



QGIS Server 3.40 User Guide

QGIS Project

avr. 18, 2025

1	Introduction	3
2	Premiers Pas	5
2.1	Installation sur systèmes Debian	5
2.1.1	Apache HTTP Server	7
2.1.2	NGINX HTTP Server	9
2.1.3	Xvfb	13
2.2	Installation sous Windows	14
2.3	Publier un projet	16
2.4	Configurer votre projet	17
2.4.1	Capacités WMS	19
2.4.2	Capacités WMTS	21
2.4.3	Capacités WFS/OAPIF	21
2.4.4	Capacités WCS	23
2.4.5	Bien configurer votre OWS	24
2.5	Intégration avec des applications tierces	24
2.5.1	Intégration avec QGIS	24
2.5.2	Intégration avec MapProxy	24
2.5.3	Intégration avec QWC2	24
3	Services	25
3.1	Les bases	25
3.1.1	SERVICE	26
3.1.2	REQUEST	26
3.1.3	MAP	26
3.1.4	FILE_NAME	26
3.1.5	Nom court	27
3.2	Web Map Service (WMS)	27
3.2.1	GetCapabilities	28
3.2.2	GetMap	28
3.2.3	GetFeatureInfo	37
3.2.4	GetLegendGraphic	40
3.2.5	GetStyle(s)	55
3.2.6	DescribeLayer	55
3.2.7	GetPrint	56
3.2.8	GetProjectSettings	60
3.2.9	GetSchemaExtension	60
3.2.10	Couches WMS externes	61
3.2.11	Redlining	61
3.3	Web Feature Service (WFS)	64
3.3.1	GetCapabilities	64

3.3.2	GetFeature	65
3.3.3	DescribeFeatureType	72
3.3.4	Transaction	73
3.4	Web Coverage Service (WCS)	75
3.4.1	GetCapabilities	75
3.4.2	DescribeCoverage	76
3.4.3	GetCoverage	78
3.5	Web Map Tile Service (WMTS)	79
3.5.1	GetCapabilities	80
3.5.2	GetTile	80
3.5.3	GetFeatureInfo	81
3.6	OGC API Features	82
3.6.1	Représentation des ressources	83
3.6.2	Points de terminaison	83
3.6.3	Pagination	87
3.6.4	Filtrage des entités	87
3.6.5	Tri des éléments	89
3.6.6	Sélection d'attribut	89
3.6.7	Personnaliser les pages HTML	89
4	Catalogue	91
5	Extensions	95
5.1	Liste d'extensions	95
5.2	Emplacement des extensions	95
5.3	Installation	95
5.3.1	Manuellement depuis un fichier ZIP	95
5.3.2	Avec un outil en ligne de commandes	96
5.4	Configuration d'un serveur HTTP	96
5.4.1	Apache	96
5.5	Comment utilisez une extension ?	97
6	Configuration avancée	99
6.1	Suivi du journal	99
6.2	Variables d'environnement	99
6.3	Résumé des paramètres	104
6.4	Connexion au fichier de service	105
6.5	Ajouter des polices à votre serveur Linux	106
7	Serveur de développement	107
8	Déploiement conteneurisé	109
8.1	Simple images docker	109
8.1.1	Premier lancement	111
8.1.2	Exemple fonctionnel	111
8.1.3	Nettoyage	112
8.2	Docker stacks	112
8.2.1	Swarm/docker-compose	112
8.2.2	Kubernetes	113
8.3	Déploiement en Cloud	117
8.3.1	Cas d'utilisation avec AWS	117
9	Foire aux questions	119

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 Server utilise QGIS comme socle 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, QGIS Server utilise les règles de cartographie comme langage de configuration, à la fois pour la configuration du serveur et pour les règles cartographiques définies 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`.

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

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 3.40.0-Bratislava
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_3.40.zip
unzip release_3.40.zip
mv QGIS-Training-Data-release_3.40/exercise_data/qgis-server-tutorial-data/world.
↳ qgs .
mv QGIS-Training-Data-release_3.40/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 proprement QGIS Server, il vous faut 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.

1. Installez Apache et `mod_fcgid` :

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

2. Vous pouvez faire fonctionner QGIS Server sur votre site web par défaut, mais nous allons plutôt procéder via la configuration d'un hôte virtuel (`virtualhost`) spécifique, dans les prochaines lignes.

1. 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
        Require all granted
    </Directory>

    <IfModule mod_fcgid.c>
        FcgidMaxRequestLen 26214400
```

(suite sur la page suivante)

(suite de la page précédente)

```
FcgidConnectTimeout 60
</IfModule>

</VirtualHost>
```

Note: Certaines des options de configuration ci-dessus sont expliquées dans les sections *variables d'environnement du serveur* et fichier `pg_service`.

2. 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`.

3. Nous pouvons maintenant activer l'hôte virtuel et le module `fcgi`, si ce n'est déjà fait :

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

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

```
systemctl restart apache2
```

5. 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`.

```
# Replace 127.0.0.1 with the IP of your server.
sh -c "echo '127.0.0.1 qgis.demo' >> /etc/hosts"
```

Important: 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 `qgis.demo` 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;
```

(suite sur la page suivante)

```
fastcgi_pass    unix:/var/run/qgisserver.socket;
}
```

FastCGI wrappers

Avertissement: `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` :

1. La première étape est l'installation du paquet :

```
apt install spawn-fcgi
```

2. 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;
}
```

3. Et redémarrez NGINX pour prendre en compte la nouvelle configuration :

```
systemctl restart nginx
```

4. 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 à l'allumage ou à tout autre moment. Par exemple, avec **systemd** :

1. Éditez le fichier `/etc/systemd/system/qgis-server.service` avec ce contenu :

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

[Service]
;; set env var as needed
```

(suite sur la page suivante)

(suite de la page précédente)

```

;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

```

2. Ensuite, activez et démarrez le service :

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

Avvertissement: 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.

1. Vous devez d'abord installer le paquet correspondant :

```
apt install fcgiwrap
```

2. 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 }

```

3. 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.

This method, to deploy QGIS Server, relies on two Systemd units to configure: a **Socket unit** and a **Service unit**.

1. **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
```

2. Maintenant, activez et démarrez les sockets :

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

3. **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
```

4. Maintenant, lancez le service de socket:

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

5. 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;
}

server {
    ...
```

(suite sur la page suivante)

(suite de la page précédente)

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

6. 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/>.

1. Pour installer ce paquet :

```
apt install xvfb
```

2. 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
```

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

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

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

- Avec Apache :

1. Add to your *Fcgid* configuration (see *Apache HTTP Server*):

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

2. Restart Apache for the new configuration to be taken into account:

```
systemctl restart apache2
```

- Avec NGINX

- 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.

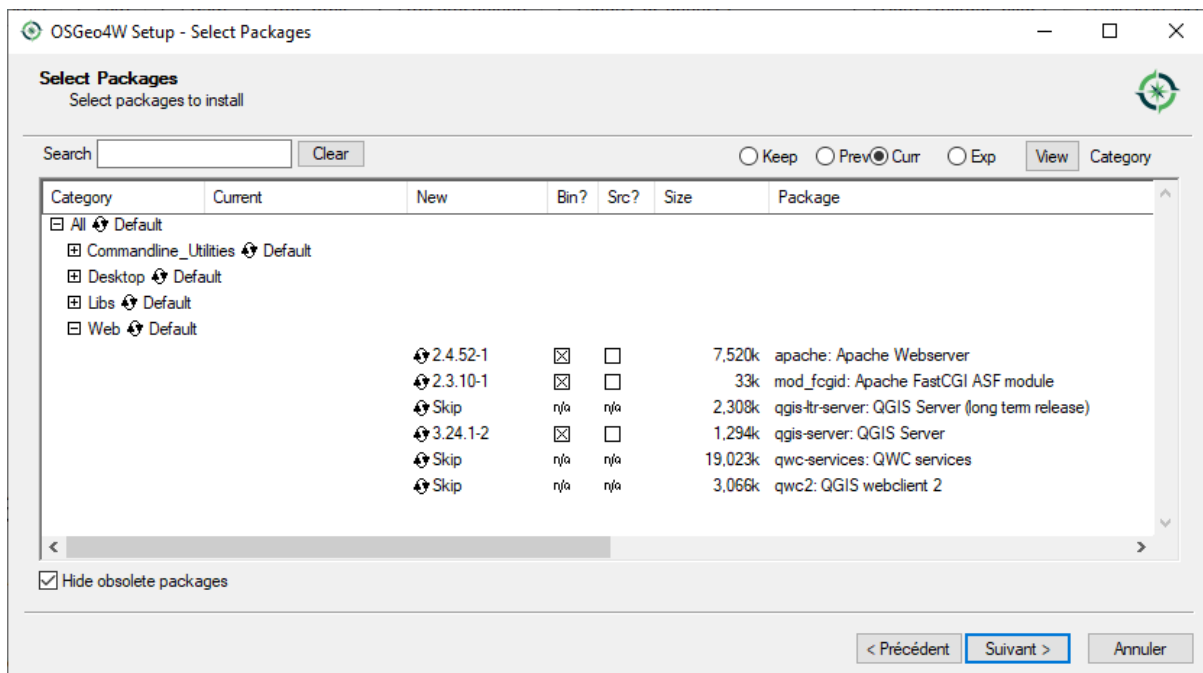
```
fastcgi_param  DISPLAY      ":99";
```

2.2 Installation sous Windows

QGIS Server can also be installed on Windows systems using the OSGeo4W network installer (<https://qgis.org/resources/installation-guide/#windows>).

Une procédure simple est la suivante:

1. Téléchargez et exécutez l'installateur OSGeo4w
2. Follow the « Advanced Install » and install the **QGIS Desktop**, **QGIS Server apache** and **mod_fcgid** packages.



3. Apache is not directly installed as a service on Windows. You need to:
 1. Right-click the OSGeo4W.bat file at the root of the C:\OSGeo4W\ folder (if the default installation paths have been used) and select *Run as administrator*
 2. In the console, run apache-install.bat, which will output

```
> apache-install.bat
Installing the 'Apache OSGeo4W Web Server' service
The 'Apache OSGeo4W Web Server' service is successfully installed.
Testing httpd.conf....
Errors reported here must be corrected before the service can be started.
...
```

The service is started as you can notice in the report. But the server may fail to run due to missing custom configuration.

4. Edit the C:\OSGeo4w\apps\apache\conf\httpd.conf file with the following changes (various other combinations are possible):

Fonction	Configuration existante	Remplacement
(Optional) Customize the address to listen to using an IP and/or port, You can and add as many entries as you wish.	<code>Listen \${SRVPORT}</code>	<code>Listen localhost:8080</code>
Indicate where to find the script files	<code>ScriptAlias /cgi-bin/ "\$ ↪{SRVROOT}/cgi-bin/"</code>	<code>ScriptAlias /cgi-bin/ ↪"C:/OSGeo4W/apps/qgis/ ↪bin/"</code>
Provide permissions on the script folder	<code><Directory "\${SRVROOT}/ ↪cgi-bin"> AllowOverride None Options None Require all granted </Directory></code>	<code><Directory "C:/OSGeo4W/ ↪apps/qgis/bin"> SetHandler cgi- ↪script AllowOverride None Options ExecCGI Require all granted </Directory></code>
Enable file extensions to use for script files. Uncomment and complete	<code>#AddHandler cgi-script . ↪cgi</code>	<code>AddHandler cgi-script . ↪cgi .exe</code>
Add more OSGeo4W custom configuration variables	<code># parse OSGeo4W apache_ ↪conf files IncludeOptional "C:/ ↪OSGeo4W/httpd.d/httpd_ ↪*.conf"</code>	<code># parse OSGeo4W apache_ ↪conf files IncludeOptional "C:/ ↪OSGeo4W/httpd.d/httpd_ ↪*.conf" SetEnv GDAL_DATA "C:/ ↪OSGeo4W/share/gdal" SetEnv QGIS_AUTH_DB_DIR_ ↪PATH "C:/OSGeo4W/apps/ ↪qgis/resources"</code>

5. Redémarrez le serveur web Apache

```
> apache-restart.bat
```

6. Open browser window to testing a GetCapabilities request to QGIS Server. Replace localhost : 8080 with the IP and port you set to listen.

```
http://localhost:8080/cgi-bin/qgis_mapserv.fcgi.exe?SERVICE=WMS&VERSION=1.3.0&  
↪REQUEST=GetCapabilities
```

A XML file with the capabilities should be returned. Your server is ready to use.

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*:

- If you followed the above instructions to install an Apache HTTP Server:

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

- If you followed the above instructions to install an NGINX HTTP Server:

```
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:

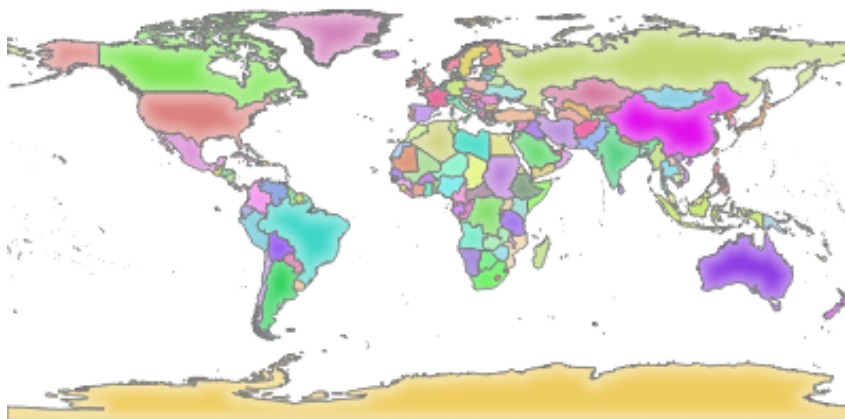


Figure2.1: Réponse du serveur à une demande GetMap de base

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

2.4 Configurer votre projet

To provide a new QGIS Server WMS, WFS, OAPIF or WCS, you have to create a QGIS project file with some data or use one of your current project. Define the colors and styles of the layers in QGIS and the project CRS, if not already defined. Then, go to the *QGIS Server* menu of the *Project ► Properties...* dialog and provide some information about the OWS in the *Service Capabilities* tab.

Project Properties — QGIS Server

Services Capabilities | WMS | WMTS | WFS/OAPIF | WCS

☒ **Enable Service Capabilities**

These parameters are used to generate the GetCapabilities document and shall be chosen carefully to avoid interoperability and security issues.

Short name: A name used to identify the root layer. The short name is a text ...

Title: The title should be brief yet descriptive enough to identify this ...

Organization: The name of the service provider.

Online resource: The web site URL of the service provider.

Person: The contact person name for the service.

Position:

E-Mail: The contact person e-mail for the service.

Phone: The contact person phone for the service.

Abstract:

Fees: Conditions Unknown

Access constraints: None

Keyword list: List of keywords separated by comma to help catalog searching.

Test Configuration

Launch

Help Apply Cancel OK

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

You have to *Enable Service Capabilities* first, if it is deactivated. This will appear in the GetCapabilities response of the WMS, WFS or WCS. If you don't check ☒ *Enable Service capabilities*, QGIS Server will use the information given in the `wms_metadata.xml` file located in the `cgi-bin` folder.

2.4.1 Capacités WMS

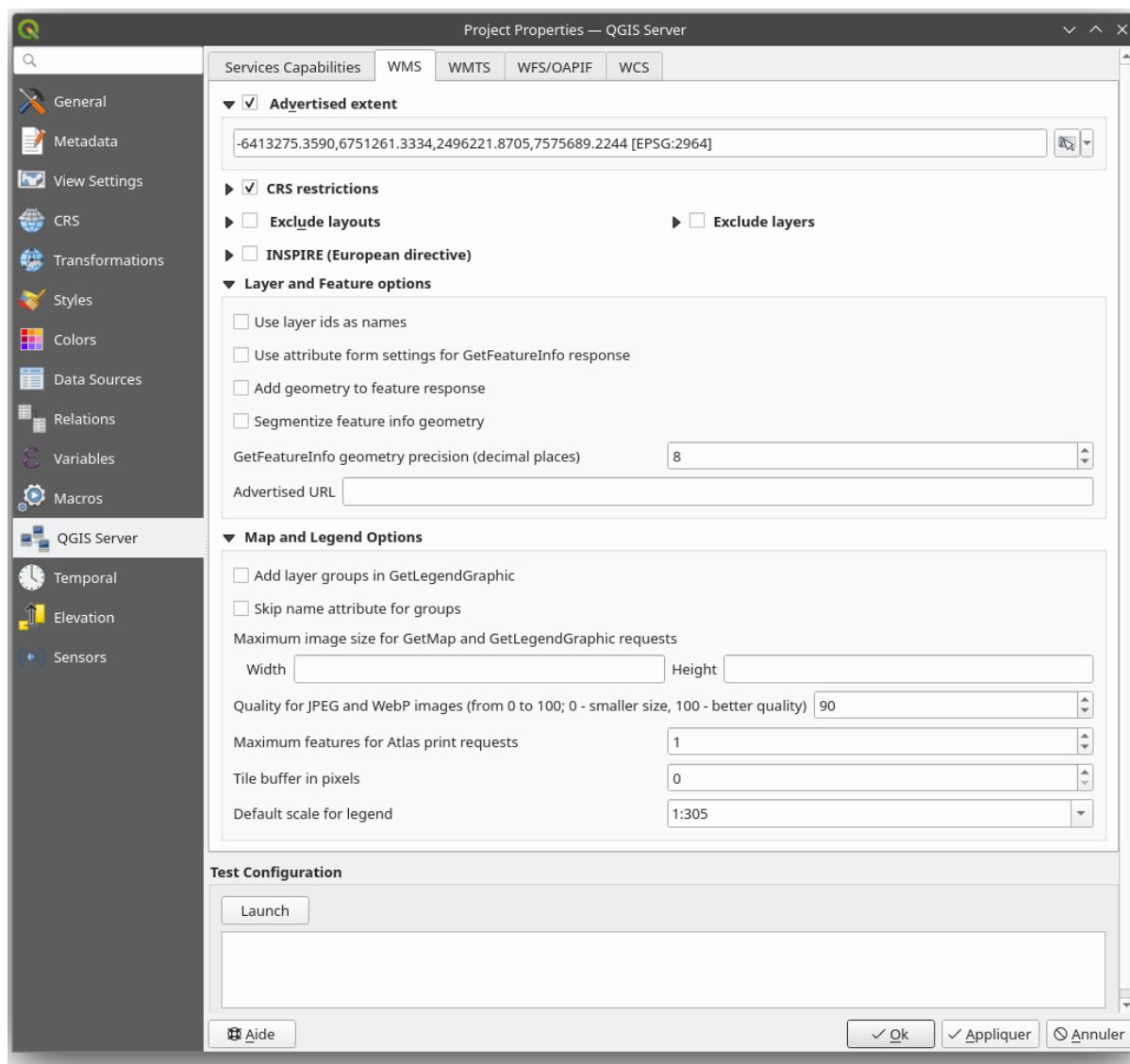


Figure2.3: Définitions dans l'onglet WMS

In the *WMS* tab, you can define the options for the WMS capabilities.

- Check *Advertised extent* to define the extent advertised in the WMS *GetCapabilities* response. The spatial extent selector widget helps you enter the extent as a *xmin*, *xmax*, *ymin*, *ymax* text or pick it from the map canvas, layers, bookmarks...
- By checking ☒ *CRS restrictions*, you can restrict in which coordinate reference systems (CRS) QGIS Server will offer to render maps. It is recommended that you restrict the offered CRS as this reduces the size of the WMS *GetCapabilities* response. Use the  button below to select those CRSs from the Coordinate Reference System Selector, or click *Used* to add the CRSs used in the QGIS project to the list.
- 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.
- *Layer and Feature Options*

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

- 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.
- If you wish, you can check  *Add geometry to feature response*. This will include the bounding box for each feature in the GetFeatureInfo response. See also the [WITH_GEOMETRY](#) parameter.
- 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 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.
- 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*.
- *Map and Legend Options*
 - When a layer group is passed to GetLegendGraphic request, all of its leaf layers are added to the legend picture (however without the groups' labels). Check the  *Add layer groups in GetLegendGraphic* option if you want to also insert the layer groups (and subgroups) names into the layer tree, just like in QGIS Desktop legend.
 - When QGIS project contains layer groups, they are listed in WMS capabilities document alongside with layers. If a group (its name as listed in capabilities) is included in WMS GetMap `LAYERS` parameter alongside with names of layers in that group, QGIS would duplicate the layers: once for the group and once for specific layer. If you check the  *Skip name attribute for groups* option, GetCapabilities will only return title attribute for the group but not its name attribute, making it impossible to include groups in list of layers of GetMap request.
 - Furthermore, you can restrict the maximum size of the maps returned by the requests by entering the maximum width and height into the respective fields under *Maximum image size for GetMap and GetLegendGraphic requests*.
 - You can change the *Quality for JPEG and WebP images* factor. The quality factor must be in the range 0 to 100. Specify 0 for maximum compression, 100 for no 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*.
 - When QGIS Server is used in tiled mode (see [TILED parameter](#)), you can set the *Tile buffer in pixels*. The recommended value is the size of the largest symbol or line width in your QGIS project.

- Depending on whether the map uses a projected CRS or a geographic CRS and if there is no information to evaluate the map unit sized symbols, you can provide reference for size through either a *Default scale for legend* or *Default map units per mm in legend*.

2.4.2 Capacités WMTS

In the *WMTS* tab you can select the layers you want to publish as WMTS and specify if you want to publish as PNG or JPEG.

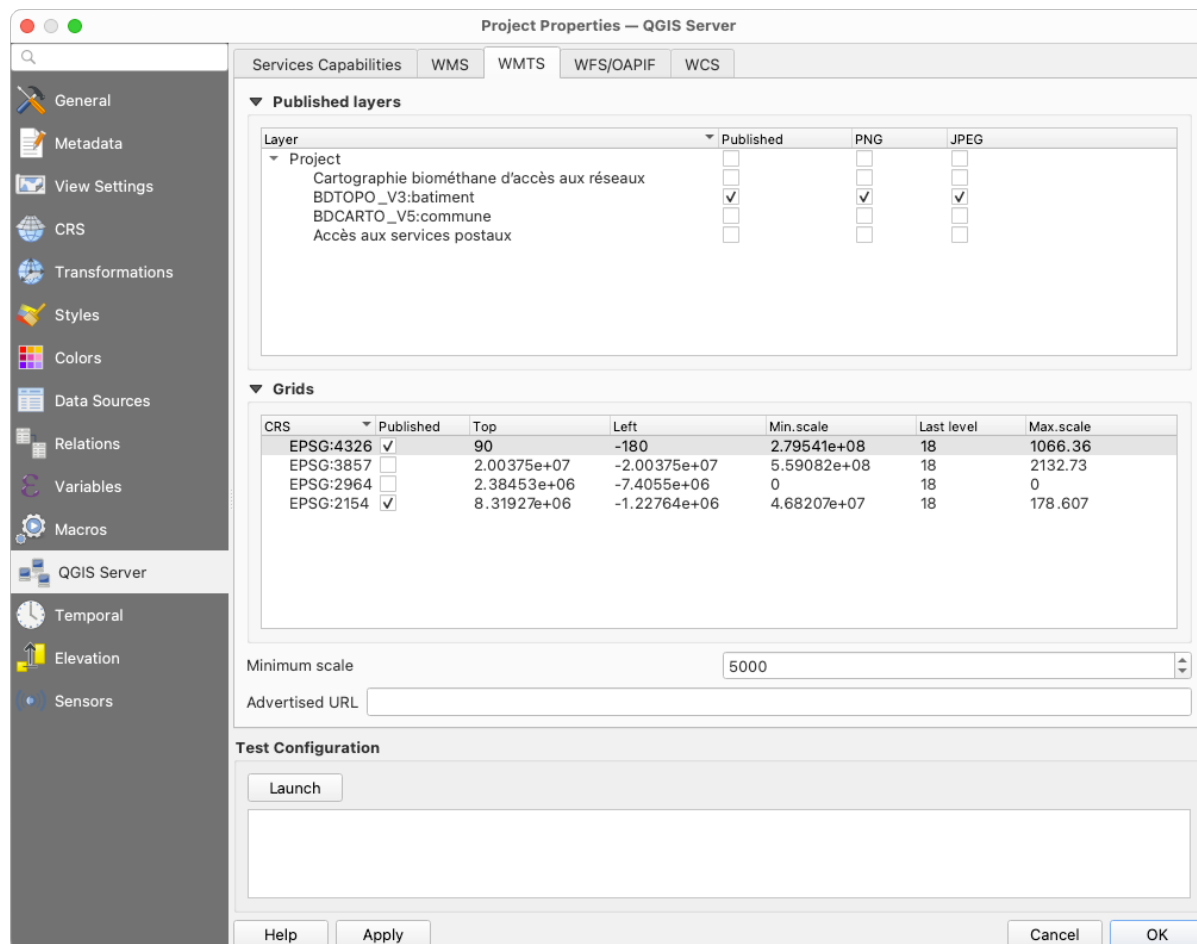


Figure2.4: Définitions dans l'onglet WMTS

If you enter a URL in the *Advertised URL* field, QGIS Server will advertise this specific URL in the WMTS GetCapabilities response.

2.4.3 Capacités WFS/OAPIF

In the *WFS/OAPIF* tab, you can select the layers you want to publish as WFS or OAPIF, and specify if they will allow update, insert and delete operations.

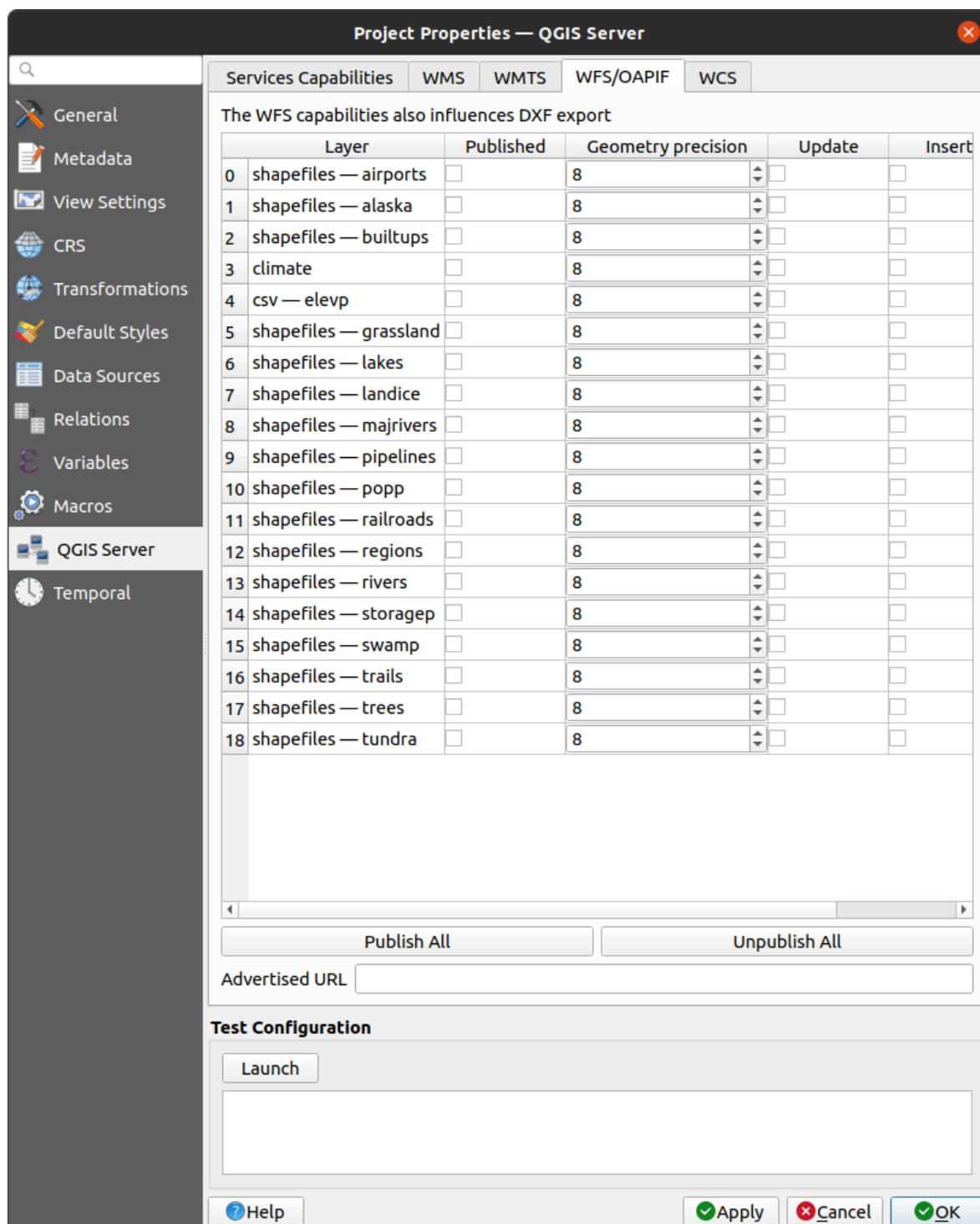


Figure2.5: Définitions dans l'onglet WFS/OAPIF

If you enter a URL in the *Advertised URL* field, QGIS Server will advertise this specific URL in the WFS GetCapabilities response.

2.4.4 Capacités WCS

In the WCS tab, you can select the layers that you want to publish as WCS.

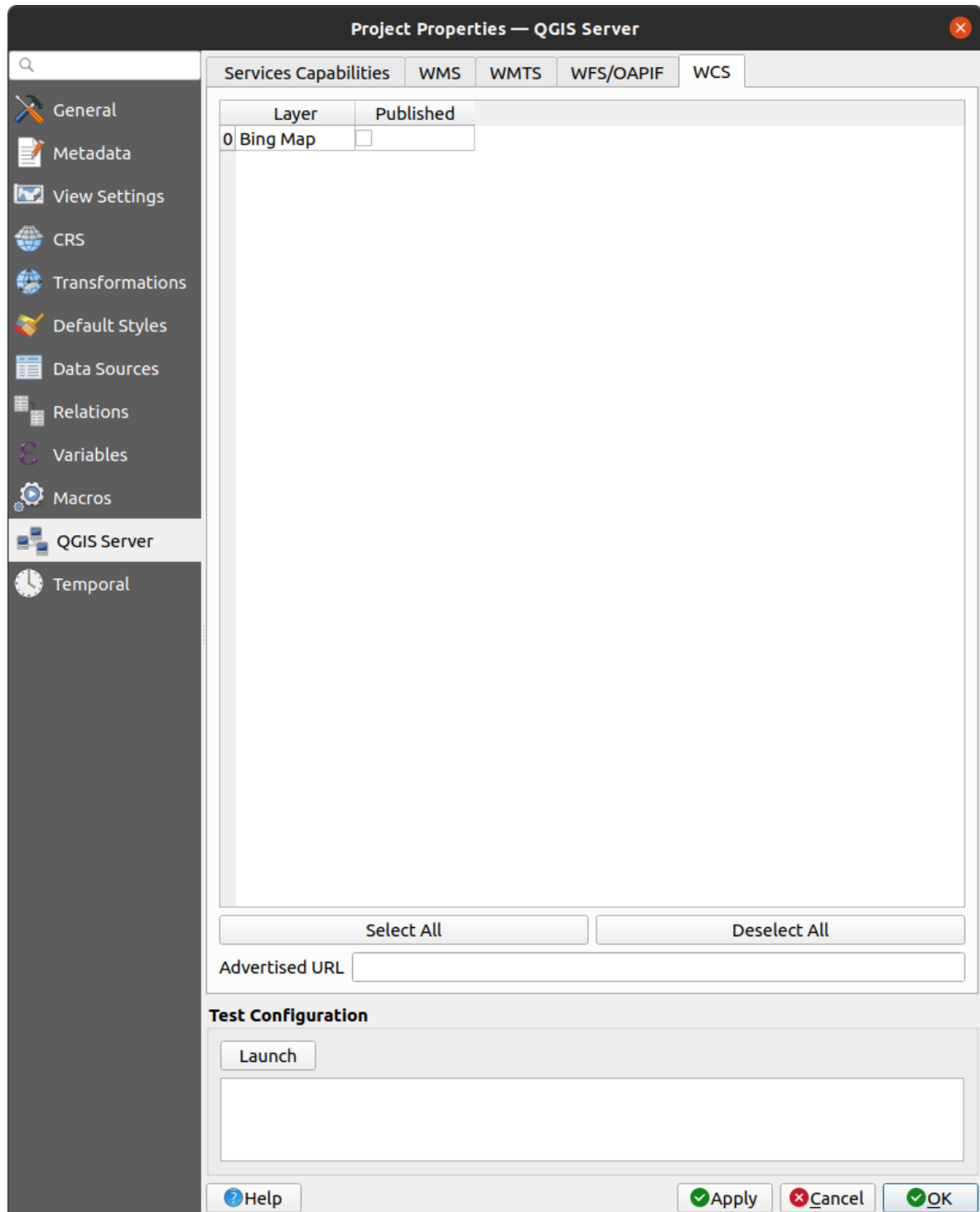


Figure2.6: Définitions dans l'onglet WCS

If you enter a URL in the *Advertised URL* field of the *WCS capabilities* section, QGIS Server will advertise this specific URL in the WCS GetCapabilities response.

2.4.5 Bien configurer votre OWS

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

- *Do not expose in WFS*
- *Do not expose in WMS*

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

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

2.5.3 Intégration avec QWC2

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

QGIS 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 1.0.0 et 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 Les bases

This section describes concepts and parameters mutually shared by services. Some of these are standard and defined in OGC specifications while others are very specific to QGIS Server.

Concepts standards :

Concept	Description
<i>SERVICE</i>	Nom du service
<i>REQUEST</i>	Nom de la requête

Concepts propres à QGIS :

Concept	Description
<i>MAP</i>	Fichier du projet QGIS
<i>FILE_NAME</i>	Nom du fichier téléchargé
<i>Nom court</i>	Short name definition

3.1.1 SERVICE

This standard parameter allows to specify the name of the service to use for a specific *request* and has to be formed like `SERVICE=NAME`.

URL example for the **WMS** service:

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

Note: Not available for REST based services like *WFS3 (OGC API Features)*.

3.1.2 REQUEST

This standard parameter allows to specify the name of the request to execute for a specific *service* and has to be formed like `REQUEST=RequestName`.

Exemple d'URL pour la requête **GetCapabilities** :

```
http://localhost/qgisserver?  
REQUEST=GetCapabilities  
&...
```

Note: Not available for REST based services like *WFS3 (OGC API Features)*.

3.1.3 MAP

This vendor parameter allows to define the QGIS project file to use. It may be an absolute path or a path relative to the location of the server executable `qgis_mapserv.fcgi`. MAP is mandatory by default because a request needs a QGIS project to actually work. However, the **QGIS_PROJECT_FILE** environment variable may be used to define a default QGIS project. In this specific case, MAP is no longer a required parameter. For further information you may refer to the *Configuration avancée* chapter.

Exemple d'URL :

```
http://localhost/qgisserver?  
MAP=/tmp/QGIS-Training-Data/exercise_data/qgis-server-tutorial-data/world.qgs  
&...
```

3.1.4 FILE_NAME

If this vendor parameter is set, the server response will be sent to the client as a file attachment with the specified file name.

URL example to save an XML **GetCapabilities** document:

```
http://localhost/qgisserver?  
SERVICE=WMS  
&REQUEST=GetCapabilities  
&FILE_NAME=wms_capabilities.xml  
&...
```

Note: Not available for REST based services like *WFS3 (OGC API Features)*.

3.1.5 Nom court

A number of elements have both a **short name** and a **title**. The short name is a text string used for machine-to-machine communication while the title is for the benefit of humans. For example, a dataset might have the descriptive title “*Maximum Atmospheric Temperature*” and be requested using the abbreviated short name “*ATMAX*”. You can set title, short name and abstract for:

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

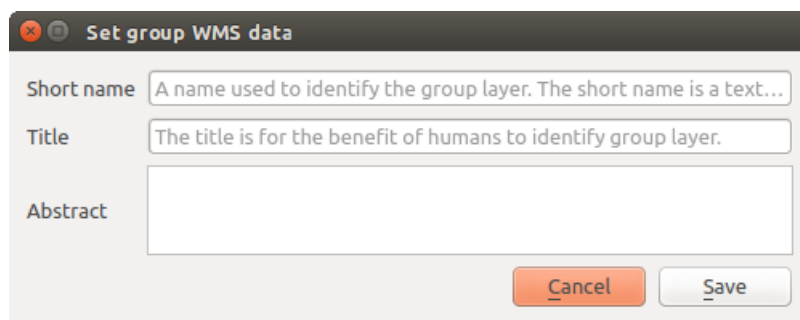


Figure3.1: Set group WMS data dialog

Thus, the short name may be used to identify these items when interacting with QGIS Server. For example with the standard `LAYERS` parameter:

```
http://localhost/qgisserver?
SERVICE=WMS
&REQUEST=GetMap
&LAYERS=shortname1,shortname2
&...
```

3.2 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>GetLegendGraphic</i>	Renvoie les symboles de légende
<i>GetStyle(s)</i>	Renvoie un document XML avec la description du style en SLD
<i>DescribeLayer</i>	Returns information about WFS and WCS availability respectively for vector and raster layers

Demandes des fournisseurs fournies par QGIS Server:

Requête	Description
<i>GetPrint</i>	Renvoie une mise en page QGIS
<i>GetProjectSettings</i>	Renvoie des informations spécifiques sur QGIS Server
<i>GetSchemaExtension</i>	Returns XML metadata about optional extended capabilities

3.2.1 GetCapabilities

Standard parameters for the **GetCapabilities** request according to the OGC WMS 1.1.1 and 1.3.0 specifications:

Paramètre	Requis	Description
<i>SERVICE</i>	Oui	Nom du service (WMS)
<i>REQUEST</i>	Oui	Nom de la requête (GetCapabilities)
<i>VERSION</i>	Non	Version du service

The **GetCapabilities** request supports as well the following vendor parameters:

Paramètre	Requis	Description
<i>MAP</i>	Oui	Fichier du projet QGIS

Exemple d'URL :

```
http://localhost/qgisserver?
SERVICE=WMS
&VERSION=1.3.0
&REQUEST=GetCapabilities
```

3.2.2 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 (WMS)
<i>REQUEST</i>	Oui	Nom de la requête (GetMap)
<i>VERSION</i>	Oui	Version du service
<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

In addition to the standard ones, QGIS Server supports *redlining*, *external WMS layers* as well as the following extra parameters:

Paramètre	Requis	Description
<i>MAP</i>	Oui	Fichier du 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/qgisserver?
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
&TRANSPARENT=TRUE
&DPI=300
&TILED=TRUE
```

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

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:

- the layer id: the project option allowing to select layers by their id is in *QGIS Server* ► *WMS* tab of the *Project Properties...* dialog. Check the *Use layer ids as names* checkbox to activate this option.
- un *nom court*

```
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 `default`.

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
- image/webp
- application/dxf: only layers that have read access in the WFS service are exported in the DXF format
- application/pdf

Exemple d'URL :

```
http://localhost/qgisserver?  
SERVICE=WMS&VERSION=1.3.0  
&REQUEST=GetMap  
&FORMAT=application/dxf  
&LAYERS=Haltungen,Normschacht,Spezialbauwerke  
&CRS=EPSG%3A21781  
&BBOX=696136.28844801,245797.12108743,696318.91114315,245939.25832905  
&WIDTH=1042  
&HEIGHT=811  
&FORMAT_OPTIONS=MODE:SYMBOLLAYERSYMBOLGY;SCALE:250  
&FILE_NAME=plan.dxf
```

TRANSPARENT

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

- TRUE
- FALSE

However, this parameter is ignored if the format of the image indicated with `FORMAT` is different from PNG.

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

Comma separated list of opacity values. Opacity can be set on layer or group level. Allowed values range from 0 (fully transparent) to 255 (fully opaque).

Exemple d'URL :

```
http://localhost/qgisserver?
SERVICE=WMS
&REQUEST=GetMap
&VERSION=1.3.0
&WIDTH=400
&HEIGHT=200
&CRS=EPSG:4326
&LAYERS=countries,places
&BBOX=42,-6,52,15
&OPACITIES=255,0
```



Figure 3.2: A gauche OPACITIES=255, 0 et à droite OPACITIES=255, 255

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
&VERSION=1.3.0
&WIDTH=400
&HEIGHT=300
&CRS=EPSG:4326
&BBOX=41,-6,52,10
&LAYERS=countries_shapeburst,countries,places
&FILTER=countries_shapeburst,countries:"name" = 'France';places: "name" = 'Paris'
```



Figure3.3: Server response to a GetMap request with FILTER parameter

In this example, the same filter "name" = 'France' is applied to layers `countries` and `countries_shapeburst`, while the filter "name" = 'Paris' is only applied to `places`.

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.



Figure3.4: Réponse du serveur à une demande GetMap avec le paramètre SELECTION

FORMAT_OPTIONS

This parameter can be used to specify options for the selected format. Only for `FORMAT=application/dxf` in GetMap request. Takes a list of key:value pairs separated by semicolon:

- **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`.
- **LAYERATTRIBUTES**: specify a field or in case of many layers a comma separated list of fields that contains values for DXF layer names - if not specified, the original QGIS layer names are used.
- **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`.

3.2.3 GetFeatureInfo

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

Paramètr	Requ	Description
<i>SERVICE</i>	Oui	Nom du service (WMS)
<i>REQUEST</i>	Oui	nom de la requete (GetFeatureInfo)
<i>VERSION</i>	Non	Version du service
<i>QUERY_L</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>TRANSPA</i>	Non	Arrière-plan transparent
<i>INFO_FOI</i>	Non	Format de sortie
<i>FEATURE</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_PR</i>	Non	La précision (nombre de chiffres) à utiliser lors du retour de la géométrie (voir comment ajouter une géométrie à une entité). 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	Fichier du 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_DISPLAY_NAME</i>	Non	Add the feature display name to the output
<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
```

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

See the `LAYERS` parameter defined in *GetMap* for more information.

FEATURE_COUNT

This parameter specifies the maximum number of features per layer to return. For example if `QUERY_LAYERS` is set to `layer1,layer2` and `FEATURE_COUNT` is set to 3 then a maximum of 3 features from `layer1` will be returned. Likewise a maximum of 3 features from `layer2` will be returned.

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_DISPLAY_NAME

This parameter specifies whether to add feature display name to the output.

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

- TRUE
- FALSE

WITH_MAPTIP

Ce paramètre spécifie s'il faut ajouter des infobulles à la sortie.

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

- TRUE
- FALSE
- **HTML_FI_ONLY_MAPTIP**: like TRUE, with the difference that the HTML response to the feature info request only contains the maptip. This gives full control over the HTML response using e.g. the built-in layer maptip editor.

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

3.2.4 GetLegendGraphic

Paramètres standards pour la requête **GetLegendGraphic** selon les spécifications OGC WMS 1.1.1 et 1.3.0 :

Paramètre	Requis	Description
<i>SERVICE</i>	Oui	Nom du service (WMS)
<i>REQUEST</i>	Oui	Nom de la requête (GetLegendGraphic)
<i>VERSION</i>	Non	Version du service
<i>LAYERS</i>	Oui	Couches à afficher
<i>STYLES</i>	Non	Style des couches
<i>SRS / CRS</i>	Non	Système de coordonnées de référence
<i>BBOX</i>	Non	Emprise de la carte
<i>WIDTH</i>	Non	Largeur de l'image en pixels
<i>HEIGHT</i>	Non	Hauteur de l'image en pixels
<i>FORMAT</i>	Non	Format de légende
<i>TRANSPARENT</i>	Non	Arrière-plan transparent

In addition to the standard ones, QGIS Server supports extra parameters to change the size of the legend elements or the font properties for layer titles and item labels:

Paramètre	Requis	Description
<i>MAP</i>	Oui	Fichier du projet QGIS
<i>SRCWIDTH</i>	Non	Largeur de carte
<i>SRCHEIGHT</i>	Non	Hauteur de carte
<i>SHOWFEATURECOUNT</i>	Non	Ajouter le décompte aux entités
<i>RULE</i>	Non	Rule symbol to render
<i>RULELABEL</i>	Non	Item labels rendering
<i>BOXSPACE</i>	Non	Space between legend frame and content (mm)
<i>LAYERSPACE</i>	Non	Vertical space between layers (mm)
<i>LAYERTITLESPACE</i>	Non	Vertical space between layer title and items (mm)
<i>SYMBOLSPACE</i>	Non	Vertical space between symbol and items (mm)
<i>ICONLABELSPACE</i>	Non	Horizontal space between symbol and label (mm)
<i>SYMBOLWIDTH</i>	Non	Width of the symbol preview (mm)
<i>SYMBOLHEIGHT</i>	Non	Height of the symbol preview (mm)
<i>LAYERTITLE</i>	Non	Layer title rendering
<i>LAYERFONTFAMILY</i>	Non	Layer font family
<i>LAYERFONTBOLD</i>	Non	Layer title bold rendering
<i>LAYERFONTSIZE</i>	Non	Layer title font size (pt)
<i>LAYERFONTITALIC</i>	Non	Layer title italic rendering
<i>LAYERFONTCOLOR</i>	Non	Layer title color
<i>ITEMFONTFAMILY</i>	Non	Item font family
<i>ITEMFONTBOLD</i>	Non	Item label bold rendering
<i>ITEMFONTSIZE</i>	Non	Item label font size (pt)
<i>ITEMFONTITALIC</i>	Non	Item label italic rendering
<i>ITEMFONTCOLOR</i>	Non	Item label color
<i>SHOWRULEDETAILS</i>	Non	Adds the rule text to JSON output
<i>ADDLAYERGROUPS</i>	Non	Adds the layer groups to JSON output

BBOX

This parameter can be used to specify the geographical area for which the legend should be built (its format is described [here](#)) but cannot be combined with the *RULE* parameter. The *SRS/CRS* parameter becomes mandatory when using the *BBOX* parameter.

Exemple d'URL :

```
http://localhost/qgisserver?
SERVICE=WMS
&REQUEST=GetLegendGraphic
&LAYERS=countries,airports
&BBOX=43.20,-2.93,49.35,8.32
&CRS=EPSG:4326
```

Note: When the *BBOX* parameter is defined, the legend is referred to as a *content based legend*.

WIDTH

This parameter is not used by default but becomes mandatory when the `RULE` parameter is set. In this case it allows to specify the width in pixels of the output image.

```
http://localhost/qgisserver?  
SERVICE=WMS  
&REQUEST=getlegendgraphic  
&LAYER=testlayer%20%C3%A8%C3%A9  
&RULE=rule1  
&WIDTH=30  
&HEIGHT=30
```

HEIGHT

This parameter is not used by default but becomes mandatory when the `RULE` parameter is set. In this case it allows to specify the height in pixels of the output image.

```
http://localhost/qgisserver?  
SERVICE=WMS  
&REQUEST=GetLegendGraphic  
&LAYER=testlayer%20%C3%A8%C3%A9  
&RULE=rule1  
&WIDTH=30  
&HEIGHT=30
```

FORMAT

This parameter may be used to specify the format of legend image. Available values are:

- `image/jpeg`
- `image/png`
- `application/json`

For JSON, symbols are encoded with Base64 and most other options related to layout or fonts are not taken into account because the legend must be built on the client side. The `RULE` parameter cannot be combined with this format.

URL example with the corresponding JSON output:

```
http://localhost/qgisserver?  
SERVICE=WMS&  
REQUEST=GetLegendGraphic&  
LAYERS=airports&  
FORMAT=application/json
```

And the corresponding JSON output:

```
{  
  "nodes": [  
    {  
      "icon": "<base64 icon>",  
      "title": "airports",  
      "type": "layer"  
    }  
  ],  
  "title": ""  
}
```

SRCWIDTH

This parameter may be defined when the `RULE` parameter is set. In this case, the `SRCWIDTH` value is forwarded to the underlying `GetMap` request as the `WIDTH` parameter while the `WIDTH` parameter of `GetLegendGraphic` is used for the image legend size.

SRCHEIGHT

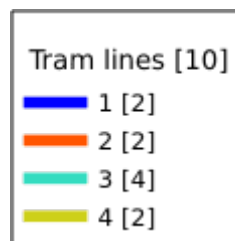
This parameter may be defined when the `RULE` parameter is set. In this case, the `SRCHEIGHT` value is forwarded to the underlying `GetMap` request as the `HEIGHT` parameter while the `HEIGHT` parameter of `GetLegendGraphic` is used for the image legend size.

SHOWFEATURECOUNT

This parameter can be used to activate feature count in the legend. Available values are (not case sensitive):

- TRUE
- FALSE

Par exemple:



RULE

This parameter is available on layers with *Rule-based* rendering and allows to build a legend with only the named rule symbol. It cannot be combined with `BBOX` parameter. `HEIGHT` and `WIDTH` must be specified.

Exemple d'URL :

```
http://localhost/qgisserver?
SERVICE=WMS
&REQUEST=GetLegendGraphic
&LAYERS=mylayer,
&RULE=myrulename,
&WIDTH=20,
&HEIGHT=20
```

RULELABEL

This parameter allows to control the item label rendering. Available values are (not case sensitive):

- TRUE : affiche l'étiquette de l'élément
- FALSE : cache l'étiquette de l'élément
- AUTO : cache l'étiquette de l'élément pour les couches ayant un rendu de type *Symbole unique*

Exemple d'URL :

```
http://localhost/qgisserver?
SERVICE=WMS
&REQUEST=GetLegendGraphic
&LAYERS=countries,airports
&BBOX=43.20,-2.93,49.35,8.32
&CRS=EPSG:4326
&TRANSPARENT=TRUE
&RULELABEL=AUTO
```



Figure3.5: Legend rendering without label for single symbol layers

BOXSPACE

This parameter allows to specify the space between legend frame and content in millimeters. By default, the space value is 2 mm.

Exemple d'URL :

```
http://localhost/qgisserver?
SERVICE=WMS
&REQUEST=GetLegendGraphic
&LAYERS=airports
&BBOX=43.20,-2.93,49.35,8.32
&CRS=EPSG:4326
&TRANSPARENT=TRUE
&BOXSPACE=0
```




Figure3.6: To the left BOXSPACE=0 and to the right BOXSPACE=15

LAYERSPACE

This parameter allows to specify the vertical space between layers in millimeters. By default, the space value is 3 mm.

Exemple d'URL :

```
http://localhost/qgisserver?
SERVICE=WMS
&REQUEST=GetLegendGraphic
&LAYERS=airports,places
&BBOX=43.20,-2.93,49.35,8.32
&CRS=EPSG:4326
&TRANSPARENT=TRUE
&LAYERSPACE=0
```



Figure3.7: To the left LAYERSPACE=0 and to the right LAYERSPACE=10

LAYERTITLESPACE

This parameter allows to specify the vertical space between layer title and items following in millimeters. By default the space value is 3 mm.

Exemple d'URL :

```
http://localhost/qgisserver?
SERVICE=WMS
&REQUEST=GetLegendGraphic
&LAYERS=airports,places
&BBOX=43.20,-2.93,49.35,8.32
&CRS=EPSG:4326
```

(suite sur la page suivante)

```
&TRANSPARENT=TRUE
&LAYERTITLESPACE=0
```

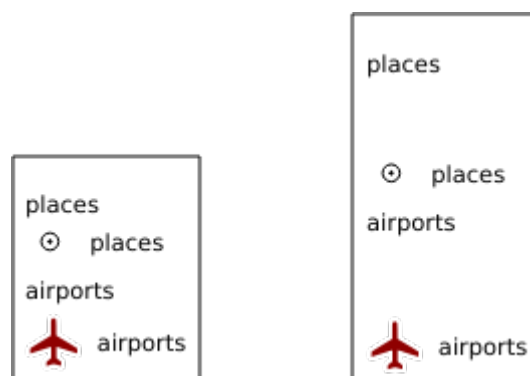


Figure3.8: To the left LAYERTITLESPACE=0 and to the right LAYERTITLESPACE=10

SYMBOLSPACE

This parameter allows to specify the vertical space between symbol and item following in millimeters. By default the space value is 2 mm.

Exemple d'URL :

```
http://localhost/qgisserver?
SERVICE=WMS
&REQUEST=GetLegendGraphic
&LAYERS=countries
&BBOX=43.20,-2.93,49.35,8.32
&CRS=EPSG:4326
&TRANSPARENT=TRUE
&SYMBOLSPACE=0
```

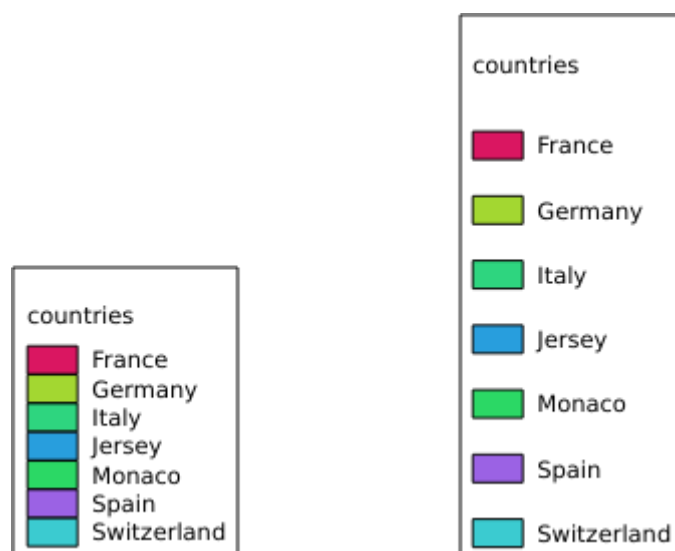


Figure3.9: To the left SYMBOLSPACE=0 and to the right SYMBOLSPACE=5

ICONLABELSPACE

This parameter allows to specify the horizontal space between symbol and label text in millimeters. By default the space value is 2 mm.

Exemple d'URL :

```
http://localhost/qgisserver?
SERVICE=WMS
&REQUEST=getlegendgraphic
&LAYERS=countries,
&BBOX=43.20,-2.93,49.35,8.32
&CRS=EPSG:4326
&TRANSPARENT=TRUE
&ICONLABELSPACE=0
```



Figure3.10: To the left ICONLABELSPACE=0 and to the right ICONLABELSPACE=10

SYMBOLWIDTH

This parameter allows to specify the width of the symbol preview in millimeters. By default the width value is 7 mm.

Exemple d'URL :

```
http://localhost/qgisserver?
SERVICE=WMS
&REQUEST=GetLegendGraphic
&LAYERS=countries,
&BBOX=43.20,-2.93,49.35,8.32
&CRS=EPSG:4326
&TRANSPARENT=TRUE
&SYMBOLWIDTH=2
```



Figure3.11: To the left SYMBOLWIDTH=2 and to the right SYMBOLWIDTH=20

SYMBOLHEIGHT

This parameter allows to specify the height of the symbol preview in millimeters. By default the height value is 4 mm.

Exemple d'URL :

```
http://localhost/qgisserver?
SERVICE=WMS
&REQUEST=GetLegendGraphic
&LAYERS=countries,
&BBOX=43.20,-2.93,49.35,8.32
&CRS=EPSG:4326
&TRANSPARENT=TRUE
&SYMBOLHEIGHT=2
```



Figure3.12: To the left SYMBOLHEIGHT=2 and to the right SYMBOLHEIGHT=6

LAYERTITLE

This parameter specifies whether to render layer title.

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

- TRUE (valeur par défaut)
- FALSE

LAYERFONTFAMILY

This parameter specifies the font family to use for rendering layer title.

```
http://localhost/qgisserver?
SERVICE=WMS
&REQUEST=GetLegendGraphic
&LAYERS=countries
&LAYERFONTFAMILY=monospace
```

LAYERFONTBOLD

This parameter specifies whether the layer title is rendered in bold. Available values are (not case sensitive):

- TRUE
- FALSE

Exemple d'URL :

```
http://localhost/qgisserver?
SERVICE=WMS
&REQUEST=GetLegendGraphic
&LAYERS=airports,places
&BBOX=43.20,-2.93,49.35,8.32
&CRS=EPSG:4326
&TRANSPARENT=TRUE
&LAYERFONTBOLD=TRUE
```



Figure3.13: Légende avec LAYERFONTBOLD=TRUE

LAYERFONTSIZE

This parameter specifies the font size for rendering layer title in point.

Exemple d'URL :

```
http://localhost/qgisserver?
SERVICE=WMS
&REQUEST=GetLegendGraphic
&LAYERS=airports,places
&BBOX=43.20,-2.93,49.35,8.32
&CRS=EPSG:4326
&TRANSPARENT=TRUE
&LAYERFONTSIZE=20
```

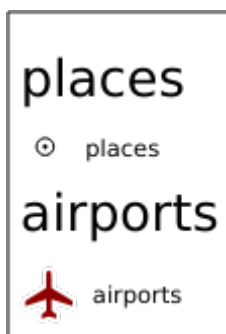


Figure3.14: Légende avec LAYERFONTSIZE=20

LAYERFONTITALIC

This parameter specifies whether the layer title is rendered in italic. Available values are (not case sensitive):

- TRUE
- FALSE

Exemple d'URL :

```
http://localhost/qgisserver?
SERVICE=WMS
&REQUEST=GetLegendGraphic
&LAYERS=airports,places
&BBOX=43.20,-2.93,49.35,8.32
&CRS=EPSG:4326
&TRANSPARENT=TRUE
&LAYERFONTITALIC=TRUE
```

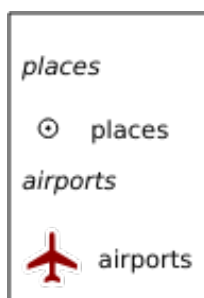


Figure3.15: Légende avec LAYERFONTITALIC=TRUE

LAYERFONTCOLOR

This parameter specifies the layer title color. The color may be literal (red, green, ..) or in hexadecimal notation (0xFF0000, 0x00FF00, ...).

Exemple d'URL :

```
http://localhost/qgisserver?
SERVICE=WMS
&REQUEST=GetLegendGraphic
&LAYERS=airports,places
&BBOX=43.20,-2.93,49.35,8.32
&CRS=EPSG:4326
&TRANSPARENT=TRUE
&LAYERFONTCOLOR=0x5f9930
```

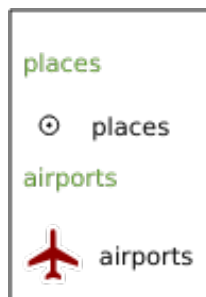


Figure3.16: Légende avec LAYERFONTCOLOR=0x5f9930

ITEMFONTFAMILY

This parameter specifies the font family to use for rendering item label.

```
http://localhost/qgisserver?
SERVICE=WMS
&REQUEST=GetLegendGraphic
&LAYERS=countries
&ITEMFONTFAMILY=monospace
```

ITEMFONTBOLD

This parameter specifies whether the item label is rendered in bold. Available values are (not case sensitive):

- TRUE
- FALSE

Exemple d'URL :

```
http://localhost/qgisserver?
SERVICE=WMS
&REQUEST=GetLegendGraphic
&LAYERS=airports,places
&BBOX=43.20,-2.93,49.35,8.32
&CRS=EPSG:4326
&TRANSPARENT=TRUE
&ITEMFONTBOLD=TRUE
```

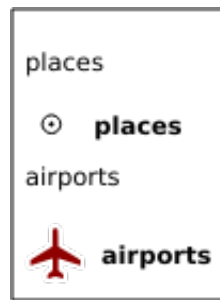


Figure3.17: Légende avec ITEMFONTBOLD=TRUE

ITEMFONTSIZE

This parameter specifies the font size for rendering layer title in point.

Exemple d'URL :

```
http://localhost/qgisserver?
SERVICE=WMS
&REQUEST=GetLegendGraphic
&LAYERS=airports,places
&BBOX=43.20,-2.93,49.35,8.32
&CRS=EPSG:4326
&TRANSPARENT=TRUE
&ITEMFONTSIZE=20
```

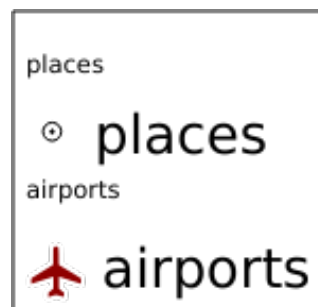


Figure3.18: Légende avec ITEMFONTSIZE=30

ITEMFONTITALIC

This parameter specifies whether the item label is rendered in italic. Available values are (not case sensitive):

- TRUE
- FALSE

Exemple d'URL :

```
http://localhost/qgisserver?
SERVICE=WMS
&REQUEST=GetLegendGraphic
&LAYERS=airports,places
&BBOX=43.20,-2.93,49.35,8.32
&CRS=EPSG:4326
&TRANSPARENT=TRUE
&ITEMFONTITALIC=TRUE
```

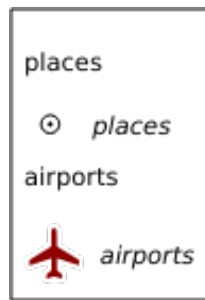



Figure3.19: Légende avec ITEMFONTITALIC=TRUE

ITEMFONTCOLOR

This parameter specifies the item label color. The color may be literal (red, green, ..) or in hexadecimal notation (0xFF0000, 0x00FF00, ...).

Exemple d'URL :

```
http://localhost/qgisserver?
SERVICE=WMS
&REQUEST=GetLegendGraphic
&LAYERS=airports,places
&BBOX=43.20,-2.93,49.35,8.32
&CRS=EPSG:4326
&TRANSPARENT=TRUE
&ITEMFONTCOLOR=0x5f9930
```

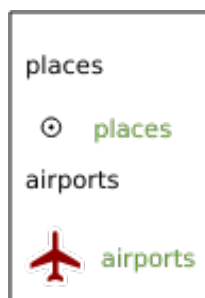


Figure3.20: Légende avec ITEMFONTCOLOR=0x5f9930

SHOWRULEDETAILS

This parameter specifies if the JSON output will also contain the details about the rule that generated the legend entry. This parameter only has effect when the renderer is rule-based or categorized.

URL example with the corresponding JSON output:

```
http://localhost/qgisserver?
SERVICE=WMS&
REQUEST=GetLegendGraphic&
LAYERS=airports&
FORMAT=application/json&
SHOWRULEDETAILS=TRUE
```

And the corresponding JSON output:

```
{
  "nodes": [
    {
      "icon": "<base64 icon>",
      "title": "airports",
      "type": "layer",
      "rule": "type = 'airport'"
    }
  ],
  "title": ""
}
```

ADDLAYERGROUPS

This parameter specifies if the JSON output will also display the names of the layers groups (and subgroups) in the legend entry. Possible values are:

- TRUE: display the groups labels
- FALSE (default): hide the groups labels

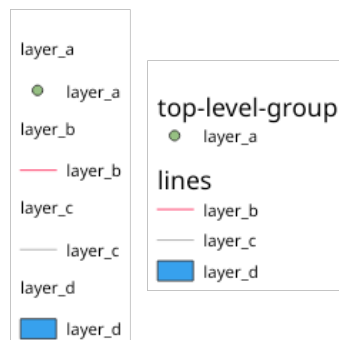


Figure3.21: Legend without (left) and with (right) layer groups labels display

The corresponding JSON output showing groups name would look like:

```
{ "nodes":
  [{ "nodes":
    [{ "icon":
      ↪ "iVBORw0KGgoAAAANSUhEUgAAABQAAAAUCAYAAACNiR0NAAAAACXBIWXMAAABY1AAAWJQFJU
      ↪ 37128evBKKisr/
      ↪ 0+a7IMiNi93C15DcUpgM4wYQ5nWuZAcMIQNvHv3LuO83C0kG0hysoFZRrJNo2AYAQC87BpkGQj1fwAAAAABJRU5Erk
      ↪ ",
      "title": "layer_a", "type": "layer"
    },
    { "nodes":
      [{ "icon":
        ↪ "iVBORw0KGgoAAAANSUhEUgAAABMAAAAUCAyAAABvVQZ0AAAAACXBIWXMAAABY1AAAWJQFJU
        ↪ 8aaEeMglEwYAAAAIoCzTtn5XoAAAAASUVORK5CYII=",
        "title": "layer_b", "type": "layer"
      },
      { "icon":
        ↪ "iVBORw0KGgoAAAANSUhEUgAAABMAAAAUCAyAAABvVQZ0AAAAACXBIWXMAAABY1AAAWJQFJU
        ↪ 8aaEeMglEwYAAAAIoCzTtn5XoAAAAASUVORK5CYII=",
        "title": "layer_c", "type": "layer"
      }
    ],
    "title": "lines", "type": "group"
  }
],
  "title": ""
}
```

(suite sur la page suivante)

(suite de la page précédente)

```

{
  "icon":
  ↪ "iVBORw0KGgoAAAANSUgAAABMAAAATCAyAAABYUDbMAAAACXBIWXMAABY1AAAWJQFJUitWAAAAKklEQVQ4jWNUV1X/
  ↪ z0AlwMLAwMDAnXmQKoYxUcWUUCNGDRs1bNSwYWYYACXDAsvQaTuVAAAAAE1FTkSuQmCC",
  "title": "layer_d", "type": "layer"
},
"title": "top-level-group", "type": "group"
}],
"title": ""
}

```

3.2.5 GetStyle(s)

Standard parameters for the **GetStyle** (or **GetStyles**) request according to the OGC WMS 1.1.1 specifications:

Paramètre	Requis	Description
<i>SERVICE</i>	Oui	Nom du service (WMS)
<i>REQUEST</i>	Oui	Nom de la requête (GetStyle ou GetStyles)
<i>LAYERS</i>	Oui	Couches à interroger

The **GetStyle** request supports as well the following vendor parameters:

Paramètre	Requis	Description
<i>MAP</i>	Oui	Fichier du projet QGIS

Exemple d'URL :

```

http://localhost/qgisserver?
SERVICE=WMS
&REQUEST=GetStyles
&LAYERS=mylayer1,mylayer2

```

3.2.6 DescribeLayer

Standard parameters for the **DescribeLayer** request according to the OGC WMS 1.1.1 and 1.3.0 specifications:

Paramètre	Requis	Description
<i>SERVICE</i>	Oui	Nom du service (WMS)
<i>REQUEST</i>	Oui	Nom de la requête (DescribeLayer)
<i>LAYERS</i>	Oui	Couches à décrire
<i>SLD_VERSION</i>	Oui	Version de SLD

The **DescribeLayer** request supports as well the following vendor parameters:

Paramètre	Requis	Description
<i>MAP</i>	Oui	Fichier du projet QGIS

Exemple d'URL :

```
http://localhost/qgisserver?
SERVICE=WMS
&REQUEST=DescribeLayer
&SLD_VERSION=1.1.0
&LAYERS=mylayer1
```

Le document XML ressemble à :

```
<DescribeLayerResponse xmlns="http://www.opengis.net/sld" xmlns:xsi="http://www.w3.
↳org/2001/XMLSchema-instance" xmlns:ows="http://www.opengis.net/ows" xmlns:xlink=
↳"http://www.w3.org/1999/xlink" xmlns:se="http://www.opengis.net/se"
↳xsi:schemaLocation="http://www.opengis.net/sld http://schemas.opengis.net/sld/1.
↳1.0/DescribeLayer.xsd">
  <Version>1.1.0</Version>
  <LayerDescription>
    <owsType>wfs</owsType>
    <se:OnlineResource xlink:href="http://localhost/qgisserver" xlink:type=
↳"simple"/>
    <TypeName>
      <se:FeatureTypeName>my_vector_layer</se:FeatureTypeName>
    </TypeName>
  </LayerDescription>
  <LayerDescription>
    <owsType>wcs</owsType>
    <se:OnlineResource xlink:href="http://localhost/qgisserver" xlink:type=
↳"simple"/>
    <TypeName>
      <se:FeatureTypeName>my_raster_layer</se:FeatureTypeName>
    </TypeName>
  </LayerDescription>
</DescribeLayerResponse>
```

SLD_VERSION

This parameter allows to specify the version of SLD. Only the value 1.1.0 is available.

3.2.7 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.

The **GetPrint** request supports *redlining*, *external WMS layers* as well as the following parameters:

Paramètre	Requis	Description
<i>MAP</i>	Oui	Fichier du projet QGIS
<i>SERVICE</i>	Oui	Nom du service (WMS)
<i>REQUEST</i>	Oui	Nom de la requête (GetPrint)
<i>VERSION</i>	Non	Version du service
<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>FORMAT_OPTIONS</i>	Non	Options du format de fichier spécifié Seulement pour FORMAT=application/pdf
<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_</i>	Non	Intervalle de grille sur l'axe x de la carte "X"
<i>mapX:GRID_INTERVAL_</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>
```

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.

FORMAT_OPTIONS

This parameter can be used to specify options for the selected format. Only for `FORMAT=application/pdf` in `GetPrint` requests. Takes a list of key:value pairs separated by semicolon:

- `RASTERIZE_WHOLE_IMAGE`: whether the whole pdf should be exported as an image. Default: false.
- `FORCE_VECTOR_OUTPUT`: whether pdf should be exported as vector. Default: false.
- `APPEND_GEOREFERENCE`: whether georeference info shall be added to the pdf. Default: true.
- `EXPORT_METADATA`: whether metadata shall be added to the pdf. Default: true.
- `TEXT_RENDER_FORMAT`: sets the text render format for pdf export. It can be `AlwaysOutlines` (default) or `AlwaysText`.
- `SIMPLIFY_GEOMETRY`: whether features geometries shall be simplified. Default: true.
- `WRITE_GEO_PDF`: whether a Geospatial PDF shall be exported. Default: false.
- `USE_ISO_32000_EXTENSION_FORMAT_GEOREFERENCING`: whether Iso32000 georeferencing shall be used. Default: false.
- `USE_OGC_BEST_PRACTICE_FORMAT_GEOREFERENCING`: whether OGC best practice georeferencing shall be used. Default: false.
- `EXPORT_THEMES`: a comma separated list of map themes to use for a Geospatial PDF export
- `PREDEFINED_MAP_SCALES`: a comma separated list of map scales to render the map
- `LOSSLESS_IMAGE_COMPRESSION`: whether images embedded in pdf must be compressed using a lossless algorithm. Default: false.
- `DISABLE_TILED_RASTER_RENDERING`: whether rasters shall be untiled in the pdf. Default: false.

Exemple d'URL :

```
http://localhost/qgisserver?
SERVICE=WMS
&VERSION=1.3.0
&REQUEST=GetPrint
&MAP=/home/qgis/projects/world.qgs
&CRS=EPSG:4326
&FORMAT=pdf
&TEMPLATE=Layout%201
&FORMAT_OPTIONS=FORCE_VECTOR_OUTPUT:TRUE;TEXT_RENDER_FORMAT:AlwaysOutlines;
↪PREDEFINED_MAP_SCALES:250
```

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

This parameter specifies the layers for a layout map item. See [GetMap Layers](#) for more information on this parameter.

mapX:STYLES

This parameter specifies the layers' styles defined in a specific layout map item. See *GetMap Styles* for more information on this parameter.

3.2.8 GetProjectSettings

This request type works similar to *GetCapabilities*, but it is more specific to QGIS Server and allows a client to read additional information which are not available in the *GetCapabilities* output:

- 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 exclusive

The **GetProjectSettings** request supports the following parameters:

Paramètre	Requis	Description
<i>MAP</i>	Oui	Fichier du projet QGIS
<i>SERVICE</i>	Oui	Nom du service (WMS)
<i>REQUEST</i>	Oui	Nom de la requête (GetProjectSettings)

3.2.9 GetSchemaExtension

The **GetSchemaExtension** request allows to retrieve optional extended capabilities and operations of the WMS service such as implemented by QGIS Server.

Paramètre	Requis	Description
<i>MAP</i>	Oui	Fichier du projet QGIS
<i>SERVICE</i>	Oui	Nom du service (WMS)
<i>REQUEST</i>	Oui	Nom de la requête (GetSchemaExtension)

Exemple d'URL :

```
http://localhost/qgisserver?
SERVICE=WMS
&REQUEST=GetSchemaExtension
```

Le document XML ressemble à :

```
<schema xmlns="http://www.w3.org/2001/XMLSchema" xmlns:wms="http://www.opengis.net/
↪wms" xmlns:qgs="http://www.qgis.org/wms" targetNamespace="http://www.qgis.org/wms
↪" elementFormDefault="qualified" version="1.0.0">
  <import namespace="http://www.opengis.net/wms" schemaLocation="http://schemas.
↪opengis.net/wms/1.3.0/capabilities_1_3_0.xsd"/>
  <element name="GetPrint" type="wms:OperationType" substitutionGroup="wms:_
↪ExtendedOperation"/>
  <element name="GetStyles" type="wms:OperationType" substitutionGroup="wms:_
↪ExtendedOperation"/>
</schema>
```


3.2.10 Couches WMS externes

QGIS Server allows including layers from external WMS servers in WMS *GetMap* and WMS *GetPrint* requests. This is especially useful if a web client uses an external background layer in the web map. For performance reasons, such layers should be directly requested by the web client (not cascaded via QGIS server). For printing however, these layers should be cascaded via QGIS server in order to appear in the printed map.

External layers can be added to the LAYERS parameter as EXTERNAL_WMS:<layername>. The parameters for the external WMS layers (e.g. url, format, dpiMode, crs, layers, styles) can later be given as service parameters <layername>:<parameter>. In a *GetMap* request, this might look like this:

```
http://localhost/qgisserver?
SERVICE=WMS
&REQUEST=GetMap
&LAYERS=EXTERNAL_WMS:basemap,layer1,layer2
&OPACITIES=100,200,255
&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
```

Similarly, external layers can be used in *GetPrint* requests:

```
http://localhost/qgisserver?
SERVICE=WMS
&REQUEST=GetPrint
&TEMPLATE=A4
&map0:layers=EXTERNAL_WMS:basemap,layer1,layer2
&map0:EXTENT=<minx,miny,maxx,maxy>
&OPACITIES=100,200,255
&basemap:url=http://externalserver.com/wms.fcgi
&basemap:format=image/jpeg
&basemap:dpiMode=7
&basemap:crs=EPSG:2056
&basemap:layers=orthofoto
&basemap:styles=default
```

3.2.11 Redlining

This feature is available and can be used with *GetMap* and *GetPrint* requests.

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.

The *GetMap* request is in the format:

```
http://localhost/qgisserver?
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></
```

(suite sur la page suivante)

(suite de la page précédente)

```

↪LineSymbolizer></Rule></FeatureTypeStyle></UserStyle></StyledLayerDescriptor>
&HIGHLIGHT_LABELSTRING=Write label here
&HIGHLIGHT_LABELSIZE=16
&HIGHLIGHT_LABELCOLOR=%23000000
&HIGHLIGHT_LABELBUFFERCOLOR=%23FFFFFF
&HIGHLIGHT_LABELBUFFERSIZE=1.5

```

The *GetPrint* equivalent is in the format (note that mapX: parameter is added to tell which map has redlining):

```

http://localhost/qgisserver?
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=%23FFFFFF
&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:



Figure3.22: Réponse du serveur à une demande GetMap avec paramètres de mise en évidence

You can see there are several parameters in this request to control the redlining feature. The full list includes:

- **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_LABELBUFFERCOLOR:** ce paramètre contrôle la couleur du tampon d'étiquette.
- **HIGHLIGHT_LABELBUFFERSIZE:** ce paramètre contrôle la taille du tampon d'étiquette.
- **HIGHLIGHT_LABELCOLOR:** ce paramètre contrôle la couleur de l'étiquette.
- **HIGHLIGHT_LABEL_DISTANCE:** controls the distance between feature (e.g. point or line) and the label in mm
- **HIGHLIGHT_LABELFONT :** Ce paramètre contrôle la police de l'étiquette (par exemple Arial)
- **HIGHLIGHT_LABEL_HORIZONTAL_ALIGNMENT:** places the label horizontally on a point using the specified alignment (e.g. "left", "center", "right")
- **HIGHLIGHT_LABEL_ROTATION:** controls the label rotation in degrees
- **HIGHLIGHT_LABELSIZE:** ce paramètre contrôle la taille de l'étiquette.
- **HIGHLIGHT_LABELSTRING:** vous pouvez transmettre votre texte d'étiquetage à ce paramètre.
- **HIGHLIGHT_LABEL_VERTICAL_ALIGNMENT:** places the label vertically on a point using the specified alignment (e.g. "top", "half", "bottom")

- **HIGHLIGHT_SYMBOL**: cela contrôle la manière dont la géométrie est définie et vous pouvez modifier la largeur, la couleur et l'opacité du trait.

3.3 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
<i>GetCapabilities</i>	Renvoie des métadonnées XML avec des informations sur le serveur
<i>GetFeature</i>	Renvoie une sélection d'entité
<i>DescribeFeatureType</i>	Renvoie une description des types d'entités et des propriétés
<i>Transaction</i>	Permet aux entités d'être insérées, mises à jour ou supprimées

3.3.1 GetCapabilities

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

Paramètre	Requis	Description
<i>SERVICE</i>	Oui	Nom du service (WFS)
<i>REQUEST</i>	Oui	Nom de la requête (GetCapabilities)
<i>VERSION</i>	Non	Version du service

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

Paramètre	Requis	Description
<i>MAP</i>	Oui	Fichier du projet QGIS

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
&...
```

3.3.2 GetFeature

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

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

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

Paramètre	Requis	Description
<i>MAP</i>	Oui	Fichier du projet QGIS
<i>STARTINDEX</i>	Non	Pagination
<i>GEOMETRYNAME</i>	Non	Type de géométrie à renvoyer
<i>EXP_FILTER</i>	Non	Expression de filtrage

TYPENAME

Ce paramètre permet de spécifier les noms des couches et est obligatoire si *FEATUREID* n'est pas défini.

Exemple d'URL :

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

FEATUREID

Ce paramètre permet de spécifier l'ID d'une entité spécifique et est formé avec *typename.fid,typename.fid,...*.

Exemple d'URL :

```
http://localhost/qgisserver?
SERVICE=WFS
&REQUEST=GetFeature
&FEATUREID=countries.0,places.1
```

réponse XML :

```
<wfs:FeatureCollection xmlns:wfs="http://www.opengis.net/wfs" xmlns:ogc="http://
↳www.opengis.net/ogc" xmlns:gml="http://www.opengis.net/gml" xmlns:ows="http://
↳www.opengis.net/ows" xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:qgs="http://
↳www.qgis.org/gml" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
↳xsi:schemaLocation="http://www.opengis.net/wfs http://schemas.opengis.net/wfs/1.
↳1.0/wfs.xsd http://www.qgis.org/gml http://192.168.1.15/qgisserver?SERVICE=WFS&
↳VERSION=1.1.0&REQUEST=DescribeFeatureType&TYPENAME=countries,places&
↳OUTPUTFORMAT=text/xml; subtype=gml/3.1.1">
  <gml:boundedBy>
    ...
  </gml:boundedBy>
  <gml:featureMember>
    <qgs:countries gml:id="countries.1">
      ...
    </qgs:countries>
  </gml:featureMember>
  <gml:featureMember>
    <qgs:places gml:id="places.1">
      ...
    </qgs:places>
  </gml:featureMember>
</wfs:FeatureCollection>
```

OUTPUTFORMAT

Ce paramètre peut être utilisé pour spécifier le format de la réponse. Si `VERSION`` est supérieur ou égal à `1.1.0`, GML3 est le format par défaut. Sinon, GML2 est utilisé.

Les valeurs disponibles sont :

- gml2
- text/xml; subtype=gml/2.1.2
- gml3
- text/xml; subtype=gml/3.1.1
- geojson
- application/vnd.geo+json,
- application/vnd.geo json
- application/geo+json
- application/geo json
- application/json

Exemple d'URL :

```
http://localhost/qgisserver?
SERVICE=WFS
&REQUEST=GetFeature
&FEATUREID=countries.0
&OUTPUTFORMAT=geojson
```

réponse GeoJSON :

```
{
  "type": "FeatureCollection",
  "bbox": [
    -180,
    -90,
    180,
    83.6236
  ],
  "features": [
    {
      "bbox": [
        -61.891113,
        16.989719,
        -61.666389,
        17.724998
      ],
      "geometry": {
        "coordinates": [
          "... "
        ],
        "type": "MultiPolygon"
      },
      "id": "countries.1",
      "properties": {
        "id": 1,
        "name": "Antigua and Barbuda"
      },
      "type": "Feature"
    }
  ]
}
```

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
&...
```

PROPERTYNAME

Ce paramètre peut être utilisé pour spécifier une propriété spécifique à retourner. Une propriété doit être associée à un TYPENAME ou à un FEATUREID:

Exemple d'URL valide :

```
http://localhost/qgisserver?  
SERVICE=WFS  
&REQUEST=GetFeature  
&PROPERTYNAME=name  
&TYPENAME=places
```

Au contraire, l'URL suivante renverra une exception :

```
http://localhost/qgisserver?  
SERVICE=WFS  
&REQUEST=GetFeature  
&PROPERTYNAME=name  
&TYPENAME=places,countries
```

```
<ServiceExceptionReport xmlns="http://www.opengis.net/ogc" version="1.2.0">  
  <ServiceException code="RequestNotWellFormed">There has to be a 1:1 mapping  
  ↪ between each element in a TYPENAME and the PROPERTYNAME list</ServiceException>  
</ServiceExceptionReport>
```

MAXFEATURES

Ce paramètre permet de limiter le nombre d'entités renvoyées par la requête.

Exemple d'URL :

```
http://localhost/qgisserver?  
SERVICE=WFS  
&REQUEST=GetFeature  
&TYPENAME=places  
&MAXFEATURES=1000
```

Note: Ce paramètre peut être utile pour améliorer les performances lorsque les couches vectorielles sous-jacentes sont lourdes.

SRSNAME

Ce paramètre permet d'indiquer le Système de Référence Spatial de sortie de la réponse ainsi que le SCR de la BBOX et doit être formaté comme EPSG:XXXX.

Exemple d'URL :

```
http://localhost/qgisserver?  
SERVICE=WFS  
&REQUEST=GetFeature  
&TYPENAME=places  
&SRSNAME=EPSG:32620
```


FILTER

This parameter allows to filter the response with the **Filter Encoding** language defined by the [OGC Filter Encoding standard](#). For example:

```
http://localhost/qgisserver?
SERVICE=WFS&
REQUEST=GetFeature&
TYPENAME=places&
FILTER=<Filter><PropertyIsEqualTo><PropertyName>name</PropertyName><Literal>Paris</
Literal></PropertyIsEqualTo></Filter>
```

In case of multiple tynenames, filters have to be enclosed in parentheses:

```
http://localhost/qgisserver?
SERVICE=WFS
&REQUEST=GetFeature
&TYPENAME=places,countries
&FILTER=(<Filter><PropertyIsEqualTo><PropertyName>name</PropertyName><Literal>Paris
</Literal></PropertyIsEqualTo></Filter>)<Filter><PropertyIsEqualTo>
<PropertyName>name</PropertyName><Literal>France</Literal></PropertyIsEqualTo></
Filter>)
```

Filtre les entités qui intersectent un polygone :

```
http://localhost/qgisserver?
SERVICE=WFS
&REQUEST=GetFeature
&VERSION=1.1.0
&TYPENAME=places
&FILTER=<Filter xmlns="http://www.opengis.net/ogc">
  <Intersects>
    <PropertyName>geometry</PropertyName>
    <Polygon xmlns="http://www.opengis.net/gml" srsName="EPSG:4326">
      <exterior>
        <LinearRing>
          <posList>
            -0.6389 42.5922
            10.2683 51.9106
            14.5196 41.0320
            -0.6389 42.5922
          </posList>
        </LinearRing>
      </exterior>
    </Polygon>
  </Intersects>
</Filter>
```

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 SRSNAME peut spécifier le SCR de l'étendue. S'il n'est pas spécifié, le SCR de la couche est utilisé.

Exemple d'URL :

```
http://localhost/qgisserver?
SERVICE=WFS
&REQUEST=GetFeature
&TYPENAME=places
&BBOX=-11.84,42.53,8.46,50.98
```

Le paramètre `FEATUREID` ne peut pas être utilisé avec le paramètre `BBOX`. Toute tentative entraînera une exception :

```
<ServiceExceptionReport xmlns="http://www.opengis.net/ogc" version="1.2.0">
  <ServiceException code="RequestNotWellFormed">FEATUREID FILTER and BBOX
  ↳ parameters are mutually exclusive</ServiceException>
</ServiceExceptionReport>
```

SORTBY

Ce paramètre permet de trier les entités résultantes selon les valeurs des propriétés et doit être formé comme `propertyname SORTRULE`.

Valeurs disponibles pour `SORTRULE` en cas de tri descendant :

- D
- +D
- DESC
- +DESC

Valeurs disponibles pour `SORTRULE` en cas de tri ascendant :

- A
- +A
- ASC
- +ASC

Exemple d'URL :

```
http://localhost/qgisserver?
SERVICE=WFS
&REQUEST=GetFeature
&TYPENAME=places
&PROPERTYNAME=name
&MAXFEATURES=3
&SORTBY=name DESC
```

Le résultat correspondant :

```
<wfs:FeatureCollection xmlns:wfs="http://www.opengis.net/wfs" xmlns:ogc="http://
↳ www.opengis.net/ogc" xmlns:gml="http://www.opengis.net/gml" xmlns:ows="http://
↳ www.opengis.net/ows" xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:qgs="http://
↳ www.qgis.org/gml" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
↳ xsi:schemaLocation="http://www.opengis.net/wfs http://schemas.opengis.net/wfs/1.
↳ 1.0/wfs.xsd http://www.qgis.org/gml http://192.168.1.15/qgisserver?SERVICE=WFS&
↳ VERSION=1.1.0&REQUEST=DescribeFeatureType&TYPENAME=places&OUTPUTFORMAT=text/xml;
↳ subtype%3Dgml/3.1.1">
  <gml:boundedBy>
    ...
  </gml:boundedBy>
  <gml:featureMember>
    <qgs:places gml:id="places.90">
      <qgs:name>Zagreb</qgs:name>
    </qgs:places>
  </gml:featureMember>
  <gml:featureMember>
    <qgs:places gml:id="places.113">
      <qgs:name>Yerevan</qgs:name>
    </qgs:places>
```

(suite sur la page suivante)

(suite de la page précédente)

```

</gml:featureMember>
<gml:featureMember>
  <qgs:places gml:id="places.111">
    <qgs:name>Yaounde</qgs:name>
  </qgs:places>
</gml:featureMember>
</wfs:FeatureCollection>

```

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 dans l'ensemble des résultats et, en combinaison avec MAXFEATURES, il offre la possibilité de paginer les résultats.

Exemple d'URL :

```

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

```

EXP_FILTER

Ce paramètre permet de filtrer les réponses à l'aide d'expressions QGIS. Le caractère ; est utilisé pour séparer les filtres lorsqu'il y en a plusieurs.

Exemple d'URL :

```

http://localhost/qgisserver?
SERVICE=WFS
&REQUEST=GetFeature
&TYPENAME=places,countries
&EXP_FILTER="name"='Paris';"name"='France'

```

3.3.3 DescribeFeatureType

Paramètres standard pour la requête **DescribeFeatureType** selon les spécifications OGC WFS 1.0.0 et 1.1.0 :

Paramètre	Requis	Description
<i>SERVICE</i>	Oui	Nom du service (WFS)
<i>REQUEST</i>	Oui	Nom de la requête (DescribeFeatureType)
<i>VERSION</i>	Non	Version du service
<i>OUTPUTFORMAT</i>	Non	Format de la réponse
<i>TYPENAME</i>	Non	Nom des couches

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

Paramètre	Requis	Description
<i>MAP</i>	Oui	Fichier du projet QGIS

Exemple d'URL :

```
http://localhost/qgisserver?
SERVICE=WFS
&VERSION=1.1.0
&REQUEST=DescribeFeatureType
&TYPENAME=countries
```

Réponse en sortie

```
<schema xmlns:ogc="http://www.opengis.net/ogc" xmlns:xsd="http://www.w3.org/2001/
↳XMLSchema" xmlns="http://www.w3.org/2001/XMLSchema" xmlns:ggs="http://www.qgis.
↳org/gml" xmlns:gml="http://www.opengis.net/gml" targetNamespace="http://www.qgis.
↳org/gml" version="1.0" elementFormDefault="qualified">
  <import schemaLocation="http://schemas.opengis.net/gml/3.1.1/base/gml.xsd"
↳namespace="http://www.opengis.net/gml"/>
  <element type="ggs:countriesType" substitutionGroup="gml:_Feature" name=
↳"countries"/>
  <complexType name="countriesType">
    <complexContent>
      <extension base="gml:AbstractFeatureType">
        <sequence>
          <element minOccurs="0" type="gml:MultiPolygonPropertyType" maxOccurs="1"
↳name="geometry"/>
          <element type="long" name="id"/>
          <element nillable="true" type="string" name="name"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
</schema>
```

3.3.4 Transaction

Cette requête permet de mettre à jour, supprimer ou ajouter une ou plusieurs entités grâce à un document XML. L'action *delete* peut être réalisée par une requête POST ainsi qu'avec le paramètre *OPERATION* tandis que les opérations *add* et *update* ne peuvent être réalisées que par une requête POST.

Paramètres standard pour la requête **Transaction** selon les spécifications OGC WFS 1.0.0 et 1.1.0:

Paramètre	Requis	Description
<i>SERVICE</i>	Oui	Nom du service (WFS)
<i>REQUEST</i>	Oui	nom de la requête (Transaction)
<i>VERSION</i>	Non	Version du service
<i>FILTER</i>	Non	Encodage du filtre OGC
<i>BBOX</i>	Non	Emprise de la carte
<i>FEATUREID</i>	Non	Filtre les entités par leur identifiant
<i>TYPENAME</i>	Non	Nom des couches

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

Paramètre	Requis	Description
<i>MAP</i>	Oui	Fichier du projet QGIS
<i>OPERATION</i>	Non	Spécifie l'opération
<i>EXP_FILTER</i>	Non	Expression de filtrage

OPERATION

Ce paramètre permet de supprimer une entité sans utiliser une requête POST avec un document XML dédié.

Exemple d'URL :

```
http://localhost/qgisserver?
SERVICE=WFS
&VERSION=1.1.0
&REQUEST=Transaction
&OPERATION=DELETE
&FEATUREID=24
```

Note: Les paramètres *FEATUREID*, *BBOX* et *FILTER* sont mutuellement exclusifs et hiérarchisés dans cet ordre.

Ajout d'entités

Exemple de requête POST

```
wget --post-file=add.xml "http://localhost/qgisserver?SERVICE=WFS&
REQUEST=Transaction"
```

avec le document *add.xml* :

```
<?xml version="1.0" encoding="UTF-8"?>
<wfs:Transaction service="WFS" version="1.0.0" xmlns:wfs="http://www.opengis.net/
wfs" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:ogc="http://www.
opengis.net/ogc" xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.
opengis.net/wfs http://schemas.opengis.net/wfs/1.0.0/WFS-capabilities.xsd">
```

(suite sur la page suivante)

(suite de la page précédente)

```
<?xml version="1.0" encoding="UTF-8"?>
<xmlns:gml="http://www.opengis.net/gml" xmlns:ows="http://www.opengis.net/ows">
  <wfs:Insert idgen="GenerateNew">
    <qgs:places>
      <qgs:geometry>
        <gml:Point srsDimension="2" srsName="http://www.opengis.net/def/crs/EPSG/0/
4326">
          <gml:coordinates decimal="." cs="," ts=" ">-4.6167,48.3833</
gml:coordinates>
        </gml:Point>
      </qgs:geometry>
      <qgs:name>Locmaria-Plouzané</qgs:name>
    </qgs:places>
  </wfs:Insert>
</wfs:Transaction>
```

Mise à jour d'entités

Exemple de requête POST

```
wget --post-file=update.xml "http://localhost/qgisserver?SERVICE=WFS&
REQUEST=Transaction"
```

avec le document *update.xml* :

```
<?xml version="1.0" encoding="UTF-8"?>
<wfs:Transaction service="WFS" version="1.0.0" xmlns:wfs="http://www.opengis.net/
wfs" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:ogc="http://www.
opengis.net/ogc" xmlns="http://www.opengis.net/wfs" updateSequence="0">
  <xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.
opengis.net/wfs http://schemas.opengis.net/wfs/1.0.0/WFS-capabilities.xsd">
  <xmlns:gml="http://www.opengis.net/gml" xmlns:ows="http://www.opengis.net/ows">
    <wfs:Update typeName="places">
      <wfs:Property>
        <wfs:Name>name</wfs:Name>
        <wfs:Value>Lutece</wfs:Value>
      </wfs:Property>
      <ogc:Filter>
        <ogc:FeatureId fid="24"/>
      </ogc:Filter>
    </wfs:Update>
  </wfs:Transaction>
```

Suppression d'entités

Exemple de requête POST

```
wget --post-file=delete.xml "http://localhost/qgisserver?SERVICE=WFS&
REQUEST=Transaction"
```

avec le document *delete.xml* :

```
<?xml version="1.0" encoding="UTF-8"?>
<wfs:Transaction service="WFS" version="1.0.0" xmlns:wfs="http://www.opengis.net/
wfs" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:ogc="http://www.
opengis.net/ogc" xmlns="http://www.opengis.net/wfs" updateSequence="0">
  <xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.
opengis.net/wfs http://schemas.opengis.net/wfs/1.0.0/WFS-capabilities.xsd">
  <xmlns:gml="http://www.opengis.net/gml" xmlns:ows="http://www.opengis.net/ows">
```

(suite sur la page suivante)

(suite de la page précédente)

```
<wfs:Delete typeName="places">
  <ogc:Filter>
    <ogc:FeatureId fid="24"/>
  </ogc:Filter>
</wfs:Delete>
</wfs:Transaction>
```

3.4 Web Coverage Service (WCS)

The **1.0.0** and **1.1.1** WCS standards implemented in QGIS Server provide a HTTP interface to access raster data, referred to as *coverage*, coming from a QGIS project.

Spécifications :

- WCS 1.0.0
- WCS 1.1.1

Requêtes standard fournies par QGIS Server:

Requête	Description
<i>GetCapabilities</i>	Renvoie des métadonnées XML avec des informations sur le serveur
<i>DescribeCoverage</i>	Retrieves XML document about additional information about coverages
<i>GetCoverage</i>	Retrieves coverage

3.4.1 GetCapabilities

Standard parameters for the **GetCapabilities** request according to the OGC WCS 1.1.1 specifications:

Paramètre	Requis	Description
<i>SERVICE</i>	Oui	Name of the service (WCS)
<i>REQUEST</i>	Oui	Nom de la requête (GetCapabilities)
<i>VERSION</i>	Non	Version du service

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

Paramètre	Requis	Description
<i>MAP</i>	Oui	Fichier du projet QGIS

Exemple d'URL :

```
http://localhost/qgisserver?
SERVICE=WCS
&VERSION=1.1.1
&REQUEST=GetCapabilities
```

XML document example when a single raster layer (named T20QPD_20171123T144719_TCI) is published in the QGIS project for the WCS service:

```

<WCS_Capabilities xmlns="http://www.opengis.net/wcs" xmlns:xlink="http://www.w3.
↪org/1999/xlink" xmlns:gml="http://www.opengis.net/gml" xmlns:xsi="http://www.w3.
↪org/2001/XMLSchema-instance" version="1.0.0" updateSequence="0"
↪xsi:schemaLocation="http://www.opengis.net/wcs http://schemas.opengis.net/wcs/1.
↪0.0/wcsCapabilities.xsd">
  <Service>
    ...
  </Service>
  <Capability>
    ...
  </Capability>
  <ContentMetadata>
    <CoverageOfferingBrief>
      <name>T20QPD_20171123T144719_TCI</name>
      <label>T20QPD_20171123T144719_TCI</label>
      <lonLatEnvelope srsName="urn:ogc:def:crs:OGC:1.3:CRS84">
        <gml:pos>-61.585973 16.331189</gml:pos>
        <gml:pos>-61.52537 16.400376</gml:pos>
      </lonLatEnvelope>
    </CoverageOfferingBrief>
  </ContentMetadata>
</WCS_Capabilities>

```

VERSION

This parameter allows to specify the version of the service to use. Currently, the version values is not internally used and always fallback to 1.1.1.

3.4.2 DescribeCoverage

This request allows to retrieve additional information about coverages like the format of the underlying datasource, the number of bands, ... Standard parameters for the **DescribeCoverage** request according to the OGC WCS 1.1.1 specifications:

Paramètre	Requis	Description
<i>SERVICE</i>	Oui	Name of the service (WCS)
<i>REQUEST</i>	Oui	Name of the request (DescribeCoverage)
<i>VERSION</i>	Non	Version du service
<i>COVERAGE</i>	Non	Specify coverage layers (WCS 1.0.0)
<i>IDENTIFIER</i>	Non	Specify coverage layers (WCS 1.1.1)

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

Paramètre	Requis	Description
<i>MAP</i>	Oui	Fichier du projet QGIS

The XML document for a 3 bands GeoTIFF raster layer looks like:

```

<CoverageDescription xmlns:xlink="http://www.w3.org/1999/xlink" xmlns="http://www.
↪opengis.net/wcs" xmlns:gml="http://www.opengis.net/gml" xmlns:xsi="http://www.w3.
↪org/2001/XMLSchema-instance" updateSequence="0" version="1.0.0"
↪xsi:schemaLocation="http://www.opengis.net/wcs http://schemas.opengis.net/wcs/1.
↪0.0/describeCoverage.xsd">
  <CoverageOffering>

```

(suite sur la page suivante)

(suite de la page précédente)

```

<name>T20QPD_20171123T144719_TCI</name>
<label>T20QPD_20171123T144719_TCI</label>
<lonLatEnvelope srsName="urn:ogc:def:crs:OGC:1.3:CRS84">
  ...
</lonLatEnvelope>
<domainSet>
  ...
</domainSet>
<rangeSet>
  <RangeSet>
    <name>Bands</name>
    <label>Bands</label>
    <axisDescription>
      <AxisDescription>
        <name>bands</name>
        <label>bands</label>
        <values>
          <singleValue>1</singleValue>
          <singleValue>2</singleValue>
          <singleValue>3</singleValue>
        </values>
      </AxisDescription>
    </axisDescription>
  </RangeSet>
</rangeSet>
<supportedCRSs>
  ...
</supportedCRSs>
<supportedFormats nativeFormat="raw binary">
  <formats>GeoTIFF</formats>
</supportedFormats>
</CoverageOffering>
</CoverageDescription>

```

COVERAGE

This parameter, defined in WCS 1.0.0, allows to specify the layers to query for additional information. Names have to be separated by a comma.

In addition, QGIS Server introduced an option to select layers by its short name. The short name of a layer may be configured through *Properties* ► *Metadata* in layer menu. If the short name is defined, then it's used by default instead of the layer's name:

```

http://localhost/qgisserver?
SERVICE=WCS
&REQUEST=DescribeCoverage
&COVERAGE=mylayer1name,mylayer2shortname

```

Note: COVERAGE is mandatory if IDENTIFIER is not set.

IDENTIFIER

This parameter replaces the *COVERAGE* parameter in WCS 1.1.1. But QGIS Server does not filter according to the *VERSION* parameter so *IDENTIFIER* and *COVERAGE* have the same effect.

Note: *IDENTIFIER* is mandatory if *COVERAGE* is not set. If both *IDENTIFIER* and *COVERAGE* parameters are defined, *COVERAGE* is always used in priority.

3.4.3 GetCoverage

This request allows to retrieve the coverage according to specific constraints like the extent or the CRS. Standard parameters for the **DescribeCoverage** request according to the OGC WCS 1.1.1 specifications:

Paramètre	Requis	Description
<i>SERVICE</i>	Oui	Name of the service (WCS)
<i>REQUEST</i>	Oui	Name of the request (GetCoverage)
<i>VERSION</i>	Non	Version du service
<i>COVERAGE</i>	Non	Specify coverage layers (WCS 1.0.0)
<i>IDENTIFIER</i>	Non	Specify coverage layers (WCS 1.1.1)
<i>WIDTH</i>	Oui	Width of the response in pixels
<i>HEIGHT</i>	Oui	Height of the response in pixels
<i>BBOX</i>	Oui	Map extent in CRS units
<i>CRS</i>	Oui	Coordinate reference system of the extent
<i>RESPONSE_CRS</i>	Non	Coordinate reference system of the response

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

Paramètre	Requis	Description
<i>MAP</i>	Oui	Fichier du projet QGIS

BBOX

This parameter allows to specify the map extent in the units of the current CRS. Coordinates have to be separated by a comma. The *BBOX* parameter is formed like *minx,miny,maxx,maxy*.

Exemple d'URL :

```
http://localhost/qgisserver?
SERVICE=WCS
&REQUEST=GetCoverage
&IDENTIFIER=T20QPD_20171123T144719_TCI
&BBOX=647533,1805950,660987,1813940
&CRS=EPSG:32620
```

CRS

This parameter allows to indicate the Spatial Reference System of the BBOX parameter and has to be formed like EPSG:XXXX.

RESPONSE_CRS

This parameter allows to indicate the output response Spatial Reference System and has to be formed like EPSG:XXXX. The CRS of the corresponding coverage layer is used by default.

WIDTH

This parameter allows to specify the width in pixels of the output image. The resolution of the response image depends on this value.

HEIGHT

This parameter allows to specify the height in pixels of the output image. The resolution of the response image depends on this value.

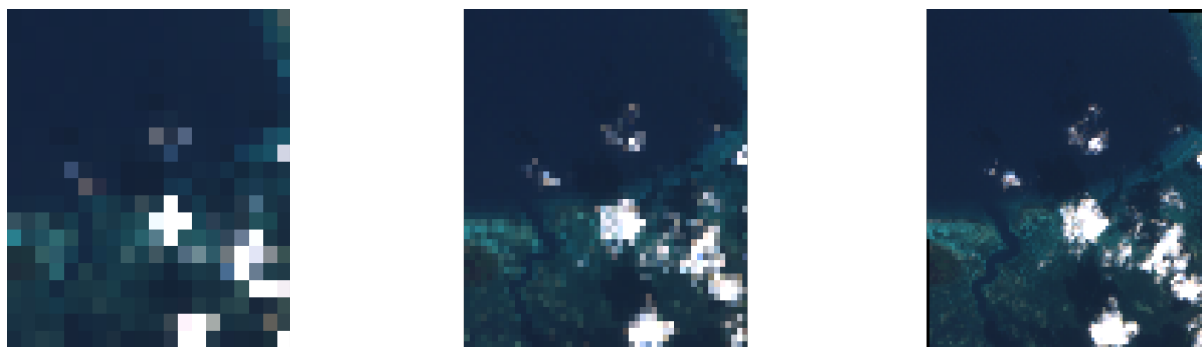


Figure3.23: From left to right: WIDTH=20&HEIGHT=20, WIDTH=50&HEIGHT=50, WIDTH=100&HEIGHT=100

3.5 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
<i>GetCapabilities</i>	Renvoie des métadonnées XML avec des informations sur le serveur
<i>GetTile</i>	Renvoie une tuile
<i>GetFeatureInfo</i>	Récupère les données (géométrie et valeurs) pour un emplacement de pixel

3.5.1 GetCapabilities

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

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

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

Paramètre	Requis	Description
<i>MAP</i>	Oui	Fichier du projet QGIS

Exemple d'URL :

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

3.5.2 GetTile

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

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

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

Paramètre	Requis	Description
<i>MAP</i>	Oui	Fichier du projet QGIS

Exemple d'URL :

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

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.

3.5.3 GetFeatureInfo

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

Paramètre	Requis	Description
<i>SERVICE</i>	Oui	Nom du service (WMTS)
<i>REQUEST</i>	Oui	nom de la requete (GetFeatureInfo)
<i>LAYER</i>	Oui	Identifiant de la couche
<i>INFOFORMAT</i>	Non	Format de sortie
<i>I</i>	Non	Coordonnée X d'un pixel
<i>J</i>	Non	Coordonnée Y d'un pixel
<i>TILEMATRIXSET</i>	Oui	Nom de la pyramide
<i>TILEMATRIX</i>	Maillage	
<i>TILEROW</i>	Oui	Coordonnées de ligne dans le maillage
<i>TILECOL</i>	Oui	Coordonnées de colonne dans le maillage

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

Paramètre	Requis	Description
<i>MAP</i>	Oui	Fichier du projet QGIS

Exemple d'URL :

```
http://localhost/?
SERVICE=WMTS
&REQUEST=GetFeatureInfo
&MAP=/home/qgis/projects/world.qgs
&LAYER=mylayer
&INFOFORMAT=image/html
&I=10
&J=5
```

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.

3.6 OGC API Features

OGC API Features (OAPIF) est la première implémentation de la nouvelle génération de protocoles OGC. Elle est décrite par le document [OGC API - Features - Part 1: Core](#).

The API can be reached on typical installations via `http://localhost/qgisserver/wfs3`.

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

- OAPIF est basé sur l'API [REST](#)
- OAPIF doit suivre les spécifications [OPENAPI](#)
- OAPIF prend en charge plusieurs formats de sortie, mais il n'en dicte aucun (seuls GeoJSON et HTML sont actuellement disponibles avec OAPIF sous QGIS) et il utilise la [négociation de contenu](#) pour déterminer le format qui doit être servi au client
- JSON et HTML sont des citoyens de première classe dans OAPIF
- OAPIF est auto-documenté (via le point de terminaison `/api`)
- OAPIF est entièrement navigable (via des liens) et explorable

Important: Alors que l'implémentation OGC API Features 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.

3.6.1 Représentation des ressources

L'implémentation de l'OGC API Features dans QGIS Server 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**

3.6.2 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 API	/conformance /api	Informations sur la conformité du service aux normes 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.)
Fonctionnali	/collections/ {collectionId}/items	Liste des entités fournies par la collection
Entité	/collections/ {collectionId}/items/ {featureId}	Informations sur une seule entité

Comme pour le WFS-T (Transactional Web Feature Service), il est possible d'ajouter, de mettre à jour et de supprimer des entités (CRUD). Les requêtes respectives sont décrites sur « `/api` ».

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`).

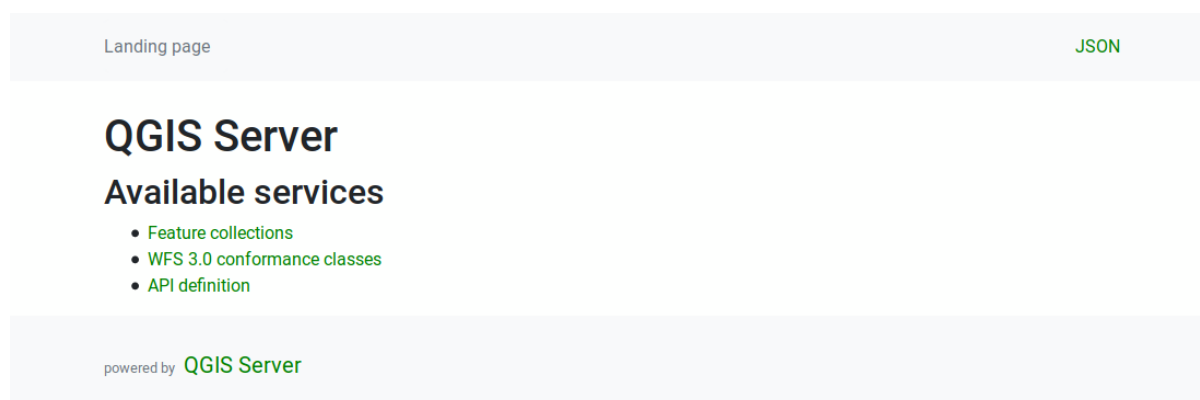


Figure3.24: Page de destination du serveur OAPIF

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/`.

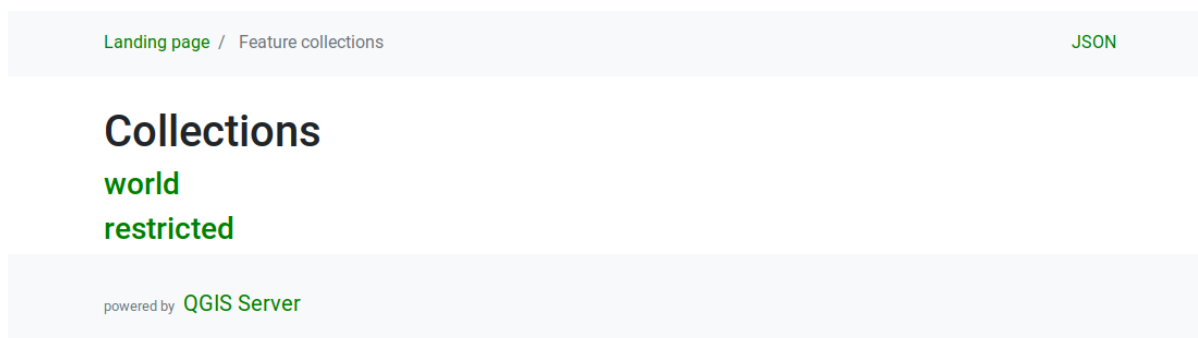


Figure3.25: Page de liste des collections du serveur OAPIF

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.

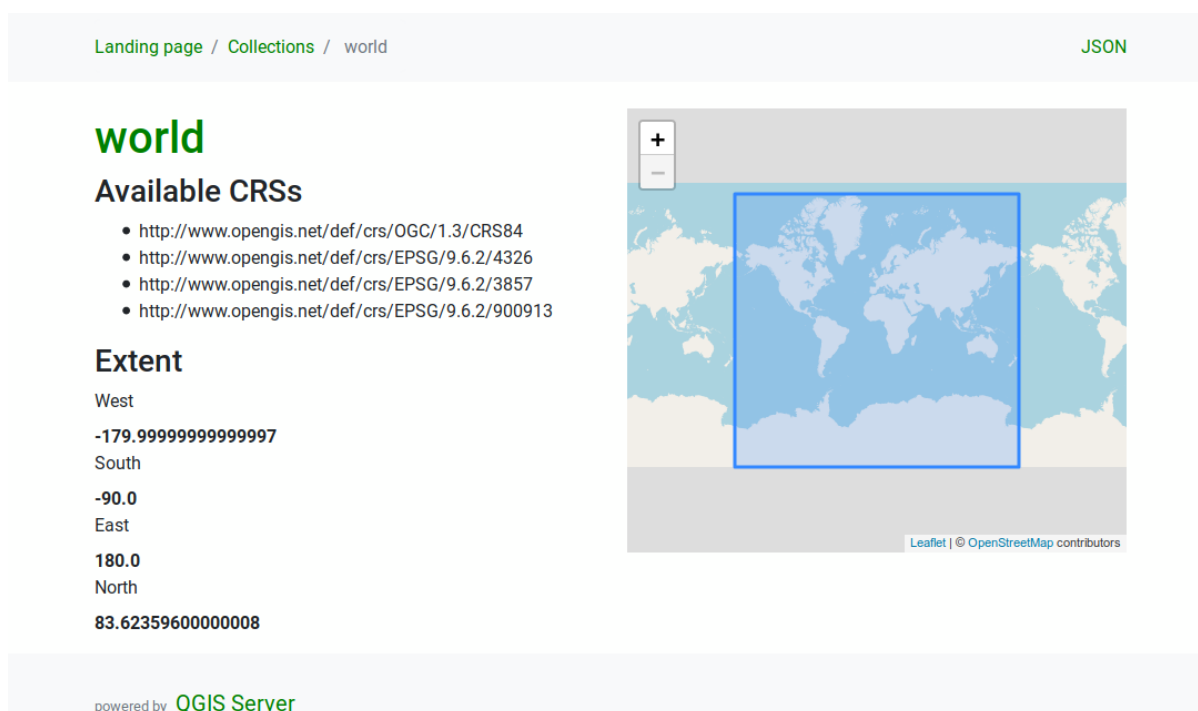


Figure3.26: Page de détail de la collection du serveur OAPIF

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.

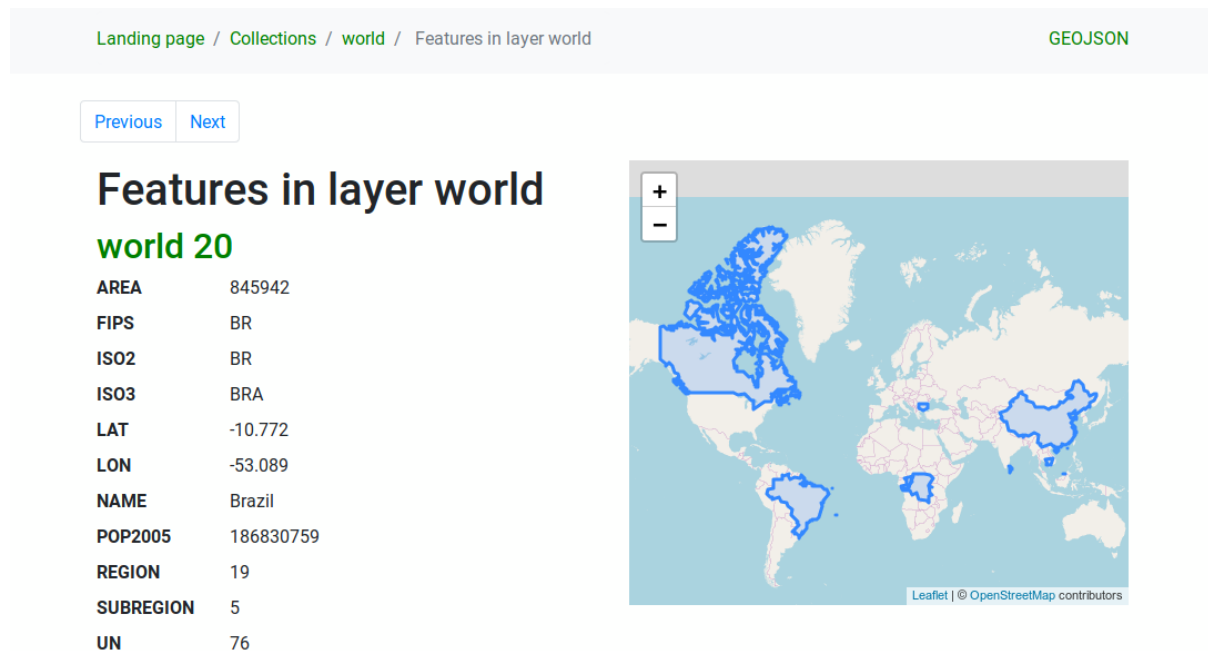


Figure3.27: Page de liste des entités du serveur OAPIF

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.



Figure3.28: Page de détail des entités du serveur OAPIF

3.6.3 Pagination

La pagination d'une longue liste d'entités est implémentée dans l'API OGC via des liens précédent (*next*) et suivant (*prev*). QGIS Server construit ces liens en ajoutant *limit* et *offset* 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*).

3.6.4 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:

Renvoie 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
```

Renvoie 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
```

Renvoie 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 « valeur »

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

3.6.5 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
```

3.6.6 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 `properties`.

Exemple d'URL :

renvoie uniquement l'attribut `name`

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

3.6.7 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` »
- `nl2br(text)` : Renvoie le texte en entrée avec les nouvelles lignes remplacées par les tags « `<br>` »
- `starts_with(string, prefix)` : renvoie vrai si un texte commence par le préfixe indiqué, sinon faux

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

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

The QGIS Server Catalog is a simple catalog that shows the list of QGIS projects served by the QGIS Server. It provides a user-friendly fully browsable website with basic mapping capabilities to quickly browse the datasets exposed through those QGIS projects. The catalog can be configured using the environment variables starting with `QGIS_SERVER_LANDING_PAGE` (see *Variables d'environnement*).

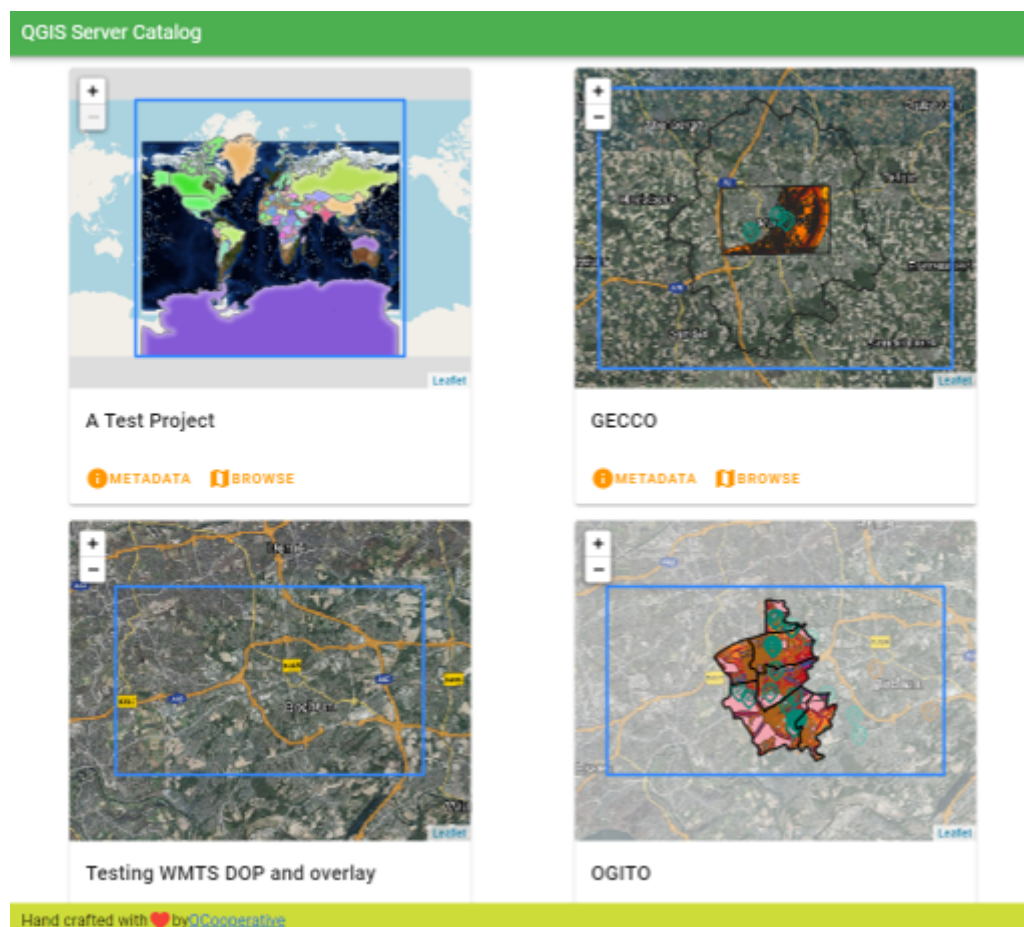


Figure4.1: Page listant les projets de QGIS Server Catalog

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

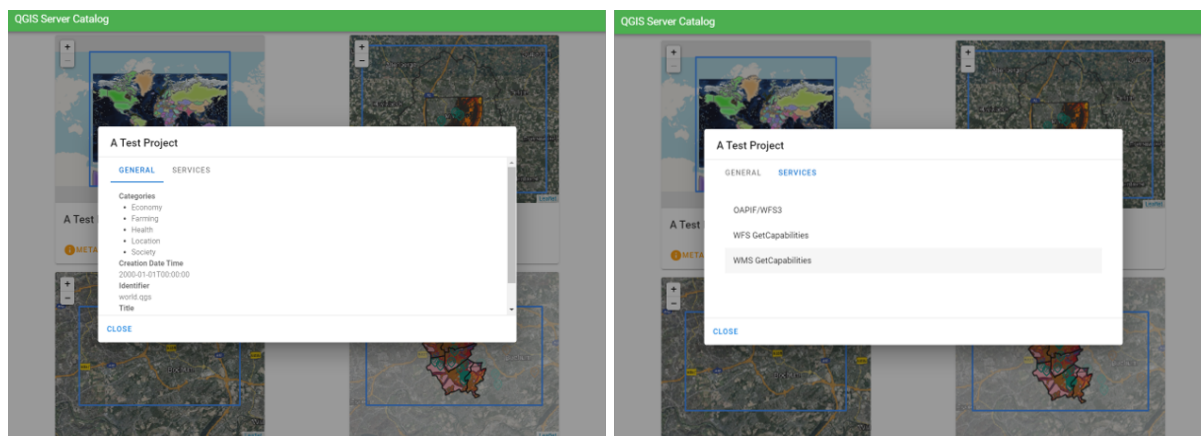


Figure4.2: Catalogue Serveur, métadonnées associées à un projet ainsi que les services qu'il propose.

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

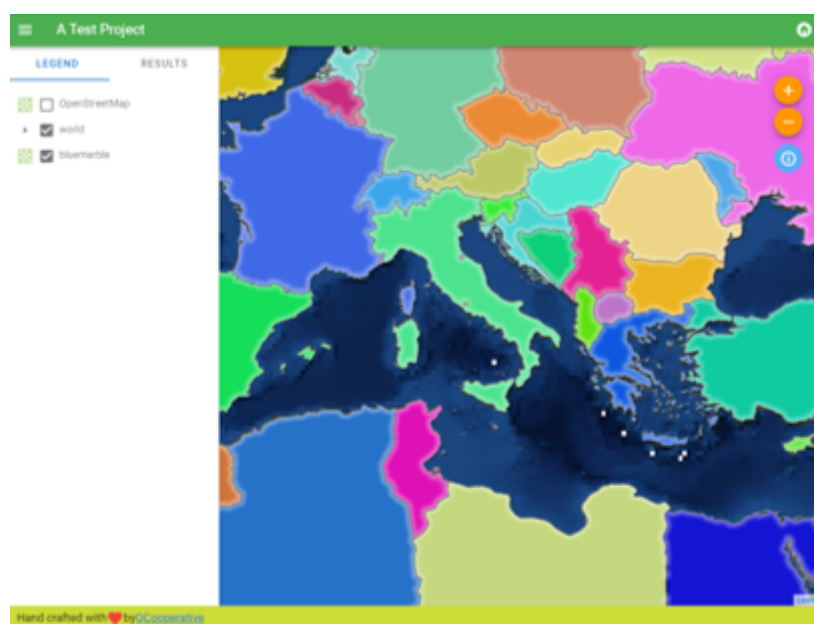


Figure4.3: Parcourir un jeu de données fourni par un projet dans le Catalogue Serveur

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

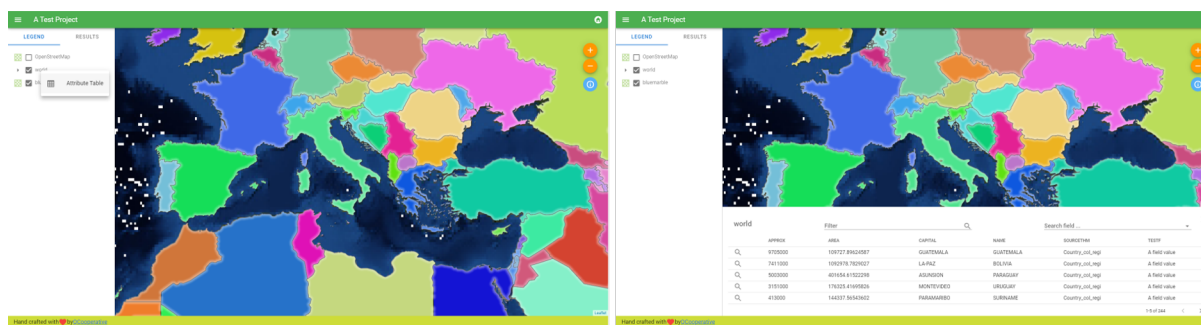


Figure4.4: Table d'attributs associée à la couche

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

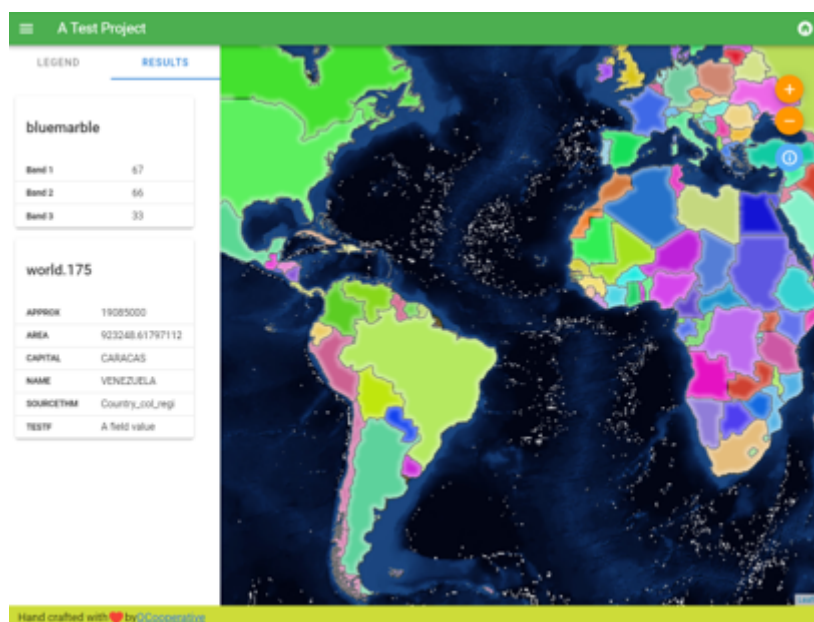


Figure4.5: Consulter les information d'une entité de la carte

5.1 Liste d'extensions

Les extensions peuvent également être installées sur QGIS Server.

Certaines extensions, destinées à un usage sur serveur, sont disponibles sur le [dépôt officiel de QGIS](#).

N'installez que les extensions dont vous avez besoin. Sur QGIS Server, les extensions sont comme des hooks, et peuvent donc modifier les éléments en entrée et sortie de QGIS Server. Elles peuvent produire des résultats non escomptés si vous ne savez pas comment fonctionne l'extension. Veuillez vous référer à leur documentation ou à celle de l'application qui nécessite des extensions de QGIS Server afin d'identifier quelle extension pourrait vous être utile.

5.2 Emplacement des extensions

Par défaut, sur les systèmes basés sur Debian, QGIS Server va rechercher les plugins situés dans `/usr/lib/qgis/plugins`. La valeur par défaut est affichée au démarrage de QGIS Server, dans les logs. Il est possible de définir un chemin personnalisé en définissant la variable d'environnement `QGIS_PLUGINPATH` dans la configuration du serveur web.

5.3 Installation

5.3.1 Manuellement depuis un fichier ZIP

Par exemple, pour installer le plugin **HelloWorld** pour tester le serveur, à l'aide un dossier spécifique, vous devez d'abord créer un répertoire pour accueillir les plugins du serveur. Ceci sera spécifié dans la configuration de l'hôte virtuel et passé au serveur via 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
```

Avertissement: D'après sa [description](#), l'extension HelloServer est à des fins de développement et de démonstration. Ne la conservez pas en production si vous n'en avez pas besoin.

5.3.2 Avec un outil en ligne de commandes

Si vous devez régulièrement installer et mettre à jour des extensions à partir du dépôt de QGIS, vous pourriez utiliser [QGIS-Plugin-Manager](#) qui est un outil d'aide à la gestion d'extensions en ligne de commande.

L'installation se fait via pip. Installer dans un environnement virtuel est une bonne pratique mais elle n'est pas obligatoire :

```
pip3 install qgis-plugin-manager
```

Pour mettre à niveau l'outil :

```
pip3 install --upgrade qgis-plugin-manager
```

Ensuite, vous pouvez utiliser l'exécutable `qgis-plugin-manager` depuis la ligne de commande :

```
cd /var/www/qgis-server/plugins
qgis-plugin-manager list
```

QGIS server version 3.19.0
List all plugins in /var/www/qgis-server/plugins

```

-----
| Folder          | Name                  | Version | Experimental |  |
|-----|-----|-----|-----|-----|
| QGIS min        | QGIS max             | Author  | Action [?]   |
|-----|-----|-----|-----|-----|
| wfsOutputExtension | wfsOutputExtension | 1.6.2   |              | 3.
| 0               | 3Liz                 |         |              |
| qgis_server_render_geojson | GeoJson Renderer | v0.4    |              | 3.
| 4               | Matthias Kuhn (OPENGIS.ch) |         |              |
| DataPlotly      | Data Plotly          | 3.7.1   |              | 3.
| 4               | 3.98                 | Matteo Ghetta (Faunalia) | Upgrade to 3.8.1 |
|-----|-----|-----|-----|-----|

```

Nous vous suggérons de lire la [documentation complète](#) afin de mieux cerner comment installer et mettre à jour les extensions avec cet outil.

5.4 Configuration d'un serveur HTTP

5.4.1 Apache

Pour être en mesure d'utiliser une extension serveur, FastCGI doit savoir où chercher. 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
```

5.5 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
```

Configuration avancée

6.1 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*

6.2 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><QGIS_OPTIONS_PATH>/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

suite sur la page suivante

Table 6.1 – suite de la page précédente

Nom	Description	Default	Services
QGIS_PROJECT_FILE	Le fichier de projet .qgs ou .qgz, 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>). Note that you may also indicate a project stored in PostgreSQL, e.g. <code>postgresql://localhost:5432?sslmode=disable&dbname=mydb&ss</code> or inside a geopackage file, e.g. <code>geopackage:/path/to/geopackage/file.gpkg?projectName=myProjectName</code> .	""	Tous
QGIS_SERVER_ALLOWED_F	Comma separated list of strings that represent the allowed extra SQL tokens accepted as components of a feature filter.	""	WMS
QGIS_SERVER_API_RESOUR	Répertoire de base pour toutes les ressources statiques OGC API (telles que OAPIF/WFS3) (modèles HTML, CSS, JS, ...).	dépend du paquet	OAPIF/WFS3
QGIS_SERVER_APPLICATION	Name of the application to be used, for instance when connecting to a database to identify the QGIS server instance connected	Serveur QGIS3	Tous
QGIS_SERVER_API_WFS3_M	Maximum value for limit in a OAPIF/WFS3 features request.	10000	OAPIF/WFS3
QGIS_SERVER_CACHE_DIR	Définit 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_CAPABILITIES	The maximum number of project capabilities to cache.	40	Tous
QGIS_SERVER_DISABLE_GET_PRINT	C'est 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 GetPrint du WMS QGIS. Activez ce drapeau de projet QGIS pour ne pas charger les mises en page.	faux	WMS
QGIS_SERVER_FORCE_READ_ONLY	Force QGIS Server to open all layers in a read only mode	faux	Tous
QGIS_SERVER_IGNORE_BAD_LAYERS	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 1 ou vrai. Dans ce cas, les « mauvaises » couches seront simplement ignorées et le projet sera considéré comme valide et disponible.	faux	Tous

myproject

suite sur la page suivante

Table 6.1 – suite de la page précédente

Nom	Description	Default	Services
QGIS_SERVER_LANDING_PATH	Préfixe de la composante « chemin » de l'URL de base de la page de destination	« «	Tous
QGIS_SERVER_LANDING_PATH	Répertoires utilisés par le service des pages d'atterrissage pour trouver les projets .qgs et .qgz	« «	Tous
QGIS_SERVER_LANDING_PATH	Chaînes de connexion PostgreSQL utilisées par le service de la page d'atterrissage pour trouver des projets	« «	Tous
QGIS_SERVER_LOG_FILE	Indique le chemin et le nom de fichier du journal. Assurez-vous que le serveur dispose des permissions adaptées pour écrire dans le fichier. Le fichier est créé automatiquement lors de l'envoi de requêtes vers le serveur. S'il n'existe pas, vérifiez les permissions.	""	Tous
Avertissement: QGIS_SERVER_LOG_FILE est obsolète depuis QGIS 3.4, utilisez plutôt QGIS_SERVER_LOG_STDERR. La prise en charge de la journalisation des fichiers sera supprimée dans QGIS 4.0.			
QGIS_SERVER_LOG_LEVEL	indique le niveau de journalisation désiré. Les valeurs disponibles sont les suivantes : • 0 ou INFO (journalise toutes les requêtes), • 1 ou WARNING • 2 ou CRITICAL (journalise uniquement les erreurs critiques, adapté à un fonctionnement en production).	0	Tous

suite sur la page suivante

Table 6.1 – suite de la page précédente

Nom	Description	Default	Services
QGIS_SERVER_LOG_PROFIL	Ajouter des informations détaillées sur le profil dans les journaux, uniquement efficace lorsque QGIS_SERVER_LOG_LEVEL=0.	faux	Tous
QGIS_SERVER_LOG_STDERR	Activez la journalisation sur stderr. Cette variable n'a aucun effet lorsque QGIS_SERVER_LOG_FILE est défini. 0 ou <code>false</code> (insensible à la casse) 1 ou <code>true</code> (insensible à la casse)	faux	Tous
QGIS_SERVER_MAX_THREADS	Nombre de threads à utiliser lorsque le rendu parallèle est activé. Si la valeur est -1, le nombre de cœurs de processeur est utilisé.	-1	Tous
QGIS_SERVER_OVERRIDE_LOCALE	Définit LOCALE à utiliser par le serveur QGIS. La valeur par défaut est vide (pas de substitution). Exemple : <code>de_CH.utf8</code>	""	Tous
QGIS_SERVER_PARALLEL_GETMAP	Active le rendu parallèle pour les demandes WMS GetMap. Il est désactivé (<code>faux</code>) par défaut. Les valeurs disponibles sont : 0 ou <code>false</code> (insensible à la casse) 1 ou <code>true</code> (insensible à la casse)	faux	WMS
QGIS_SERVER_PROJECT_CACHE_STRATEGY	Controls the periodic strategy interval for cache invalidation, in milliseconds. Defaults to 0 which selects the legacy File system watcher.		Tous
QGIS_SERVER_PROJECT_CACHE_STRATEGY	Defines method for invalidating the project cache. Available strategies are: <code>filesystem</code>: uses the file system watcher strategy <code>periodic</code>: uses the last modified value of a project for checking changes on project configuration. Convenient on atypical file systems, such as NFS, or when the project file is stored in a database system like PostgreSQL. <code>off</code>: disables internal cache invalidation completely	filesystem	Tous

suite sur la page suivante

Table 6.1 – suite de la page précédente

Nom	Description	Default	Services
QGIS_SERVER_SERVICE_URL	This is an option to set the service URL if it is not present in the project. The service URL is defined from (in order of precedence): • Value defined in the project per service • The <code>QGIS_SERVER_<service>_SERVICE_URL</code> environment variable • The <code>QGIS_SERVER_SERVICE_URL</code> environment variable • The <code>X-Qgis-<service>-Service-Url</code> header • The <code>X-Qgis-Service-Url</code> header • Build from the <code>Forwarded-Host</code> header • Build from the <code>X-Forwarded-Host</code> and <code>X-Forwarded-Proto</code> headers • Build from the <code>Host</code> header and the server protocol • Build from the server name and the server protocol. In the last four cases, the resulting Service URL is based on the <code>MAP</code> parameter provided in the query string and on the incoming path request.	""	Tous
QGIS_SERVER_SHOW_GROUPS	Définit si un séparateur de groupe (par exemple mille séparateurs) doit être utilisé pour les valeurs numériques (par exemple dans les réponses <code>GetFeatureInfo</code>). La valeur par défaut est 0. • 0 ou <code>false</code> (insensible à la casse) • 1 ou <code>true</code> (insensible à la casse)	faux	WMS
QGIS_SERVER_TRUST_LAYERS	Il s'agit d'une option au niveau du projet pour améliorer le temps de lecture du projet en utilisant les étendues de la couche vecteur définies dans les métadonnées du projet et en désactivant la vérification de l'unicité de la clé primaire de la couche PostgreSQL/PostGIS. On peut forcer la confiance dans les métadonnées de la couche en fixant cette variable à « 1 » ou « true ». L'étendue de la couche vecteur sera alors celle définie dans le projet, et la clé primaire de la couche PostgreSQL/PostGIS définie dans la source de données est considérée comme unique sans contrôle. Ne l'utilisez pas si l'étendue des couches n'est pas fixée pendant l'utilisation du projet.	faux	Tous

suite sur la page suivante

Table 6.1 – suite de la page précédente

Nom	Description	Default	Services
QGIS_SERVER_WCS_SERVIC	This is an option to set the service URL if it is not present in the project. See QGIS_SERVER_SERVICE_URL for more information.	""	WCS
QGIS_SERVER_WFS_SERVIC	This is an option to set the service URL if it is not present in the project. See QGIS_SERVER_SERVICE_URL for more information.	""	WFS
QGIS_SERVER_WMS_MAX_I / QGIS_SERVER_WMS_MAX_I	Hauteur/largeur maximale d'une requête WMS. La valeur la plus conservatrice entre la requête et le projet est utilisée. Une valeur de -1 signifie qu'il n'y a pas de maximum défini.	-1	WMS
QGIS_SERVER_WMS_SERVI	This is an option to set the service URL if it is not present in the project. See QGIS_SERVER_SERVICE_URL for more information.	""	WMS
QGIS_SERVER_WMTS_SERV	This is an option to set the service URL if it is not present in the project. See QGIS_SERVER_SERVICE_URL for more information.	""	WMTS
QUERY_STRING	La chaîne de caractères de la requête, habituellement transmise par le serveur Web. Cette variable peut être utile pour tester le binaire de QGIS Server depuis la ligne de commande. Par exemple, pour tester une demande GetCapabilities sur la ligne de commande d'un projet qui nécessite également une connexion PostgreSQL définie dans un fichier pg_service.conf: <pre>PGSERVICEFILE=/etc/pg_service. ↪conf \ QUERY_STRING="MAP=/home/ ↪projects/world.qgs& ↪SERVICE=WMS& ↪REQUEST=GetCapabilities" \ /usr/lib/cgi-bin/qgis_mapserv. ↪fcgi</pre> Le résultat doit être le contenu de la réponse GetCapabilities ou, si quelque chose ne va pas, un message d'erreur.	""	Tous

6.3 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
```

(suite sur la page suivante)

(suite de la page précédente)

```
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.

6.4 Connexion au fichier de service

Pour faire en sorte qu'Apache puisse accéder au fichier de service PostgreSQL (consultez la section 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
  . . . . .
```

6.5 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 Server 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 Server.

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

8.1 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:bookworm-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 \
    && localedef -i en_US -f UTF-8 en_US.UTF-8 \
```

(suite sur la page suivante)

```
# Add the current key for package downloading
# Please refer to QGIS install documentation (https://www.qgis.org/fr/site/forusers/alldownloads.html#debian-ubuntu)
&& mkdir -m755 -p /etc/apt/keyrings \
&& wget -O /etc/apt/keyrings/qgis-archive-keyring.gpg https://download.qgis.org/downloads/qgis-archive-keyring.gpg \
# Add repository for latest version of qgis-server
# Please refer to QGIS repositories documentation if you want other version
(https://qgis.org/resources/installation-guide/#repositories)
&& echo "deb [signed-by=/etc/apt/keyrings/qgis-archive-keyring.gpg] https://qgis.org/debian bookworm main" | tee /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 v0.19.0
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 .
```

8.1.1 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			

8.1.2 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)

(suite de la page précédente)

```

    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>

8.1.3 Nettoyage

Pour nettoyer les images en cours, tapez :

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

8.2 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.

8.2.1 Swarm/docker-compose

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

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

```

(suite sur la page suivante)

(suite de la page précédente)

```
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			
10v2e7cl43u3	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
```

8.2.2 Kubernetes

Installation

If you have a **Docker Desktop** installation, using Kubernetes (aka k8s) is pretty straight forward: [enable 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: Always
          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/
          volumes:
            - name: nginx-conf
              configMap:
                name: nginx-configuration
---
kind: ConfigMap
apiVersion: v1
metadata:
  name: nginx-configuration
data:
  nginx.conf: |
    server {
      listen 80;
      server_name _;
      location / {
        root /usr/share/nginx/html;
        index index.html index.htm;
      }
      location /qgis-server {
        proxy_buffers 16 16k;
        proxy_buffer_size 16k;
        gzip off;
        include fastcgi_params;
        fastcgi_pass qgis-server:5555;
      }
    }

```

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:
  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 -f ./
```

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)
service/Kubernetes	ClusterIP	10.152.183.1	<none>	443/TCP
service/qgis-exec-server	ClusterIP	10.152.183.218	<none>	5555/TCP

(suite sur la page suivante)

(suite de la page précédente)

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 service/qgis-server service/qgis-nginx deployment/qgis-nginx
↪ deployment/qgis-server configmap/nginx-configuration
```

8.3 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*).

8.3.1 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.

To use docker-compose alike functionalities, you need to install the **ecs-cli** client and have proper permissions / roles. Then, with the help of the **ecs-cli compose** commands, you can reuse the *stack description*.

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

- *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.