



# **QGIS Server 3.16 User Guide**

**QGIS Project**

**mars 24, 2022**



<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Premiers Pas</b>	<b>3</b>
2.1	Installation sur systèmes Debian . . . . .	3
2.1.1	Apache HTTP Server . . . . .	5
2.1.2	NGINX HTTP Server . . . . .	7
2.1.3	Xvfb . . . . .	11
2.2	Installation sous Windows . . . . .	12
2.3	Publier un projet . . . . .	15
2.4	Configurer votre projet . . . . .	16
2.4.1	Capacités WMS . . . . .	18
2.4.2	Capacités WFS . . . . .	19
2.4.3	Capacités WCS . . . . .	19
2.4.4	Bien configurer votre OWS . . . . .	19
2.5	Intégration avec des applications tierces . . . . .	19
2.5.1	Intégration avec QGIS . . . . .	19
2.5.2	Intégration avec MapProxy . . . . .	20
2.5.3	Intégration avec QWC2 . . . . .	20
<b>3</b>	<b>Services</b>	<b>21</b>
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) . . . . .	37
3.2.1	GetFeature . . . . .	37
3.3	Web Map Tile Service (WMTS) . . . . .	39
3.3.1	GetCapabilities . . . . .	40
3.3.2	GetTile . . . . .	40
3.3.3	GetFeatureInfo . . . . .	42
3.4	WFS3 (OGC API Features) . . . . .	44
3.4.1	Représentation des ressources . . . . .	45
3.4.2	Points de terminaison . . . . .	45
3.4.3	Pagination . . . . .	49
3.4.4	Filtrage des entités . . . . .	49
3.4.5	Tri des éléments . . . . .	50
3.4.6	Sélection d'attribut . . . . .	50
3.4.7	Personnaliser les pages HTML . . . . .	51
3.5	Paramètres supplémentaires pris en charge par tous les types de demande . . . . .	52

3.6	REDLINING . . . . .	52
3.7	Couches WMS externes . . . . .	54
3.8	QGIS Server catalogue . . . . .	55
<b>4</b>	<b>Extensions</b>	<b>59</b>
4.1	Installation . . . . .	59
4.2	Configuration d'un serveur HTTP . . . . .	59
4.2.1	Apache . . . . .	59
4.3	Comment utilisez une extension ? . . . . .	60
<b>5</b>	<b>Configuration avancée</b>	<b>61</b>
5.1	Suivi du journal . . . . .	61
5.2	Variables d'environnement . . . . .	61
5.3	Résumé des paramètres . . . . .	63
5.4	Nom court pour les couches, les groupes et le projet . . . . .	64
5.5	Connexion au fichier de service . . . . .	64
5.6	Ajouter des polices à votre serveur Linux . . . . .	65
<b>6</b>	<b>Serveur de développement</b>	<b>67</b>
<b>7</b>	<b>Déploiement conteneurisé</b>	<b>69</b>
7.1	Simple images docker . . . . .	69
7.1.1	Premier lancement . . . . .	71
7.1.2	Exemple fonctionnel . . . . .	71
7.1.3	Nettoyage . . . . .	72
7.2	Docker stacks . . . . .	72
7.2.1	Swarm/docker-compose . . . . .	72
7.2.2	Kubernetes . . . . .	73
7.3	Déploiement en Cloud . . . . .	77
7.3.1	Cas d'utilisation avec AWS . . . . .	77
<b>8</b>	<b>Foire aux questions</b>	<b>79</b>

QGIS Server est une implémentation open source de WMS, WFS, OGC API for Features 1.0 (WFS3) et WCS qui, en outre, met en œuvre des fonctionnalités cartographiques avancées pour la cartographie thématique. QGIS Server est une application FastCGI/CGI (Common Gateway Interface) écrite en C++ qui fonctionne avec un serveur web (par exemple, Apache, Nginx). Il dispose d'un support d'extension Python permettant un développement et un déploiement rapide et efficace de nouvelles fonctionnalités.

QGIS Serveur utilise QGIS comme backend pour la logique des couches SIG et le rendu cartographique. La bibliothèque Qt est utilisée pour l'interface et la programmation multiplateforme en C++. À la différence des autres serveurs WMS, le QGIS Server utilise les règles de cartographie comme langage de configuration, à la fois pour la configuration du serveur et pour les règles cartographiques définie par l'utilisateur.

Etant donné que QGIS Desktop et QGIS Server utilisent les mêmes bibliothèques de visualisation, les cartes publiées sur le web ont le même aspect que sous le SIG Desktop.

Dans les sections suivantes, nous fournirons un exemple de configuration pour configurer un QGIS Server sur Linux (Debian, Ubuntu et dérivés) et sur Windows. Pour plus d'informations sur le développement de plugins serveur, veuillez lire `server_plugins`.

La permission de copier, distribuer, modifier ce document est accordée sous les termes de la GNU Free Documentation License, dans sa version 1.3 ou plus récente telle que publiée par la Free Software Foundation ; sans modification de son contenu, sans ajouts la précédant ou la suivant.

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



### 2.1 Installation sur systèmes Debian

Nous donnons un guide d'installation court et simple pour une configuration de travail minimale sur les systèmes basés sur Debian (y compris Ubuntu et dérivés). Cependant, de nombreuses autres distributions et systèmes d'exploitation fournissent des packages pour QGIS Server.

**Note:** Dans Ubuntu, vous pouvez utiliser votre utilisateur le plus couramment utilisé, en ajoutant `sudo` aux commandes nécessitant des autorisations administrateur. Dans Debian, vous pouvez travailler en tant qu'administrateur (`root`), sans utiliser `sudo`.

Les prérequis et étapes pour ajouter des dépôts officiels de QGIS afin d'installer QGIS Server sur un système basé sur Debian sont indiqués dans la [page des installateurs QGIS](#). Nous vous conseillons d'installer la version long terme.

Une fois que le dépôt de la version cible est configuré et QGIS Server installé, vous pouvez tester votre installation à l'aide de:

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

Si vous obtenez la sortie suivante, le serveur est correctement installé.

**Note:** Selon votre version de QGIS, de petites différences peuvent apparaître dans le rapport fourni par la commande `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
```

(suite sur la page suivante)

(suite de la page précédente)

```
<ServiceExceptionReport version="1.3.0" xmlns="https://www.opengis.net/ogc">
  <ServiceException code="Service configuration error">Service unknown or
  ↪ unsupported</ServiceException>
</ServiceExceptionReport>
```

**Note:** Comme on le voit ci-dessous, QGIS renvoie un code de statut 400, qui identifie correctement que la requête a échoué, du fait de l'absence d'une session http active. Ce n'est pas un bug; cela indique que le serveur fonctionne correctement.

---

```
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"}]
```

Ajoutons un exemple de projet. Vous pouvez utiliser les vôtres ou l'une des données de démonstration de la formation :

```
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 .
```

Bien entendu, vous pouvez utiliser votre logiciel SIG préféré pour ouvrir ce fichier et jeter un coup d'œil à la configuration et aux couches disponibles.

Pour déployer correctement QGIS Server, vous avez besoin d'un serveur HTTP. Les choix recommandés sont **Apache** ou **NGINX**.



## 2.1.1 Apache HTTP Server

**Note:** Dans ce qui suit, veuillez remplacer `qgis.demo` par le nom ou l'adresse IP de votre serveur.

Installez Apache et `mod_fcgid`:

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

Vous pouvez faire fonctionner QGIS Server sur votre site web par défaut, ou alors configurer un hôte virtuel (virtualhost), comme cela est présenté ci-dessous.

Dans le répertoire `/etc/apache2/sites-available`, créons un fichier nommé `qgis.demo.conf`, avec ce contenu :

```
<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
```

(suite sur la page suivante)

(suite de la page précédente)

```
FcgidConnectTimeout 60
</IfModule>

</VirtualHost>
```

Pour aller plus loin :

- *QGIS Server logging*
- pg-service-file dans QGIS Server

Vous pouvez faire ce qui précède dans un système Linux Desktop en collant et en sauvegardant la configuration ci-dessus après avoir fait :

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

---

**Note:** Certaines des options de configuration sont expliquées dans la section des *variables d'environnement du serveur*.

---

Créons maintenant les répertoires qui stockeront les logs du serveur QGIS et ceux de la base de données d'authentification :

```
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
```

---

**Note:** `www-data` est l'utilisateur Apache sur les systèmes basés sur Debian et il faut qu'Apache accède à ces répertoires et aux fichiers qu'ils contiennent. La commande `chown www-data . . .` attribue la propriété des fichiers et répertoires à `www-data`.

---

Nous pouvons maintenant activer l'hôte virtuel, ainsi que le mode `fcgid` si ce n'est pas déjà le cas.

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

Redémarrez maintenant Apache pour que la nouvelle configuration soit prise en compte :

```
systemctl restart apache2
```

Maintenant qu'Apache sait qu'il doit répondre aux demandes adressées à <http://qgis.demo>, nous devons également configurer le système client de manière à ce qu'il sache qui est `qgis.demo`. Pour ce faire, nous ajoutons `127.0.0.1 qgis.demo` dans le fichier `hosts`. Nous pouvons le faire avec `sh -c "echo '127.0.0.1 qgis.demo' >> /etc/hosts"`. Remplacez `127.0.0.1` par l'IP de votre serveur.

---

**Note:** Rappelez-vous que les fichiers `qgis.demo.conf` et `/etc/hosts` doivent être configurés pour que votre installation fonctionne. Vous pouvez également tester l'accès QGIS Server à partir d'autres clients sur le réseau (par exemple des machines Windows ou macOS) en allant dans leur fichier `/etc/hosts` et en faisant pointer le nom `myhost` vers l'adresse IP de la machine serveur sur le réseau (non pas sur `127.0.0.1` car c'est l'IP locale, uniquement accessible depuis la machine locale). Sur les machines `*nix`, le fichier `hosts` est situé dans le répertoire `/etc`, alors que sous Windows, il se trouve dans le répertoire `C:\Windows\System32\drivers\etc`. Sous Windows, vous devez démarrer votre éditeur de texte avec des privilèges d'administrateur avant d'ouvrir le fichier `hosts`.

---

QGIS Server est maintenant disponible sur <http://qgis.demo>. Pour vérifier, tapez dans un navigateur, comme dans ce simple cas :

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

## 2.1.2 NGINX HTTP Server

**Note:** Dans ce qui suit, veuillez remplacer `qgis.demo` par le nom ou l'adresse IP de votre serveur.

Vous pouvez également utiliser QGIS Server avec NGINX. Contrairement à Apache, NGINX ne génère pas automatiquement des processus FastCGI. Les processus FastCGI doivent être démarrés autrement.

Pour installer NGINX :

```
apt install nginx
```

- Comme première option, vous pouvez utiliser **spawn-fcgi** ou **fcgiwrap** pour démarrer et gérer les processus de QGIS Server. Des paquets Debian officiels existent pour les deux. Lorsque vous n'avez pas X serveur en fonctionnement et que vous avez besoin, par exemple, d'imprimer, vous pouvez utiliser *xvfb*.
- Une autre option consiste à s'appuyer sur **Systemd**, le système d'initialisation pour GNU / Linux que la plupart des distributions Linux utilisent aujourd'hui. L'un des avantages de cette méthode est qu'elle ne nécessite aucun autre composant ou processus. Il est censé être simple, mais robuste et efficace pour les déploiements de production.

### Configuration de NGINX

L'**include fastcgi\_params**; utilisé dans la configuration précédente est important, car il ajoute les paramètres de `/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;
```

De plus, vous pouvez utiliser certaines *Variables d'environnement* pour configurer QGIS Server. Dans le fichier de configuration NGINX, `/etc/nginx/nginx.conf`, vous devez utiliser l'instruction `fastcgi_param` pour définir ces variables comme indiqué ci-dessous :

```
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

**Attention:** `fcgiwrap` est plus facile à mettre en place que `spawn-fcgi`, car il est déjà intégré dans un service Systemd. Mais il conduit aussi à une solution beaucoup plus lente que l'utilisation de `spawn-fcgi`. Avec `fcgiwrap`, un nouveau processus QGIS Server est créé à chaque demande, ce qui signifie que le processus d'initialisation QGIS Server, qui comprend la lecture et l'analyse du fichier de projet QGIS, est effectué à chaque demande. Avec `spawn-fcgi`, le processus QGIS Server reste en fonction entre les demandes, ce qui se traduit par une bien meilleure performance. C'est pourquoi il est recommandé d'utiliser `spawn-fcgi` en production.

### spawn-fcgi

Si vous voulez utiliser `spawn-fcgi`, la première étape est d'installer le paquet :

```
apt install spawn-fcgi
```

Ensuite, introduisez le bloc suivant dans votre configuration de serveur NGINX :

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

Et redémarrez NGINX pour prendre en compte la nouvelle configuration :

```
systemctl restart nginx
```

Enfin, étant donné qu'il n'y a pas de fichier de service par défaut pour `spawn-fcgi`, vous devez démarrer manuellement QGIS Server dans votre terminal :

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

QGIS Server est maintenant disponible sur <http://qgis.demo/qgisserver>.

**Note:** Lorsque vous utilisez `spawn-fcgi`, vous pouvez définir directement les variables d'environnement avant de lancer le serveur. Par exemple : `export QGIS_SERVER_LOG_STDERR=1`

Bien entendu, vous pouvez ajouter un script d'initialisation pour démarrer QGIS Server au démarrage ou à tout moment. Par exemple avec **systemd**, éditez le fichier `/etc/systemd/system/qgis-server.service` avec ce contenu :

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

(suite sur la page suivante)

(suite de la page précédente)

```

[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

```

Ensuite, activez et démarrez le service :

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

**Avertissement:** Avec les commandes ci-dessus, spawn-fcgi n'engendre qu'un seul process QGIS Server.

## fcgiwrap

L'utilisation de fcgiwrap est beaucoup plus facile à configurer que **spawn-fcgi** mais elle est beaucoup plus lente. Vous devez d'abord installer le package correspondant :

```
apt install fcgiwrap
```

Ensuite, introduisez le bloc suivant dans votre configuration de serveur NGINX :

```

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  }

```

Enfin, redémarrez NGINX et **fcgiwrap** pour prendre en compte la nouvelle configuration :

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

QGIS Server est maintenant disponible sur <http://qgis.demo/qgisserver>.

## Systemd

QGIS Server a besoin d'un Serveur X en fonctionnement pour être utilisable, notamment pour l'impression. Si vous disposez déjà d'un serveur X en fonctionnement, vous pouvez utiliser les services Systemd.

Cette méthode, pour déployer QGIS Server, s'appuie sur deux services Systemd :

- un **Socket unit**
- et un **Service unit**.

**QGIS Server Socket** définit et crée une socket de système de fichiers, utilisée par NGINX pour démarrer et communiquer avec QGIS Server. L'unité Socket doit être configurée avec `Accept=false`, ce qui signifie que

les appels à l'appel système `accept()` sont délégués au processus créé par Service unit. Il se trouve dans `/etc/systemd/system/qgis-server@.socket`, qui est en fait un modèle:

```
[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
```

Maintenant, activez et démarrez les sockets :

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

**Service unit QGIS Server** définit et démarre le processus QGIS Server. L'important est que l'entrée standard du processus de service soit connectée à la prise définie par l'unité Socket. Cela doit être configuré en utilisant `StandardInput = socket` dans la configuration Service unit située dans `/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
```

---

**Note:** Les *variables d'environnement* de QGIS Server sont définies dans un fichier séparé, `/etc/qgis-server/env`. Il pourrait ressembler à ceci :

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

---

Maintenant, lancez le service de socket:

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

Enfin, pour le serveur HTTP NGINX, introduisons la configuration de ce paramétrage :

```
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;
}
```

(suite sur la page suivante)

(suite de la page précédente)

```
server {
    ...

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

Redémarrez maintenant NGINX pour que la nouvelle configuration soit prise en compte:

```
systemctl restart nginx
```

Merci à Oslandia d'avoir partagé leur [tutoriel](#).

### 2.1.3 Xvfb

QGIS Server a besoin d'un serveur X en cours d'exécution pour être pleinement utilisable, en particulier pour l'impression. Sur les serveurs, il est généralement recommandé de ne pas l'installer, vous pouvez donc utiliser `xvfb` pour avoir un environnement X virtuel.

Si vous exécutez le serveur dans un environnement graphique/X11 alors pas besoin d'installer `xvfb`. Plus d'infos à <https://www.itopen.it/qgis-server-setup-notes/>.

Pour installer ce paquet :

```
apt install xvfb
```

Créez le fichier de service, `/etc/systemd/system/xvfb.service`, avec ce contenu :

```
[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
```

Activer, démarrer et vérifier le statut de `xvfb.service`:

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

Ensuite, selon votre serveur HTTP, vous devez configurer le paramètre **DISPLAY** ou utiliser directement **xvfb-run**.

### Avec Apache

Ensuite, vous pouvez configurer le paramètre **DISPLAY**.

Avec Apache, il vous suffit d'ajouter à votre configuration *FastCGI* (voir ci-dessus) :

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

Redémarrez maintenant Apache pour que la nouvelle configuration soit prise en compte :

```
systemctl restart apache2
```

### Avec NGINX

Vous pouvez alors utiliser directement **xvfb-run** ou configurer le paramètre **DISPLAY**.

- Avec `spawn-fcgi` utilisant `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
```

- Avec la variable d'environnement **DISPLAY** dans la configuration du serveur HTTP.

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

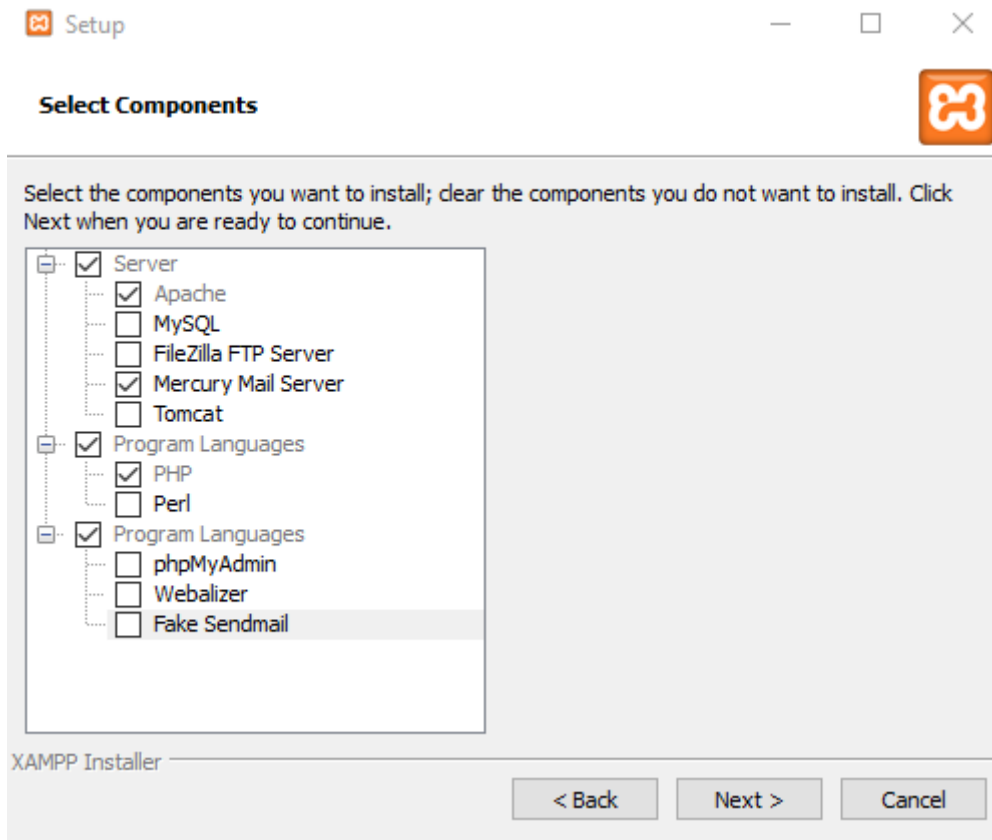
## 2.2 Installation sous Windows

QGIS Server peut également être installé sur les systèmes Windows. Bien que le package QGIS Server soit disponible dans la version 64 bits du programme d'installation réseau OSGeo4W (<https://qgis.org/en/site/forusers/download.html>), aucun package Apache (ou autre serveur Web) n'est disponible, donc cela doit être installé par d'autres moyens.

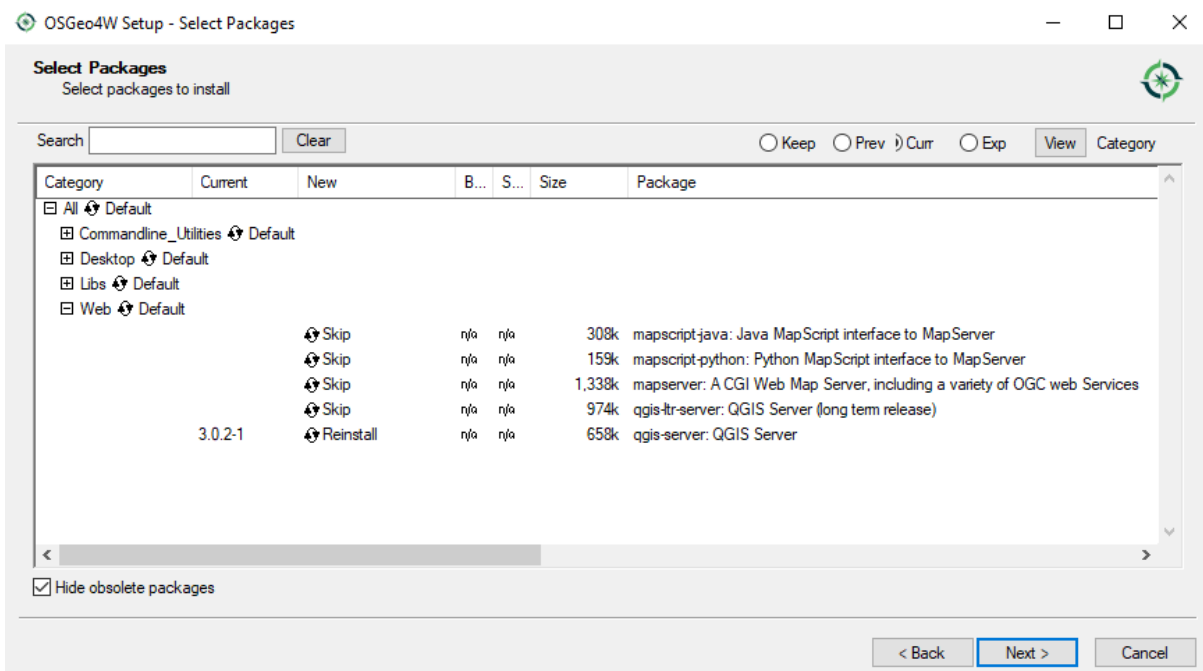
Une procédure simple est la suivante:

1. Téléchargez le programme d'installation de XAMPP (<https://www.apachefriends.org/download.html>) pour Windows et installez Apache





2. Téléchargez le programme d'installation d'OSGeo4W, suivez « Installation avancée » et installez les packages QGIS Desktop et QGIS Server



3. Modifiez le fichier httpd.conf ( C : \xampp\apache\conf\httpd.conf si les chemins d'installation par défaut ont été utilisés) et apportez les modifications suivantes:

De :

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

À :

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

De :

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

À :

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

De :

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

À :

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

#### 4. Puis ajoutez à la fin de httpd.conf :

```
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. Redémarrez le serveur Web Apache à partir du panneau de configuration XAMPP et ouvrez la fenêtre du navigateur pour tester une demande GetCapabilities vers QGIS Server

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

## 2.3 Publier un projet

Maintenant que QGIS Server est installé et fonctionne, il ne nous reste plus qu'à l'utiliser.

De toute évidence, nous avons besoin d'un projet QGIS pour travailler. Bien sûr, vous pouvez personnaliser entièrement votre projet en définissant les coordonnées, préciser certaines restrictions sur CRS ou même exclure certaines couches. Tout ce que vous devez savoir à ce sujet est décrit plus loin dans *Configurer votre projet*.

Mais pour l'instant, nous allons utiliser un projet simple déjà configuré et préalablement téléchargé dans `/home/qgis/projects/world.qgs`, comme décrit ci-dessus.

En ouvrant le projet et en jetant un coup d'œil sur les couches, nous savons que 4 couches sont actuellement disponibles:

- airports
- places
- countries
- countries\_shapeburst

Vous n'avez pas besoin de comprendre la demande complète pour l'instant mais vous pouvez récupérer une carte avec certaines des couches précédentes grâce à QGIS Server en faisant quelque chose comme ceci dans votre navigateur Web pour récupérer la couche *countries*:

```
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
```

Si vous obtenez l'image suivante, QGIS Server s'exécute correctement:

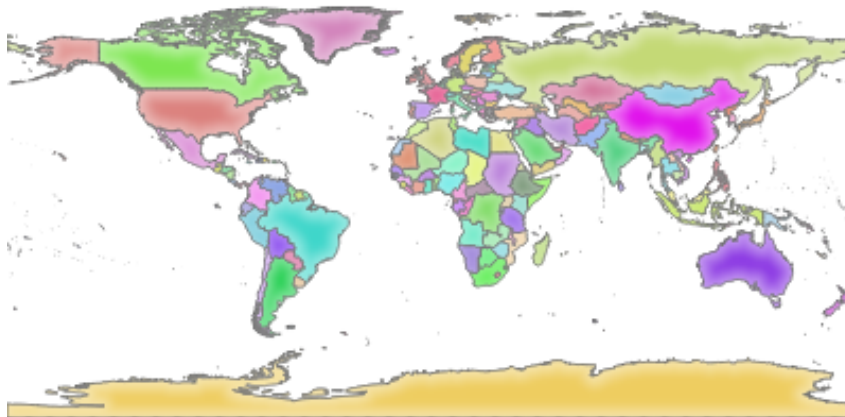


Figure2.1: Réponse du serveur à une demande GetMap de base

Notez que vous pouvez définir la variable d'environnement **QGIS\_PROJECT\_FILE** afin d'utiliser un projet par défaut au lieu de donner un paramètre **MAP** (voir *Variables d'environnement*).

Par exemple avec `spawn-fcgi` :

```
export QGIS_PROJECT_FILE=/home/qgis/projects/world.qgs
spawn-fcgi -f /usr/lib/bin/cgi-bin/qgis_mapserv.fcgi \
```

(suite sur la page suivante)

(suite de la page précédente)

```
-s /var/run/qgisserver.socket \  
-U www-data -G www-data -n
```

## 2.4 Configurer votre projet

Pour fournir des services WMS, WFS ou WCS via QGIS Server, il vous faut un fichier de projet QGIS avec des données à paramétrer. Définissez d'abord les couleurs et les styles des couches dans QGIS et le SCR du projet, s'ils ne sont pas déjà définis.

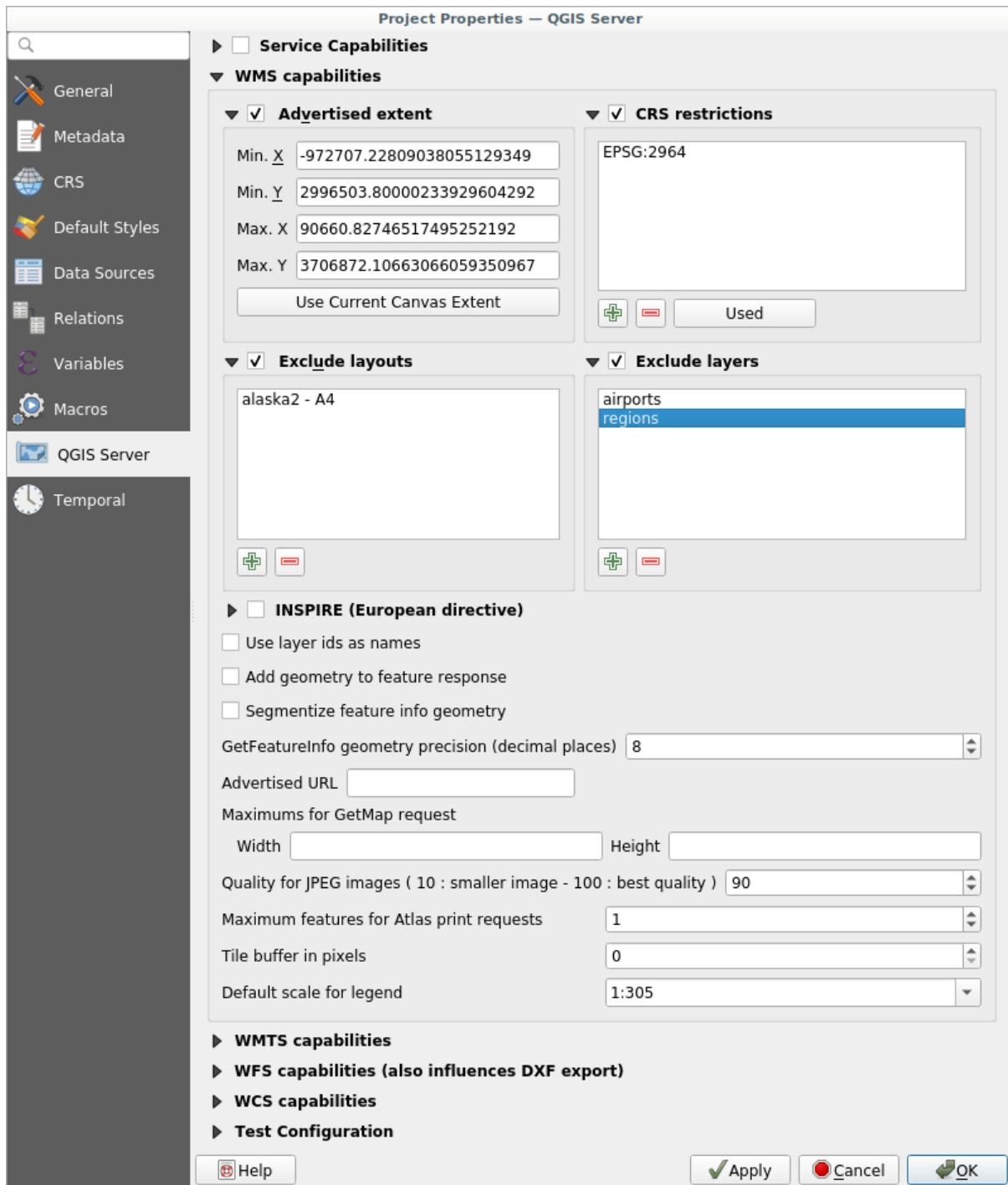





Figure2.2: Définitions pour un projet WMS/WFS/WCS de QGIS Server

Ensuite, allez dans le menu *QGIS Server* de la boîte de dialogue *Projet* ► *Propriétés...* et fournissez des informations sur OWS dans les champs sous *Capacités du Service*. Cela apparaîtra dans la réponse *GetCapabilities* du WMS, WFS ou WCS. Si vous ne cochez pas  *Capacités du service*, QGIS Server utilisera les informations fournies dans le fichier `wms_metadata.xml` situé dans le dossier `cgi-bin`.

## 2.4.1 Capacités WMS

Dans la section *Capacités WMS*, vous pouvez définir l'étendue publiée dans la réponse WMS *GetCapabilities* en entrant les valeurs X et Y minimales et maximales dans les champs sous *Etendue publiée*. En cliquant sur *Utiliser l'étendue actuelle du canevas* cela définit les valeurs actuellement affichées dans le canevas de carte QGIS. En cochant  *CRS restrictions*, vous pouvez restreindre dans quels systèmes de référence de coordonnées (CRS) QGIS Server proposera de rendre les cartes. Il est recommandé de restreindre le CRS proposé car cela réduit la taille de la réponse WMS *GetCapabilities*. Utilisez le  bouton ci-dessous pour sélectionner ces CRS dans le sélecteur de système de référence de coordonnées, ou cliquez sur *Utilisé* pour ajouter les CRS utilisés dans le projet QGIS à la liste.

Si vous avez défini des mises en page dans votre projet, elles seront listées dans la réponse *GetProjectSettings* et peuvent être utilisées par la requête *GetPrint* pour créer des impressions, en utilisant l'une des mises en page d'impression comme modèle. Il s'agit d'une extension spécifique à QGIS de la spécification WMS 1.3.0. Si vous souhaitez exclure toute mise en page d'impression de la publication WMS, cochez la case  *Exclure des mises en page* et cliquez sur le bouton  ci-dessous. Ensuite, sélectionnez une mise en page dans la boîte de dialogue *Sélectionner mise en page* afin de l'ajouter à la liste des mises en page exclues.

Si vous souhaitez exclure une couche ou un groupe de couches de la publication WMS,  *Exclure couches* et cliquez sur le  bouton ci-dessous. Cela ouvre la boîte de dialogue *Sélectionner les couches et groupes restreints*, qui vous permet de choisir les couches et les groupes que vous ne souhaitez pas publier. Utilisez la touche *Shift* ou *Ctrl* si vous souhaitez sélectionner plusieurs entrées. Il est recommandé d'exclure de la publication les couches dont vous n'avez pas besoin car cela réduit la taille de la réponse WMS *GetCapabilities*, ce qui entraîne des temps de chargement plus rapides côté client.

Si vous cochez  *utiliser les id des couches*, les ids de couche seront utilisés pour référencer les couches dans la réponse *GetCapabilities* ou le paramètre *GetMap* `LAYERS`. Sinon, le nom de la couche ou le nom court s'il est défini (voir *vectorservermenu*) est utilisé.

Vous pouvez recevoir les *GetFeatureInfo* en texte brut, XML et GML. La valeur par défaut est XML.

Si vous le souhaitez, vous pouvez cocher la case  *Ajouter la géométrie à la réponse d'entité*. Cela comprendra l'étendue pour chaque entité dans la réponse *GetFeatureInfo*. Voir aussi le paramètre `WITH_GEOMETRY`.

Comme de nombreux clients Web ne peuvent pas afficher d'arcs de cercle dans les géométries, vous avez la possibilité de segmenter la géométrie avant de l'envoyer au client dans une réponse *GetFeatureInfo*. Cela permet à ces clients d'afficher toujours la géométrie d'une entité (par exemple pour mettre en évidence l'entité). Vous devez  *Segmenter la géométrie des informations d'entité* pour activer l'option.

Vous pouvez également utiliser l'option *Précision de la géométrie GetFeatureInfo*. Cela vous permet d'économiser de la bande passante lorsque vous n'avez pas besoin de la précision max.

Si vous souhaitez que QGIS Server publie des URL de demandes spécifiques dans la réponse WMS *GetCapabilities*, entrez l'URL correspondante dans le champ *URL publiée*.

De plus, vous pouvez restreindre la taille maximale des cartes renvoyées par la demande *GetMap* en entrant la largeur et la hauteur maximales dans les champs respectifs sous *Maximum de requêtes GetMap*.

Vous pouvez modifier le facteur *Qualité d'image JPEG*. Le paramètre de qualité doit être compris entre 0 et 100. Précisez 0 pour une compression maximale, 100 pour aucune compression.

Vous pouvez modifier la limite des entités de l'atlas à imprimer dans une requête en définissant le champ *Nombre Entités maximums pour l'impression en atlas*.

Lorsque QGIS Server est utilisé en mode tuilé (voir le paramètre `TILED`), vous pouvez définir le *Tampon de tuiles en pixels*. La valeur recommandée est la taille du plus grand symbole ou de la plus grande largeur de ligne dans votre projet QGIS.

Si une de vos couches utilise l'affichage d'indication de carte (pour afficher du texte issu d'expressions), cette dernière sera listée dans la sortie *GetFeatureInfo*. Si la couche utilise une Valeur Relationnelle pour un de ses attributs, cette information sera également renvoyée dans la sortie *GetFeatureInfo*.

## 2.4.2 Capacités WFS

Dans la partie *Capacités WFS*, vous pouvez sélectionner les couches que vous voulez publier en WFS et indiquer si elle permettent les opérations de mise à jour, d'insertion et de suppression. Si vous ajoutez une URL dans le champ *URL publiée* des *Capacités WFS*, QGIS Server annoncera cette URL spécifique dans la réponse WFS GetCapabilities.

## 2.4.3 Capacités WCS

Dans la partie *Capacités WCS*, vous pouvez sélectionner les couches qui vous voulez publier en WCS. Si vous indiquez une URL dans le champ *URL publiée* de la section *Capacités WCS*, QGIS Server annoncera cette URL spécifique dans la réponse WCS GetCapabilities.

## 2.4.4 Bien configurer votre OWS

Pour les couches vectorielles, le menu *Champs* de la boîte de dialogue *Couche ► Propriétés de la couche* vous permet de définir pour chaque attribut s'il sera publié ou non. Par défaut, tous les attributs sont publiés en WMS et WFS. Si vous ne voulez pas qu'un attribut spécifique soit publié, décochez la case correspondante dans la colonne *WMS* ou *WFS*.

Vous pouvez superposer des filigranes sur les cartes produites par votre WMS en ajoutant des annotations de texte ou des annotations SVG au fichier de projet. Voir la section *sec\_annotations* pour des instructions sur la création d'annotations. Pour que les annotations soient affichées sous forme de filigranes sur la sortie WMS, la case à cocher *Position de la carte fixe* dans la boîte de dialogue *Texte d'annotation* doit être décochée. Vous pouvez y accéder en double-cliquant sur l'annotation alors que l'un des outils d'annotation est actif. Pour les annotations SVG, vous devrez soit définir le projet pour enregistrer des chemins absolus (dans le menu *Général* de *Projet ► Propriétés...*) soit modifier manuellement le chemin d'accès à l'image SVG afin qu'elle représente un chemin relatif valide.

## 2.5 Intégration avec des applications tierces

QGIS Server fournit des services cartographiques web conformes aux standards de l'OGC tels que le [WMS](#), [WFS](#), [etc.](#). De ce fait, il peut être utilisé par une grande variété de logiciels.

### 2.5.1 Intégration avec QGIS

QGIS Desktop est l'outil de « design » de la carte alors que QGIS Server permet de publier celle-ci. Les cartes ou projets QGIS seront publiés par QGIS Server selon les protocoles et standards OGC. Ces projets QGIS peuvent être sous forme de fichiers sur disque ou d'entrées de bases de données (via *Projet ► Enregistrer sous ► PostgreSQL* depuis QGIS Desktop).

En outre, une procédure dédiée de mise à jour doit être établie pour rafraîchir le projet utilisé par QGIS Server (c'est-à-dire, copier les fichiers projets dans l'emplacement du serveur et redémarrer QGIS Server). Pour le moment, les processus automatisés (tels que le rechargement du serveur par dessus le service de file d'attente de messages) ne sont pas implémentés.

### 2.5.2 Intégration avec MapProxy

MapProxy est un serveur de cache de tuiles pouvant se connecter à tout serveur cartographique respectant les protocoles WMS/WMTS. De ce fait, il peut se connecter directement aux services web cartographiques de QGIS Server et ainsi améliorer l'expérience utilisateur.

### 2.5.3 Intégration avec QWC2

QWC2 est une application web responsive dédiée à QGIS Server. Il vous aide à construire un visualiseur de carte hautement personnalisé avec sélection de couches, informations sur les entités... En outre, de multiples extensions sont disponibles notamment pour l'authentification, l'impression... La liste complète est disponible sur ce [dépôt](#).



QGIS Server est capable de servir des données selon des protocoles standard tels que décrits par l'Open Geospatial Consortium (OGC):

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

Des paramètres et des demandes supplémentaires du fournisseur sont pris en charge en plus de la norme d'origine qui améliorent considérablement les possibilités de personnalisation de son comportement grâce au moteur de rendu QGIS.

### 3.1 Web Map Service (WMS)

Les normes WMS 1.1.1 et 1.3.0 mises en œuvre dans QGIS Server fournissent une interface HTTP pour demander des images de cartes ou de légendes générées à partir d'un projet QGIS. Une requête WMS typique définit le projet QGIS à utiliser, les couches à rendre ainsi que le format d'image à générer. Un support de base est également disponible pour les **Styled Layer Descriptor (SLD)**.

Spécifications :

- WMS 1.1.1
- WMS 1.3.0
- SLD 1.1.0 WMS profile

Requêtes standard fournies par QGIS Server:

Requête	Description
<i>GetCapabilities</i>	Renvoie des métadonnées XML avec des informations sur le serveur
<i>GetMap</i>	Renvoie une carte
<i>GetFeatureInfo</i>	Récupère les données (géométrie et valeurs) pour un emplacement de pixel
<i>GetLegendGraphics</i>	Renvoie les symboles de légende

Demandes des fournisseurs fournies par QGIS Server:

Requête	Description
<i>GetPrint</i>	Renvoie une composition QGIS
<i>GetProjectSettings</i>	Renvoie des informations spécifiques sur QGIS Server

### 3.1.1 GetMap

Paramètres standard pour la requête **GetMap** selon les spécifications OGC WMS 1.1.1 et 1.3.0 :

Paramètre	Requis	Description
<i>SERVICE</i>	Oui	Nom du service ( <b>WMS</b> )
<i>VERSION</i>	Oui	Version du service
<i>REQUEST</i>	Oui	Nom de la requete ( <b>GetMap</b> )
<i>LAYERS</i>	Non	Couches à afficher
<i>STYLES</i>	Non	Style des couches
<i>SRS / CRS</i>	Oui	Système de coordonnées de référence
<i>BBOX</i>	Oui	Emprise de la carte
<i>WIDTH</i>	Oui	Largeur de l'image en pixels
<i>HEIGHT</i>	Oui	Hauteur de l'image en pixels
<i>FORMAT</i>	Non	Format de l'image
<i>TRANSPARENT</i>	Non	Arrière-plan transparent
<i>SLD</i>	Non	URL d'un SLD à utiliser pour la symbologie
<i>SLD_BODY</i>	Non	Contenu de SLD (XML) à utiliser pour la symbologie

Outre les paramètres standard, QGIS prend en charge les paramètres suivants :

Paramètre	Requis	Description
<i>MAP</i>	Oui	Spécifiez le fichier de projet QGIS
<i>BGCOLOR</i>	Non	Spécifiez la couleur d'arrière-plan
<i>DPI</i>	Non	Spécifiez la résolution de sortie
<i>IMAGE_QUALITY</i>	Non	Compression JPEG
<i>OPACITIES</i>	Non	Opacité pour une couche ou un groupe
<i>FILTER</i>	Non	Sous-ensemble d'entités
<i>SELECTION</i>	Non	Mettre en évidence d'entités
<i>FILE_NAME</i>	Non	Nom du fichier téléchargé Seulement pour <i>FORMAT=application/dxf</i>
<i>FORMAT_OPTIONS</i>	Non	Options du format de fichier spécifié Seulement pour <i>FORMAT=application/dxf</i>
<i>TILED</i>	Non	Travailler en <i>mode tuile</i>

Exemple d'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
```

(suite sur la page suivante)

(suite de la page précédente)

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

## SERVICE

Ce paramètre doit être WMS.

Par exemple:

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

## VERSION

Ce paramètre permet de spécifier la version du service à utiliser. Les valeurs disponibles pour le paramètre VERSION sont:

- 1.1.1
- 1.3.0

Selon le numéro de version, de légères différences sont à prévoir comme expliqué plus loin pour les paramètres suivants:

- CRS / SRS
- BBOX

## REQUEST

Ce paramètre est GetMap en cas de requête **GetMap**.

## LAYERS

Ce paramètre permet de spécifier les couches à afficher sur la carte. Les noms doivent être séparés par une virgule.

De plus, QGIS Server a introduit certaines options pour sélectionner des couches en:

- un nom court
- l'identifiant de la couche

Le nom court d'une couche peut être configuré via les *Propriétés* ► *Métadonnées* de la couche. Si le nom court est défini, il est utilisé par défaut au lieu du nom de la couche:

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

De plus, il existe une option de projet permettant de sélectionner les couches par leur identifiant dans *OWS Server* -> *WMS capacités* du *Projet* -> *Propriétés ...* . Pour activer cette option, la case à cocher *:guilabel:'Utiliser les identifiants des couches comme noms* doit être cochée.

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

### STYLES

Ce paramètre peut être utilisé pour spécifier le style d'une couche pour l'étape de rendu. Les styles doivent être séparés par une virgule. Le nom du style par défaut est `par défaut`.

### SRS / CRS

Ce paramètre permet d'indiquer le système de référence spatiale de la sortie de la carte dans **WMS 1.1.1** et doit être formé comme `EPSG:XXXX`. Notez que `CRS` est également supporté si la version actuelle est **1.1.1**.

Pour **WMS 1.3.0**, le paramètre `CRS` est préférable mais `SRS` est également pris en charge.

Notez que si les paramètres `CRS` et `SRS` sont indiqués dans la demande, c'est la version actuelle indiquée dans le paramètre `VERSION` qui est déterminante.

Dans le cas suivant, le paramètre `SRS` est conservé quel que soit le paramètre `VERSION` car `CRS` n'est pas indiqué:

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

Dans le cas suivant, le paramètre `SRS` est conservé au lieu de `CRS` à cause du paramètre `VERSION`:

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

Dans le cas suivant, le paramètre `CRS` est conservé au lieu de `SRS` à cause du paramètre `VERSION`:

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

## BBOX

Ce paramètre permet de spécifier l'étendue de la carte avec des unités en fonction du SCR actuel. Les coordonnées doivent être séparées par une virgule.

Le paramètre BBOX est formé par `min_a`, `min_b`, `max_a`, `max_b` mais les axes a et ``b`` peuvent varier en fonction du paramètre actuel de `VERSION`

- En WMS **1.1.1**, l'ordre des axes est toujours est/nord
- En WMS **1.3.0**, l'ordre des axes dépend du SCR

Par exemple, dans le cas d'un SCR EPSG : 4326 et du protocole WMS en version **1.1.1**, a est la longitude (est) et b la latitude (nord), la requête est donc la suivante :

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

Mais dans le cas du protocole WMS en version **1.3.0**, l'ordre des axes définis dans la base EPSG est nord/est donc a est la latitude et b la longitude :

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

## WIDTH

Ce paramètre permet de spécifier la largeur en pixels de l'image de sortie.

## HEIGHT

Ce paramètre permet de spécifier la hauteur en pixels de l'image de sortie.

## FORMAT

Ce paramètre peut être utilisé pour spécifier le format de l'image de la carte. Les valeurs disponibles sont:

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

- `application/dxf` Seules les couches qui ont un accès en lecture dans le service WFS sont exportées au format DXF.

Exemple d'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:SYMBOLLAYERSYMBOLOLOGY;SCALE:250&FILE_NAME=plan.dxf
```

### TRANSPARENT

Ce paramètre booléen peut être utilisé pour spécifier la transparence d'arrière-plan. Les valeurs disponibles sont (non sensibles à la casse):

- TRUE
- FALSE

Cependant, ce paramètre est ignoré si le format de l'image de la carte indiqué par `FORMAT` est différent de PNG.

### MAP

Ce paramètre permet de définir le fichier projet QGIS à utiliser.

Comme mentionné dans la [table des paramètres GetMap](#), `MAP` est obligatoire car une requête a besoin d'un projet QGIS pour fonctionner. Cependant, la variable d'environnement `QGIS_PROJECT_FILE` peut être utilisée pour définir un projet QGIS par défaut. Dans ce cas spécifique, `MAP` n'est plus un paramètre obligatoire. Pour plus d'informations, vous pouvez vous référer à [Configuration avancée](#).

### BGCOLOR

Ce paramètre permet d'indiquer une couleur de fond pour l'image de la carte. Cependant, il ne peut pas être combiné avec le paramètre `TRANSPARENT` dans le cas d'images PNG (la transparence est prioritaire). La couleur peut être littérale ou en notation hexadécimale.

Exemple d'URL avec la notation littérale:

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

Exemple d'URL avec la notation hexadécimale:

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

## DPI

Ce paramètre peut être utilisé pour spécifier la résolution de sortie demandée.

## IMAGE\_QUALITY

Ce paramètre n'est utilisé que pour les images JPEG. Par défaut, la compression JPEG est -1.

Vous pouvez modifier la valeur par défaut par projet QGIS dans le menu *OWS Server -> WMS capabilities* du menu *Projet -> Propriétés ...*. Si vous souhaitez le remplacer dans une demande *GetMap*, vous pouvez le faire en utilisant le paramètre `IMAGE_QUALITY`.

## OPACITIES

Liste des valeurs d'opacité séparées par des virgules. L'opacité peut être définie au niveau de la couche ou du groupe. Les valeurs autorisées vont de 0 (totalement transparent) à 255 (totalement opaque).

## FILTER

Un sous-ensemble de couches peut être sélectionné avec le paramètre `FILTER`. La syntaxe est fondamentalement la même que pour la chaîne de sous-ensemble QGIS. Cependant, il existe certaines restrictions pour éviter les injections SQL dans les bases de données via QGIS Server. Si une chaîne dangereuse est trouvée dans le paramètre, QGIS Server renverra l'erreur suivante:

```
<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>
```

Exemple d'URL :

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

Dans cet exemple, le même filtre (le champ `col2` est égal à la chaîne `blabla`) est appliqué aux couches `mylayer1` et `mylayer2`, tandis que le filtre sur `col1` est uniquement appliqué à `mylayer1`.

---

**Note:** Il est possible d'effectuer des recherches d'attributs via `GetFeatureInfo` et d'omettre le paramètre `X / Y` si un `FILTRE` est là. QGIS Server renvoie ensuite des informations sur les entités correspondantes et génère un cadre de délimitation combiné dans la sortie XML.

---

### SELECTION

Le paramètre `SELECTION` peut mettre en évidence les entités d'une ou plusieurs couches. Les entités vectorielles peuvent être sélectionnées en passant des listes séparées par des virgules avec des identifiants d'entités.

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

L'image suivante présente la réponse d'une demande `GetMap` en utilisant l'option `SELECTION` par ex. `http://myserver.com/...&SELECTION=countries: 171,65`.

Comme ces identifiants d'entités correspondent dans l'ensemble de données source à **France** et **Roumanie**, ils sont surlignés en jaune.



Figure3.1: Réponse du serveur à une demande `GetMap` avec le paramètre `SELECTION`



## FORMAT-OPTIONS

Ce paramètre peut être utilisé pour spécifier des options pour le format sélectionné. Uniquement pour « FORMAT=application/dxf ». Une liste de paires clé:valeur séparées par un point-virgule :

- **ÉCHELLE**: à utiliser pour les règles de symbologie, les filtres et les styles (pas la mise à l'échelle réelle des données - les données restent à l'échelle d'origine).
- **MODE**: correspond aux options d'exportation proposées dans la boîte de dialogue d'exportation QGIS Desktop DXF. Les valeurs possibles sont `NOSYMBOLLOGY`, `FEATURESYMBOLLOGY` et `SYMBOLLAYERSYMBOLLOGY`.
- **LAYERSATTRIBUTES**: spécifiez un champ qui contient des valeurs pour les noms de couche DXF - s'il n'est pas spécifié, les noms de couche QGIS d'origine sont utilisés.
- **USE\_TITLE\_AS\_LAYERNAME**: si activé, le titre de la couche sera utilisé comme nom de la couche.
- **CODEC**: spécifiez un codec à utiliser pour l'encodage. La valeur par défaut est `ISO-8859-1`, vérifiez la boîte de dialogue d'exportation DXF du bureau QGIS pour les valeurs valides.
- **NO\_MTEXT** : Utilisez `TEXT` au lieu de `MTEXT` pour les étiquettes.
- **FORCE\_2D** : Force la sortie 2D. Ceci est nécessaire pour la largeur de la polyligne.

## TILED

Pour des raisons de performance, QGIS server peut être utilisé en mode « tiled ». Dans ce mode, le client demande plusieurs petites tuiles de taille fixe, et les assemblent pour former la carte entière. Ce faisant, les symboles à la limite ou près de la limite entre deux tuiles peuvent apparaître coupés, car ils ne sont présents que dans une des tuiles.

Mettez le paramètre `TILED` à `TRUE` pour indiquer au logiciel QGIS Server de travailler en mode *tiled*, et d'appliquer le *Tile buffer* configuré dans le projet QGIS (voir *Configurer votre projet*).

Lorsque `TILED` est `VRAI` et lorsqu'un tampon de tuile différent de zéro est configuré dans le projet QGIS, les entités en dehors de l'étendue de tuile sont dessinées pour éviter les symboles coupés aux limites des tuiles.

`TILED` est par défaut `FALSE`.

### 3.1.2 GetFeatureInfo

Paramètres standard pour la requête **GetFeatureInfo** selon les spécifications OGC WMS 1.1.1 et 1.3.0 :

Paramètre	Requis	Description
<i>SERVICE</i>	Oui	Nom du service ( <b>WMS</b> )
<i>VERSION</i>	Non	Version du service
<i>REQUEST</i>	Oui	nom de la requete ( <b>GetFeatureInfo</b> )
<i>QUERY_LAYERS</i>	Oui	Couches à interroger
<i>LAYERS</i>	Oui	Couches de données à afficher (identique à <i>QUERY_LAYERS</i> )
<i>STYLES</i>	Non	Style des couches
<i>SRS</i> / <i>CRS</i>	Oui	Système de coordonnées de référence
<i>BBOX</i>	Non	Emprise de la carte
<i>WIDTH</i>	Oui	Largeur de l'image en pixels
<i>HEIGHT</i>	Oui	Hauteur de l'image en pixels
<i>TRANSPARENT</i>	Non	Arrière-plan transparent
<i>INFO_FORMAT</i>	Non	Format de sortie
<i>FEATURE_COUNT</i>	Non	Nombre maximum d'entités à renvoyer
<i>I</i>	Non	Colonne de pixels du point à interroger
<i>X</i>	Non	Identique au paramètre <i>I</i> , mais dans WMS 1.1.1
<i>J</i>	Non	Ligne de pixels du point à interroger
<i>Y</i>	Non	Identique au paramètre <i>J</i> , mais dans WMS 1.1.1
<i>WMS_PRECISION</i>	Non	La précision (nombre de chiffres) à utiliser lors du retour de la géométrie (voir <i>how to add geometry to feature response</i> ). La valeur par défaut est « -1 », ce qui signifie que la précision définie dans le projet est utilisée.

Outre les paramètres standard, QGIS prend en charge les paramètres suivants :

Paramètre	Requis	Description
<i>MAP</i>	Oui	Spécifiez le fichier de projet QGIS
<i>FILTER</i>	Non	Sous-ensemble d'entités
<i>FI_POINT_TOLERANCE</i>	Non	Tolérance en pixels pour les couches de points
<i>FI_LINE_TOLERANCE</i>	Non	Tolérance en pixels pour les couches de lignes
<i>FI_POLYGON_TOLERANCE</i>	Non	Tolérance en pixels pour les couches de polygones
<i>FILTER_GEOM</i>	Non	Filtrage géométrique
<i>WITH_MAPTIP</i>	Non	Ajouter des conseils de carte à la sortie
<i>WITH_GEOMETRY</i>	Non	Ajouter une géométrie à la sortie

Exemple d'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

Ce paramètre est `GetFeatureInfo` en cas de requête **GetFeatureInfo**.

## INFO\_FORMAT

Ce paramètre peut être utilisé pour spécifier le format du résultat. Les valeurs disponibles sont:

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

## QUERY\_LAYERS

Ce paramètre spécifie les couches à afficher sur la carte. Les noms sont séparés par une virgule.

De plus, QGIS Server introduit des options pour sélectionner des couches en:

- nom court
- identifiant de couche

Voir le paramètre `LAYERS` défini dans *GetMap* pour plus d'informations.

## FEATURE\_COUNT

Ce paramètre spécifie le nombre maximal d'entités par couche à renvoyer. Par exemple, si `QUERY_LAYERS` est défini sur `layer1, layer2` et que `FEATURE_COUNT` est défini sur 3, un maximum de 3 entités de `layer1` seront renvoyées. De même, un maximum de 3 entités de la couche 2 sera retourné.

Par défaut, une seule entité par couche est renvoyée.

## I

Ce paramètre, défini dans WMS 1.3.0, vous permet de spécifier la colonne de pixels du point de requête.

## X

Même paramètre que `I`, mais défini dans WMS 1.1.1.

## J

Ce paramètre, défini dans WMS 1.3.0, vous permet de spécifier la ligne de pixels du point de requête.

### Y

Même paramètre que J, mais défini dans WMS 1.1.1.

### FI\_POINT\_TOLERANCE

Ce paramètre spécifie la tolérance en pixels pour les couches de points.

### FI\_LINE\_TOLERANCE

Ce paramètre spécifie la tolérance en pixels pour les couches de lignes.

### FI\_POLYGON\_TOLERANCE

Ce paramètre spécifie la tolérance en pixels pour les couches de polygones.

### FILTER\_GEOM

Ce paramètre spécifie une géométrie WKT avec laquelle les entités doivent se croiser.

### WITH\_MAPTIP

Ce paramètre spécifie s'il faut ajouter des conseils de mappage à la sortie.

Les valeurs disponibles sont (non sensibles à la casse):

- TRUE
- FALSE

### WITH\_GEOMETRY

Ce paramètre spécifie s'il faut ajouter des géométries à la sortie. Pour l'utiliser, il vous faut d'abord activer l'option *Ajouter la géométrie à la réponse d'une entité* dans le projet QGIS. Voir *Configurer son projet*.

Les valeurs disponibles sont (non sensibles à la casse):

- TRUE
- FALSE

## 3.1.3 GetPrint

QGIS Server a la capacité de créer une sortie de mise en page d'impression au format pdf ou pixel. Les fenêtres de mise en page d'impression dans le projet publié sont utilisées comme modèles. Dans la demande **GetPrint**, le client a la possibilité de spécifier les paramètres des mises en page et des étiquettes contenues.

Paramètres de la demande **GetPrint**:

Paramètre	Requis	Description
<i>MAP</i>	Oui	Spécifiez le fichier de projet QGIS
<i>SERVICE</i>	Oui	Nom du service (WMS)
<i>VERSION</i>	Non	Version du service
<i>REQUEST</i>	Oui	Nom de la requête ( <b>GetPrint</b> )
<i>LAYERS</i>	Non	Couches à afficher
<i>TEMPLATE</i>	Oui	Modèle de mise en page à utiliser
<i>SRS / CRS</i>	Oui	Système de coordonnées de référence
<i>FORMAT</i>	Non	Format de sortie
<i>ATLAS_PK</i>	Non	Entité atlas
<i>STYLES</i>	Non	Style des couches
<i>TRANSPARENT</i>	Non	Arrière-plan transparent
<i>OPACITIES</i>	Non	Opacité pour une couche ou un groupe
<i>SELECTION</i>	Non	Mettre en évidence d'entités
<i>mapX:EXTENT</i>	Non	Étendue de la carte "X"
<i>mapX:LAYERS</i>	Non	Couches de la carte "X"
<i>mapX:STYLES</i>	Non	Style de couche de la carte "X"
<i>mapX:SCALE</i>	Non	Échelle des couches de la carte "X"
<i>mapX:ROTATION</i>	Non	Rotation de la carte "X"
<i>mapX:GRID_INTERVAL_X</i>	Non	Intervalle de grille sur l'axe x de la carte "X"
<i>mapX:GRID_INTERVAL_Y</i>	Non	Intervalle de grille sur l'axe y de la carte "X"

Exemple d'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
```

Notez que le modèle de mise en page peut contenir plusieurs cartes. De cette façon, si vous souhaitez configurer une carte spécifique, vous devez utiliser les paramètres `mapX:` où X est un nombre positif que vous pouvez récupérer grâce à la requête **GetProjectSettings**.

Par exemple:

```
<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

Ce paramètre doit être `GetPrint` pour la demande **GetPrint**.

### TEMPLATE

Ce paramètre peut être utilisé pour spécifier le nom d'un modèle de mise en page à utiliser pour l'impression.

### FORMAT

Ce paramètre spécifie le format de l'image de la carte. Les valeurs disponibles sont:

- `png` (valeur par défaut)
- `image/png`
- `jpg`
- `jpeg`
- `image/jpeg`
- `svg`
- `image/svg`
- `image/svg+xml`
- `pdf`
- `application/pdf`

Si le paramètre `FORMAT` est différent de l'une de ces valeurs, une exception est renvoyée.

### ATLAS\_PK

Ce paramètre permet d'activer le rendu Atlas en indiquant quelles entités nous voulons imprimer. Afin de récupérer un atlas avec toutes les fonctionnalités, le symbole `*` peut être utilisé (selon le nombre maximum d'entités autorisé dans la configuration du projet).

Lorsque `FORMAT` est `pdf`, un seul document PDF combinant les pages d'entités est renvoyé. Pour tous les autres formats, une seule page est renvoyée.

### mapX:EXTENT

Ce paramètre spécifie l'étendue d'un élément de mise en page sous la forme `xmin, ymin, xmax, ymax`.

### mapX:ROTATION

Ce paramètre spécifie la rotation de la carte en degrés.

### mapX:GRID\_INTERVAL\_X

Ce paramètre spécifie la densité du quadrillage dans la direction X.

### mapX:GRID\_INTERVAL\_Y

Ce paramètre spécifie la densité de la ligne de grille dans la direction Y.

### mapX:SCALE

Ce paramètre spécifie l'échelle de la carte pour un élément de la mise en page. Ceci est utile pour assurer une visibilité basée sur l'échelle des couches et des étiquettes même si le client et le serveur peuvent avoir des algorithmes différents pour calculer le dénominateur d'échelle.

### mapX:LAYERS

Ce paramètre spécifie les couches pour un élément de la carte dans la mise en page. Voir *GetMap Layers* pour plus d'informations sur ce paramètre.

### mapX:STYLES

Ce paramètre spécifie les styles de couches définis dans un élément spécifique de la carte de mise en page. Voir *GetMap Styles* pour plus d'informations sur ce paramètre.

## 3.1.4 GetLegendGraphics

Plusieurs paramètres supplémentaires sont disponibles pour modifier la taille des éléments de légende:

- **BOXSPACE** espace entre le cadre de légende et le contenu (mm)
- **FORMAT**, image/jpeg, image/png ou application/json. Pour le JSON, les symboles sont encodés en Base64 et la plupart des autres options relatives à la mise en page ou aux polices ne sont pas prises en compte car la légende doit être construite côté client.
- **LAYERSPACE** espace vertical entre les couches (mm)
- **LAYERTITLESPACE** espace vertical entre le titre de la couche et les éléments suivants (mm)
- **SYMBOLSPACE** espace vertical entre le symbole et l'élément suivant (mm)
- **ICONLABELSPACE** espace horizontal entre le symbole et le texte de l'étiquette (mm)
- **SYMBOLWIDTH** largeur de l'aperçu du symbole (mm)
- **SYMBOLHEIGHT** hauteur de l'aperçu du symbole (mm)

Ces paramètres modifient les propriétés de police des titres la couche et des éléments étiquettes :

- **LAYERFONTFAMILY/ITEMFONTFAMILY** famille de polices pour le titre de la couche / le texte de l'élément
- **LAYERFONTBOLD/ITEMFONTBOLD** TRUE pour utiliser une police en gras
- **LAYERFONTSIZE/ITEMFONTSIZE** Taille de police en point
- **LAYERFONTITALIC/ITEMFONTITALIC** TRUE pour utiliser la police italique
- **LAYERFONTCOLOR/ITEMFONTCOLOR** Code couleur hexadécimal (par exemple #FF0000 pour le rouge)
- **LAYERTITLE** « FALSE » pour obtenir uniquement les graphiques de la légende sans le titre de la couche

- **RULELABEL** :
  - FALSE pour obtenir uniquement les graphiques de légende sans les étiquettes
  - AUTO pour cacher l'étiquette de l'élément, pour les couches utilisant le rendu *Symbole unique*

Légende basée sur le contenu. Ces paramètres permettent au client de demander une légende affichant uniquement les symboles des entités se trouvant dans la zone demandée:

- **BBOX** la zone géographique pour laquelle la légende doit être construite
- **CRS / SRS** le référentiel de coordonnées adopté pour définir les coordonnées BBOX
- **SRCWIDTH / SRCHEIGHT** si elles sont définies, elles doivent correspondre aux paramètres **WIDTH** et **HEIGHT** de la requête `GetMap`, pour permettre à QGIS Server de mettre à l'échelle les symboles en fonction de la taille de l'image de la carte.

Les fonctionnalités de légende basées sur le contenu sont basées sur l'implémentation de UMN MapServer: <<https://www.mapserver.org/development/rfc/ms-rfc-101.html>> \_

- **SHOWFEATURECOUNT** s'il est défini sur `TRUE` ajoute dans la légende le nombre d'entités comme dans l'image suivante:



- **RULE** à définir sur un nom de règle donné pour n'obtenir que le symbole de règle nommé
- **WIDTH/HEIGHT** la taille de l'image de légende générée si le paramètre **RULE** est défini

### 3.1.5 GetProjectSettings

Ce type de demande fonctionne de manière similaire à **GetCapabilities**, mais il est plus spécifique à QGIS Server et permet à un client de lire des informations supplémentaires qui ne sont pas disponibles dans la sortie `GetCapabilities`:

- visibilité initiale des couches
- information sur les attributs des vecteurs et leur type d'éditeur
- information sur l'ordre des couches et de leur rendu
- liste des couches publiées en WFS
- montrer si un groupe dans l'arbre des couches est mutuellement exclusif



## 3.2 Web Feature Service (WFS)

Les normes **1.0.0** et **1.1.0** WFS implémentées dans QGIS Server fournissent une interface HTTP pour interroger les entités géographiques d'un projet QGIS. Une demande WFS typique définit le projet QGIS à utiliser et la couche à interroger.

Document de spécifications en fonction du numéro de version du service:

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

Requêtes standard fournies par QGIS Server:

Requête	Description
GetCapabilities	Renvoie des métadonnées XML avec des informations sur le serveur
GetFeature	Renvoie une sélection d'entité
DescribeFeatureType	Renvoie une description des types d'entités et des propriétés
Transaction	Permet aux entités d'être insérées, mises à jour ou supprimées

### 3.2.1 GetFeature

Paramètres standard pour la demande **GetFeature** selon les spécifications OGC WFS 1.0.0 et 1.1.0:

Paramètre	Requis	Description
SERVICE	Oui	Nom du service
VERSION	Non	Version du service
REQUEST	Oui	Nom de la requête
TYPENAME	Non	Nom des couches
OUTPUTFORMAT	Non	Format de sortie
RESULTTYPE	Non	Type de résultat
PROPERTYNAME	Non	Nom des propriétés à renvoyer
MAXFEATURES	Non	Nombre maximum d'entités à renvoyer
SRSNAME	Non	Système de coordonnées de référence
FEATUREID	Non	Filtre les entités par leur identifiant
FILTER	Non	Encodage du filtre OGC
BBOX	Non	Emprise de la carte
SORTBY	Non	Trie les résultats

Outre les paramètres standard, QGIS prend en charge les paramètres suivants :

Paramètre	Requis	Description
MAP	Oui	Spécifiez le fichier de projet QGIS
STARTINDEX	Non	Pagination
GEOMETRYNAME	Non	Type de géométrie à renvoyer
EXP_FILTER	Non	Expression de filtrage

### SERVICE

Ce paramètre doit être `WFS` en cas de requête **GetFeature**.

Par exemple:

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

### VERSION

Ce paramètre permet de spécifier la version du service à utiliser. Les valeurs disponibles pour le paramètre `VERSION` sont:

- 1.0.0
- 1.1.0

Si aucune version n'est indiquée dans la requête, alors 1.1.0 est utilisée par défaut.

Exemple d'URL :

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

### REQUEST

Ce paramètre est `GetFeature` en cas de requête **GetFeature**.

Exemple d'URL :

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

### RESULTTYPE

Ce paramètre peut être utilisé pour indiquer le genre de résultat attendu. Les valeurs possibles sont :

- `results`: le comportement par défaut
- `hits`: renvoie seulement le décompte des entités

Exemple d'URL :

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

## GEOMETRYNAME

Ce paramètre peut être utilisé pour indiquer le type de géométrie attendu pour les entités. Les valeurs possibles sont :

- extent
- centroid
- none

Exemple d'URL :

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

## STARTINDEX

Ce paramètre est standard dans WFS 2.0, mais c'est une extension pour WFS 1.0.0. En fait, il peut être utilisé pour ignorer certaines entités de l'ensemble de résultats et en combinaison avec MAXFEATURES, il offre la possibilité de parcourir les résultats.

Exemple d'URL :

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

## 3.3 Web Map Tile Service (WMTS)

La norme **1.0.0** WMTS implémentée dans QGIS Server fournit une interface HTTP pour demander des images de carte tuilées générées à partir d'un projet QGIS. Une demande WMTS typique a défini le projet QGIS à utiliser, certains paramètres WMS comme les couches à rendre, ainsi que les paramètres de tuile.

Document de spécifications du service:

- [WMTS 1.0.0](#)

Requêtes standard fournies par QGIS Server:

Requête	Description
GetCapabilities	Renvoie des métadonnées XML avec des informations sur le serveur
GetTile	Renvoie une tuile
GetFeatureInfo	Récupère les données (géométrie et valeurs) pour un emplacement de pixel

### 3.3.1 GetCapabilities

Paramètres standard pour la demande **GetCapabilities** selon les spécifications OGC WMTS 1.0.0:

Paramètre	Requis	Description
SERVICE	Oui	Nom du service (WMTS)
REQUEST	Oui	Nom de la requête (GetCapabilities)

Outre les paramètres standard, QGIS prend en charge les paramètres suivants :

Paramètre	Requis	Description
MAP	Oui	Spécifiez le fichier de projet QGIS

Exemple d'URL :

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

#### SERVICE

Ce paramètre doit être WMTS en cas de requête **GetCapabilities**.

#### REQUEST

Ce paramètre est *GetCapabilities* en cas de requête **GetCapabilities**.

#### MAP

Ce paramètre permet de définir le fichier projet QGIS à utiliser.

### 3.3.2 GetTile

Paramètres standard pour la demande **GetTile** selon les spécifications OGC WMTS 1.0.0:

Paramètre	Requis	Description
SERVICE	Oui	Nom du service (WMTS)
REQUEST	Oui	Nom de la requête (GetTile)
LAYER	Oui	Identifiant de la couche
FORMAT	Oui	Format de sortie de la tuile
TILEMATRIXSET	Oui	Nom de la pyramide
TILEMATRIX	Oui	Maillage
TILEROW	Oui	Coordonnées de ligne dans le maillage
TILECOL	Oui	Coordonnées de colonne dans le maillage

Outre les paramètres standard, QGIS prend en charge les paramètres suivants :

Paramètre	Requis	Description
MAP	Oui	Spécifiez le fichier de projet QGIS

Exemple d'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

Ce paramètre doit être WMTS en cas de requête **GetTile**.

## REQUEST

Ce paramètre est GetTile en cas de requête **GetTile**.

## LAYER

Ce paramètre permet de spécifier la couche à afficher sur la tuile.

De plus, QGIS Server a introduit quelques options pour sélectionner une couche en:

- un nom court
- l'identifiant de la couche

Le nom court d'une couche peut être configuré via les *Propriétés* ► *Métadonnées* de la couche. Si le nom court est défini, il est utilisé par défaut au lieu du nom de la couche:

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

De plus, il y a une option de projet permettant de sélectionner les couches par leur identifiant dans *OWS Server* -> *WMS capabilities Projet* -> *propriétés projet*. Pour activer cette option, la case à cocher *Utiliser les identifiants des couches comme noms* doit être cochée.

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

## FORMAT

Ce paramètre peut être utilisé pour spécifier le format de l'image de tuile. Les valeurs disponibles sont:

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

Si le paramètre FORMAT est différent de l'une de ces valeurs, le format par défaut PNG est utilisé à la place.

## TILEMATRIXSET

Ce paramètre permet de définir le SCR à utiliser pour calculer la pyramide sous-jacente. Format : EPSG:XXXX.

## TILEMATRIX

Ce paramètre permet de définir la matrice à utiliser pour la tuile de sortie.

## TILEROW

Ce paramètre permet de sélectionner la ligne de la tuile pour entrer dans la matrice.

## TILECOL

Ce paramètre permet de sélectionner la colonne de la tuile pour entrer dans la matrice.

## MAP

Ce paramètre permet de définir le fichier projet QGIS à utiliser.

Comme mentionné dans la [table des paramètres GetMap](#), MAP est obligatoire car une requête a besoin d'un projet QGIS pour fonctionner. Cependant, la variable d'environnement QGIS\_PROJECT\_FILE peut être utilisée pour définir un projet QGIS par défaut. Dans ce cas spécifique, MAP n'est plus un paramètre obligatoire. Pour plus d'informations, vous pouvez vous référer à [Configuration avancée](#).

### 3.3.3 GetFeatureInfo

Paramètres standard pour la requête **GetFeatureInfo** selon la spécification OGC WMTS 1.0.0 :

Paramètre	Requis	Description
SERVICE	Oui	Nom du service (WMTS)
REQUEST	Oui	Nom de la requête (GetFeatureInfo)
LAYER	Oui	Identifiant de la couche
INFOFORMAT	Non	Format de sortie
I	Non	Coordonnée X d'un pixel
J	Non	Coordonnée Y d'un pixel
TILEMATRIXSET	Oui	<i>voir GetTile</i>
TILEMATRIX	Oui	<i>voir GetTile</i>
TILEROW	Oui	<i>voir GetTile</i>
TILECOL	Oui	<i>voir GetTile</i>

Outre les paramètres standard, QGIS prend en charge les paramètres suivants :

Paramètre	Requis	Description
MAP	Oui	Spécifiez le fichier de projet QGIS

Exemple d'URL :

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

(suite sur la page suivante)

(suite de la page précédente)

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

## SERVICE

Ce paramètre doit être WMTS en cas de requête **GetFeatureInfo**.

## REQUEST

Ce paramètre est `GetFeatureInfo` en cas de requête **GetFeatureInfo**.

## MAP

Ce paramètre permet de définir le fichier projet QGIS à utiliser.

Comme mentionné dans la [table des paramètres GetMap](#), MAP est obligatoire car une requête a besoin d'un projet QGIS pour fonctionner. Cependant, la variable d'environnement `QGIS_PROJECT_FILE` peut être utilisée pour définir un projet QGIS par défaut. Dans ce cas spécifique, MAP n'est plus un paramètre obligatoire. Pour plus d'informations, vous pouvez vous référer à [Configuration avancée](#).

## LAYER

Ce paramètre permet de spécifier la couche à afficher sur la tuile.

De plus, QGIS Server a introduit quelques options pour sélectionner une couche en:

- un nom court
- l'identifiant de la couche

Le nom court d'une couche peut être configuré via les *Propriétés* ► *Métadonnées* de la couche. Si le nom court est défini, il est utilisé par défaut au lieu du nom de la couche:

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

De plus, il y a une option de projet permettant de sélectionner les couches par leur identifiant dans *OWS Server* -> *WMS capabilities* *Projet* -> *propriétés projet*. Pour activer cette option, la case à cocher *Utiliser les identifiants des couches comme noms* doit être cochée.

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

### INFOFORMAT

Ce paramètre permet de définir le format de sortie du résultat. Les valeurs disponibles sont:

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

La valeur par défaut est `text/plain`.

### I

Ce paramètre permet de définir la coordonnée X du pixel pour lequel nous voulons récupérer les informations sous-jacentes.

### J

Ce paramètre permet de définir la coordonnée Y du pixel pour lequel nous voulons récupérer les informations sous-jacentes.

## 3.4 WFS3 (OGC API Features)

WFS3 est la première implémentation de la nouvelle génération de protocoles OGC. Il est décrit par le document *OGC API - Features - Part 1: Core* <<http://docs.opengeospatial.org/is/17-069r3/17-069r3.html>> \_ document.

Voici un résumé informel rapide des différences les plus importantes entre le protocole WFS bien connu et WFS3:

- WFS3 est basé sur une API REST <[https://en.wikipedia.org/wiki/Representational\\_state\\_transfer](https://en.wikipedia.org/wiki/Representational_state_transfer)> \_ \_
- L'API WFS3 doit suivre les spécifications *OPENAPI* <[https://en.wikipedia.org/wiki/OpenAPI\\_Specification](https://en.wikipedia.org/wiki/OpenAPI_Specification)> \_
- WFS3 prend en charge plusieurs formats de sortie, mais il n'en dicte aucun (seuls GeoJSON et HTML sont actuellement disponibles dans QGIS WFS3) et il utilise la  *négociation de contenu* <[https://en.wikipedia.org/wiki/Content\\_negotiation](https://en.wikipedia.org/wiki/Content_negotiation)> \_ pour déterminer le format doit être servi au client
- JSON et HTML sont des citoyens de première classe dans WFS3
- WFS3 est auto-documenté (via le point de terminaison `/api`)
- WFS3 est entièrement navigable (via des liens) et navigable

---

**Important:** Alors que l'implémentation WFS3 dans QGIS peut utiliser le paramètre `MAP` pour spécifier le fichier de projet, aucun paramètre de requête supplémentaire n'est autorisé par la spécification OPENAPI. Pour cette raison, il est fortement recommandé que `MAP` ne soit pas exposé dans l'URL et que le fichier de projet soit spécifié dans l'environnement par d'autres moyens (c'est-à-dire en définissant `QGIS_PROJECT_FILE` dans l'environnement via une règle de réécriture de serveur Web).

---

---

**Note:** Le point de terminaison **API** fournit une documentation complète de tous les paramètres et formats de sortie pris en charge de votre service. Les paragraphes suivants ne décrivent que les plus importants.

---



### 3.4.1 Représentation des ressources

L'implémentation de QGIS Server WFS3 prend actuellement en charge les formats de représentation (sortie) de ressources suivants:

- HTML
- JSON

Le format réellement servi dépendra de la négociation de contenu, mais un format spécifique peut être explicitement demandé en ajoutant un spécificateur de format aux points de terminaison.

Les extensions de spécificateur de format prises en charge sont:

- .json
- .html

Des alias de spécificateur de format supplémentaires peuvent être définis par des points de terminaison spécifiques:

- .openapi: alias pour .json pris en charge par le point de terminaison **API**
- .geojson: alias pour .json pris en charge par les points de terminaison **Features** et **Feature**

### 3.4.2 Points de terminaison

L'API fournit une liste de points de terminaison que les clients peuvent récupérer. Le système est conçu de telle manière que chaque réponse fournit un ensemble de liens pour naviguer à travers toutes les ressources fournies.

Les points de terminaison fournis par l'implémentation de QGIS sont:

Nom	Chemin	Description
Page de destination	/	Informations générales sur le service et fournit des liens vers tous les points de terminaison disponibles
Conformité	/conformance	Informations sur la conformité du service aux normes
API	/api	Description complète des noeuds finaux fournis par le service et de la structure des documents retournés
Les collections	/collections	Liste de toutes les collections (c'est-à-dire couches vectorielles) fournies par le service
Collection	/collections/{collectionId}	Informations sur une collection (nom, métadonnées, étendue, etc.)
Fonctionnalités	/collections/{collectionId}/items	Liste des entités fournies par la collection
Entité	/collections/{collectionId}/items/{featureId}	Informations sur une seule entité

#### Page de destination

Le critère d'évaluation principal est la **page destination**. À partir de cette page, il est possible de naviguer vers tous les points de terminaison de service disponibles. La **page de destination** doit fournir des liens vers

- la définition de l'API (chemin d'accès /api relations de liaison service-desc et service-doc),
- la déclaration de conformité (chemin /conformance, relation de liaison conformance), et
- les Collections (chemin /collections, relation de lien data).

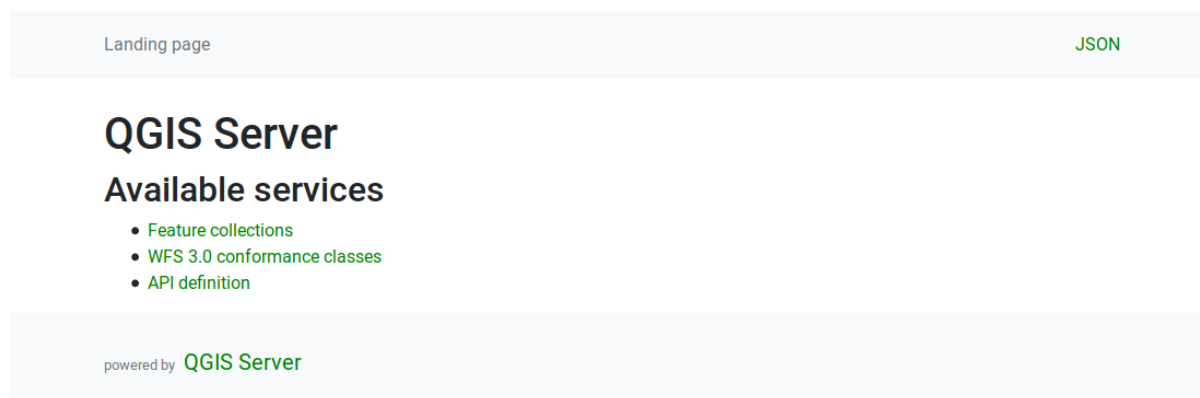


Figure3.2: Page de destination du serveur WFS3

### Définition de l'API

La **Définition API** est une description conforme à OPENAPI de l'API fournie par le service. Dans sa représentation HTML, il s'agit d'une page consultable où tous les points de terminaison et leurs formats de réponse sont répertoriés et documentés avec précision. Le chemin de ce point de terminaison est `/api`.

La définition de l'API fournit une documentation complète et faisant autorité du service, y compris tous les paramètres pris en charge et les formats renvoyés.

---

**Note:** Ce point de terminaison est analogue aux `GetCapabilities` de WFS

---

### Liste des collections

Le point de terminaison des collections fournit une liste de toutes les collections disponibles dans le service. Étant donné que le service « serves » un seul projet QGIS, les collections sont les couches vectorielles du projet en cours (si elles ont été publiées en tant que WFS dans les propriétés du projet). Le chemin de ce point de terminaison est `/collections/`.

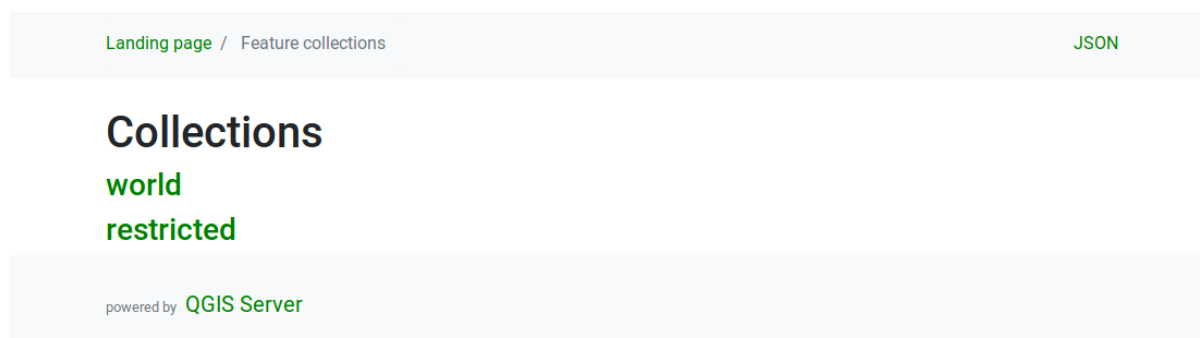


Figure3.3: Page de liste des collections du serveur WFS3

## Détail de la collection

Bien que le point de terminaison des collections ne fournisse pas d'informations détaillées sur chaque collection disponible, ces informations sont disponibles dans les points de terminaison `/collections/{collectionId}`. Les informations typiques incluent l'étendue, une description, les SCR et autres métadonnées.

La représentation HTML fournit également une carte consultable avec les entités disponibles.

The screenshot shows a web page for a WFS3 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 CRS 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 coordinates: West: -179.99999999999997, South: -90.0, East: 180.0, North: 83.62359600000008. To the right is a map of the world with a blue rectangular bounding box covering the entire visible area. The map includes a zoom control and a 'Leaflet | © OpenStreetMap contributors' footer. At the bottom, it says 'powered by QGIS Server'.

Figure3.4: Page de détail de la collecte du serveur WFS3

## Liste des entités

Ce point de terminaison fournit une liste de toutes les entités d'une collection connaissant l'ID de la collection. Le chemin de ce noeud final est `/collections/{collectionId}/items`.

La représentation HTML fournit également une carte consultable avec les entités disponibles.

---

**Note:** Ce point de terminaison est analogue à `GetFeature` dans WFS 1 et WFS 2.

---

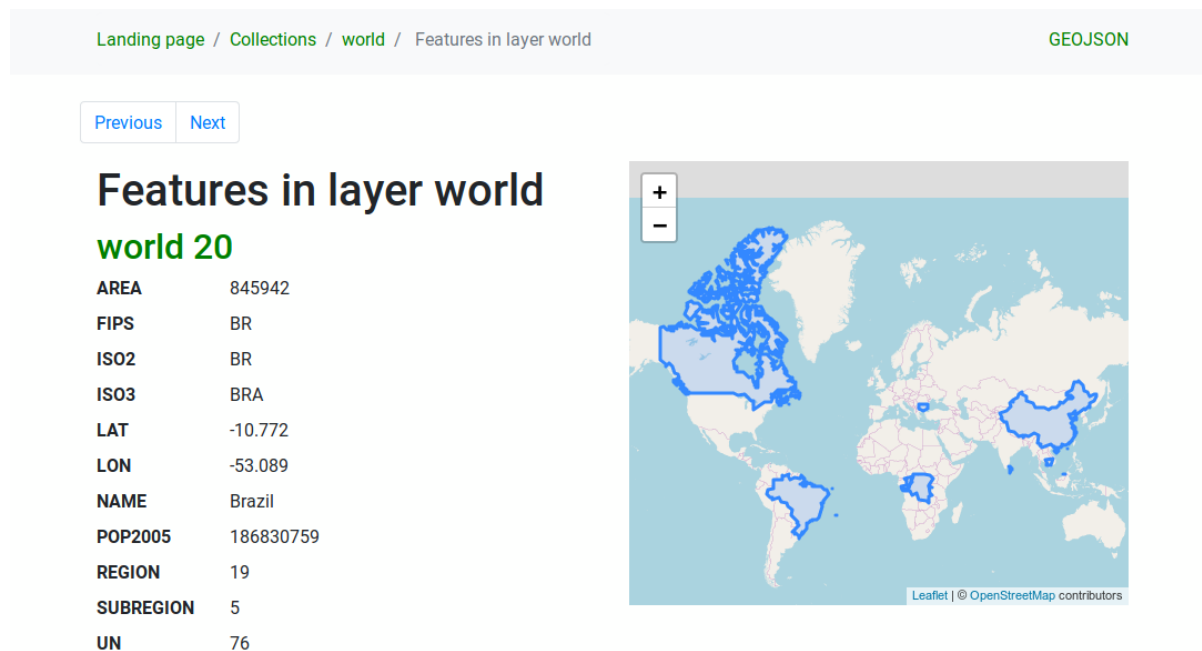


Figure3.5: Page de liste des entités du serveur WFS3

### Détail des entités

Ce point de terminaison fournit toutes les informations disponibles sur une seule entité, y compris les attributs de l'entité et sa géométrie. Le chemin de ce point de terminaison est `/collections/{collectionId}/items/{itemId}`.

La représentation HTML fournit également une carte consultable avec la géométrie de l'entité.



Figure3.6: Page de détail des entités du serveur WFS3

### 3.4.3 Pagination

La pagination d'une longue liste d'entités est implémentée dans l'API OGC via des liens suivant et précédent, QGIS serveur construit ces liens en ajoutant `limite` et `décalage` comme paramètres de chaîne de requête.

Exemple d'URL :

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

**Note:** La valeur maximale acceptable pour `limit` peut être configurée avec le paramètre de configuration du serveur `QGIS_SERVER_API_WFS3_MAX_LIMIT` (voir *Variables d'environnement*).

### 3.4.4 Filtrage des entités

Les entités disponibles dans une collection peuvent être filtrées / recherchées en spécifiant un ou plusieurs filtres.

#### Filtre date et heure

Les collections avec des attributs `date` et / ou `datetime` peuvent être filtrées en spécifiant un argument `datetime` dans la chaîne de requête. Par défaut, le premier champ `date` / `datetime` est utilisé pour le filtrage. Ce comportement peut être configuré en définissant une dimension « Date » ou « Heure » dans *QGIS Server -> Dimension* de la boîte de dialogue des propriétés de la couche.

La syntaxe de filtrage de la date et de l'heure est entièrement décrite dans *Définition de l'API* et prend également en charge les plages (les valeurs de début et de fin sont incluses) en plus des valeurs uniques.

Exemples d'URL:

Revoie uniquement les entités dont la dimension de date correspond à 2019-01-01

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

Revoie uniquement les entités dont la dimension `datetime` correspond à 2019-01-01T01: 01: 01

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

Revoie uniquement les entités dont la dimension `datetime` se situe dans la plage 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
```

#### Filtre de boîte englobante

Un filtre spatial de boîte englobante peut être spécifié avec le paramètre `bbox` :

L'ordre des éléments séparés par des virgules est le suivant:

- Coin inférieur gauche, longitude WGS 84
- Coin inférieur gauche, latitude WGS 84
- Coin supérieur droit, longitude WGS 84
- Coin supérieur droit, latitude WGS 84

**Note:** Les spécifications OGC autorisent également un spécificateur bbox à 6 éléments où les troisième et sixième éléments sont les composants Z, ce qui n'est pas encore pris en charge par le serveur QGIS.

---

Exemple d'URL :

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

Si le CRS de la boîte englobante n'est pas WGS 84, un CRS différent peut être spécifié en utilisant le paramètre optionnel bbox-crs. L'identificateur de format du CRS doit être au format OGC URI :

Exemple d'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
```

### Filtres d'attributs

Les filtres d'attribut peuvent être combinés avec le filtre de boîte englobante et ils se présentent sous la forme générale: 1=2. Plusieurs filtres peuvent être combinés à l'aide de l'opérateur AND.

Exemple d'URL :

filtre toutes les entités où l'attribut `name` est égal à « ma valeur »

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

Les correspondances partielles sont également prises en charge en utilisant un opérateur \* (« étoile »):

Exemple d'URL :

filtre toutes les entités où l'attribut name se termine par « value »

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

### 3.4.5 Tri des éléments

Il est possible d'ordonner le résultat par valeur de champ en utilisant le paramètre d'interrogation « sortby ».

Par défaut, les résultats sont triés par ordre croissant. Pour trier les résultats par ordre décroissant, un drapeau booléen (sortdesc) peut être activé :

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

### 3.4.6 Sélection d'attribut

Les attributs d'entité renvoyés par un appel *Liste des entités* peuvent être limités en ajoutant une liste de noms d'attributs séparés par des virgules dans l'argument de chaîne de requête facultatif propriétés.

Exemple d'URL :

renvoie uniquement l'attribut name

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

### 3.4.7 Personnaliser les pages HTML

La représentation HTML utilise un ensemble de modèles HTML pour générer la réponse. Le modèle est analysé par un moteur de modèle appelé *inja*. Les modèles peuvent être personnalisés en les remplaçant (voir *Remplacements de modèle*). Le modèle a accès aux mêmes données disponibles pour la représentation JSON, avec quelques fonctions supplémentaires :

#### Fonctions de modèle personnalisées

- `path_append(path)` : ajoute un chemin de répertoire à l'URL actuelle
- `path_chomp(n)` : supprime le nombre spécifié « n » de composants de répertoire du chemin d'URL actuel
- `json_dump()` : affiche les données JSON transmises au modèle
- `static(path)` : renvoie l'URL complète du chemin statique spécifié. Par exemple: « `static(« /style/black.css »)` » avec un chemin racine « `http://localhost/qgisserver/wfs3` » renverra « `http://localhost/qgisserver/wfs3/static/style/black.css` ».
- `links_filter(links, key, value)` : retourne les liens filtrés d'une liste de liens
- `content_type_name(content_type)` : renvoie un nom court à partir d'un type de contenu, par exemple « `text / html` » renverra « `HTML` »

#### Remplacements de modèle

Les modèles et les actifs statiques sont stockés dans des sous-répertoires du répertoire de ressources de l'API par défaut du serveur QGIS ( `/usr/share/qgis/ressources/server/api/` sur un système Linux), le répertoire de base peut être personnalisé en modifiant le variable d'environnement `QGIS_SERVER_API_RESOURCES_DIRECTORY`.

Une installation Linux typique aura l'arborescence de répertoires suivante:

```

/usr/share/qgis/ressources/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

```

Pour remplacer les modèles, vous pouvez copier l'arborescence entière vers un autre emplacement et pointer vers le nouvel emplacement. `QGIS_SERVER_API_RESOURCES_DIRECTORY`

## 3.5 Paramètres supplémentaires pris en charge par tous les types de demande

Les paramètres supplémentaires suivants sont pris en charge par tous les protocoles.

- **FILE\_NAME**: si défini, la réponse du serveur sera envoyée au client sous forme de pièce jointe avec le nom de fichier spécifié.

---

**Note:** Non disponible pour WFS3.

---

- **MAP**: similaire à MapServer, le paramètre MAP peut être utilisé pour spécifier le chemin d'accès au fichier de projet QGIS. Vous pouvez spécifier un chemin absolu ou un chemin relatif à l'emplacement de l'exécutable du serveur (`qgis_mapserv.fcgi`). S'il n'est pas spécifié, QGIS Server recherche les fichiers `.qgs` dans le répertoire où se trouve l'exécutable du serveur.

Exemple:

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

---

**Note:** Vous pouvez définir un **QGIS\_PROJECT\_FILE** en tant que variable d'environnement pour indiquer à l'exécutable du serveur où trouver le fichier de projet QGIS. Cette variable sera l'emplacement où QGIS recherchera le fichier de projet. S'il n'est pas défini, il utilisera le paramètre MAP dans la demande et examinera enfin le répertoire exécutable du serveur.

---

## 3.6 REDLINING

Cette fonctionnalité est disponible et peut être utilisée avec les requêtes `GetMap` et `GetPrint`.

La fonction de mise en évidence peut être utilisée pour transmettre des géométries et des étiquettes dans la demande qui se chevauchent par le serveur sur l'image retournée standard (carte). Cela permet à l'utilisateur de mettre l'accent ou peut-être d'ajouter des commentaires (étiquettes) à certaines zones, emplacements, etc. qui ne figurent pas sur la carte standard.

La requête `GetMap` est au format :

```
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'équivalent de `GetPrint` au format (notez que le paramètre `mapX` : est ajouté pour dire quelle carte a un redlining):

:



```

http://qgisplatform.demo/cgi-bin/qgis_mapserv.fcgi?map=/world.qgs&SERVICE=WMS&
↵VERSION=1.3.0&
REQUEST=GetPrint
...
&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><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>
&map0:HIGHLIGHT_LABELSTRING=Write label here
&map0:HIGHLIGHT_LABELSIZE=16
&map0:HIGHLIGHT_LABELCOLOR=%23000000
&map0:HIGHLIGHT_LABELBUFFERCOLOR=%23FFFFFFF
&map0:HIGHLIGHT_LABELBUFFERSIZE=1.5

```

Voici l'image produite par la demande ci-dessus dans laquelle un polygone et une étiquette sont dessinés au-dessus de la carte normale:



Figure3.7: Réponse du serveur à une demande GetMap avec paramètres de mise en évidence

Vous pouvez voir qu'il y a plusieurs paramètres dans cette demande:

- **HIGHLIGHT\_GEOM:** Vous pouvez ajouter POINT, MULTILINESTRING, POLYGON etc. Il prend en charge les géométries en plusieurs parties. Voici un exemple: HIGHLIGHT\_GEOM =

MULTILINESTRING((0 0, 0 1, 1 1)). Les coordonnées doivent être dans le CRS de la demande GetMap / GetPrint.

- **HIGHLIGHT\_SYMBOL**: cela contrôle la manière dont la géométrie est définie et vous pouvez modifier la largeur, la couleur et l'opacité du trait.
- **HIGHLIGHT\_LABELSTRING**: vous pouvez transmettre votre texte d'étiquetage à ce paramètre.
- **HIGHLIGHT\_LABELSIZE**: ce paramètre contrôle la taille de l'étiquette.
- **HIGHLIGHT\_LABELFONT** : Ce paramètre contrôle la police de l'étiquette (par exemple Arial)
- **HIGHLIGHT\_LABELCOLOR**: ce paramètre contrôle la couleur de l'étiquette.
- **HIGHLIGHT\_LABELBUFFERCOLOR**: ce paramètre contrôle la couleur du tampon d'étiquette.
- **HIGHLIGHT\_LABELBUFFERSIZE**: ce paramètre contrôle la taille du tampon d'étiquette.

## 3.7 Couches WMS externes

QGIS Server permet d'inclure des couches de serveurs WMS externes dans les requêtes WMS GetMap et WMS GetPrint. Cela est particulièrement utile si un client Web utilise une couche d'arrière-plan externe dans la carte Web. Pour des raisons de performances, ces couches doivent être demandées directement par le client Web (et non en cascade via QGIS Server). Cependant, pour l'impression, ces couches doivent être mises en cascade via le serveur QGIS afin d'apparaître sur la carte imprimée.

Des couches externes peuvent être ajoutées au paramètre LAYERS sous la forme EXTERNAL\_WMS:1. Les paramètres des couches WMS externes (par exemple, url, format, dpiMode, crs, couches, styles) peuvent être donnés ultérieurement comme paramètres de service 2: 3. Dans une demande GetMap, cela pourrait ressembler à ceci:

```
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
```

De même, des couches externes peuvent être utilisées dans les demandes 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 catalogue

QGIS Serveur Catalogue est un simple catalogue listant les projets QGIS diffusés par QGIS Serveur. Il offre une interface web conviviale, entièrement navigable et dotée de fonctionnalités cartographiques de base permettant de parcourir les jeux de données des projets QGIS.

QGIS Serveur catalogue utilise les variables QGIS\_SERVER\_LANDING\_PAGE\_PROJECTS\_DIRECTORIES et QGIS\_SERVER\_LANDING\_PAGE\_PROJECTS\_PG\_CONNECTIONS (voir *Variables d'environnement*)

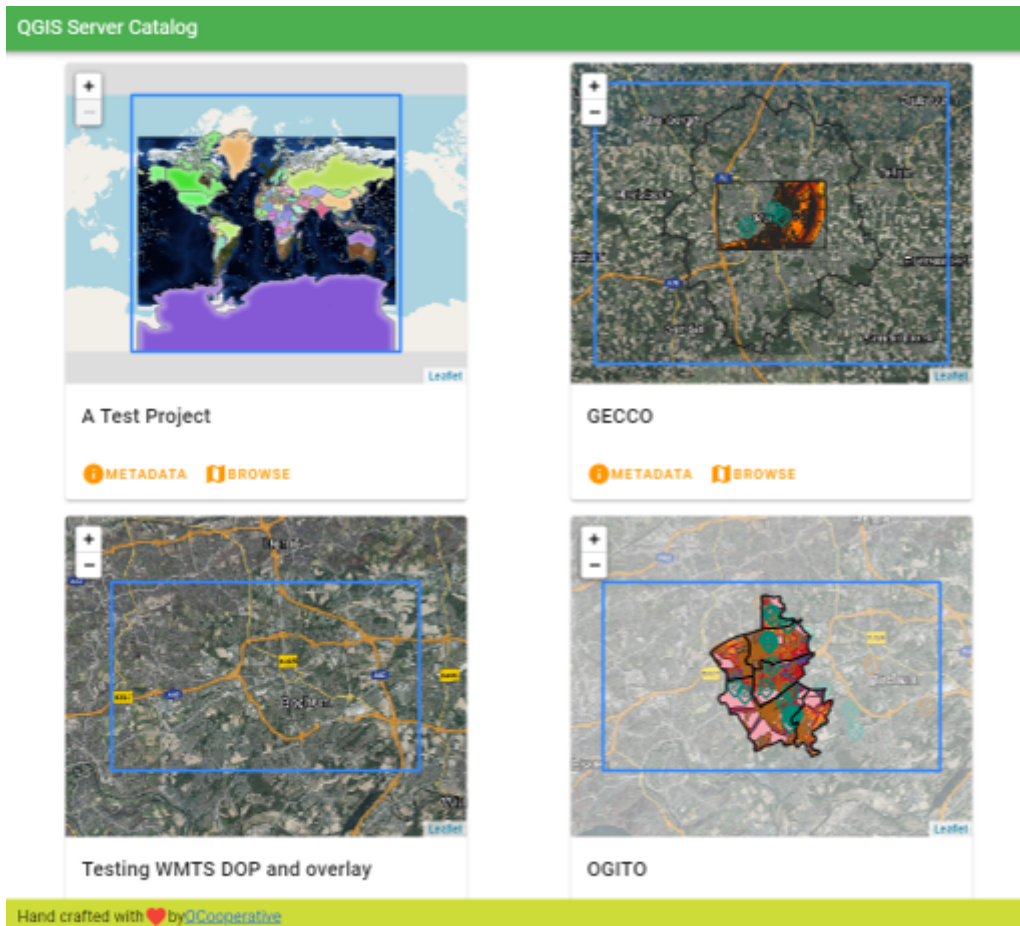


Figure 3.8: Page listant les projets de QGIS Serveur catalogue

Vous pouvez consulter les métadonnées associées à un projet ainsi que les services proposés. Les liens vers ces services sont également donnés.

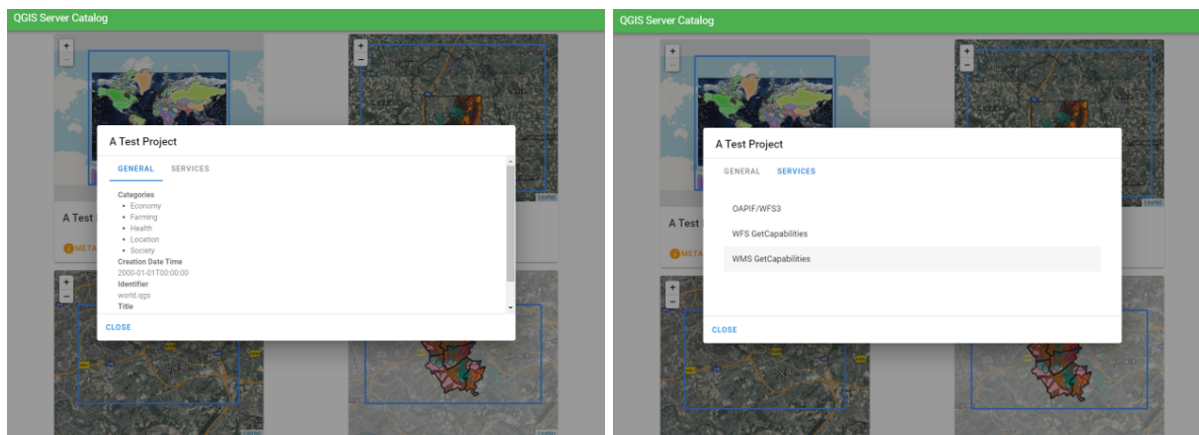


Figure3.9: Catalogue Serveur, métadonnées associées à un projet ainsi que les services qu'il propose.

En parcourant un projet, les jeux de données disponibles sont listés.

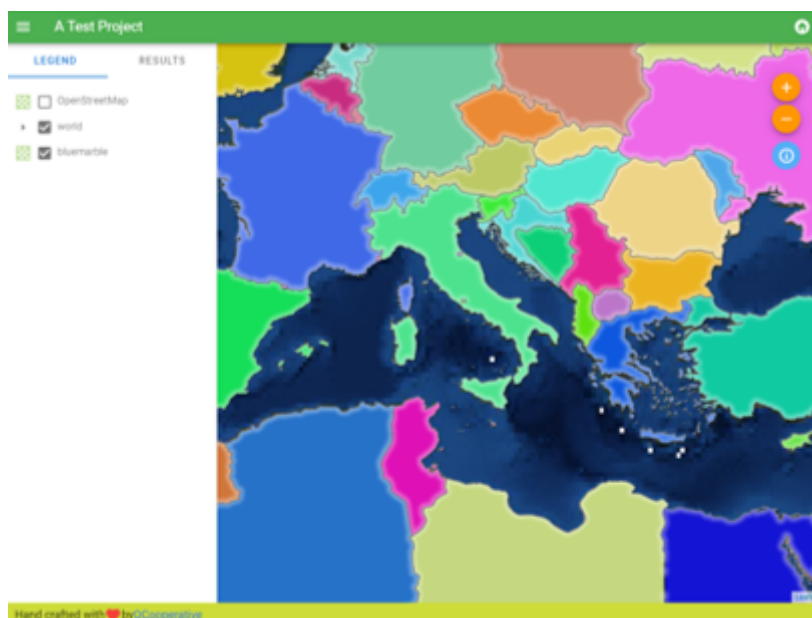


Figure3.10: Parcourir un jeu de données fourni par un projet dans le Catalogue Serveur

Faites un clic droit sur une couche pour afficher sa table attributaire.

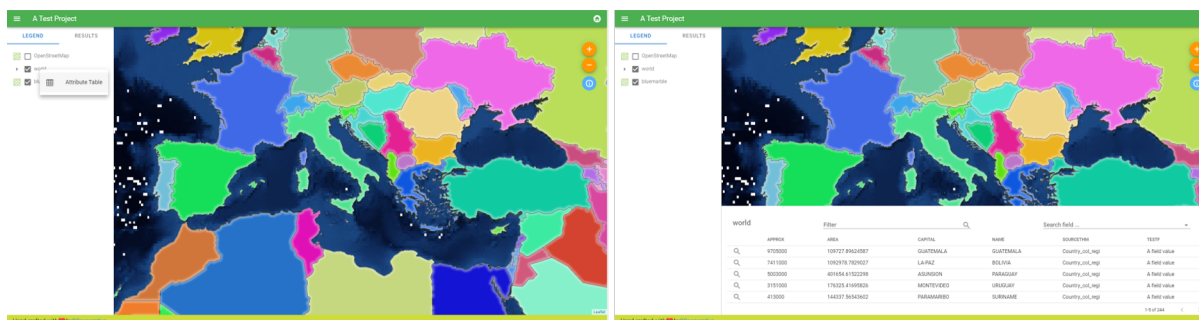


Figure3.11: Table d'attributs associée à la couche

Il est possible de consulter les informations des entités de la carte comme le montre l'image ci-dessous :

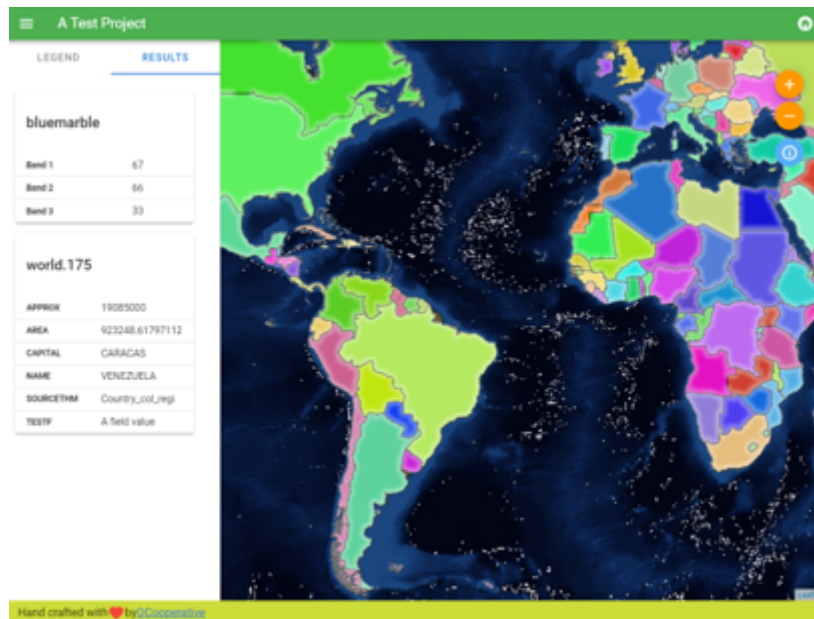


Figure3.12: Consulter les information d'une entité de la carte



## 4.1 Installation

Pour installer par exemple l'extension HelloWorld pour tester les serveurs, vous devez tout d'abord créer un répertoire pour accueillir les extensions serveur. Cela sera spécifié dans la configuration de l'hôte virtuel et passé au serveur par le biais d'une variable d'environnement :

```
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 Configuration d'un serveur HTTP

### 4.2.1 Apache

Pour être en mesure d'utiliser une extension serveur, FastCGI doit savoir où regarder. Donc, nous devons modifier le fichier de configuration Apache pour renseigner la variable d'environnement **QGIS\_PLUGINPATH** à FastCGI :

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

De plus, une autorisation basique HTTP est nécessaire pour utiliser l'extension HelloWorld citée précédemment. Donc, nous devons mettre à jour une dernière fois le fichier de configuration Apache :

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

Puis, redémarrez Apache :

```
systemctl restart apache2
```

## 4.3 Comment utilisez une extension ?

Testez le serveur avec l'extension HelloWorld :

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

Vous pouvez consulter les GetCapabilities par défaut du serveur QGIS sur:

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



### 5.1 Suivi du journal

Pour enregistrer les requêtes envoyées au serveur, vous devez paramétrer la variable d'environnement suivante :

- *QGIS\_SERVER\_LOG\_STDERR*

Une plus grande personnalisation est possible avec les variables suivantes :

- *QGIS\_SERVER\_LOG\_LEVEL*
- *QGIS\_SERVER\_LOG\_PROFILE*

### 5.2 Variables d'environnement

Vous pouvez configurer certains aspects de QGIS Server en définissant **des variables d'environnement**.

Selon le serveur HTTP et la façon dont vous exécutez QGIS Server, il existe plusieurs façons de définir ces variables. Ceci est décrit en détail dans *Apache HTTP Server*.

Nom	Description	Default	Services
QGIS_OPTIONS_PATH	Indique le chemin vers le répertoire des paramètres. Elle fonctionne de la même manière que l'option <code>--optionspath</code> de l'application QGIS. Elle recherche le fichier de paramètres dans <code>&lt;QGIS_OPTIONS_PATH&gt;/QGIS/QGIS3.ini</code> .	""	Tous
QGIS_PLUGINPATH	Utile si vous utilisez des extensions Python pour le serveur, le répertoire indiqué est celui qui sera utilisé pour la recherche des extensions Python.	""	Tous
QGIS_PROJECT_FILE	Le fichier de projet <code>`.qgs`</code> ou <code>.qgz</code> , habituellement transmis sous forme de paramètre dans la chaîne de la requête (avec <i>MAP</i> ), vous pouvez également la paramétrer comme une variable d'environnement (par exemple, en utilisant le module Apache <i>mod_rewrite</i> ). Notez que vous pouvez également indiquer un projet stocké dans PostgreSQL, par exemple <code>postgresql://localhost:5432?sslmode=disable&amp;dbname=mydb&amp;schema=myschema&amp;project=myproject</code> .	""	Tous
QGIS_SERVER_API_RESOURCES_DIRECTORY	Base pour toutes les ressources statiques OGC API (telles que OAPIF/WFS3) (modèles HTML, CSS, JS, ...).	dépend du package 1	WFS
QGIS_SERVER_API_WFS3_MAXIMUM_ITEMS	Valeur maximale pour <code>limit</code> dans une demande d'entités.	10000	WFS
QGIS_SERVER_CACHE_DIRECTORY	Le dossier du cache réseau dans le système de fichier.	cache dans le répertoire du profil	Tous
QGIS_SERVER_CACHE_SIZE	Définit la taille du cache réseau en Mo.	50 Mo	Tous
QGIS_SERVER_DISABLE_GETPRINT	Une option au niveau du projet qui permet d'améliorer le temps de lecture du projet en désactivant le chargement des mises en page. L'activation de cette option désactive la demande de <code>GetPrint</code> du WMS QGIS. Activez ce drapeau de projet QGIS pour ne pas charger les mises en page.	faux	WMS
QGIS_SERVER_IGNORE_BADLAYERS	Les « mauvaises » couches sont des couches qui ne peuvent pas être chargées. Le comportement par défaut de QGIS Server consiste à considérer le projet comme non disponible s'il contient une mauvaise couche. Le comportement par défaut peut être remplacé en définissant cette variable sur <code>1</code> ou <code>vrai</code> . Dans ce cas, les « mauvaises » couches seront simplement ignorées et le projet sera considéré comme valide et disponible.	faux	Tous
QGIS_SERVER_LANDING_PAGE_PROJECTS_DIRECTORIES	Répertoires utilisés par le service des pages d'atterrissage pour trouver les projets <code>.qgs</code> et <code>.qgz</code>	<code>/qgis/server_projects_directories</code>	Tous
QGIS_SERVER_LANDING_PAGE_PG_CONNECTIONS	Chaînes de connexion PostgreSQL utilisées par le service de la page d'atterrissage pour trouver des projets	<code>/qgis/server_projects_pg_connections</code>	Tous
62	<b>Chapter 5. Configuration avancée</b>		
QGIS_SERVER_LOG_FILE	Indique le chemin et le nom de fichier du journal. Assurez-vous que le serveur dispose des permissions adaptées pour	""	Tous

## 5.3 Résumé des paramètres

Au démarrage de QGIS Server, vous disposez d'un résumé de tous les paramètres configurables grâce aux variables d'environnement. De plus, la valeur actuellement utilisée et l'origine sont également affichées.

Par exemple avec 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
```

Dans ce cas particulier, nous savons que les valeurs **QGIS\_SERVER\_MAX\_THREADS** et **QGIS\_SERVER\_PARALLEL\_RENDERING** sont lues à partir du fichier ini situé dans le répertoire **QGIS\_OPTIONS\_PATH** (qui est défini via une variable d'environnement). Les entrées correspondantes dans le fichier ini sont **/qgis/max\_threads** et **/qgis/parallel\_rendering** et leurs valeurs sont **true** et **4** threads.

## 5.4 Nom court pour les couches, les groupes et le projet

Un certain nombre d'éléments disposent d'un <Name> et d'un <Title>. Le nom est une chaîne de caractères utilisée dans la communication de machine à machine alors que le titre est utilisé pour les êtres humains.

Par exemple, un ensemble de données peut avoir le titre descriptif « Température atmosphérique maximale » et être demandé en utilisant le **nom abrégé** « ATMAX ». L'utilisateur peut définir le titre pour les couches, les groupes et les projets.

Le nom de l'OWS est basé sur le nom utilisé dans l'arbre des couches. Ce nom est plus une étiquette pour les humains qu'un nom pour la communication de machine à machine. Vous pouvez définir un **nom court** pour les couches, groupes ou projets, qui sera utilisé par le serveur QGIS comme nom d'identification de la couche (dans *LAYERS* par exemple).

Vous pouvez définir un titre, un nom abrégé et un résumé pour :

- **Layers** : cliquez avec le bouton droit de la souris sur une couche et choisissez *Propriétés...* ► *Serveur QGIS* ► *Description*.
- **Groupes** : cliquez avec le bouton droit de la souris sur un groupe et sélectionnez *Définir les données WMS du groupe*.

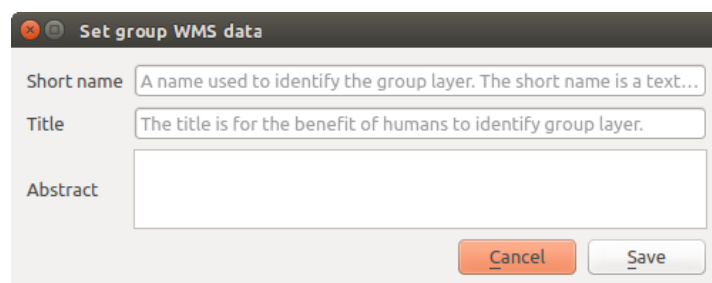


Figure5.1: Définir la boîte de dialogue des données associées au groupe WMS

- **Projet** : allez à *Projet* ► *Propriétés...* ► *Serveur QGIS* ► *Capacités du service*.

## 5.5 Connexion au fichier de service

Pour faire en sorte qu'Apache puisse accéder au fichier de service PostgreSQL (consultez la section *ref:pg-service-file*), vous devez modifier vos fichiers `*.conf` de la manière suivante:

```
SetEnv PGSERVICEFILE /home/web/.pg_service.conf

<Directory "/home/web/apps2/bin/">
  AllowOverride None
  ....
```

## 5.6 Ajouter des polices à votre serveur Linux

Gardez à l'esprit que vous pouvez utiliser des projet QGIS qui utilisent des polices qui n'existent pas par défaut sur les autres machines. Cela signifie que si vous partagez le projet, il pourra apparaître de manière différente sur d'autres machines (si les polices n'existent pas sur la machine cible).

Pour s'assurer que cela n'arrive pas, vous devez simplement installer les polices manquantes sur la machine cible. En règle général, le faire sur des systèmes bureautiques est relativement simple (double-cliquer sur des polices).

Sous Linux, si vous n'avez pas d'environnement de bureau d'installé (ou que vous préférez la ligne de commande), vous devrez:

- Sur les systèmes basés sur 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
```

- Pour les systèmes basés sur 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
```



---

## Serveur de développement

---

Une installation et un déploiement en production de QGIS Serveur implique généralement la mise en place d'un serveur web (par exemple Apache ou Nginx) qui peut transmettre les requêtes HTTP provenant des clients à l'application binaire FastCGI de QGIS Serveur.

Si vous souhaitez tester rapidement le serveur QGIS sur votre machine locale sans avoir à configurer et à installer une stack de serveur web complète, vous pouvez utiliser le serveur autonome de développement QGIS.

Il s'agit d'une application indépendante qui fournit un serveur web très simple prêt à servir vos fichiers de projet.

**Avertissement:** Le Serveur de Développement Autonome n'a pas été développé dans le but d'être utilisé en production, il n'a pas été vérifié pour les vulnérabilités de sécurité ou pour d'autres conditions de stress qui se produisent normalement sur un serveur exposé publiquement.

Pour lancer le serveur :

```
$ qgis_mapserver
```

Le port par défaut que le serveur de développement écoute est 8000. Exemple de sortie :

```
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
```

Le serveur dispose de quelques options qui peuvent être passées en argument de ligne de commande. Vous pouvez les voir toutes en invoquant le serveur avec `-h`.

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

(suite sur la page suivante)

(suite de la page précédente)

```
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
```



---

## Déploiement conteneurisé

---

Il existe de nombreuses façons d'utiliser les applications conteneurisées, des plus simples (simples images Docker) aux plus sophistiquées (Kubernetes, etc.).

---

**Note:** Ce type de déploiement nécessite l'installation et l'exécution de l'application `docker`. Consultez ce [tutoriel](#).

---

---

**Indication:** Docker exécute des applications pré-packagées (alias images) qui peuvent être récupérées comme sources (Dockerfile et ressources) à construire ou déjà construites à partir de registres (privés ou publics).

---

---

**Note:** Les téléchargements de paquets QGIS Debian-Ubuntu nécessitent une clé d'authentification gpg valide. Veuillez vous référer aux pages d'installation <<https://www.qgis.org/fr/site/forusers/alldownloads.html#debian-ubuntu>>`\_ pour mettre à jour le Dockerfile suivant avec la dernière empreinte de la clé

---

### 7.1 Simples images docker

Comme l'image du docker n'existe pas dans un dépôt public, vous devrez la construire. Pour ce faire, créez un dossier `qgis-server` et à l'intérieur de ce dossier :

- créer un fichier `Dockerfile` avec ce contenu :

```
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 \
```

(suite sur la page suivante)

```

    && 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"]

```

- créer un fichier cmd.sh avec ce contenu :

```

#!/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

```

- construire l'image avec:

```

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

```

### 7.1.1 Premier lancement

Pour faire fonctionner le serveur, vous aurez besoin d'un fichier de projet QGIS. Vous pouvez utiliser l'un des vôtres ou choisir [cette exemple](#).

Pour ce faire, créez un dossier `data` dans le dossier `qgis-server` et copiez votre fichier dedans. Pour respecter les explications suivantes, renommez-le en `osm.qgs`.

**Note:** 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 *Configurer votre projet* and subsequent.

Maintenant, vous pouvez faire fonctionner le serveur avec :

```
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
```

Options utilisées :

- **-d** : exécuter en arrière-plan
- **-rm** : supprimer le conteneur lorsqu'il est arrêté
- **-name** : nom du conteneur à créer
- **-net** : (précédemment créé) nom du sous-réseau
- **-hostname** : nom d'hôte du conteneur, pour référence ultérieure
- **-v** : dossier de données locales à monter dans le conteneur
- **-p** : mappage des ports hôtes/conteneurs
- **-e** : variable d'environnement à utiliser dans le conteneur

Pour vérifier, tapez `docker ps | grep qgis-server` et vous devriez voir une ligne avec **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 Exemple fonctionnel

Comme le serveur n'accepte que les connexions fastcgi, vous avez besoin d'un serveur HTTP qui gère ce protocole. Pour ce faire, nous devons créer un simple fichier de configuration Nginx et lancer une image Nginx.

Créez un fichier `nginx.conf` dans le dossier courant avec ce contenu :

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

(suite sur la page suivante)

```

proxy_buffer_size 16k;
gzip off;
include fastcgi_params;
fastcgi_pass qgis-server:5555;
}
}

```

Et taper cette commande:

```

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

```

Pour vérifier la disponibilité des capacités du serveur, tapez dans l'url d'un navigateur <http://localhost:8080/qgis-server/?SERVICE=WMS&VERSION=1.3.0&REQUEST=GetCapabilities>

### 7.1.3 Nettoyage

Pour nettoyer les images en cours, tapez :

```

docker stop qgis-server nginx

```

## 7.2 Docker stacks

La méthode précédente est scriptable, mais elle n'est ni facile à mettre en œuvre, ni standardisée, ni facilement gérable.

Pour travailler avec un jeu d'images Docker, vous pourriez utiliser une stack Docker gérée par un orchestrateur. Dans une pile, les images fonctionnent dans le même réseau privé, et vous pouvez démarrer/arrêter toute la pile ou déployer la pile à d'autres workers. Il existe de nombreux orchestrateurs, par exemple Swarm, Kubernetes et Mesos.

Dans ce qui suit, nous présenterons des configurations simples à des fins de test. Elles ne sont pas adaptées à la production.

### 7.2.1 Swarm/docker-compose

Docker a maintenant son propre orchestrateur : Swarm (compatible avec les fichiers docker-compose). Vous devez l'activer <<https://docs.docker.com/get-started/orchestration/#enable-docker-swarm>>\_ (la version Mac fonctionnera également avec des conteneurs Linux).

#### Description de la Stack

Maintenant que Swarm fonctionne, créez le service grâce au fichier (voir [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)

```

(suite de la page précédente)

```
- DEBUG=1                                # display env before spawning QGIS Server

nginx:
  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
```

Pour déployer (ou mettre à jour) la stack, tapez :

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

Vérifiez le statut de déploiement de la stack jusqu'à ce que vous obteniez **1/1** dans la colonne **replicas** :

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

Quelque chose comme :

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				

Pour vérifier les capacités du serveur WMS, tapez dans un navigateur web <http://localhost:8080/qgis-server/?SERVICE=WMS&VERSION=1.3.0&REQUEST=GetCapabilities>

## Nettoyage

Pour nettoyer, tapez :

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

## 7.2.2 Kubernetes

### Installation

Si vous avez une installation **Docker Desktop**, l'utilisation de Kubernetes (alias k8s) est assez simple : [activez k8s](#).

Sinon, suivez le tutoriel [minikube](#) ou [microk8s](#) pour Ubuntu.

Comme l'installation de Kubernetes peut être vraiment complexe, nous nous concentrerons uniquement sur les aspects utilisés dans cette démo. Pour de plus amples informations, consultez la [documentation officielle](#).

### microk8s

microk8s nécessite des étapes supplémentaires : vous devez activer le registre et marquer l'image du serveur qgis afin que Kubernetes puisse trouver les images créées.

Tout d'abord, activez le dépôt :

```
microk8s enable dashboard dns registry
```

Ensuite, vous devez étiqueter et pousser l'image vers votre dépôt nouvellement créé :

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

Enfin, ajoutez ou complétez le fichier `/etc/docker/daemon.json` pour que votre dépôt **127.0.0.1:32000** soit listé dans le champ **insecure-registries** :

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

### Créer des manifests

Kubernetes décrit les objets à déployer dans les manifests yaml. Il en existe de nombreux types différents, mais nous n'utiliserons que les déploiements (handle pods, c'est-à-dire les images de docker) et les services pour exposer les déploiements à des fins internes ou externes.

### Déploiement des manifests

Créez un fichier `deployments.yaml` avec ce contenu :

```
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"
```

(suite sur la page suivante)

(suite de la page précédente)

```

    ports:
      - containerPort: 5555
    volumeMounts:
      - name: qgis-data
        mountPath: /data/
  volumes:
    - name: qgis-data
      hostPath:
        path: REPLACE_WITH_FULL_PATH/data
---
apiVersion: apps/v1
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

Créez un fichier `services.yaml` avec ce contenu :

```

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:

```

(suite sur la page suivante)

```

name: qgis-nginx
namespace: default
spec:
  type: NodePort
  selector:
    myLabel: qgis-nginx
  ports:
    - port: 80
      targetPort: 80
      nodePort: 30080
    
```

### Déploiement des manifests

Pour déployer les images et les services dans Kubernetes, on peut utiliser le tableau de bord (cliquez sur le + en haut à droite) ou la ligne de commande.

**Note:** Lorsque vous utilisez la ligne de commande avec microk8s, vous devez préfixer chaque commande par « microk8s ».

Pour déployer ou mettre à jour vos manifests :

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

Pour vérifier ce qui est actuellement déployé :

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

Vous devriez obtenir quelque chose comme :

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

Pour lire les journaux nginx/qgis, tapez :

```
kubectl logs -f POD_NAME
```

Pour vérifier les capacités du serveur WMS, tapez dans un navigateur web <http://localhost:30080/qgis-server/?SERVICE=WMS&VERSION=1.3.0&REQUEST=GetCapabilities>



## Nettoyage

Pour nettoyer, tapez :

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

## 7.3 Déploiement en Cloud

La gestion de votre propre grappe de serveurs pour assurer le déploiement d'applications conteneurisées est une tâche complexe. Vous devez gérer de multiples questions, telles que le matériel, la bande passante et la sécurité à différents niveaux.

Les solutions de déploiement dans le Cloud peuvent être une bonne alternative lorsque vous ne voulez pas vous concentrer sur la gestion de l'infrastructure.

Un déploiement dans le Cloud peut utiliser des mécanismes propriétaires, mais ils sont également compatibles avec les étapes expliquées précédemment (*docker images* and *stack management*).

### 7.3.1 Cas d'utilisation avec AWS

Avec Amazon AWS, grâce aux fonctionnalités de l'ECS (Elastic Container Service) <<https://console.aws.amazon.com/ecs/home>>, vous pouvez utiliser docker-compose ou Kubernetes pour gérer votre stack. Vous devrez créer un registre d'images pour que vos images personnalisées soient accessibles.

Pour utiliser les mêmes fonctionnalités que docker-compose, vous devez installer le client **ecs-cli** et avoir les permissions / rôles. Ensuite, à l'aide des commandes *ecs-cli compose* (voir le manuel *ecs-cli compose* <<https://docs.aws.amazon.com/AmazonECS/latest/developerguide/cmd-ecs-cli-compose.html>> et le tutoriel *ecs-cli*), vous pouvez réutiliser la description de *stack*.

Pour utiliser Kubernetes, vous pouvez utiliser la console web AWS ou l'outil en ligne de commande **eksctl** et avoir les permissions / rôles appropriés. Ensuite, avec un environnement kubectl bien configuré, vous pouvez réutiliser les *Kubernetes manifests*.



## Foire aux questions

- *Quelles sont les différences entre QGIS Desktop et QGIS Server ?*

QGIS Desktop dispose d'une interface graphique et permet de créer et modifier des cartes. QGIS Server est une application serveur qui sert vos projets QGIS à des applications d'utilisateur final via des services web OGC comme WMS, WFS, etc..

- *Qu'est-ce que l'OGC ?*

L'OGC (Open Geospatial Consortium) est une organisation à but non lucratif qui se consacre à la production de standards ouverts de qualité pour la communauté géospatiale mondiale.

- *Citez d'autres serveurs de cartographie web ?*

ArcGIS server, Geoserver, Mapserver, Mapnik etc.

- *Comment comparer QGIS Server à d'autres serveurs de cartographie web ? (01/01/2021)*

Fonctionnalités	QGIS Server	GeoServer	ArcGIS Server
Depuis	2006	2001	1999
Licence	GPL	GPL	commercial
Support commercial	Sociétés multiples	Sociétés multiples	ESRI et son réseau de revendeurs
Technologie	C++/python	Java	C++
Cache de tuilage	oui	oui (via GeoWebCache)	oui
3D	Non	Non	Oui
Requêtage	filtres FES (2.0) et OGC (1.0)	filtres CQL et OGC	filtres OGC
Génération de rapports	oui	oui	oui
Administration du serveur	oui grâce à des logiciels tiers (LizMap, QWC2, etc.)	web + API REST	web + API REST
Couche de projet de SIG/édition de symbologie	complète grâce à une interface graphique dédiée	simple grâce à une interface web	complète grâce à une interface graphique dédiée

- *Quelles sont les versions de spécifications OGC implémentées dans QGIS server comparées aux autres serveurs de cartographie web? (01/01/2021)*

Standards OGC	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 (alias WFS3)	1.0.0	non	non
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)	non	1.0.0	1.0.0
CSW (Catalogue Service for the Web)	non	2.0.2	non
SLD (Styled Layer Descriptor)	oui	oui	oui

- *Qu'est-ce qu'un cache de tuiles*

Les cartes sont souvent statiques. Comme la plupart des clients produisent un rendu des données WMS (Web Map Service) à chaque requête, cela peut entraîner du traitement superflu et des temps d'attente.

La cache de tuiles optimise cette expérience en mémorisant (cachant) les images de cartes, les tuiles, lorsqu'elles sont demandées, ce qui le fait se comporter comme un proxy entre le client (tel que OpenLayers ou Google Maps) et le serveur (tout serveur supportant WMS). Lorsque de nouvelles cartes et tuiles sont demandées, QGIS Server intercepte ces appels et renvoie les tuiles déjà rendues si mémorisées, ou bien interroge le moteur QGIS pour effectuer le rendu de nouvelles tuiles si besoin. Par conséquent, une fois les tuiles mémorisées, la vitesse de rendu de la carte s'accroît énormément, produisant une grande amélioration de l'expérience utilisateur.

- *Qu'est ce que PostgreSQL ?*

**PostgreSQL** est un puissant compagnon de base de données relationnelle-objet open source pour QGIS.

- *Qu'est-ce que PostGIS ?*

**PostGIS** est une extension de base de données spatiales pour la base de données relationnelle-objet PostgreSQL. Il ajoute le support des objets géographiques permettant d'exécuter des requêtes de localisation en SQL.

- A suivre...