS Server	GeoServer	ArcGIS Server
Since	2006	2001	1999
Licence	GPL	GPL	commercial
Commercial support	Multiple companies	Multiple companies	ESRI and its vendors network
Technology	C++/python	Java	C++
Tile cache	yes	yes (via GeoWebCache)	yes
3D	No	No	Yes
Querying	FES (2.0) and OGC (1.0) filters	CQL and OGC filters	OGC filters
Report generation	yes	yes	yes
Server administration	yes via third parties (LizMap, QWC2, etc.)	web + API REST	web + API REST
GIS project Layer/symbology edition	complete via dedicated GUI	simple via web interface	complete via dedicated GUI

- *What are the OGC specification versions implemented in QGIS server compared to other web mapping servers? (2021/01/01)*

OGC standards	QGIS Server	GeoServer	ArcGIS Server
WMS (Web Map Service)	1.3.0 - 1.1.1	1.3.0 - 1.1.1	1.3.0 - 1.1.1
WFS (Web Feature Service)	1.1.0 - 1.0.0	2.0.0 - 1.0.0	2.0.0 - 1.0.0
OAPIF (aka WFS3)	1.0.0	no	no
WMTS (Web Map Tile Service)	1.0.0	1.0.0	1.0.0
WCS (Web Coverage Service)	1.0.0	2.0.1 - 1.0.0	2.0.1 - 1.0.0
WPS (Web Processing Service)	no	1.0.0	1.0.0
CSW (Catalogue Service for the Web)	no	2.0.2	no
SLD (Styled Layer Descriptor)	yes	yes	yes

- *What is a tile cache?*

Maps are often static. As most mapping clients render WMS (Web Map Service) data every time they are queried, this can result in unnecessary processing and increased wait times.

The tile cache optimizes this experience by saving (caching) map images, or tiles, as they are requested, in effect acting as a proxy between client (such as OpenLayers or Google Maps) and server (any WMS-compliant server). As new maps and tiles are requested, QGIS server intercepts these calls and returns pre-rendered tiles if stored, or calls the QGIS engine to render new tiles as necessary. Thus, once tiles are stored, the speed of map rendering increases by many times, creating a much improved user experience.

- *What is PostgreSQL?*

PostgreSQL is a powerful, open source object-relational database companion for QGIS.

- *What is PostGIS?*

PostGIS is a spatial database extender for PostgreSQL object-relational database. It adds support for geographic objects allowing location queries to be run in SQL.

- To be continued...