# QGIS Server 3.16 User Guide

**QGIS Project**

**2022 年 04 月 02 日**

# 目次

# 第 1 章

# はじめに

QGIS Server is an open source WMS, WFS, OGC API for Features 1.0 (WFS3) and WCS implementation that, in addition, implements advanced cartographic features for thematic mapping. QGIS Server is a FastCGI/CGI (Common Gateway Interface) application written in C++ that works together with a web server (e.g., Apache, Nginx). It has Python plugin support allowing for fast and efficient development and deployment of new features.

QGIS サーバーは、GIS ロジックのためのバックエンドとして、地図のレンダリングに QGIS を使用しています。また、Qt ライブラリは、グラフィックスおよびプラットフォームに依存しない C ++ プログラミングのために使用されます。他の WMS のソフトウェアとは対照的に、QGIS サーバーは、サーバーの構成およびユーザー定義の地図作成ルールの両方、設定言語として地図作成規則を使用します。

QGIS のデスクトップと QGIS サーバーは同じ可視化ライブラリを使用するように、ウェブ上で公開されている地図は、デスクトップ GIS と同じに見えます。

次のセクションでは、Linux（Debian、Ubuntu および派生物）と Windows に QGIS サーバをセットアップするためのサンプル構成を提供します。サーバプラグイン開発の詳細は server_plugins を参照。

# 第 2 章

# はじめに

## 2.1 Debian のベースのシステムへのインストール

We will give a short and simple installation how-to for a minimal working configuration on Debian based systems (including Ubuntu and derivatives). However, many other distributions and OSs provide packages for QGIS Server.

**注釈:** In Ubuntu you can use your regular user, prepending `sudo` to commands requiring admin permissions. In Debian you can work as admin (`root`), without using `sudo`.

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

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

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

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

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

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

```

(次のページに続く)

(前のページからの続き)

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

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

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

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

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

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

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

### 2.1.1 Apache HTTP Server

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

Install Apache and mod_fcgid:

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

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

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

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

  DocumentRoot /var/www/html

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

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

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

  # QGIS log
  FcgidInitialEnv QGIS_SERVER_LOG_STDERR 1
  FcgidInitialEnv QGIS_SERVER_LOG_LEVEL 0

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

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

  # Set pg access via pg_service file
```

(次のページに続く)

(前のページからの続き)

```
  SetEnv PGSERVICEFILE /home/qgis/.pg_service.conf
  FcgidInitialEnv PGPASSFILE "/home/qgis/.pgpass"

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

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

</VirtualHost>
```

Further readings:

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

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

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

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

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

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

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

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

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

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

```
systemctl restart apache2
```

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

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

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

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

### 2.1.2 NGINX HTTP Server

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

You can also use QGIS Server with NGINX. Unlike Apache, NGINX does not automatically spawn FastCGI processes. The FastCGI processes are to be started by something else.

Install NGINX:

```
apt install nginx
```

- As a first option, you can use **spawn-fcgi** or **fcgiwrap** to start and manage the QGIS Server processes.

Official Debian packages exist for both. When you have no X server running and you need, for example, printing, you can use *xvfb*.

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

### NGINX Configuration

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

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

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

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

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

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

Moreover, you can use some **環境変数** to configure QGIS Server. In the NGINX configuration file, /etc/nginx/nginx.conf, you have to use fastcgi_param instruction to define these variables as shown below:

```
location /qgisserver {
     gzip           off;
     include        fastcgi_params;
     fastcgi_param  QGIS_SERVER_LOG_STDERR  1;
```

(次のページに続く)

(前のページからの続き)

```
    fastcgi_param  QGIS_SERVER_LOG_LEVEL   0;
    fastcgi_pass   unix:/var/run/qgisserver.socket;
}
```

### FastCGI wrappers

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

### spawn-fcgi

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

```
apt install spawn-fcgi
```

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

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

And restart NGINX to take into account the new configuration:

```
systemctl restart nginx
```

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

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

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

**注釈:** When using spawn-fcgi, you may directly define environment variables before running the server. For example: `export QGIS_SERVER_LOG_STDERR=1`

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

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

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

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

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

Then enable and start the service:

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

**警告:** With the above commands spawn-fcgi spawns only one QGIS Server process.

### fcgiwrap

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

```
apt install fcgiwrap
```

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

```
location /qgisserver {
    gzip           off;
```

(次のページに続く)

(前のページからの続き)

```
    include         fastcgi_params;
    fastcgi_pass    unix:/var/run/fcgiwrap.socket;
    fastcgi_param   SCRIPT_FILENAME /usr/lib/cgi-bin/qgis_mapserv.fcgi;
}
```

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

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

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

### Systemd

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

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

- a Socket unit
- and a Service unit.

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

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

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

[Install]
WantedBy=sockets.target
```

Now enable and start sockets:

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

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

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

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

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

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

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

Now start socket service:

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

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

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

server {
   …
```

(次のページに続く)

(前のページからの続き)

```

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

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

```
systemctl restart nginx
```

Thanks to Oslandia for sharing their tutorial.

### 2.1.3 Xvfb

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

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

To install the package:

```
apt install xvfb
```

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

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

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

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

`xvfb.service` の状態を有効にし、開始し、チェックします：

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

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

### With Apache

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

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

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

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

```
systemctl restart apache2
```

### With NGINX

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

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

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

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

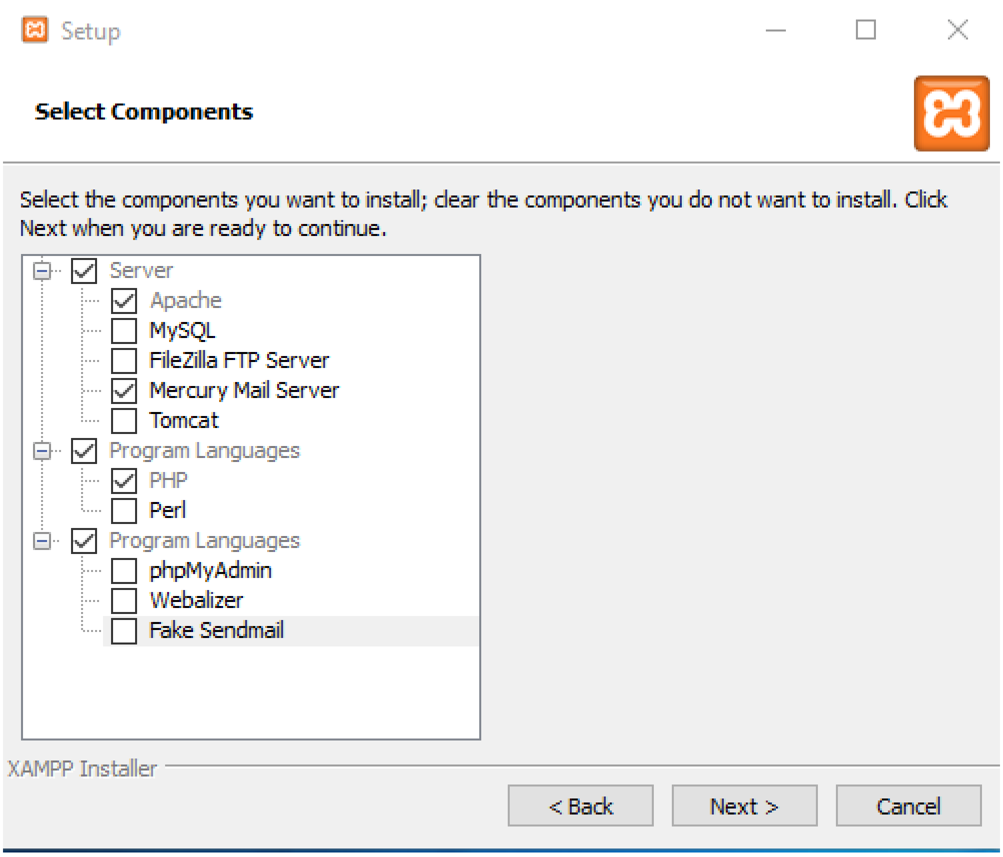
```
fastcgi_param  DISPLAY       ":99";
```

## 2.2 Installation on Windows

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

A simple procedure is the following:

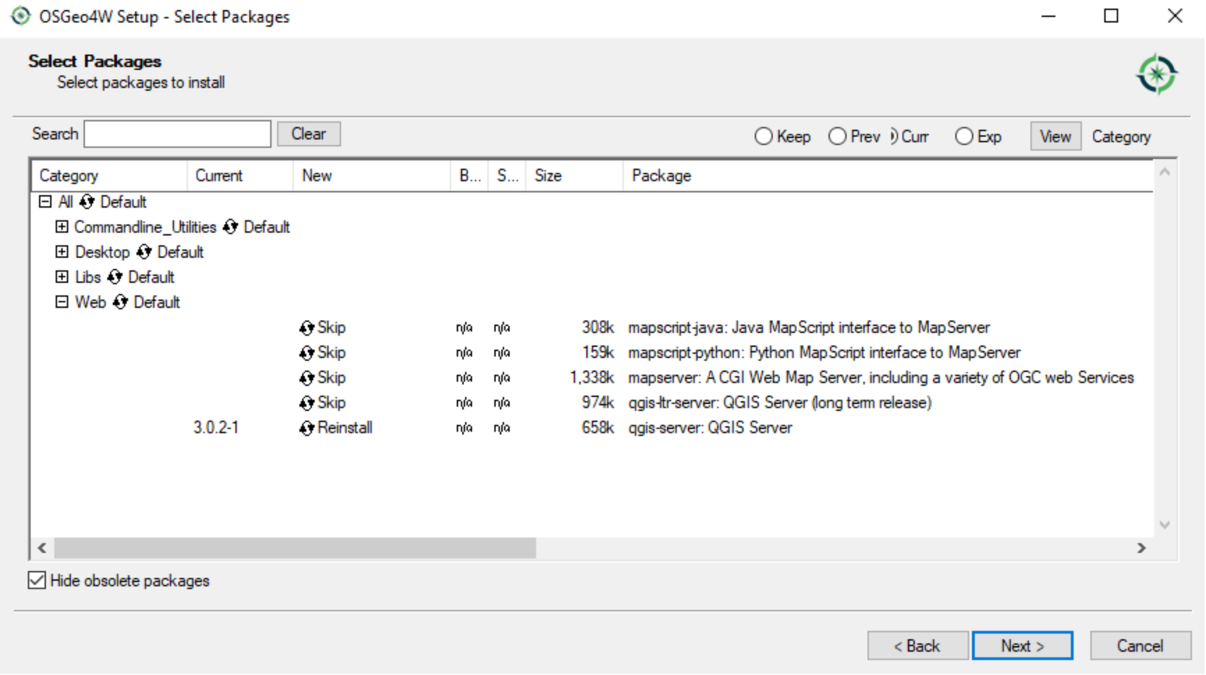
1. Download the XAMPP installer (https://www.apachefriends.org/download.html) for Windows and install Apache



Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

- ☑ Server
  - ☑ Apache
  - □ MySQL
  - □ FileZilla FTP Server
  - ☑ Mercury Mail Server
  - □ Tomcat
- ☑ Program Languages
  - ☑ PHP
  - □ Perl
- ☑ Program Languages
  - □ phpMyAdmin
  - □ Webalizer
  - □ Fake Sendmail

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



| Category | Current | New | B... | S... | Size | Package |
|---|---|---|---|---|---|---|
| ⊟ All Default | | | | | | |
| ⊞ Commandline_Utilities Default | | | | | | |
| ⊞ Desktop Default | | | | | | |
| ⊞ Libs Default | | | | | | |
| ⊟ Web Default | | | | | | |
| | | Skip | n/a | n/a | 308k | mapscript-java: Java MapScript interface to MapServer |
| | | Skip | n/a | n/a | 159k | mapscript-python: Python MapScript interface to MapServer |
| | | Skip | n/a | n/a | 1,338k | mapserver: A CGI Web Map Server, including a variety of OGC web Services |
| | | Skip | n/a | n/a | 974k | qgis-ltr-server: QGIS Server (long term release) |
| | 3.0.2-1 | Reinstall | n/a | n/a | 658k | qgis-server: QGIS Server |

☑ Hide obsolete packages

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

From:

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

To:

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

From:

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

To:

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

From:

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

To:

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

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

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

5. Restart the Apache web server from the XAMPP Control Panel and open browser window to testing a GetCapabilities request to QGIS Server

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

## 2.3 Serve a project

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

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

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

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

- airports
- places
- countries
- countries_shapeburst

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

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

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

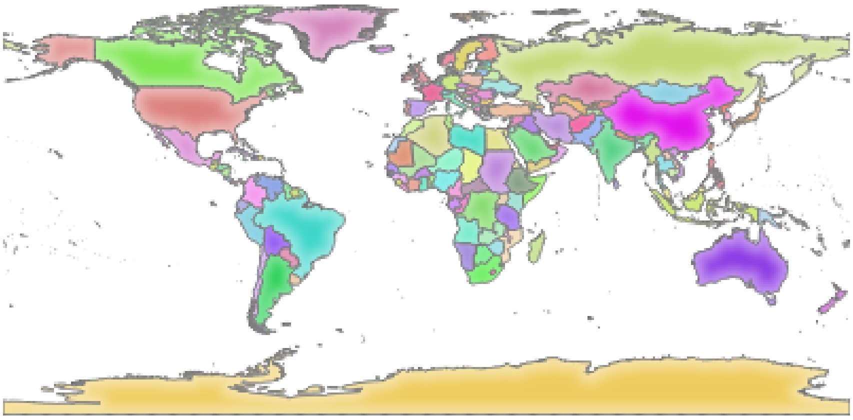


図 2.1　Server response to a basic GetMap request

Note that you may define **QGIS_PROJECT_FILE** environment variable to use a project by default instead of giving a **MAP** parameter (see **環境変数**).

For example with spawn-fcgi:

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

## 2.4 Configure your project

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