



# **QGIS Server 3.22 User Guide**

**QGIS Project**

**04 dez. 2023**



---

## Sumário

---

<b>1</b>	<b>Introdução</b>	<b>1</b>
<b>2</b>	<b>Iniciando</b>	<b>3</b>
2.1	Installation on Debian-based systems . . . . .	3
2.1.1	Servidor HTTP Apache . . . . .	4
2.1.2	Servidor HTTP NGINX . . . . .	6
2.1.3	Xvfb . . . . .	11
2.2	Instalação no Windows . . . . .	12
2.3	Serve a project . . . . .	14
2.4	Configurar seu projeto . . . . .	15
2.4.1	Recursos WMS . . . . .	17
2.4.2	WFS capabilities . . . . .	18
2.4.3	WCS capabilities . . . . .	18
2.4.4	Sintonia fina do seu OWS . . . . .	18
2.5	Integração com terceiros . . . . .	18
2.5.1	Integração com o QGIS Desktop . . . . .	18
2.5.2	Integração com MapProxy . . . . .	18
2.5.3	Integração com QWC2 . . . . .	19
<b>3</b>	<b>Serviços</b>	<b>21</b>
3.1	Fundamentos . . . . .	21
3.1.1	SERVIÇO . . . . .	22
3.1.2	REQUISITAR . . . . .	22
3.1.3	MAPA . . . . .	22
3.1.4	NOME_ARQUIVO . . . . .	22
3.1.5	Nome curto . . . . .	23
3.2	Web Map Service (WMS) . . . . .	23
3.2.1	GetCapabilities . . . . .	24
3.2.2	GetMap . . . . .	24
3.2.3	GetFeatureInfo . . . . .	33
3.2.4	GetLegendGraphics . . . . .	36
3.2.5	GetStyle(s) . . . . .	49
3.2.6	DescribeLayer . . . . .	49
3.2.7	GetPrint . . . . .	50
3.2.8	GetProjectSettings . . . . .	52
3.2.9	GetSchemaExtension . . . . .	53
3.2.10	External WMS layers . . . . .	53
3.2.11	Redlining . . . . .	54
3.3	Web Feature Service (WFS) . . . . .	56
3.3.1	GetCapabilities . . . . .	56
3.3.2	GetFeature . . . . .	57

3.3.3	DescribeFeatureType . . . . .	63
3.3.4	Transação . . . . .	64
3.4	Web Coverage Service (WCS) . . . . .	66
3.4.1	GetCapabilities . . . . .	66
3.4.2	DescribeCoverage . . . . .	67
3.4.3	GetCoverage . . . . .	69
3.5	Web Map Tile Service (WMTS) . . . . .	70
3.5.1	GetCapabilities . . . . .	71
3.5.2	GetTile . . . . .	71
3.5.3	GetFeatureInfo . . . . .	72
3.6	Funções OGC API . . . . .	73
3.6.1	Representação de recursos . . . . .	74
3.6.2	Endpoints . . . . .	74
3.6.3	Paginação . . . . .	78
3.6.4	Feature filtering . . . . .	78
3.6.5	Feature sorting . . . . .	79
3.6.6	Seleção de atributo . . . . .	79
3.6.7	Personalizar as páginas HTML . . . . .	80
<b>4</b>	<b>Catálogo</b>	<b>81</b>
<b>5</b>	<b>Plugins</b>	<b>85</b>
5.1	Lista de plugins . . . . .	85
5.2	Localização dos plugins . . . . .	85
5.3	Instalação . . . . .	85
5.3.1	Manualmente com um ZIP . . . . .	85
5.3.2	Com uma ferramenta de linha de comando . . . . .	86
5.4	Configuração do Servidor HTTP . . . . .	86
5.4.1	Apache . . . . .	86
5.5	Como usar um plug-in . . . . .	87
<b>6</b>	<b>Configuração avançada</b>	<b>89</b>
6.1	Registrando . . . . .	89
6.2	Variáveis ambientais . . . . .	89
6.3	Settings summary . . . . .	91
6.4	Connection to service file . . . . .	92
6.5	Adicione fontes ao seu servidor linux . . . . .	92
<b>7</b>	<b>Servidor de desenvolvimento</b>	<b>93</b>
<b>8</b>	<b>Containerized deployment</b>	<b>95</b>
8.1	Imagens simples do estivador . . . . .	95
8.1.1	Primeira execução . . . . .	97
8.1.2	Amostra utilizável . . . . .	97
8.1.3	Limpeza . . . . .	98
8.2	Pilhas de docas . . . . .	98
8.2.1	Enxame/doca-composição . . . . .	98
8.2.2	Kubernetes . . . . .	99
8.3	Implantação de nuvens . . . . .	103
8.3.1	AWS usecase . . . . .	103
<b>9</b>	<b>Perguntas mais frequentes</b>	<b>105</b>

# CAPÍTULO 1

---

## Introdução

---

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 Server uses QGIS as back end for the GIS logic and for map rendering. Furthermore, the Qt library is used for graphics and for platform-independent C++ programming. In contrast to other WMS software, the QGIS Server uses cartographic rules as a configuration language, both for the server configuration and for the user-defined cartographic rules.

Como o desktop QGIS e o Servidor QGIS usam as mesmas bibliotecas de visualização, os mapas que são publicados na web têm a mesma aparência do desktop GIS.

In the following sections, we will provide a sample configuration to set up a QGIS Server on Linux (Debian, Ubuntu and derivatives) and on Windows. For more information about server plugin development, please read `server_plugins`.

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

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



# CAPÍTULO 2

---

## Iniciando

---

### 2.1 Installation on Debian-based systems

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.

---

**Nota:** No Ubuntu, você pode usar seu usuário comum, acrescentando `sudo` aos comandos que requerem permissões de administrador. No Debian você pode trabalhar como administrador (`root`), sem usar `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.

Uma vez configurado o repositório da versão desejada e instalado o QGIS Server, você pode testar a instalação com:

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

Se você obtiver a seguinte saída, o servidor está corretamente instalado.

---

**Nota:** Dependendo da versão do QGIS, você poderá ver uma saída ligeiramente diferente informada quando executar `qgis_mapserv.fcgi``.

---

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

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

**Nota:** Como visto abaixo, o QGIS informa um código de Status 400, que identifica corretamente que a solicitação falhou porque não há sessão http ativa. Isto não é um bug e indica que o servidor está funcionando corretamente.

```
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 3.22.6-Białowieża
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_3.22.zip
unzip release_3.22.zip
mv QGIS-Training-Data-release_3.22/exercise_data/qgis-server-tutorial-data/world.
↪qgs .
mv QGIS-Training-Data-release_3.22/exercise_data/qgis-server-tutorial-data/
↪naturalearth.sqlite .
```

Certamente, você pode usar seu software GIS favorito para abrir este arquivo e dar uma olhada na configuração e nas camadas disponíveis.

Para implantar corretamente o servidor QGIS, você precisa de um servidor HTTP. As escolhas recomendadas são [Apache](#) ou [NGINX](#).

### 2.1.1 Servidor HTTP Apache

---

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

---

1. Instalar Apache e `mod_fcgid`:

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

2. Você pode executar o QGIS Server em seu site padrão, mas vamos configurar um `virtual host` especificamente para isto, como segue.

1. No diretório `:file: '/etc/apache2/sites-available'`, crie um arquivo chamado `:arquivo: 'qgis.demo.conf'`, com este conteúdo:

```
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
```

(continua na próxima página)

(continuação da página anterior)

```

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
FcgidConnectTimeout 60
</IfModule>

</VirtualHost>

```

**Nota:** Algumas das opções de configuração acima são explicadas nas seções Servidor *variáveis de ambiente* e *pg\_service file*

- Vamos agora criar os diretórios que irão armazenar os logs e o banco de dados de autenticação do Servidor QGIS:

```

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

```

(continua na próxima página)

(continuação da página anterior)

```
chown www-data:www-data /home/qgis/qgisserverdb
```

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

3. Agora podemos habilitar o host virtual e o `fcgid mod`, se isso ainda não tiver sido feito:

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

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

```
systemctl restart apache2
```

5. Agora que o Apache sabe que deve responder aos pedidos para `http://qgis.demo` também precisamos configurar o sistema do cliente para que ele saiba quem é `qgis.demo`. Fazemos isso adicionando `127.0.0.1 qgis.demo` no arquivo `hosts`.

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

**Importante:** 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 `qgis.demo` 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 Servidor HTTP NGINX

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

Você também pode utilizar o Servidor QGIS com **NGINX**. Ao contrário do Apache, NGINX não gera automaticamente processos FastCGI. Os processos FastCGI devem ser iniciados por outra coisa.

Instalar 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.

## Configuração NGINX

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

```

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

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

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

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

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

```

Moreover, you can use some *Variáveis ambientais* to configure QGIS Server. In the NGINX configuration file, /etc/nginx/nginx.conf, you have to use **fastcgi\_param** instruction to define these variables as shown below:

```

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

```

## FastCGI wrappers

**Aviso:** **fciwrap** 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 fciwrap, 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.

### gerar-fcgi

Se você quiser utilizar **spawn-fcgi**:

- O primeiro passo é instalar o pacote:

```
apt install spawn-fcgi
```

2. Em seguida, introduzir o seguinte bloco na configuração do seu servidor NGINX:

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

3. E reiniciar o NGINX para levar em conta a nova configuração:

```
systemctl restart nginx
```

4. 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
```

O Servidor QGIS está agora disponível em <http://qgis.demo/qgisserver>.

---

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

---

É claro, você pode adicionar um roteiro de inicialização para iniciar o QGIS Server no momento da inicialização ou sempre que quiser. Por exemplo, com **systemd**:

1. Edite o arquivo:`/etc/systemd/system/qgis-server.service` com este conteúdo:

```
[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
```

2. Em seguida, habilitar e iniciar o serviço:

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

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

## fcgiwrap

Utilizar **fcgiwrap** é muito mais fácil de configurar do que **spawn-fcgi** mas é muito mais lento.

1. Primeiro você precisa instalar o pacote correspondente:

```
apt install fcgiwrap
```

2. Em seguida, introduzir o seguinte bloco na configuração do seu servidor 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. Finally, restart NGINX and **fcgiwrap** to take into account the new configuration:

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

O Servidor QGIS está agora disponível em <http://qgis.demo/qgisserver>.

## Sistemad

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.

Este método, para implantar o Servidor QGIS, conta com duas unidades Systemd para configurar: uma **Unidade de bolso** e uma **Unidade de serviço**.

1. 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
```

2. Agora ative e inicie as tomadas:

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

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

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

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

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

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

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

4. Now start socket service:

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

5. Finalmente, para o servidor HTTP NGINX, vamos apresentar a configuração desta instalação:

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

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

6. 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/>.

1. Para instalar o pacote:

```
apt install xvfb
```

2. 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
```

3. Ative, inicie e verifique o status do xvfb.service:

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

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

- Usando Apache:

1. Adicione à sua configuração *Fcgid* (ver *Servidor HTTP Apache*):

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

2. Reinicie o Apache para que a nova configuração seja considerada:

```
systemctl restart apache2
```

- Usando NGINX

- 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.

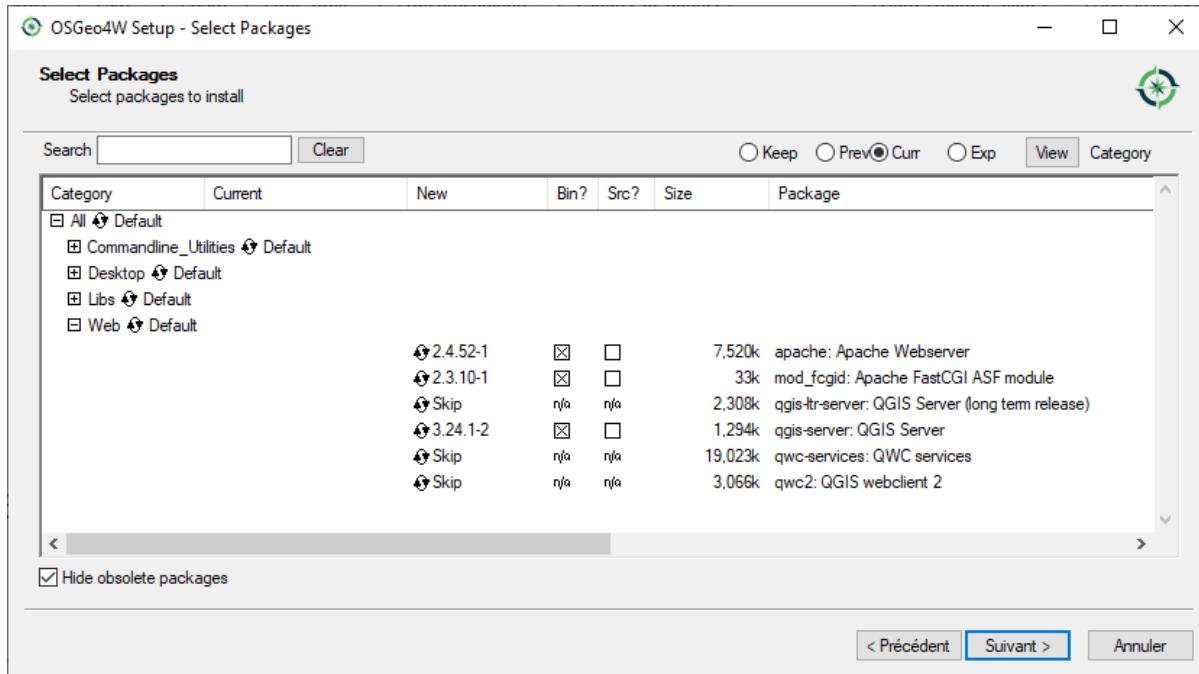
```
fastcgi_param DISPLAY      ":99";
```

## 2.2 Instalação no Windows

O QGIS Server também pode ser instalado em sistemas Windows usando a versão 64 bit do instalador de rede OSGeo4W (<https://qgis.org/en/site/forusers/download.html>).

Um procedimento simples é o seguinte:

1. Baixe e execute o instalador OSGeo4W
2. Siga a “Instalação Avançada” e instale os pacotes **QGIS Desktop**, **QGIS Server apache** e **mod\_fcgid**.



3. O Apache não é instalado diretamente como um serviço no Windows. Você precisa fazê-lo:
  1. Clique com o botão direito do mouse no arquivo OSGeo4W.bat na raiz da pasta C:\OSGeo4W64\ (se os caminhos de instalação padrão tiverem sido utilizados) e selecione *Executar como administrador*
  2. No console, execute apache-install.bat, que terá saída

```
> 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.
...
```

O serviço está iniciado, como você pode notar no relatório. Mas o servidor pode não funcionar devido à falta de configuração personalizada.

4. Edite o :arquivo:`C:\OSGeo4W64\apps\apache\conf\httpd.conf` arquivo com as seguintes mudanças (várias outras combinações são possíveis):

Finalidade	Configuração existente	Substituição
(Opcional) Personalize o endereço para ouvir usando um IP e/ou porta, Você pode e adicionar quantas entradas desejar.	<code>Listen \${SRVPORT}</code>	<code>Listen localhost:8080</code>
Indique onde encontrar os arquivos de script	<code>ScriptAlias /cgi-bin/ "\${SRVROOT}/cgi-bin/"</code>	<code>ScriptAlias /cgi-bin/ "C:/OSGeo4W64/apps/qgis/bin/"</code>
Fornecer permissões na pasta de scripts	<code>&lt;Directory "\${SRVROOT}/cgi-bin"&gt; AllowOverride None Options None Require all granted &lt;/Directory&gt;</code>	<code>&lt;Directory "C:/OSGeo4W64/apps/qgis/bin"&gt; SetHandler cgi-script AllowOverride None Options ExecCGI Require all granted &lt;/Directory&gt;</code>
Habilitar extensões de arquivo para uso em arquivos de script. Descomentar e completar	<code>#AddHandler cgi-script .cgi</code>	<code>AddHandler cgi-script .cgi .exe</code>
Adicionar mais variáveis de configuração personalizadas OSGeo4W	<code># parse OSGeo4W apache.conf files IncludeOptional "C:/OSGeo4W64/httpd.d/httpd_*.conf"</code>	<code># parse OSGeo4W apache.conf files IncludeOptional "C:/OSGeo4W64/httpd.d/httpd_*.conf" SetEnv GDAL_DATA "C:/OSGeo4W64/share/gdal" SetEnv QGIS_AUTH_DB_DIR "C:/OSGeo4W64/apps/qgis/resources"</code>

5. Reiniciar o servidor web Apache

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

6. Abra a janela do navegador para testar uma solicitação de GetCapabilities no Servidor QGIS. Substitua localhost:8080 pelo IP e porta que você definir para ouvir.

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

Um :arquivo:`XML` arquivo com as capacidades deve ser devolvido. Seu servidor está pronto para uso.

## 2.3 Serve a project

Agora que o Servidor QGIS está instalado e funcionando, só temos que usá-lo.

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 [Configurar seu projeto](#).

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.

Ao abrir o projeto e dar uma olhada rápida nas camadas, sabemos que 4 camadas estão disponíveis no momento:

- aeroportos
- lugares
- países
- countries\_shapeburst

Você não precisa entender a solicitação completa por enquanto, mas pode recuperar um mapa com algumas das camadas anteriores graças ao Servidor QGIS fazendo algo assim em seu navegador da web para recuperar a camada *países*:

```
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
```

Se você obtiver a próxima imagem, então o Servidor QGIS está rodando corretamente:

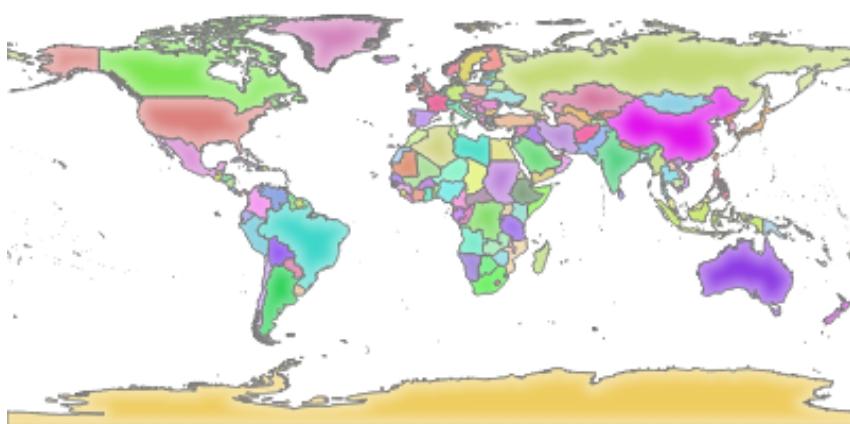


Figura2.1: Server response to a basic GetMap request

Note that you may define **QGIS\_PROJECT\_FILE** environment variable to use a project by default instead of giving a **MAP** parameter (see [Variáveis ambientais](#)).

Por exemplo com spawn-fcgi:

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

(continua na próxima página)

(continuação da página anterior)

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

## 2.4 Configurar seu projeto

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.

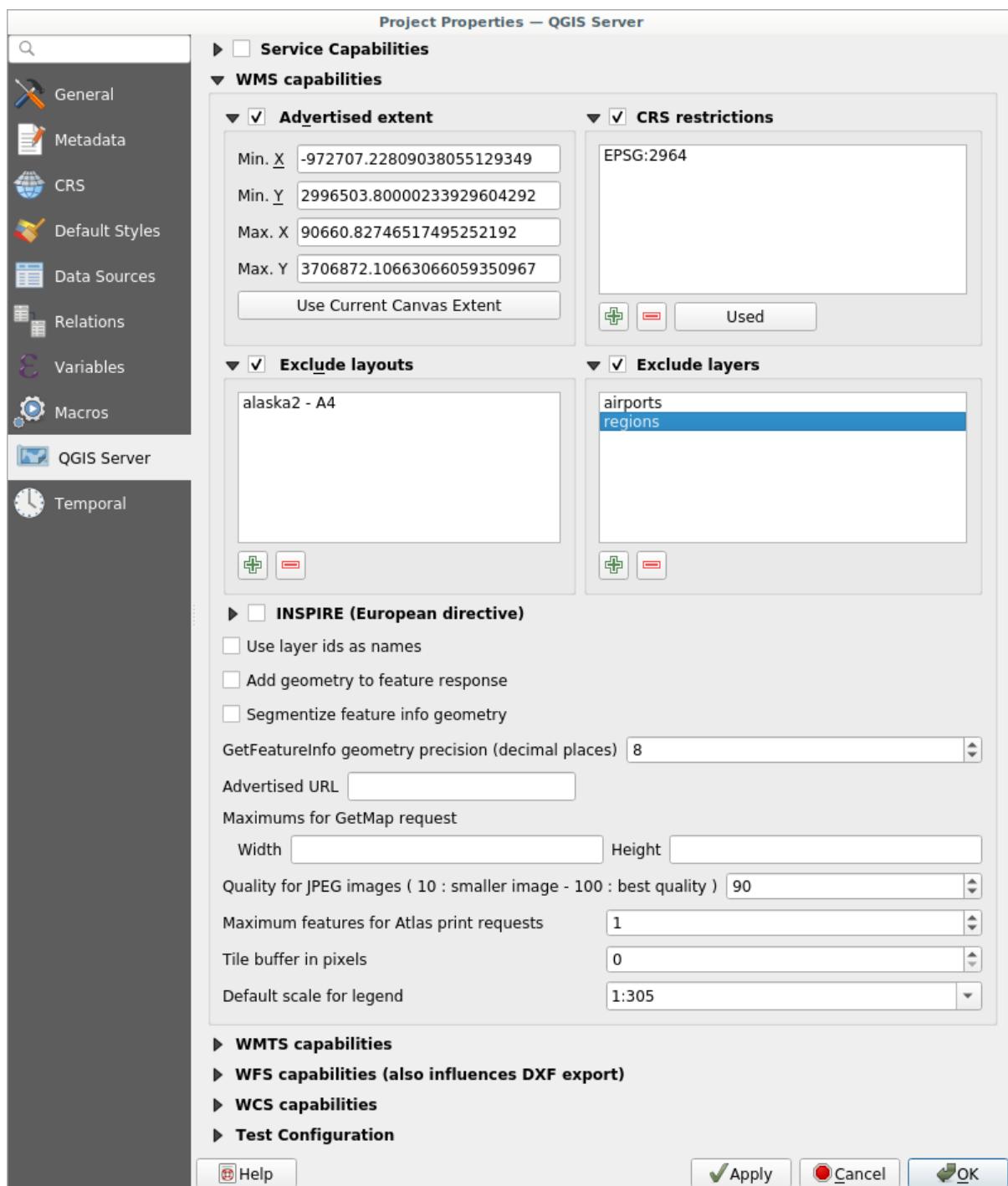


Figura2.2: Definitions for a QGIS Server WMS/WFS/WCS project

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

## 2.4.1 Recursos WMS

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

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

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

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

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

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

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

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

Se você quiser que o QGIS Server anuncie URLs de solicitação específicas na resposta do WMS `GetCapabilities`, digite a URL correspondente no campo *URL anunciada*

Além disso, você pode restringir o tamanho máximo dos mapas devolvidos pelo pedido `GetMap` inserindo a largura e altura máximas nos respectivos campos sob :guilabel:`Solicitação máxima do GetMap`.

Você pode mudar o fator *Qualidade para imagens JPEG*. O fator de qualidade deve estar na faixa de 0 a 100. Especifique 0 para compressão máxima, 100 para nenhuma compressão.

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

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

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

## 2.4.2 WFS capabilities

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

## 2.4.3 WCS capabilities

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

## 2.4.4 Sintonia fina do seu OWS

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

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

## 2.5 Integração com terceiros

O Servidor QGIS fornece serviços web padrão OGC como WMS, WFS, etc. assim ele pode ser utilizado por uma grande variedade de ferramentas para o usuário final.

### 2.5.1 Integração com o QGIS Desktop

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

Além disso, deve ser estabelecido um fluxo de trabalho dedicado de atualização para atualizar um projeto utilizado por um Servidor QGIS (ou seja, copiar arquivos de projeto para a localização do servidor e reiniciar o Servidor QGIS). Por enquanto, os processos automatizados (como serviço de recarga de mensagens do servidor em fila de espera) ainda não estão implementados.

### 2.5.2 Integração com 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 Integração com QWC2

QWC2 <<https://github.com/qgis/qwc2>> é uma aplicação web responsiva e dedicada ao Servidor QGIS. Ele ajuda você a construir um visualizador de mapas altamente personalizado com seleção de camadas, informações de características, etc... Também estão disponíveis muitos plugins como autenticação ou serviço de impressão, a lista completa está disponível neste **repositório <[>](https://github.com/qwc-services)**.



# CAPÍTULO 3

---

## Serviços

---

QGIS Server is able to serve data according to standard protocols as described by the **Open Geospatial Consortium (OGC)**:

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

Extra vendor parameters and requests are supported in addition to the original standard that greatly enhance the possibilities of customizing its behavior thanks to the QGIS rendering engine.

### 3.1 Fundamentos

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.

Conceitos padrão:

Conceito	Descrição
<i>SERVIÇO</i>	Nome do serviço
<i>REQUISIÇÃO</i>	Nome da requisição

Conceitos de fornecedor:

Conceito	Descrição
<i>MAPA</i>	arquivo de projeto QGIS
<i>FILE_NAME</i>	File name of the downloaded file
<i>Nome curto</i>	Definição de nome curto

### 3.1.1 SERVIÇO

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  
&...
```

---

**Nota:** Not available for REST based services like [WFS3 \(OGC API Features\)](#).

---

### 3.1.2 REQUISITAR

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.

URL example for the **GetCapabilities** request:

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

---

**Nota:** Not available for REST based services like [WFS3 \(OGC API Features\)](#).

---

### 3.1.3 MAPA

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 [Configuração avançada](#) chapter.

Exemplo de URL:

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

### 3.1.4 NOME\_ARQUIVO

Se este parâmetro do fornecedor for definido, a resposta do servidor será enviada ao cliente como um anexo de arquivo com o nome do arquivo especificado.

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

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

---

**Nota:** Not available for REST based services like [WFS3 \(OGC API Features\)](#).

---

### 3.1.5 Nome curto

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*.

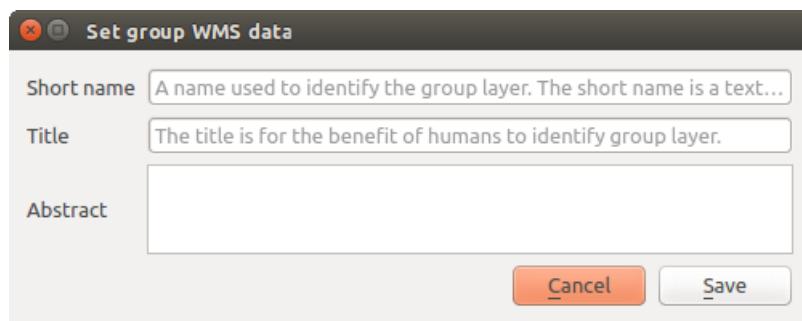


Figura3.1: Set group WMS data dialog

Assim, o nome abreviado pode ser usado para identificar esses itens ao interagir com o Servidor QGIS. Por exemplo, com o parâmetro padrão CAMADAS:

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

## 3.2 Web Map Service (WMS)

The **1.1.1** and **1.3.0** WMS standards implemented in QGIS Server provide a HTTP interface to request map or legend images generated from a QGIS project. A typical WMS request defines the QGIS project to use, the layers to render as well as the image format to generate. Basic support is also available for **Styled Layer Descriptor (SLD)**.

Especificações:

- [WMS 1.1.1](#)
- [WMS 1.3.0](#)
- [SLD 1.1.0 WMS profile](#)

Standard requests provided by QGIS Server:

Solicitação	Descrição
<a href="#">GetCapabilities</a>	Retorna metadados XML com informações sobre o servidor
<a href="#">GetMap</a>	Retorna um mapa
<a href="#">GetFeatureInfo</a>	Recupera dados (geometria e valores) para uma localização de pixel
<a href="#">GetLegendGraphics</a>	Retorna os símbolos das legendas
<a href="#">GetStyle(s)</a>	Returns XML document with style description in SLD
<a href="#">DescribeLayer</a>	Returns information about WFS and WCS availability respectively for vector and raster layers

Vendor requests provided by QGIS Server:

Solicitação	Descrição
<a href="#">GetPrint</a>	Retorna um layout QGIS
<a href="#">GetProjectSettings</a>	Returns specific information about QGIS Server
<a href="#">GetSchemaExtension</a>	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:

Parâmetros	Requerido	Descrição
<a href="#">SERVICE</a>	Sim	Nome do serviço ( <b>WMS</b> )
<a href="#">REQUEST</a>	Sim	Name of the request ( <b>GetCapabilities</b> )
<a href="#">VERSION</a>	Não	Versão do serviço

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

Parâmetros	Requerido	Descrição
<a href="#">MAP</a>	Sim	arquivo de projeto QGIS

Exemplo de URL:

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

### 3.2.2 GetMap

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

Parâmetros	Requerido	Descrição
<i>SERVICE</i>	Sim	Nome do serviço ( <b>WMS</b> )
<i>REQUEST</i>	Sim	Name of the request ( <b>GetMap</b> )
<i>VERSION</i>	Sim	Versão do serviço
<i>LAYERS</i>	Não	Layers to display
<i>STYLES</i>	Não	Estilo das camadas
<i>SRS / CRS</i>	Sim	Sistema de referência de coordenadas
<i>BBOX</i>	Sim	Extensão do mapa
<i>WIDTH</i>	Sim	Largura da imagem em pixels
<i>HEIGHT</i>	Sim	Altura da imagem em pixels
<i>FORMAT</i>	Não	Formato de imagem
<i>TRANSPARENT</i>	Não	Plano de fundo transparente
<i>SLD</i>	Não	URL of an SLD to be used for styling
<i>SLD_BODY</i>	Não	In-line SLD (XML) to be used for styling

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

Parâmetros	Requerido	Descrição
<i>MAP</i>	Sim	arquivo de projeto QGIS
<i>BGCOLOR</i>	Não	Especificar a cor de fundo
<i>DPI</i>	Não	Especificar a resolução de saída
<i>IMAGE_QUALITY</i>	Não	Compressão JPEG
<i>OPACITIES</i>	Não	Opacidade para camada ou grupo
<i>FILTER</i>	Não	Subset of features
<i>SELECTION</i>	Não	Highlight features
<i>NOME_ARQUIVO</i>	Não	File name of the downloaded file Only for <i>FORMAT=application/dxf</i>
<i>FORMAT_OPTIONS</i>	Não	Opções do formato de arquivo especificado Only for <i>FORMAT=application/dxf</i>
<i>TILED</i>	Não	Working in <i>tiled mode</i>

Exemplo de URL:

```
http://localhost/qgis_server?
SERVICE=WMS
&VERSION=1.3.0
&REQUEST=GetMap
&MAP=/home/qgis/projects/world.qgs
&LAYERS=mylayer1,mylayer2,mylayer3
&STYLES=style1,default,style3
&OPACITIES=125,200,125
&CRS=EPSG:4326
&WIDTH=400
&HEIGHT=400
&FORMAT=image/png
&TRANSPARENT=TRUE
&DPI=300
&TILED=TRUE
```

## VERSÃO

Este parâmetro permite especificar a versão do serviço a ser utilizada. Os valores disponíveis para o parâmetro VERSÃO são:

- 1.1.1
- 1.3.0

According to the version number, slight differences have to be expected as explained later for the next parameters:

- CRS / SRS
- BBOX

## CAMADAS

Este parâmetro permite especificar as camadas a serem exibidas no mapa. Os nomes devem ser separados por vírgula.

In addition, QGIS Server introduced some options to select layers by:

- a *short name*
- o ID da camada

The project option allowing to select layers by their id is in *QGIS Server ▶ WMS Capabilities* menu of the *Project ▶ Properties...* dialog. Check the *Use layer ids as names* checkbox to activate this option.

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

## ESTILOS

Este parâmetro pode ser usado para especificar o estilo de uma camada para a etapa de renderização. Os estilos devem ser separados por vírgula. O nome do estilo padrão é padrão.

## SRS / CRS

This parameter allows to indicate the map output Spatial Reference System in WMS **1.1.1** and has to be formed like EPSG:XXXX. Note that CRS is also supported if current version is **1.1.1**.

For WMS **1.3.0**, CRS parameter is preferable but SRS is also supported.

Note that if both CRS and SRS parameters are indicated in the request, then it's the current version indicated in VERSION parameter which is decisive.

In the next case, the SRS parameter is kept whatever the VERSION parameter because CRS is not indicated:

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

In the next case, the SRS parameter is kept instead of CRS because of the VERSION parameter:

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

In the next case, the CRS parameter is kept instead of SRS because of the VERSION parameter:

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

## BBOX

Este parâmetro permite especificar a extensão do mapa com unidades de acordo com o SRC atual. As coordenadas devem ser separadas por vírgula.

The BBOX parameter is formed like `min_a,min_b,max_a,max_b` but a and b axis definition is different according to the current VERSION parameter:

- in WMS **1.1.1**, the axis ordering is always east/north
- in WMS **1.3.0**, the axis ordering depends on the CRS authority

For example in case of EPSG:4326 and WMS **1.1.1**, a is the longitude (east) and b the latitude (north), leading to a request like:

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

But in case of WMS **1.3.0**, the axis ordering defined in the EPSG database is north/east so a is the latitude and b the longitude:

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

## LARGURA

Este parâmetro permite especificar a largura em pixels da imagem de saída.

## ALTURA

Este parâmetro permite especificar a altura em pixels da imagem de saída.

## FORMATO

Este parâmetro pode ser usado para especificar o formato da imagem do mapa. Os valores disponíveis são:

- jpg
- jpeg
- imagem/jpeg
- imagem/png
- image/png; mode=1bit
- image/png; mode=8bit
- image/png; mode=16bit
- application/dxf: only layers that have read access in the WFS service are exported in the DXF format

Exemplo de 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:SYMBOLLAYERSYMBOLITY;SCALE:250  
&FILE_NAME=plan.dxf
```

## TRANSPARENTE

Este parâmetro booleano pode ser usado para especificar a transparência do plano de fundo. Os valores disponíveis são (não diferenciam maiúsculas de minúsculas):

- VERDADEIRO
- FALSO

No entanto, este parâmetro é ignorado se o formato da imagem indicado com FORMATO for diferente de PNG.

## BGCOLOR

This parameter allows to indicate a background color for the map image. However it cannot be combined with TRANSPARENT parameter in case of PNG images (transparency takes priority). The colour may be literal or in hexadecimal notation.

Exemplo de URL com a notação literal:

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

Exemplo de URL com a notação hexadecimal:

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

## DPI

Este parâmetro pode ser usado para especificar a resolução de saída solicitada.

## QUALIDADE\_IMAGEM

Este parâmetro é usado apenas para imagens JPEG. Por padrão, a compressão JPEG é -1.

You can change the default per QGIS project in the *OWS Server ▶ WMS capabilities* menu of the *Project ▶ Properties...* dialog. If you want to override it in a GetMap request you can do it using the IMAGE\_QUALITY parameter.

## OPACIDADES

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).

Exemplo de 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
```

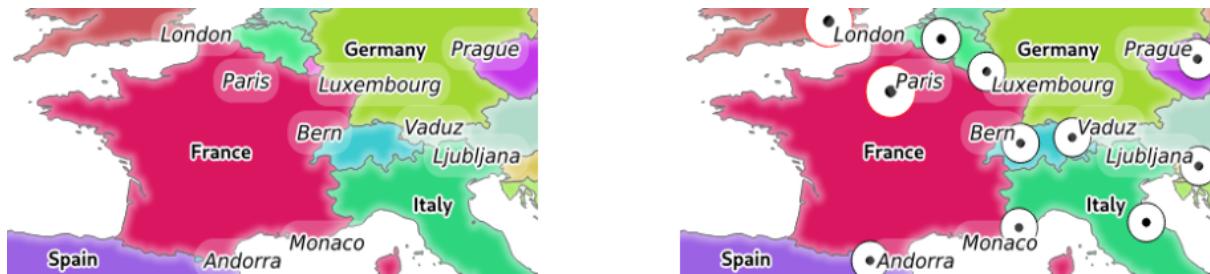


Figura 3.2: À esquerda OPACIDADES=255, 0 e à direita OPACIDADES=255, 255

## FILTRO

A subset of layers can be selected with the FILTER parameter. The syntax is basically the same as for the QGIS subset string. However, there are some restrictions to avoid SQL injections into databases via QGIS Server. If a dangerous string is found in the parameter, QGIS Server will return the next error:

```
<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>
```

Exemplo de 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'
```



Figura3.3: 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.

---

**Nota:** It is possible to make attribute searches via GetFeatureInfo and omit the X/Y parameter if a FILTER is there. QGIS Server then returns info about the matching features and generates a combined bounding box in the XML output.

---

## SELEÇÃO

The SELECTION parameter can highlight features from one or more layers. Vector features can be selected by passing comma separated lists with feature ids.

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

The following image presents the response from a GetMap request using the SELECTION option e.g. `http://myserver.com/...&SELECTION=countries:171,65`.

As those features id's correspond in the source dataset to **France** and **Romania** they're highlighted in yellow.



Figura 3.4: Server response to a GetMap request with SELECTION parameter

## FORMATO-OPÇÕES

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

- `SCALE`: to be used for symbology rules, filters and styles (not actual scaling of the data - data remains in the original scale).
- `MODE`: corresponds to the export options offered in the QGIS Desktop DXF export dialog. Possible values are `NOSYMBOLS`, `FEATURESYMBOLS` and `SYMBOLLAYERSYMBOLS`.
- `LAYERSATTRIBUTES`: specify a field that contains values for DXF layer names - if not specified, the original QGIS layer names are used.
- `USE_TITLE_AS_LAYERNAME`: if enabled, the title of the layer will be used as layer name.
- `CODEC`: specify a codec to be used for encoding. Default is `ISO-8859-1` check the QGIS desktop DXF export dialog for valid values.
- `NO_MTEXT`: Use `TEXT` instead of `MTEXT` for labels.
- `FORCE_2D`: Force 2D output. This is required for polyline width.

## TILED

For performance reasons, QGIS Server can be used in tiled mode. In this mode, the client requests several small fixed size tiles, and assembles them to form the whole map. Doing this, symbols at or near the boundary between two tiles may appear cut, because they are only present in one of the tile.

Set the `TILED` parameter to `TRUE` to tell QGIS Server to work in *tiled* mode, and to apply the *Tile buffer* configured in the QGIS project (see [Configurar seu projeto](#)).

When `TILED` is `TRUE` and when a non-zero Tile buffer is configured in the QGIS project, features outside the tile extent are drawn to avoid cut symbols at tile boundaries.

`TILED` defaults to `FALSE`.

### 3.2.3 GetFeatureInfo

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

Parâmetros	Requerido	Descrição
<code>SERVICE</code>	Sim	Nome do serviço ( <b>WMS</b> )
<code>REQUEST</code>	Sim	Name of the request ( <b>GetFeatureInfo</b> )
<code>VERSION</code>	Não	Versão do serviço
<code>QUERY_LAYERS</code>	Sim	Layers to query
<code>LAYERS</code>	Sim	Layers to display (identical to <code>QUERY_LAYERS</code> )
<code>STYLES</code>	Não	Estilo das camadas
<code>SRS / CRS</code>	Sim	Sistema de referência de coordenadas
<code>BBOX</code>	Não	Extensão do mapa
<code>WIDTH</code>	Sim	Largura da imagem em pixels
<code>HEIGHT</code>	Sim	Altura da imagem em pixels
<code>TRANSPARENTE</code>	Não	Plano de fundo transparente
<code>INFO_FORMAT</code>	Não	Formato de saída
<code>FEATURE_COUNT</code>	Não	Número máximo de feições a serem retornadas
<code>I</code>	Não	Pixel column of the point to query
<code>X</code>	Não	Igual ao parâmetro <code>I</code> , mas em WMS 1.1.1
<code>J</code>	Não	Pixel row of the point to query
<code>Y</code>	Não	Igual ao parâmetro <code>J</code> , mas em WMS 1.1.1
<code>WMS_PRECISION</code>	Não	The precision (number of digits) to be used when returning geometry (see <a href="#">how to add geometry to feature response</a> ). The default value is <code>-1</code> meaning that the precision defined in the project is used.

Além dos padrões, o QGIS Server suporta os seguintes parâmetros extras:

Parâmetros	Requerido	Descrição
<i>MAP</i>	Sim	arquivo de projeto QGIS
<i>FILTER</i>	Não	Subset of features
<i>FL_POINT_TOLERANCE</i>	Não	Tolerância em pixels para camadas de ponto
<i>FL_LINE_TOLERANCE</i>	Não	Tolerância em pixels para camadas de linha
<i>FL_POLYGON_TOLERANCE</i>	Não	Tolerância em pixels para camadas de polígono
<i>FILTER_GEOM</i>	Não	Filtragem de geometria
<i>WITH_MAPTIP</i>	Não	Add map tips to the output
<i>WITH_GEOMETRY</i>	Não	Adicionar geometria à saída

Exemplo de 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

Este parâmetro pode ser usado para especificar o formato do resultado. Os valores disponíveis são:

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

## QUERY\_LAYERS

Este parâmetro especifica as camadas a serem exibidas no mapa. Os nomes são separados por uma vírgula.

In addition, QGIS Server introduces options to select layers by:

- nome curto
- camada id

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.

Por padrão, apenas 1 feição por camada é retornada.

I

Este parâmetro, definido no WMS 1.3.0, permite especificar a coluna de pixel do ponto de consulta.

X

Mesmo parâmetro que I, mas definido no WMS 1.1.1.

J

Este parâmetro, definido no WMS 1.3.0, permite especificar a linha de pixel do ponto de consulta.

Y

Mesmo parâmetro que J, mas definido no WMS 1.1.1.

## FI\_POINT\_TOLERANCE

Este parâmetro especifica a tolerância em pixels para camadas de pontos.

## FI\_LINE\_TOLERANCE

Este parâmetro especifica a tolerância em pixels para camadas de linha.

## FI\_POLYGON\_TOLERANCE

Este parâmetro especifica a tolerância em pixels para camadas de polígono.

## FILTRAR\_GEO

This parameter specifies a WKT geometry with which features have to intersect.

## WITH\_MAPTIP

This parameter specifies whether to add map tips to the output.

Available values are (not case sensitive):

- VERDADEIRO
- FALSO

## COM\_GEOMETRIA

This parameter specifies whether to add geometries to the output. To use this feature you must first enable the *Add geometry to feature response* option in the QGIS project. See [Configure your project](#).

Available values are (not case sensitive):

- VERDADEIRO
- FALSO

### 3.2.4 GetLegendGraphics

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

Parâmetros	Requerido	Descrição
<i>SERVICE</i>	Sim	Nome do serviço ( <b>WMS</b> )
<i>REQUEST</i>	Sim	Name of the request ( <b>GetLegendGraphics</b> )
<i>VERSION</i>	Não	Versão do serviço
<i>LAYERS</i>	Sim	Layers to display
<i>STYLES</i>	Não	Estilo das camadas
<i>SRS / CRS</i>	Não	Sistema de referência de coordenadas
<i>BBOX</i>	Não	Extensão do mapa
<i>WIDTH</i>	Não	Largura da imagem em pixels
<i>HEIGHT</i>	Não	Altura da imagem em pixels
<i>FORMAT</i>	Não	Formato da legenda
<i>TRANSPARENT</i>	Não	Plano de fundo transparente

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:

Parâmetros	Requerido	Descrição
<i>MAP</i>	Sim	arquivo de projeto QGIS
<i>SRCPWIDTH</i>	Não	Largura do mapa
<i>SRCPHEIGHT</i>	Não	Altura do mapa
<i>SHOWFEATURECOUNT</i>	Não	Add feature count of features
<i>RULE</i>	Não	Rule symbol to render
<i>RULELABEL</i>	Não	Item labels rendering
<i>BOXSPACE</i>	Não	Espaço entre o quadro da legenda e o conteúdo (mm)
<i>LAYERSPACE</i>	Não	Espaço vertical entre camadas (mm)
<i>LAYERTITLESPACE</i>	Não	Espaço vertical entre o título da camada e os itens (mm)
<i>SYMBOLSPACE</i>	Não	Espaço vertical entre o símbolo e os itens (mm)
<i>ICONLABELSPACE</i>	Não	Espaço horizontal entre o símbolo e o rótulo (mm)
<i>SYMBOLWIDTH</i>	Não	Largura da pré-visualização do símbolo (mm)
<i>SYMBOLHEIGHT</i>	Não	Altura da visualização do símbolo (mm)
<i>LAYERTITLE</i>	Não	Layer title rendering
<i>LAYERFONTFAMILY</i>	Não	Layer font family
<i>LAYERFONTBOLD</i>	Não	Layer title bold rendering
<i>LAYERFONTSIZE</i>	Não	Tamanho da fonte do título da camada (pt)
<i>LAYERFONTITALIC</i>	Não	Layer title italic rendering
<i>LAYERFONTCOLOR</i>	Não	Cor do título da camada
<i>ITEMFONTFAMILY</i>	Não	Item font family
<i>ITEMFONTBOLD</i>	Não	Item label bold rendering
<i>ITEMFONTSIZE</i>	Não	Tamanho da fonte do rótulo do item (pt)
<i>ITEMFONTITALIC</i>	Não	Item label italic rendering
<i>ITEMFONTCOLOR</i>	Não	Cor do rótulo do item

## 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.

Exemplo de URL:

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

---

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

---

## LARGURA

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=GetLegendGraphics
&LAYER=testlayer%20%C3%A8%C3%A9
&RULE=rule1
&WIDTH=30
&HEIGHT=30
```

## ALTURA

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=GetLegendGraphics
&LAYER=testlayer%20%C3%A8%C3%A9
&RULE=rule1
&WIDTH=30
&HEIGHT=30
```

## FORMATO

Este parâmetro pode ser usado para especificar o formato da imagem da legenda. Os valores disponíveis são:

- imagem/jpeg
- imagem/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=GetLegendGraphics&  
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 GetLegendGraphics is used for the image legend size.

## SRCHEIGHT

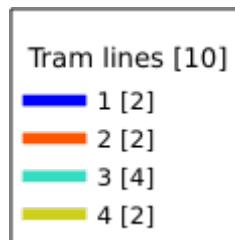
This parameter may be defined when the RULE parameter is set. In this case, the SRCHEIGHT value is forwarded to the underlying GetMap request as the HEIGHT parameter while the HEIGHT parameter of GetLegendGraphics is used for the image legend size.

## SHOWFEATURECOUNT

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

- VERDADEIRO
- FALSO

Por exemplo:



## REGRA

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 nor the JSON format.

Exemplo de URL:

```
http://localhost/qgisserver?
SERVICE=WMS
&REQUEST=GetLegendGraphics
&LAYERS=mylayer,
&RULE=myrulename
```

## RULELABEL

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

- `VERDADEIRO`: exibe o rótulo do item
- `FALSO`: oculta o rótulo do item
- `AUTO`: hide item label for layers with *Single symbol* rendering

Exemplo de URL:

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

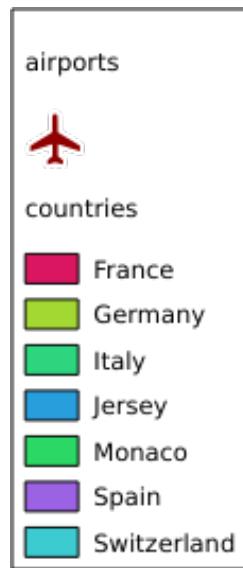


Figura3.5: Legend rendering without label for single symbol layers

## BOXSPACE

Este parâmetro permite especificar o espaço entre o quadro da legenda e o conteúdo em milímetros. Por padrão, o valor do espaço é 2 mm.

Exemplo de URL:

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



Figura3.6: To the left `BOXSPACE=0` and to the right `BOXSPACE=15`

## ESPAÇOCAMADA

Este parâmetro permite especificar o espaço vertical entre as camadas em milímetros. Por padrão, o valor do espaço é 3 mm.

Exemplo de URL:

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

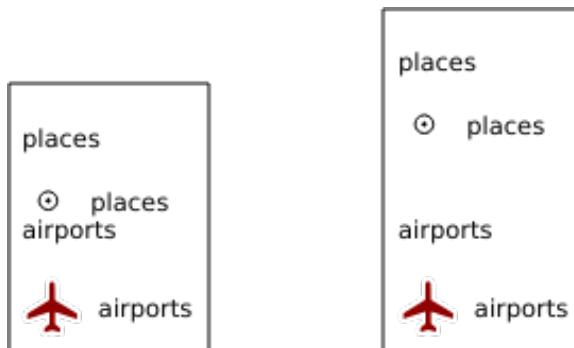


Figura3.7: To the left `LAYERSPACE=0` and to the right `LAYERSPACE=10`

## LAYERTITLESPACE

Este parâmetro permite especificar o espaço vertical entre o título da camada e os itens seguintes em milímetros. Por padrão, o valor do espaço é 3 mm.

Exemplo de URL:

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

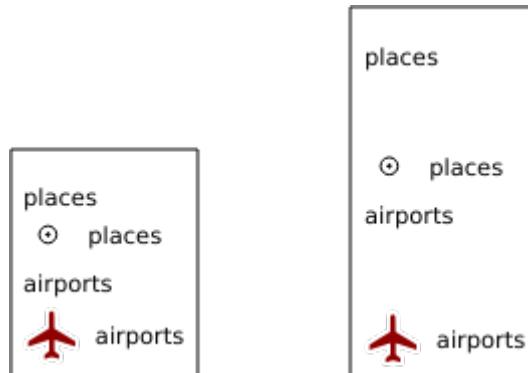


Figura 3.8: To the left LAYERTITLESPACE=0 and to the right LAYERTITLESPACE=10

## ESPAÇOSÍMBOLO

Este parâmetro permite especificar o espaço vertical entre o símbolo e o item seguinte em milímetros. Por padrão, o valor do espaço é 2 mm.

Exemplo de URL:

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



Figura3.9: To the left SYMBOLSPACE=0 and to the right SYMBOLSPACE=5

## ICONLABELSPACE

Este parâmetro permite especificar o espaço horizontal entre o símbolo e o texto do rótulo em milímetros. Por padrão, o valor do espaço é 2 mm.

Exemplo de URL:

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

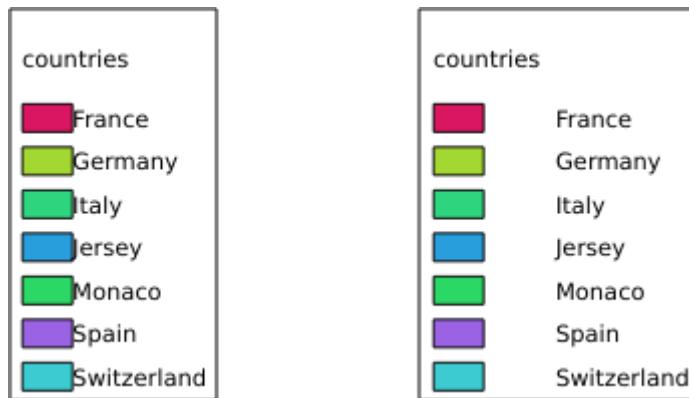


Figura3.10: To the left ICONLABELSPACE=0 and to the right ICONLABELSPACE=10

## LARGURASÍMBOLO

Este parâmetro permite especificar a largura da visualização do símbolo em milímetros. Por padrão, o valor da largura é 7 mm.

Exemplo de URL:

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



Figura 3.11: To the left SYMBOLWIDTH=2 and to the right SYMBOLWIDTH=20

## ALTURASÍMBOLO

Este parâmetro permite especificar a altura da visualização do símbolo em milímetros. Por padrão, o valor da altura é 4 mm.

Exemplo de URL:

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



Figura3.12: To the left SYMBOLHEIGHT=2 and to the right SYMBOLHEIGHT=6

## TITULOCAMADA

This parameter specifies whether to render layer title.

Available values are (not case sensitive):

- VERDADEIRO (valor padrão)
- FALSO

## LAYERFONTFAMILY

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

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

## CAMADAFONTENEGRITO

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

- VERDADEIRO
- FALSO

Exemplo de URL:

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



Figura3.13: Legend with LAYERFONTBOLD=TRUE

## CAMADAFONTETAMANHO

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

Exemplo de URL:

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



Figura3.14: Legend with LAYERFONTSIZE=20

## CAMADAFONTEITÁLICO

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

- VERDADEIRO
- FALSO

Exemplo de URL:

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

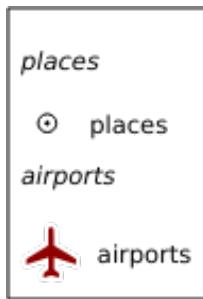


Figura3.15: Legend with LAYERFONTITALIC=TRUE

## CAMADAFONTECOR

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

Exemplo de URL:

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



Figura3.16: Legend with LAYERFONTCOLOR=0x5f9930

## ITEMFONTFAMILY

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

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

## ITEMFONTENEGRO

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

- VERDADEIRO
- FALSO

Exemplo de URL:

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



Figura3.17: Legend with ITEMFONTBOLD=TRUE

## ITEMFONTEMANHO

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

Exemplo de URL:

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

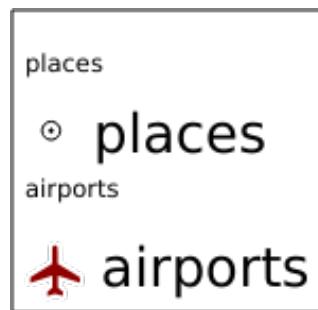


Figura3.18: Legend with ITEMFONTSIZE=30

## ITEMFONTEITALICO

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

- VERDADEIRO
- FALSO

Exemplo de URL:

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

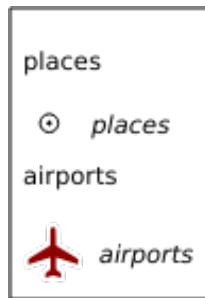


Figura 3.19: Legend with ITEMFONTITALIC=TRUE

## ITEMFONTECOR

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

Exemplo de URL:

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



Figura 3.20: Legend with ITEMFONTCOLOR=0x5f9930

### 3.2.5 GetStyle(s)

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

Parâmetros	Requerido	Descrição
<i>SERVICE</i>	Sim	Nome do serviço (WMS)
<i>REQUEST</i>	Sim	Name of the request ( <b>GetStyle</b> or <b>GetStyles</b> )
<i>LAYERS</i>	Sim	Layers to query

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

Parâmetros	Requerido	Descrição
<i>MAP</i>	Sim	arquivo de projeto QGIS

Exemplo de 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:

Parâmetros	Requerido	Descrição
<i>SERVICE</i>	Sim	Nome do serviço (WMS)
<i>REQUEST</i>	Sim	Name of the request ( <b>DescribeLayer</b> )
<i>LAYERS</i>	Sim	Camadas para descrever
<i>SLD_VERSION</i>	Sim	Versão SLD

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

Parâmetros	Requerido	Descrição
<i>MAP</i>	Sim	arquivo de projeto QGIS

Exemplo de URL:

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

The XML document looks like:

```
<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/qgis_server" xlink:type="simple"/>
```

(continua na próxima página)

(continuação da página anterior)

```

<TypeName>
  <se:FeatureTypeName>my_vector_layer</se:FeatureTypeName>
</TypeName>
</LayerDescription>
<LayerDescription>
  <owsType>wcs</owsType>
  <se:OnlineResource xlink:href="http://localhost/qgis_server" xlink:type=
→"simple"/>
  <TypeName>
    <se:FeatureTypeName>my_raster_layer</se:FeatureTypeName>
  </TypeName>
</LayerDescription>
</DescribeLayerResponse>

```

## VERSÃO\_SLD

Este parâmetro permite especificar a versão do SLD. Apenas o valor 1.1.0 está disponível.

### 3.2.7 GetPrint

QGIS Server has the capability to create print layout output in pdf or pixel format. Print layout windows in the published project are used as templates. In the **GetPrint** request, the client has the possibility to specify parameters of the contained layout maps and labels.

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

Parâmetros	Requerido	Descrição
<i>MAP</i>	Sim	arquivo de projeto QGIS
<i>SERVICE</i>	Sim	Nome do serviço ( <b>WMS</b> )
<i>REQUEST</i>	Sim	Name of the request ( <b>GetPrint</b> )
<i>VERSION</i>	Não	Versão do serviço
<i>LAYERS</i>	Não	Layers to display
<i>TEMPLATE</i>	Sim	Layout template to use
<i>SRS / CRS</i>	Sim	Sistema de referência de coordenadas
<i>FORMAT</i>	Não	Formato de saída
<i>ATLAS_PK</i>	Não	Feições Atlas
<i>STYLES</i>	Não	Estilo das camadas
<i>TRANSPARENT</i>	Não	Plano de fundo transparente
<i>OPACITIES</i>	Não	Opacidade para camada ou grupo
<i>SELECTION</i>	Não	Highlight features
<i>mapX:EXTENT</i>	Não	Extensão do mapa 'X'
<i>mapX:LAYERS</i>	Não	Camadas do mapa 'X'
<i>mapX:STYLES</i>	Não	Layers' style of the map 'X'
<i>mapX:SCALE</i>	Não	Layers' scale of the map 'X'
<i>mapX:ROTATION</i>	Não	Rotação do mapa 'X'
<i>mapX:GRID_INTERVAL_X</i>	Não	Intervalo da grade no eixo y do mapa 'X'
<i>mapX:GRID_INTERVAL_Y</i>	Não	Intervalo da grade no eixo y do mapa 'X'

Exemplo de URL:

```

http://localhost/qgisserver?
SERVICE=WMS
&VERSION=1.3.0
&REQUEST=GetPrint
&MAP=/home/qgis/projects/world.qgs

```

(continua na próxima página)

(continuação da página anterior)

```
&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
```

Note that the layout template may contain more than one map. In this way, if you want to configure a specific map, you have to use `mapX:` parameters where X is a positive number that you can retrieve thanks to the **GetProjectSettings** request.

Por exemplo:

```
<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>
```

## MODELO

This parameter can be used to specify the name of a layout template to use for printing.

## FORMATO

Este parâmetro especifica o formato da imagem do mapa. Os valores disponíveis são:

- png (valor padrão)
- imagem/png
- jpg
- jpeg
- imagem/jpeg
- svg
- imagem/svg
- image/svg+xml
- pdf
- application/pdf

Se o parâmetro FORMATO for diferente de um desses valores, uma exceção será retornada.

## **ATLAS\_PK**

This parameter allows activation of Atlas rendering by indicating which features we want to print. In order to retrieve an atlas with all features, the \* symbol may be used (according to the maximum number of features allowed in the project configuration).

When FORMAT is pdf, a single PDF document combining the feature pages is returned. For all other formats, a single page is returned.

## **mapX:EXTENT**

This parameter specifies the extent for a layout map item as xmin,ymin,xmax,ymax.

## **mapX:ROTATION**

Este parâmetro especifica a rotação do mapa em graus.

## **mapX:GRID\_INTERVAL\_X**

This parameter specifies the grid line density in the X direction.

## **mapX:GRID\_INTERVAL\_Y**

Este parâmetro especifica a densidade da linha de grade na direção Y.

## **mapaX:ESCALA**

This parameter specifies the map scale for a layout map item. This is useful to ensure scale based visibility of layers and labels even if client and server may have different algorithms to calculate the scale denominator.

## **mapaX:CAMADAS**

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

## **mapaX:ESTILOS**

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:

- visibilidade inicial das camadas
- informações sobre atributos vetoriais e seus tipos de edição
- information about layer order and drawing order
- list of layers published in WFS
- show if a group in the layer tree is mutually exclusive

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

Parâmetros	Requerido	Descrição
<i>MAP</i>	Sim	arquivo de projeto QGIS
<i>SERVICE</i>	Sim	Nome do serviço (WMS)
<i>REQUEST</i>	Sim	Name of the request ( <b>GetProjectSettings</b> )

### 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.

Parâmetros	Requerido	Descrição
<i>MAP</i>	Sim	arquivo de projeto QGIS
<i>SERVICE</i>	Sim	Nome do serviço (WMS)
<i>REQUEST</i>	Sim	Name of the request ( <b>GetSchemaExtension</b> )

Exemplo de URL:

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

The XML document looks like:

```
<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 External WMS layers

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
```

(continua na próxima página)

(continuação da página anterior)

```
&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.

The redlining feature can be used to pass geometries and labels in the request which are overlapped by the server over the standard returned image (map). This permits the user to put emphasis or maybe add some comments (labels) to some areas, locations etc. that are not in the standard map.

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></
˓→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
```

(continua na próxima página)

(continuação da página anterior)

```
&map0:HIGHLIGHT_LABELSIZE=16
&map0:HIGHLIGHT_LABELCOLOR=%23000000
&map0:HIGHLIGHT_LABELBUFFERCOLOR=%23FFFFFF
&map0:HIGHLIGHT_LABELBUFFERSIZE=1.5
```

Here is the image outputted by the above request in which a polygon and a label are drawn on top of the normal map:



Figura 3.21: Server response to a GetMap request with redlining parameters

Você pode ver que há vários parâmetros nesta solicitação:

- **HIGHLIGHT\_GEO**: You can add POINT, MULTILINESTRING, POLYGON etc. It supports multipart geometries. Here is an example: HIGHLIGHT\_GEO=MULTILINESTRING((0 0, 0 1, 1 1)). The coordinates should be in the CRS of the GetMap/GetPrint request.
- **HIGHLIGHT\_SYMBOL**: This controls how the geometry is outlined and you can change the stroke width, color and opacity.
- **HIGHLIGHT\_LABELSTRING**: You can pass your labeling text to this parameter.
- **HIGHLIGHT\_LABELSIZE**: This parameter controls the size of the label.
- **HIGHLIGHT\_LABELFONT**: This parameter controls the font of the label (e.g. Arial)
- **HIGHLIGHT\_LABELCOLOR**: This parameter controls the label color.
- **HIGHLIGHT\_LABELBUFFERCOLOR**: This parameter controls the label buffer color.

- **HIGHLIGHT\_LABELBUFFERSIZE**: This parameter controls the label buffer size.

## 3.3 Web Feature Service (WFS)

The **1.0.0** and **1.1.0** WFS standards implemented in QGIS Server provide a HTTP interface to query geographic features from a QGIS project. A typical WFS request defines the QGIS project to use and the layer to query.

Documento de especificações de acordo com o número da versão do serviço:

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

Requisitos padrões fornecidos pelo Servidor QGIS:

Solicitação	Descrição
<i>GetCapabilities</i>	Retorna metadados XML com informações sobre o servidor
<i>GetFeature</i>	Retorna uma seleção de feições
<i>DescribeFeatureType</i>	Retorna uma descrição dos tipos de características e propriedades
<i>Transaction</i>	Permite que feições sejam inseridas, atualizadas ou excluídas

### 3.3.1 GetCapabilities

Standard parameters for the **GetCapabilities** request according to the OGC WFS 1.0.0 and 1.1.0 specifications:

Parâmetros	Requerido	Descrição
<i>SERVICE</i>	Sim	Nome do serviço ( <b>WFS</b> )
<i>REQUEST</i>	Sim	Nome da requisição ( <b>GetCapabilities</b> )
<i>VERSION</i>	Não	Versão do serviço

Além dos padrões, o QGIS Server suporta os seguintes parâmetros extras:

Parâmetros	Requerido	Descrição
<i>MAP</i>	Sim	arquivo de projeto QGIS

### VERSION

Este parâmetro permite especificar a versão do serviço a ser utilizada. Os valores disponíveis para o parâmetro VERSÃO são:

- 1.0.0
- 1.1.0

Se nenhuma versão for indicada na solicitação, 1.1.0 será usado por padrão.

Exemplo de URL:

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

### 3.3.2 GetFeature

Parâmetros padrão para a solicitação **GetFeature** de acordo com as especificações OGC WFS 1.0.0 e 1.1.0:

Parâmetros	Requerido	Descrição
<i>SERVICE</i>	Sim	Nome do serviço ( <b>WFS</b> )
<i>REQUEST</i>	Sim	Nome da requisição ( <b>GetFeature</b> )
<i>VERSION</i>	Não	Versão do serviço
<i>TYPENAME</i>	Não	Nome das camadas
<i>FEATUREID</i>	Não	Filtrar as feições por ids
<i>OUTPUTFORMAT</i>	Não	Formato de Saída
<i>RESULTTYPE</i>	Não	Tipo do resultado
<i>PROPERTYNAME</i>	Não	Nome das propriedades a serem retornadas
<i>MAXFEATURES</i>	Não	Número máximo de feições a serem retornadas
<i>SRSNAME</i>	Não	Sistema de referência de coordenadas
<i>FILTER</i>	Não	Codificação de filtros OGC
<i>BBOX</i>	Não	Extensão do Mapa
<i>SORTBY</i>	Não	Classificar os resultados

Além dos padrões, o QGIS Server suporta os seguintes parâmetros extras:

Parâmetros	Requerido	Descrição
<i>MAP</i>	Sim	arquivo de projeto QGIS
<i>STARTINDEX</i>	Não	Paginação
<i>GEOMETRYNAME</i>	Não	Tipo de geometria para retornar
<i>EXP_FILTER</i>	Não	Filtragem de expressão

#### TYPENAME

Este parâmetro permite especificar nomes de camadas e é obrigatório se ID DA FEIÇÃO não estiver definido.

Exemplo de URL:

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

#### FEATUREID

Este parâmetro permite especificar a identificação de uma característica específica e é formado como `^type-name fid,typename fid,...^`.

Exemplo de URL:

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

Resposta em 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%3Dgml/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

Este parâmetro pode ser usado para especificar o formato da resposta. Se “VERSÃO” for maior ou igual a “1.1.0”, GML3 é o formato padrão. Caso contrário, é utilizado GML2.

Os valores disponíveis são:

- gml2
- texto/xml; subtipo=gml/2.1.2
- gml3
- texto/xml; subtipo=gml/3.1.1
- geojson
- application/vnd.geo+json,
- application/vnd.geo json
- application/geo+json
- application/geo json
- aplicação/json

Exemplo de URL:

```

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

```

Resposta de GeoJSON:

```
{
  "type": "FeatureCollection",
  "bbox": [
    -180,
    100
  ],
  "features": [
    {
      "type": "Feature"
    }
  ]
}
```

(continua na próxima página)

(continuação da página anterior)

```

    -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

Este parâmetro pode ser usado para especificar o tipo de resultado a ser retornado. Os valores disponíveis são:

- resultados: o comportamento padrão
- “acertos”: retorna apenas uma contagem de recursos

Exemplo de URL:

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

## PROPERTYNAME

This parameter may be used to specify a specific property to return. A property needs to be mapped with a TYPE-NAME or a FEATUREID:

Exemplo de URL válido:

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

Pelo contrário, a próxima URL retornará uma exceção:

```
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

Este parâmetro permite limitar o número de feições devolvidas pelo pedido.

---

**Nota:** Este parâmetro pode ser útil para melhorar o desempenho quando as camadas vetoriais subjacentes são pesadas.

---

## SRSNAME

This parameter allows to indicate the response output Spatial Reference System as well as the BBOX CRS and has to be formed like EPSG:XXXX.

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

## FILTER

Este parâmetro permite filtrar a resposta com a linguagem **Codificação de filtro** definida pelo [Codificação de filtro padrão OGC](#).

Exemplo de URL:

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

## BBOX

Este parâmetro permite especificar a extensão do mapa com unidades de acordo com o SRC atual. As coordenadas devem ser separadas por vírgula.

The SRSNAME parameter may specify the CRS of the extent. If not specified, the CRS of the layer is used.

Exemplo de URL:

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

The FEATUREID parameter cannot be used with the BBOX. Any attempt will result in an 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

Este parâmetro permite ordenar as características resultantes de acordo com os valores de propriedade e deve ser formado como nomPropriedade REGRA DE ORGANIZAÇÃO.

Valores disponíveis para REGRA DE ORGANIZAÇÃO em caso de triagem descendente:

- D
- +D
- DESC
- +DESC

Valores disponíveis para REGRA DE ORGANIZAÇÃO em caso de triagem ascendente:

- A
- +A
- ASC
- +ASC

Exemplo de URL:

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

O resultado correspondente:

```
<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>
```

(continua na próxima página)

(continuação da página anterior)

```

</qgs:places>
</gml:featureMember>
<gml:featureMember>
  <qgs:places gml:id="places.113">
    <qgs:name>Yerevan</qgs:name>
  </qgs:places>
</gml:featureMember>
<gml:featureMember>
  <qgs:places gml:id="places.111">
    <qgs:name>Yaounde</qgs:name>
  </qgs:places>
</gml:featureMember>
</wfs:FeatureCollection>

```

## GEOMETRYNAME

Este parâmetro pode ser usado para especificar o tipo de geometria a ser retornada para as feições. Os valores disponíveis são:

- extent
- centroid
- none

Exemplo de URL:

```

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

```

## STARTINDEX

Este parâmetro é padrão no WFS 2.0, mas é uma extensão para o WFS 1.0.0.

Na verdade, ele pode ser utilizado para pular algumas características no conjunto de resultados e, em combinação com “MAXFEIÇÕES”, ele fornece a capacidade de paginar através de resultados.

Exemplo de URL:

```

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

```

## EXP\_FILTER

Este parâmetro permite filtrar a resposta com expressões QGIS.

Exemplo de URL:

```
http://localhost/qgisserver?
SERVICE=WFS&
REQUEST=GetFeature&
TYPENAME=places&
EXP_FILTER="name"='Paris'
```

### 3.3.3 DescribeFeatureType

Standard parameters for the **DescribeFeatureType** request according to the OGC WFS 1.0.0 and 1.1.0 specifications:

Parâmetros	Requerido	Descrição
<i>SERVICE</i>	Sim	Nome do serviço ( <b>WFS</b> )
<i>REQUEST</i>	Sim	Name of the request ( <b>DescribeFeatureType</b> )
<i>VERSION</i>	Não	Versão do serviço
<i>OUTPUTFORMAT</i>	Não	Formato da resposta
<i>TYPENAME</i>	Não	Nome das camadas

Além dos padrões, o QGIS Server suporta os seguintes parâmetros extras:

Parâmetros	Requerido	Descrição
<i>MAP</i>	Sim	arquivo de projeto QGIS

Exemplo de URL:

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

Resposta de saída:

```
<schema xmlns:ogc="http://www.opengis.net/ogc" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://www.w3.org/2001/XMLSchema" xmlns:qgs="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="qgs: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>
    </element>
  </schema>
```

(continua na próxima página)

(continuação da página anterior)

```
</complexType>
</schema>
```

### 3.3.4 Transação

Esta solicitação permite atualizar, apagar ou adicionar uma ou várias características graças a um documento XML. A ação *deletar* pode ser realizada com uma solicitação POST, bem como com o parâmetro *OPERAÇÃO* enquanto que as operações *adicionar* e a *atualizar* podem ser realizadas apenas através de uma solicitação POST.

Parâmetros padrão para a solicitação da **Transação** de acordo com as especificações OGC WFS 1.0.0 e 1.1.0:

Parâmetros	Requerido	Descrição
<i>SERVICE</i>	Sim	Nome do serviço ( <b>WFS</b> )
<i>REQUEST</i>	Sim	Nome da solicitação ( <b>Transação</b> )
<i>VERSION</i>	Não	Versão do serviço
<i>FILTER</i>	Não	Codificação de filtros OGC
<i>BBOX</i>	Não	Extensão do Mapa
<i>FEATUREID</i>	Não	Filtrar as feições por ids
<i>TYPENAME</i>	Não	Nome das camadas

Além dos padrões, o QGIS Server suporta os seguintes parâmetros extras:

Parâmetros	Requerido	Descrição
<i>MAP</i>	Sim	arquivo de projeto QGIS
<i>OPERATION</i>	Não	Especifique a operação
<i>EXP_FILTER</i>	Não	Filtragem de expressão

## OPERAÇÃO

Este parâmetro permite excluir um recurso sem utilizar um pedido POST com um documento XML dedicado.

Exemplo de URL:

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

**Nota:** FEATUREID, BBOX and FILTER parameters are mutually exclusive and prioritized in this order.

## Adicionar feições

Exemplo de solicitação de POST:

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

com o documento *adiciona.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: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>
```

## Atualizar feições

Exemplo de solicitação de POST:

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

com o documento *atualiza.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>
```

## Excluir feições

Exemplo de solicitação de POST:

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

com o documento *deleta.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: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.

Especificações:

- WCS 1.0.0
- WCS 1.1.1

Standard requests provided by QGIS Server:

Solicitação	Descrição
<i>GetCapabilities</i>	Returns XML metadata with information about the server
<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:

Parâmetros	Requerido	Descrição
<i>SERVICE</i>	Sim	Name of the service ( <b>WCS</b> )
<i>REQUEST</i>	Sim	Name of the request ( <b>GetCapabilities</b> )
<i>VERSION</i>	Não	Versão do serviço

Além dos padrões, o QGIS Server suporta os seguintes parâmetros extras:

Parâmetros	Requerido	Descrição
<i>MAP</i>	Sim	Arquivo de projeto QGIS

Exemplo de URL:

```

http://localhost/qgis_server?
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>

```

## VERSÃO

Este parâmetro permite especificar a versão do serviço a ser utilizada. Atualmente, os valores de versão não são usados internamente e sempre retornam para 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:

Parâmetros	Requerido	Descrição
<i>SERVICE</i>	Sim	Name of the service ( <b>WCS</b> )
<i>REQUEST</i>	Sim	Name of the request ( <b>DescribeCoverage</b> )
<i>VERSION</i>	Não	Versão do serviço
<i>COVERAGE</i>	Não	Specify coverage layers (WCS 1.0.0)
<i>IDENTIFIER</i>	Não	Especificar camadas de cobertura (WCS 1.1.1)

Além dos padrões, o QGIS Server suporta os seguintes parâmetros extras:

Parâmetros	Requerido	Descrição
<i>MAP</i>	Sim	Arquivo de projeto QGIS

O documento XML para uma camada raster GeoTIFF de 3 bandas se parece com:

```

<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>
    <name>T20QPD_20171123T144719_TCI</name>
    <label>T20QPD_20171123T144719_TCI</label>

```

(continua na próxima página)

(continuação da página anterior)

```

<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>

```

## COBERTURA

Este parâmetro, definido no WCS 1.0.0, permite especificar as camadas a serem consultadas por informações adicionais. Os nomes devem ser separados por vírgula.

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

```

---

**Nota:** COVERAGE is mandatory if IDENTIFIER is not set.

---

## IDENTIFICADOR

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.

**Nota:** `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:

Parâmetros	Requerido	Descrição
<code>SERVICE</code>	Sim	Name of the service ( <b>WCS</b> )
<code>REQUEST</code>	Sim	Name of the request ( <b>GetCoverage</b> )
<code>VERSION</code>	Não	Versão do serviço
<code>COVERAGE</code>	Não	Specify coverage layers (WCS 1.0.0)
<code>IDENTIFIER</code>	Não	Especificar camadas de cobertura (WCS 1.1.1)
<code>WIDTH</code>	Sim	Largura da resposta em pixels
<code>HEIGHT</code>	Sim	Altura da resposta em pixels
<code>BBOX</code>	Sim	Extensão do mapa em unidades SRC
<code>CRS</code>	Sim	Sistema de Referência de Coordenadas da extensão
<code>RESPONSE_CRS</code>	Não	Sistema de referência de coordenadas da resposta

Além dos padrões, o QGIS Server suporta os seguintes parâmetros extras:

Parâmetros	Requerido	Descrição
<code>MAP</code>	Sim	Arquivo de projeto 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`.

Exemplo de URL:

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

**SRC**

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.

**LARGURA**

Este parâmetro permite especificar a largura em pixels da imagem de saída. A resolução da imagem de resposta depende desse valor.

**ALTURA**

Este parâmetro permite especificar a altura em pixels da imagem de saída. A resolução da imagem de resposta depende desse valor.



Figura3.22: From left to right: WIDTH=20&HEIGHT=20, WIDTH=50&HEIGHT=50, WIDTH=100&HEIGHT=100

## 3.5 Web Map Tile Service (WMTS)

The **1.0.0** WMTS standard implemented in QGIS Server provides a HTTP interface to request tiled map images generated from a QGIS project. A typical WMTS request defined the QGIS project to use, some WMS parameters like layers to render, as well as tile parameters.

Documento de especificações do serviço:

- [WMTS 1.0.0](#)

Standard requests provided by QGIS Server:

Solicitação	Descrição
<a href="#">GetCapabilities</a>	Returns XML metadata with information about the server
<a href="#">GetTile</a>	Retorna um mosaico
<a href="#">GetFeatureInfo</a>	Recupera dados (geometria e valores) para um local de pixel

### 3.5.1 GetCapabilities

Standard parameters for the **GetCapabilities** request according to the OGC WMTS 1.0.0 specifications:

Parâmetros	Requerido	Descrição
<i>SERVICE</i>	Sim	Name of the service ( <b>WMTS</b> )
<i>REQUEST</i>	Sim	Name of the request ( <b>GetCapabilities</b> )

In addition to the standard ones, QGIS Server supports the following extra parameters:

Parâmetros	Requerido	Descrição
<i>MAP</i>	Sim	arquivo de projeto QGIS

URL example:

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

### 3.5.2 GetTile

Standard parameters for the **GetTile** request according to the OGC WMTS 1.0.0 specifications:

Parâmetros	Requerido	Descrição
<i>SERVICE</i>	Sim	Name of the service ( <b>WMTS</b> )
<i>REQUEST</i>	Sim	Name of the request ( <b>GetTile</b> )
<i>LAYER</i>	Sim	Identificador de camada
<i>FORMAT</i>	Sim	Output format of the tile
<i>TILEMATRIXSET</i>	Sim	Nome da pirâmide
<i>TILEMATRIX</i>	Sim	Meshing
<i>TILEROW</i>	Sim	Coordenada de linha na malha
<i>TILECOL</i>	Sim	Coordenada da coluna na malha

In addition to the standard ones, QGIS Server supports the following extra parameters:

Parâmetros	Requerido	Descrição
<i>MAP</i>	Sim	arquivo de projeto QGIS

URL example:

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

## FORMATO

Este parâmetro pode ser usado para especificar o formato da imagem do bloco. Os valores disponíveis são:

- jpg
- jpeg
- imagem/jpeg
- imagem/png

Se o parâmetro FORMATO for diferente de um desses valores, então o formato padrão PNG será usado.

## TILEMATRIXSET

This parameter defines the CRS to use when computing the underlying pyramid. Format: EPSG:XXXX.

## TILEMATRIX

This parameter allows to define the matrix to use for the output tile.

## TILEROW

This parameter allows to select the row of the tile to get within the matrix.

## TILECOL

This parameter allows to select the column of the tile to get within the matrix.

### 3.5.3 GetFeatureInfo

Standard parameters for the **GetFeatureInfo** request according to the OGC WMTS 1.0.0 specification:

Parâmetros	Requerido	Descrição
<i>SERVICE</i>	Sim	Name of the service ( <b>WMTS</b> )
<i>REQUEST</i>	Sim	Name of the request ( <b>GetFeatureInfo</b> )
<i>LAYER</i>	Sim	Identificador de camada
<i>INFOFORMAT</i>	Não	Formato de saída
<i>I</i>	Não	Coordenada X de um pixel
<i>J</i>	Não	Coordenada Y de um pixel
<i>TILEMATRIXSET</i>	Sim	Nome da pirâmide
<i>TILEMATRIX</i>	Meshing	
<i>TILEROW</i>	Sim	Coordenada de linha na malha
<i>TILECOL</i>	Sim	Coordenada da coluna na malha

In addition to the standard ones, QGIS Server supports the following extra parameters:

Parâmetros	Requerido	Descrição
<i>MAP</i>	Sim	arquivo de projeto QGIS

URL example:

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

## INFOFORMAT

Este parâmetro permite definir o formato de saída do resultado. Os valores disponíveis são:

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

The default value is `text/plain`.

## I

Este parâmetro permite definir a coordenada X do pixel para o qual queremos recuperar a informação subjacente.

## J

Este parâmetro permite definir a coordenada Y do pixel para o qual queremos recuperar a informação subjacente.

## 3.6 Feições OGC API

OGC API Features (OAPIF) is the first implementation of the new generation of OGC protocols. It is described by the [OGC API - Features - Part 1: Core](#) document.

Aqui está um rápido resumo informal das diferenças mais importantes entre o bem conhecido protocolo WFS e a OAPIF:

- OAPIF é baseado em uma [API REST](#)
- OAPIF deve seguir as [OPENAPI](#) especificações
- OAPIF supports multiple output formats but it does not dictate any (only GeoJSON and HTML are currently available in QGIS OAPIF) and it uses [content negotiation](#) to determine which format is to be served to the client
- JSON e HTML são cidadãos de primeira classe na OAPIF
- OAPIF é autodocumentação (através do endpoint `/api`)
- OAPIF é totalmente navegável (através de links) e buscável

---

**Importante:** While the OGC API Features implementation in QGIS can make use of the `MAP` parameter to specify the project file, no extra query parameters are allowed by the OPENAPI specification. For this reason it is strongly recommended that `MAP` is not exposed in the URL and the project file is specified in the environment by other means (i.e. setting `QGIS_PROJECT_FILE` in the environment through a web server rewrite rule).

---

**Nota:** The **API** endpoint provides comprehensive documentation of all supported parameters and output formats of your service. The following paragraphs will only describe the most important ones.

---

### 3.6.1 Representação de recursos

The implementation of OGC API Features in QGIS Server currently supports the following resource representation (output) formats:

- HTML
- JSON

The format that is actually served will depend on content negotiation, but a specific format can be explicitly requested by appending a format specifier to the endpoints.

Supported format specifier extensions are:

- `.json`
- `.html`

Additional format specifier aliases may be defined by specific endpoints:

- `.openapi`: alias for `.json` supported by the **API** endpoint
- `.geojson`: alias for `.json` supported by the **Features** and **Feature** endpoints

### 3.6.2 Endpoints

The API provides a list of endpoints that the clients can retrieve. The system is designed in such a way that every response provides a set of links to navigate through all the provided resources.

Endpoints points provided by the QGIS implementation are:

Nome	Caminho	Descrição
Landing Page	/	General information about the service and provides links to all available endpoints
Con-formi-dade	/conformidade	Informações sobre a conformidade do serviço com as normas
API	/api	Full description of the endpoints provided by the service and the returned documents structure
Coleções	/coleções	List of all collections (i.e. ‘vector layers’) provided by the service
Coleção	/collections/{collectionId}	Informações sobre uma coleção (nome, metadados, extensão etc.)
Feições	/collections/{collectionId}/items	Lista das feições fornecidas pela coleção
Feição	/collections/{collectionId}/items/{featureId}	Informações sobre uma feição única

## Landing Page

The main endpoint is the **Landing Page**. From that page it is possible to navigate to all the available service endpoints. The **Landing Page** must provide links to

- the API definition (path `/api` link relations `service-desc` and `service-doc`),
- the Conformance declaration (path `/conformance`, link relation `conformance`), and
- the Collections (path `/collections`, link relation `data`).

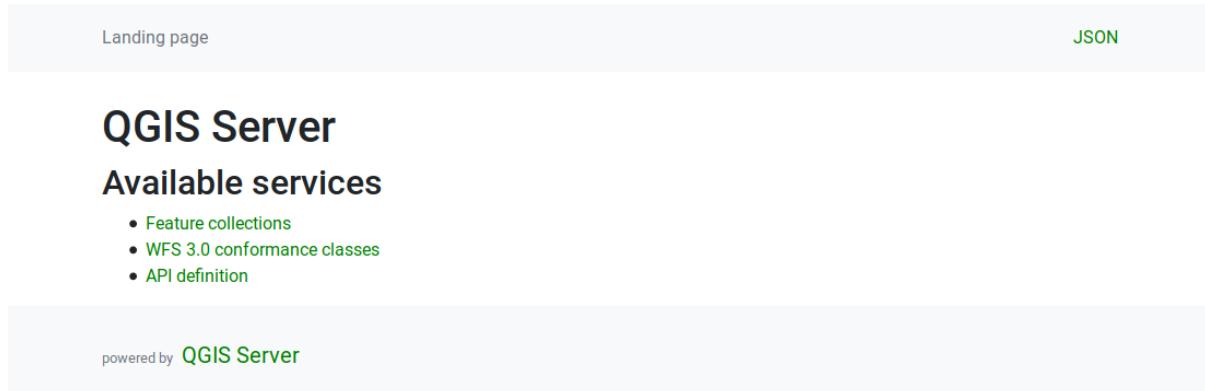


Figura3.23: Página principal do servidor OAPIF

## API Definition

The **API Definition** is an OPENAPI-compliant description of the API provided by the service. In its HTML representation it is a browsable page where all the endpoints and their response formats are accurately listed and documented. The path of this endpoint is `/api`.

The API definition provides a comprehensive and authoritative documentation of the service, including all supported parameters and returned formats.

---

**Nota:** This endpoint is analogue to WFS's GetCapabilities

---

## Lista de coleções

The collections endpoint provides a list of all the collections available in the service. Since the service “serves” a single QGIS project the collections are the vector layers from the current project (if they were published as WFS in the project properties). The path of this endpoint is `/collections/`.

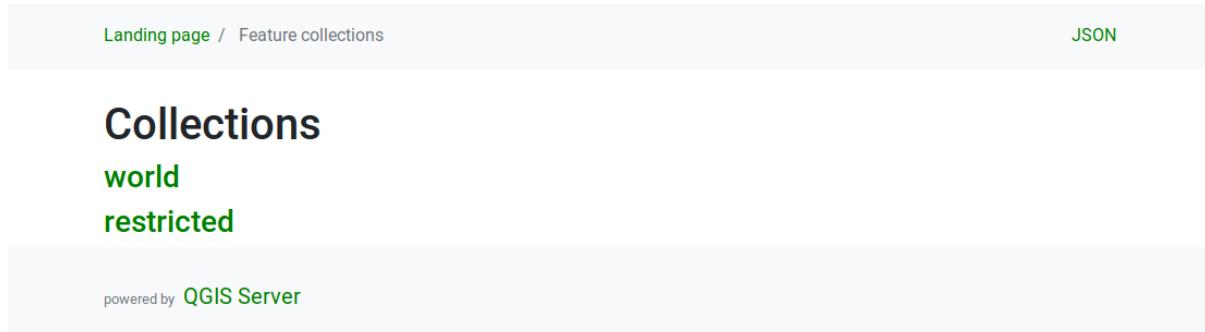


Figura3.24: Server OAPIF collections list page

## Collection detail

While the collections endpoint does not provide detailed information about each available collection, that information is available in the `/collections/{collectionId}` endpoints. Typical information includes the extent, a description, CRSs and other metadata.

The HTML representation also provides a browsable map with the available features.

Landing page / Collections / world JSON

## world

### Available CRSs

- <http://www.opengis.net/def/crs/OGC/1.3/CRS84>
- <http://www.opengis.net/def/crs/EPSG/9.6.2/4326>
- <http://www.opengis.net/def/crs/EPSG/9.6.2/3857>
- <http://www.opengis.net/def/crs/EPSG/9.6.2/900913>

### Extent

West  
**-179.99999999999997**

South  
**-90.0**

East  
**180.0**

North  
**83.62359600000008**

powered by **QGIS Server**

Figura3.25: Página de detalhes da coleta do Servidor OAPIF

## Lista de feições

This endpoint provides a list of all features in a collection knowing the collection ID. The path of this endpoint is `/collections/{collectionId}/items`.

The HTML representation also provides a browsable map with the available features.

---

**Nota:** This endpoint is analogue to GetFeature in WFS 1 and WFS 2.

---

Landing page / Collections / world / Features in layer world GEOJSON

[Previous](#) [Next](#)

## Features in layer world

### world 20

AREA	845942
FIPS	BR
ISO2	BR
ISO3	BRA
LAT	-10.772
LON	-53.089
NAME	Brazil
POP2005	186830759
REGION	19
SUBREGION	5
UN	76

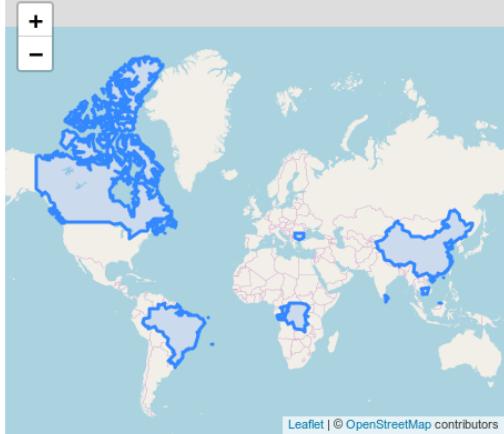

Leaflet | © OpenStreetMap contributors

Figura3.26: Página de lista de recursos do servidor OAPIF

### Detalhe da feição

This endpoint provides all the available information about a single feature, including the feature attributes and its geometry. The path of this endpoint is `/collections/{collectionId}/items/{itemId}`.

The HTML representation also provides a browsable map with the feature geometry.

Landing page / Collections / world / Items of world / world - feature 20 GEOJSON

## world - feature 20

AREA	845942
FIPS	BR
ISO2	BR
ISO3	BRA
LAT	-10.772
LON	-53.089
NAME	Brazil
POP2005	186830759
REGION	19
SUBREGION	5
UN	76


Leaflet | © OpenStreetMap contributors

powered by [QGIS Server](#)

Figura3.27: Página de detalhes do servidor OAPIF

### 3.6.3 Paginação

Pagination of a long list of features is implemented in the OGC API through `next` and `prev` links, QGIS server constructs these links by appending `limit` and `offset` as query string parameters.

URL example:

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

---

**Nota:** The maximum acceptable value for `limit` can be configured with the `QGIS_SERVER_API_WFS3_MAX_LIMIT` server configuration setting (see: [Variáveis ambientais](#)).

---

### 3.6.4 Feature filtering

The features available in a collection can be filtered/searched by specifying one or more filters.

#### Filtro de data e hora

Collections with date and/or datetime attributes can be filtered by specifying a `datetime` argument in the query string. By default the first date/datetime field is used for filtering. This behavior can be configured by setting a “Date” or “Time” dimension in the *QGIS Server ▶ Dimension* section of the layer properties dialog.

The date and time filtering syntax is fully described in the [API Definition](#) and also supports ranges (begin and end values are included) in addition to single values.

Exemplo de URL:

Returns only the features with date dimension matching 2019-01-01

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

Returns only the features with datetime dimension matching 2019-01-01T01:01:01

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

Returns only the features with datetime dimension in the range 2019-01-01T01:01:01 - 2019-01-01T12:00:00

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

#### Bounding box filter

A bounding box spatial filter can be specified with the `bbox` parameter:

A ordem dos elementos separados por vírgula é:

- Lower left corner, WGS 84 longitude
- Lower left corner, WGS 84 latitude
- Upper right corner, WGS 84 longitude
- Upper right corner, WGS 84 latitude

---

**Nota:** The OGC specifications also allow a 6 item bbox specifier where the third and sixth items are the Z components, this is not yet supported by QGIS server.

---

URL example:

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

If the *CRS* of the bounding box is not [WGS 84](#), a different CRS can be specified by using the optional parameter `bbox-crs`. The CRS format identifier must be in the [OGC URI](#) format:

URL example:

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

## Filtros de atributos

Attribute filters can be combined with the bounding box filter and they are in the general form: `<attribute name>=<attribute value>`. Multiple filters can be combined using the AND operator.

URL example:

filters all features where attribute name equals “my value”

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

Partial matches are also supported by using a \* (“star”) operator:

URL example:

filters all features where attribute name ends with “value”

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

## 3.6.5 Feature sorting

It is possible to order the result set by field value using the `sortby` query parameter.

The results are sorted in ascending order by default. To sort the results in descending order, a boolean flag (`sortdesc`) can be set:

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

## 3.6.6 Seleção de atributo

The feature attributes returned by a [Lista de feições](#) call can be limited by adding a comma separated list of attribute names in the optional `properties` query string argument.

URL example:

retorna apenas o atributo nome

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

### 3.6.7 Personalizar as páginas HTML

The HTML representation uses a set of HTML templates to generate the response. The template is parsed by a template engine called `inja`. The templates can be customized by overriding them (see: [Substituição de modelos](#)). The template has access to the same data that are available to the JSON representation and a few additional functions are available to the template:

#### Funções de modelo personalizado

- `path_append( path )`: appends a directory path to the current url
- `path_chomp( n )`: removes the specified number “n” of directory components from the current url path
- `json_dump( )`: prints the JSON data passed to the template
- `estático( caminho )`: retorna a URL completa para o caminho estático especificado. Por exemplo: “`estático(“/style/black.css”)`” com um caminho raiz “`http://localhost/qgisserver/oapif`” retornará “`http://localhost/qgisserver/oapif/static/style/black.css`”.
- `links_filter( links, key, value )`: Returns filtered links from a link list
- `content_type_name( content_type )`: Returns a short name from a content type, for example “text/html” will return “HTML”
- `n12br( texto )`: Retorna o texto de entrada com todas as novas linhas substituídas por “`<br>`” tags

#### Substituição de modelos

Templates and static assets are stored in subdirectories of the QGIS server default API resource directory (`/usr/share/qgis/resources/server/api` on a Linux system), the base directory can be customized by changing the environment variable `QGIS_SERVER_API_RESOURCES_DIRECTORY`.

A typical Linux installation will have the following directory tree:

```
/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
```

To override the templates you can copy the whole tree to another location and point `QGIS_SERVER_API_RESOURCES_DIRECTORY` to the new location.

# CAPÍTULO 4

---

## Catálogo

---

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 QGIS Server catalog uses the variables QGIS\_SERVER\_LANDING\_PAGE\_PROJECTS\_DIRECTORIES and QGIS\_SERVER\_LANDING\_PAGE\_PROJECTS\_PG\_CONNECTIONS (see *Variáveis ambientais*)

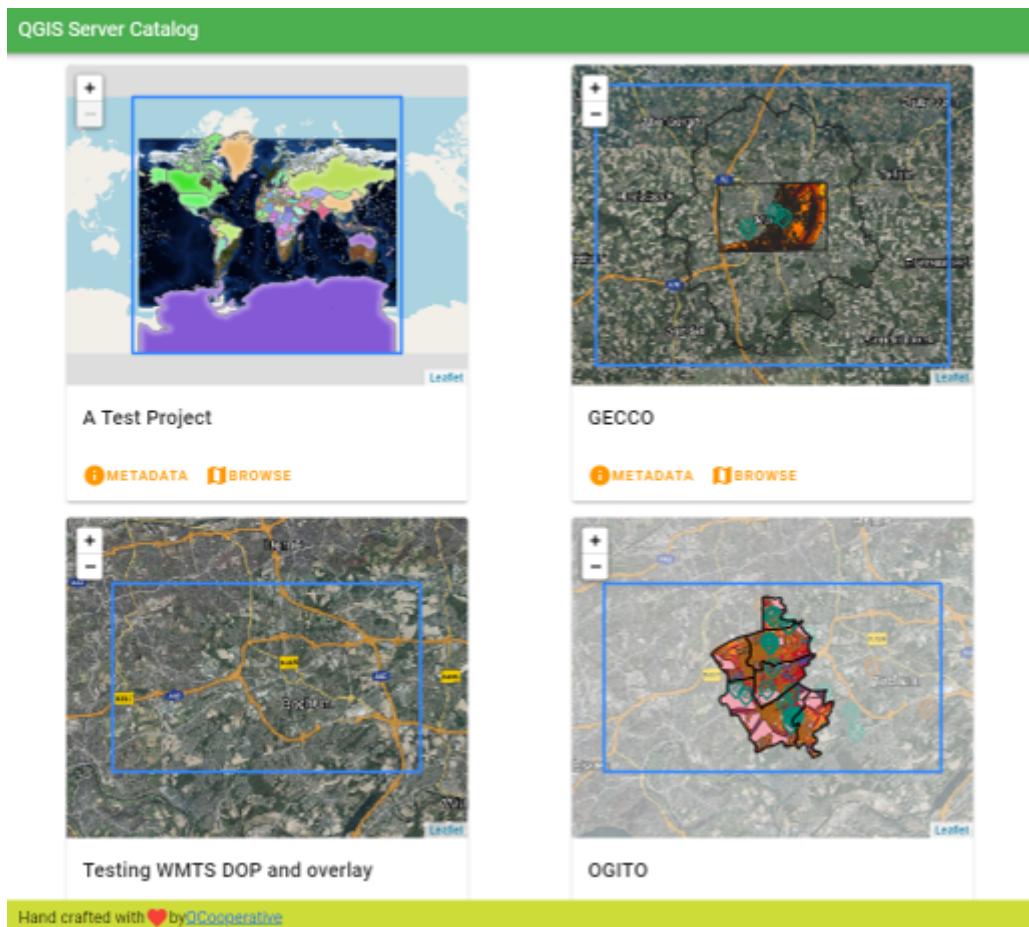


Figura4.1: Server Catalog project list page

Você pode consultar os metadados associados a um projeto e os serviços que ele fornece. Links para esses serviços também são fornecidos.

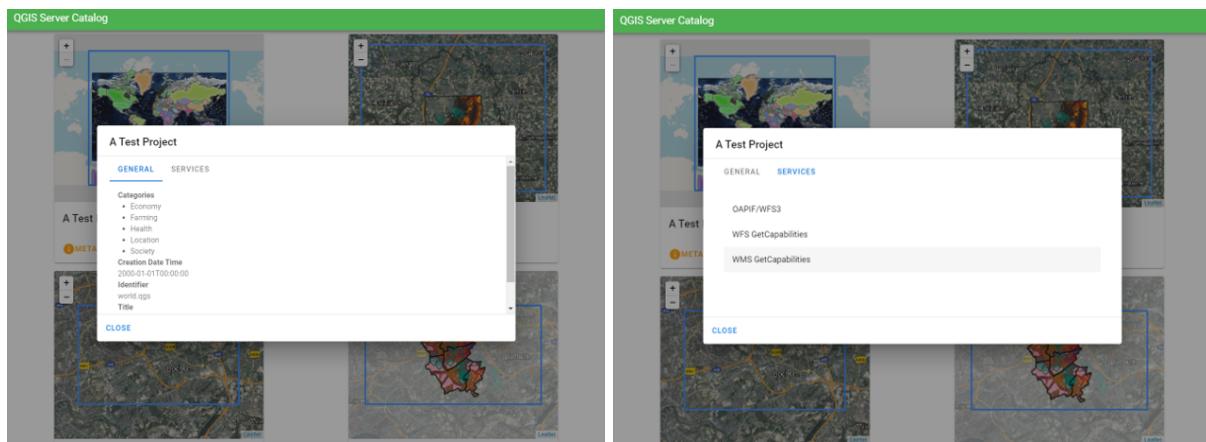


Figura4.2: Server Catalog, metadata associated to a project and services (links to) that it provides.

By browsing a project, it is listed the dataset that it serves.



Figura4.3: Browsing a dataset served by a project in the Server Catalog

Use o clique direito sobre uma camada para exibir a tabela de atributos associada a ela.

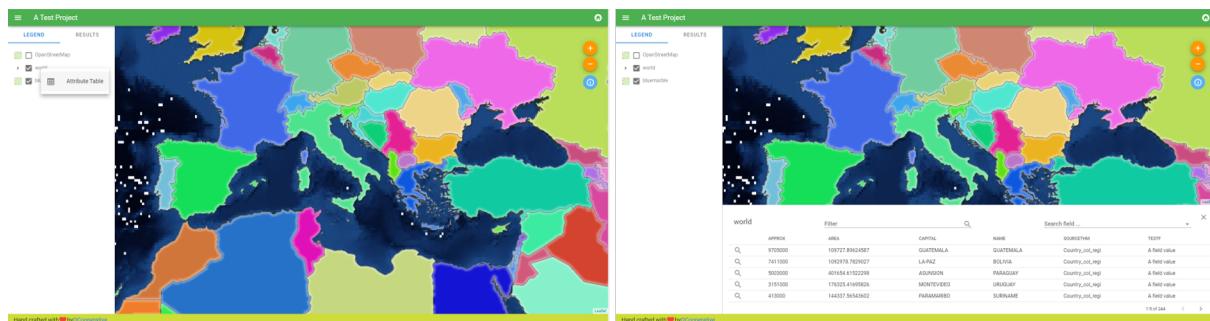


Figura4.4: Tabela de atributos associados a uma camada

É possível consultar as informações dos elementos do mapa, como mostrado na imagem abaixo:

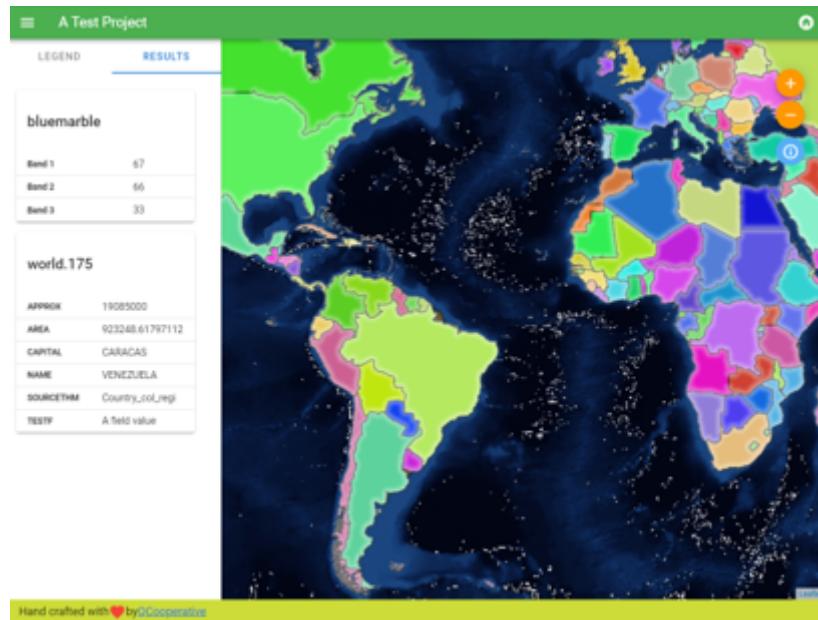


Figura4.5: Consulting information of a map element

# CAPÍTULO 5

---

## Plugins

---

### 5.1 Lista de plugins

Plugins também podem ser instalados no Servidor QGIS.

Alguns plugins projetados para servidor podem ser encontrados no repositório oficial [QGIS](#).

Instalar apenas plugins que você precisa para seu próprio propósito. No servidor QGIS, os plugins são como ganchos no servidor QGIS, eles podem alterar entradas ou saídas do servidor QGIS. Eles podem produzir resultados inesperados se você não souber como o plugin funciona. Por favor, consulte a respectiva documentação ou o aplicativo que precisa de plugins do servidor QGIS para saber qual plugin pode ser útil para você.

### 5.2 Localização dos plugins

By default, on Debian based systems, QGIS Server will look for plugins located in `/usr/lib/qgis/plugins`. The default value is displayed when QGIS Server is starting, in the logs. It's possible to set a custom path by defining the environment variable `QGIS_PLUGINPATH` in the web server configuration.

### 5.3 Instalação

#### 5.3.1 Manualmente com um ZIP

Como exemplo, para instalar o plugin **HelloWorld** para testar o servidor, usando uma pasta específica, primeiro é necessário criar um diretório para manter os plugins do servidor. Isto será especificado na configuração do host virtual e passado para o servidor através de uma variável de ambiente:

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

**Aviso:** De acordo com sua [descrição](#), o plugin HelloServer é projetado para fins de desenvolvimento e demonstração. Não guarde este plugin para produção se você não precisar dele.

### 5.3.2 Com uma ferramenta de linha de comando

Se você precisar instalar e atualizar regularmente plugins que são armazenados no repositório de plugins QGIS, você pode utilizar o [QGIS-Plugin-Manager](#). É uma ferramenta para ajudá-lo a gerenciar os plugins a partir da linha de comando.

A instalação está usando pip. Instalar em um ambiente virtual é uma boa prática, mas não é obrigatório:

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

Para atualizar a ferramenta:

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

Então, você pode utilizar o executável `qgis-plugin-manager` da linha de comando:

```
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 |
|-----|
|-----|
| wfsOutputExtension | wfsOutputExtension | 1.6.2 | | 3.0 |
|-----|
| qgis_server_render_geojson | GeoJson Renderer | v0.4 | | 3.4 |
|-----|
| DataPlotly | Data Plotly | 3.7.1 | | 3.4 |
|-----|
| 3.98 | Matteo Ghetta (Faunalia) | Upgrade to 3.8.1 | |
```

Sugerimos que você leia a documentação completa no [arquivo leia-me](#) para saber como instalar ou atualizar plugins com esta ferramenta.

## 5.4 Configuração do Servidor HTTP

### 5.4.1 Apache

Para poder usar plugins de servidor, FastCGI precisa saber onde procurar. Portanto, temos que modificar o arquivo de configuração do Apache para indicar a variável de ambiente `QGIS_PLUGINPATH` para FastCGI:

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

Moreover, a basic HTTP authorization is necessary to play with the HelloWorld plugin previously introduced. So we have to update the Apache configuration file a last time:

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

Em seguida, reinicie o Apache:

```
systemctl restart apache2
```

## 5.5 Como usar um plug-in

Teste o servidor com o plugin HelloWorld:

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

You can have a look at the default GetCapabilities of the QGIS server at:

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



# CAPÍTULO 6

---

## Configuração avançada

---

### 6.1 Registrando

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

- *QGIS\_SERVER\_LOG\_STDERR*

With the following variables the logging can be further customized:

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

### 6.2 Variáveis ambientais

Você pode configurar alguns aspectos do QGIS Server configurando **environment variables**.

According to the HTTP server and how you run QGIS Server, there are several ways to define these variables. This is fully described in [Servidor HTTP Apache](#).

Nome	Descrição	Padrão	Serviços
QGIS_OPÇÕES_CAMINHO	Specifies the path to the directory with settings. It works the same way as QGIS application --optionspath option. It is looking for settings file in <QGIS_OPTIONS_PATH>/QGIS/QGIS3.ini.	"	Tudo
QGIS_PLUGINPATH	Useful if you are using Python plugins for the server, this sets the folder that is searched for Python plugins.	"	Tudo
QGIS_PROJETO_ARQUIVO	The .qgs or .qgz project file, normally passed as a parameter in the query string (with MAP), you can also set it as an environment variable (for example by using mod_rewrite Apache module). Note that you may also indicate a project stored in PostgreSQL, e.g. postgresql://localhost:5432?sslmode=disable&dbname=mydb&schema=myschema&project=myp project.	"	Tudo
QGIS_SERVER_API_RESOURCE_DIRECTORY	Defines the directory for all static OGC API resources (such as OAPIF/WFS3) (HTML models, CSS, JS, ...)	depende do pacote	OAPIF/WFS3
QGIS_SERVER_API_WFS3_MAXLIMIT	MaxLimit value for limit in a OAPIF/WFS3 features request.	10000	OAPIF/WFS3
QGIS_SERVER_CACHE_DIRECTORY	Defines the network cache directory on the filesystem.	cache in profile directory	Tudo
QGIS_SERVER_CACHE_SIZE	Sets the network cache size in MB.	50 MB	Tudo
QGIS_SERVER_DISABLE_GETPRINT	TERMINA opção no nível do projeto para melhorar o tempo de leitura do projeto desativando o carregamento de layouts. Activating this option disables the QGIS WMS GetPrint request. Set this QGIS project flag to not load layouts.	falso	WMS
QGIS_SERVER_IGNORE_BADLAYERS	“BAD LAYERS” are layers that cannot be loaded. The default behavior of QGIS Server is to consider the project as not available if it contains a bad layer. The default behavior can be overridden by setting this variable to 1 or true. In this case, “bad” layers will just be ignored, and the project will be considered valid and available.	falso	Tudo
QGIS_SERVER_LANDING_PAGE_PREFIX	Componente do caminho da URL base da página inicial	""	Tudo
QGIS_SERVER_LANDING_PAGE_PROJECTS_DIRECTORIES	Directories used by the landing page service to find .qgs and .qgz projects	""	Tudo
QGIS_SERVER_LANDING_PAGE_PG_CONNECTIONS	PostgreSQL connection strings used by the landing page service to find projects	""	Tudo
QGIS_SERVER_LOG_FILE	Especificar o caminho e o nome do arquivo. Certifique-se de que o servidor tenha as permissões adequadas para gravar no arquivo. O arquivo deve ser criado automaticamente, basta enviar algumas requisições para o servidor. Se não estiver lá, verifique as permissões.	"	Tudo
Aviso: QGIS_SERVER_LOG_FILE is de- re- cated			

## 6.3 Settings summary

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

Por exemplo com spawn-fcgi:

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

QGIS Server Settings:

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

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

  - QGIS_SERVER_MAX_THREADS / '/qgis/max_threads' (Number of threads to use when_
↪parallel rendering is activated): '4' (read fromINI_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 fromINI_FILE)

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

In this particular case, we know that **QGIS\_SERVER\_MAX\_THREADS** and **QGIS\_SERVER\_PARALLEL\_RENDERING** values are read from the ini file found in **QGIS\_OPTIONS\_PATH** directory (which is defined through an environment variable). The corresponding entries in the ini file are **/qgis/max\_threads** and **/qgis/parallel\_rendering** and their values are **true** and **4** threads.

## 6.4 Connection to service file

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

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

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

## 6.5 Adicione fontes ao seu servidor linux

Tenha em mente que você pode usar projetos QGIS que apontam para fontes que podem não existir por padrão em outras máquinas. Isso significa que se você compartilhar o projeto, ele poderá parecer diferente em outras máquinas (se as fontes não existirem na máquina de destino).

Para garantir que isso não aconteça, você só precisa instalar as fontes ausentes na máquina de destino. Fazer isso em sistemas desktop geralmente é trivial (clique duas vezes nas fontes).

Para linux, se você não tiver um ambiente de desktop instalado (ou preferir a linha de comando), precisará:

- On Debian based systems:

```
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
```

- On Fedora based systems:

```
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
```

# CAPÍTULO 7

## Servidor de desenvolvimento

A production installation and deployment of QGIS Server usually involves setting up a web server component (e.g. Apache or Nginx) that can forward the HTTP requests coming from the clients to the QGIS Server FastCGI binary application.

If you want to quickly test QGIS Server on your local machine without configuring and installing a full web server stack you can use the QGIS Development Standalone server.

Esta é uma aplicação independente que fornece um servidor web muito simples, pronto para servir seus arquivos de projeto.

**Aviso:** The Standalone Development Server has not been developed with the purpose of being used in production, it was not checked for security vulnerabilities or for other stress conditions that normally will occur on a publicly exposed server.

Para lançar o servidor:

```
$ qgis_mapserver
```

A porta padrão que o Servidor de Desenvolvimento escuta é 8000. Exemplo de saída:

```
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
```

O servidor tem algumas opções que podem ser passadas como argumentos de linha de comando. Você pode vê-las todas invocando o servidor com `-h`.

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

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

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

# CAPÍTULO 8

## Containerized deployment

There are many ways to use containerized application, from the most simple (simple Docker images) to sophisticated (Kubernetes and so on).

**Nota:** Este tipo de implantação precisa da aplicação docker para ser instalado e executado. Verifique este [tutorial <https://www.docker.com/101-tutorial>>](https://www.docker.com/101-tutorial).

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

**Nota:** Os downloads de pacotes QGIS Debian-Ubuntu precisam de uma chave de autenticação gpg válida. Favor consultar as [páginas de instalação](#) para atualizar o seguinte arquivo Docker.

### 8.1 Imagens simples do estivador

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

- create a file Dockerfile with this content:

```
FROM debian:bullseye-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 \
```

(continua na próxima página)

(continuação da página anterior)

```

    && localedef -i en_US -f UTF-8 en_US.UTF-8 \
    # 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/en/site/forusers/alldownloads.html#repositories)
    && echo "deb [signed-by=/etc/apt/keyrings/qgis-archive-keyring.gpg] https://qgis.org/debian bullseye 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"]

```

- criar um arquivo cmd.sh com este conteúdo:

```

#!/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

```

- construir a imagem com:

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

### 8.1.1 Primeira execução

Para executar o servidor, você precisará de um arquivo de projeto QGIS. Você pode utilizar um dos seus ou escolher esta amostra.

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

---

**Nota:** You may need to add advertised URLs under the *QGIS Server* tab of the *Project ▶ Properties* if the Get-Capabilites 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 *Configurar seu projeto* and subsequent.

---

Agora, você pode executar o servidor com:

```
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
```

Opções utilizadas:

- **-d\***: rodar em segundo plano
- **-rm**: remover o container quando parar
- **-nome**: nome do container a ser criado
- **-net\***: (anteriormente criada) sub-rede
- **-hostname**: nome da hostname do recipiente, para referência posterior
- **-v**: diretório de dados local a ser montado no contêiner
- **\*\*-p**: mapeamento da porta
- **-e\***: variável de ambiente a ser utilizada no contêiner

Para verificar, digite `docker ps | grep qgis-server` e você deve ver uma linha com **qgis-server**:

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

### 8.1.2 Amostra utilizável

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

Criar um arquivo `nginx.conf` no diretório atual com este conteúdo:

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

(continua na próxima página)

(continuação da página anterior)

```

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

```

E digite este comando:

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

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

### 8.1.3 Limpeza

Para limpar as imagens em execução, digite:

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

## 8.2 Pilhas de docas

O método anterior é fácil de ser roteirizado, mas não pode ser facilmente compactado, nem padronizado ou facilmente administrado.

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

A seguir, apresentaremos configurações simples para fins de teste. Elas não são adequadas para produção.

### 8.2.1 Enxame/doca-composição

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).

#### Descrição da pilha

Agora que o Swarm está funcionando, crie o arquivo de serviço (veja 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

```

(continua na próxima página)

(continuação da página anterior)

```
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
```

Para implantar (ou atualizar) a pilha, digite:

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

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

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

Algo parecido com:

ID	NAME	MODE	REPLICAS	
→IMAGE				→
gmx7ewlvwsqt	qgis_nginx	replicated	1 / 1	→
→nginx:1.13	*:8080→80/tcp			
10v2e7cl43u3	qgis_qgis-server	replicated	1 / 1	→
→qgis-server:latest				

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

## Limpeza

Para limpeza, digite:

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

## 8.2.2 Kubernetes

### Instalação

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

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

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

### microk8s

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

Primeiro, habilite o registro:

```
microk8s enable dashboard dns registry
```

Em seguida, etiqueite e passe a imagem para seu registro recém-criado:

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

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

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

### Criando manifestos

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

### Manifestações de implantação

Criar um arquivo **:arquivo:`deployments.yaml`** com este conteúdo:

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

(continua na próxima página)

(continuação da página anterior)

```

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

---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: qgis-nginx
  namespace: default
spec:
  replicas: 1
  selector:
    matchLabels:
      myLabel: qgis-nginx
  template:
    metadata:
      labels:
        myLabel: qgis-nginx
    spec:
      containers:
        - name: qgis-nginx
          image: nginx:1.13
          ports:
            - containerPort: 80
          volumeMounts:
            - name: nginx-conf
              mountPath: /etc/nginx/conf.d/default.conf
      volumes:
        - name: nginx-conf
          hostPath:
            path: REPLACE_WITH_FULL_PATH/nginx.conf

```

## Manifestos de serviço

Criar um arquivo :arquivo:`services.yaml` com este conteúdo:

```

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:

```

(continua na próxima página)

(continuação da página anterior)

```

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

```

## Implantação de manifestos

Para implantar as imagens e serviços em Kubernetes, pode-se usar o painel de controle (clique no **+**\* no canto superior direito) ou a linha de comando.

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

Para implantar ou atualizar seus manifestos:

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

Para verificar o que está atualmente implantado:

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

Você deveria obter algo como:

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
<b>NAME</b>				
service/Kubernetes	ClusterIP	10.152.183.1	<none>	443/TCP
service/qgis-exec-server	ClusterIP	10.152.183.218	<none>	5555/TCP
service/qgis-nginx	NodePort	10.152.183.234	<none>	80:30080/TCP
service/qgis-server	ClusterIP	10.152.183.132	<none>	5555/TCP
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

Para ler os logs nginx/qgis, digite:

```
kubectl logs -f POD_NAME
```

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

## Limpeza

Para limpar, digite:

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

## 8.3 Implantação de nuvens

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

Cloud deployment solutions can be a good alternative when you do not want to focus on infrastructure management. A cloud deployment may use proprietary mechanisms, but they are also compatible with the stages explained previously (*docker images* and *stack management*).

### 8.3.1 AWS usecase

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

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

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



# CAPÍTULO 9

## Perguntas mais frequentes

- *Quais são as diferenças entre o QGIS Desktop e o Servidor QGIS?*

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

- *What is OGC?*

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

- *Name some other web mapping servers?*

ArcGIS server, Geoserver, Mapserver, Mapnik etc.

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

Feições	Servidor QGIS	GeoServer	Servidor ArcGIS
Desde	2006	2001	1999
Licença	GPL	GPL	comercial
Suporte comercial	Multiple companies	Multiple companies	ESRI and its vendors network
Tecnologia	C++/python	Java	C++
Tile cache	sim	yes (via GeoWebCache)	sim
3D	Não	Não	Sim
Consultas	FES (2.0) and OGC (1.0) filters	CQL and OGC filters	OGC filters
Report generation	sim	sim	sim
Administração do servidor	yes via third parties (LizMap, QWC2, etc.)	web + API REST	web + API REST
Edição de camada/simbologia do projeto GIS	complete via dedicated GUI	simple via web interface	complete via dedicated GUI

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

OGC standards	Servidor QGIS	GeoServer	Servidor ArcGIS
WMS (Web Map Service)	1.3.0 - 1.1.1	1.3.0 - 1.1.1	1.3.0 - 1.1.1
SRW (Serviço de Recursos da Web)	1.1.0 - 1.0.0	2.0.0 - 1.0.0	2.0.0 - 1.0.0
OAPIF (aka WFS3)	1.0.0	não	não
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)	não	1.0.0	1.0.0
SCW (Serviço de Catálogo para a Web)	não	2.0.2	não
SLD (Styled Layer Descriptor)	sim	sim	sim

- *What is a tile cache?*

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

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

- *What is PostgreSQL?*

PostgreSQL <<https://www.postgresql.org/>> é um poderoso e de código aberto companheiro de banco de dados objeto-relacional para QGIS.

- *O que é PostGIS?*

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

- Continua...