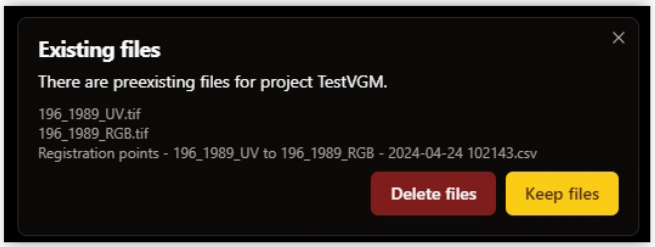
Addendum: “How to …” with the XRF Explorer V2

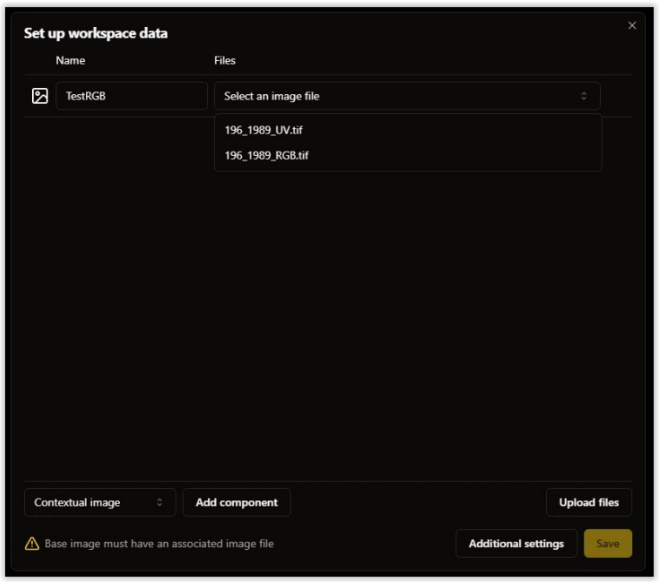
[1] How to startup a new project with data files already on the webserver

The normal use case would be to start a new ‘project’ (also known as: workspace, or data source) in the Menu > File > New project. When a project name is chosen that has no folder with the same name on the webserver, then this folder is created by the system and a next screen will popup to manage the uploading of files.

But what happens in case a folder is already present on the webserver? This may be useful, because files can be big and uploading them using an FTP client could be a more reliable way to do that. So, alternatively, we can create a folder first on the webserver (in the right location, as directed by the pathname mentioned in /root/xrf-explorer/config/backend.yml) with a chosen the project name. We can upload files in that folder. Then, we can go to the app and use Menu > File > New project, and enter the folder name. The following popup is the response:



When clicking on Delete files, the folder will be emptied (we do not want it in this case), when clicking on Keep files, then the folder gets connected, and a JSON workspace file will be created in that folder. Also the next popup window will be shown for setting up the workspace data, and already in the dropdown boxes the available files will be shown (see below).



From this point onwards, the setup works as usual.

[2] How to connect an existing URL to the webapp on the webserver

During the development phase a temporary URL was created to connect to the web app. This was highly useful, because this gave the opportunity to easily access it for testing. But the temp URL needed to be replaced by some other URL and another temporary solution was chosen, being: https://xrf.paintingexplorer.nl

To be able to use that URL that redirects to the web app using secure connections also, support was asked by the TUE development team. Their first response is included here:

To point the domain towards xrf-explorer and handle https termination (as xrf-explorer only handles http connections) it would be easiest to install a reverse proxy such as nginx proxy manager https://nginxproxymanager.com/guide/#quick-setup using docker compose. As this uses the default 80 and 443 http ports, visiting xrf.paintingexplorer.nl should then show some error code from nginx proxy manager. If this is successful, then the reverse proxy is set up correctly (otherwise, make sure the firewall is set up to allow traffic to ports 80, 81 and 443) and the connection to xrf-explorer can be made by visiting :81 and setting up a proxy host for xrf.paintingexplorer.nl that points to localhost:8080 over http. Additionally, https setup and simple password authentication can be set up using nginx proxy manager, however that should in large parts be fairly self explanatory.

This can already be a challenge when lacking experience on this matter, so we asked their practical support to realize the rerouting. This was done, with complementary documentation, as follows:

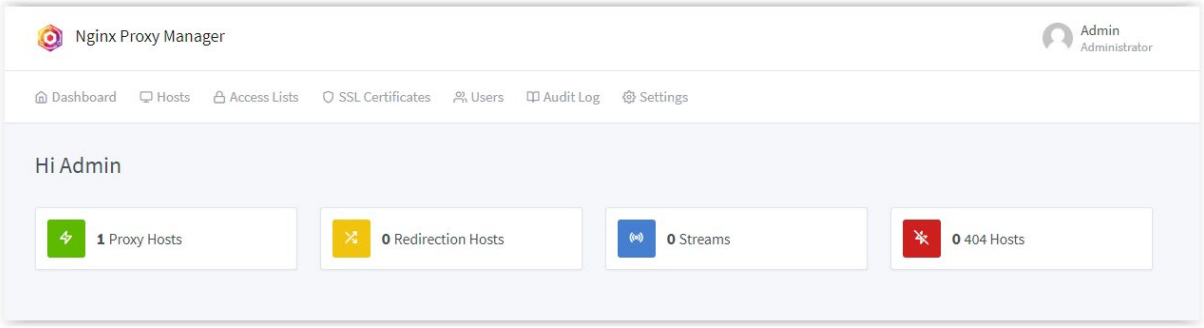
Server installation notes

A number of programs are installed on the webserver that make the connection from URL to starting up the software work. Below we will discuss each of them, a short (but hopefully complete) overview, so that problems can be solved and configuration can be adjusted without too much effort.

Nginx Proxy Manager (NPM)

Admin interface is available on <ip address of the webserver vps>:81 with the following login:

> Email: admin@paintingexplorer.nl
> Password: <ask the admin>



In Nginx also a free SSL (secure socket layer) certificate by 'Let's Encrypt' is configured for the incoming call of xrf.paintingexplorer.nl

In Nginx a proxy host is configured that redirects the incoming https://xrf.paintingexplorer.nl to http://localhost:8080, where the XRF Explorer program is running

NPM itself is installed via docker compose. The configuration is located on the webserver in `/opt/npm/docker-compose.yml` and can be started via

```bash
cd /opt/npm
docker compose up -d
```

##XRF Explorer securities

The XRF-Explorer program itself is secured with a user and password, configured in Nginx Proxy Manager.
At the moment, this is

> Username: xrf
> Password: <ask the admin>

XRF Explorer program itself is installed in the directory on the webserver `/opt/xrf-explorer`, all data is in `/var/opt/xrf-explorer/data` and the configuration is read from `/etc/xrf-explorer/config.yml`.

The program is managed by SystemD, so starting/restarting/stopping can be done with `systemctl start/stop/restart xrf-explorer`. The SystemD configuration is located in `/etc/systemd/system/xrf-explorer.service`.

The program is run by SystemD as the `xrf` user on the server, so it is important that all relevant files are also owned by the `xrf` user. Therefore, updates and such must be performed as the `xrf` user. You can switch users from `root` to `xrf` with the webserver command: `su xrf`. If a file is not owned by `xrf` while it is necessary, this can be remedied with the command `chown xrf:xrf /opt/xrf-explorer -R` and `chmod xrf:xrf /var/opt/xrf- explorer -R`.

##XRF firewall issues

The only thing that isn't quite right yet is the firewall. You can usually set up a firewall with a VPS via the provider's web interface, at the moment everything is open and since that also means that SSH, XRF Explorer and the NPM admin interface are open, this is not very safe. I would recommend ensuring that only ports 80 (http port) and 443 (https port) allow access for incoming connections via this web interface and then only opening ports 22 (ssh port) and 81 (alternative to 80) when necessary to perform maintenance or manually transfer files to the server via SFTP and then close these ports again.

[Hostkey response: Unfortunately, we do not provide free server administration services to our clients. After the server has been deployed – its management becomes your responsibility. It is recommended to check the guides on the open sources.]

[MarcoR to JanB: Maybe install ufw on the VPS and close ports? Does that make sense?]

##XRF Explorer update

In case a new version of XRF Explorer is available on Github, and you want to install that, then use the following webserver commands, as the default 'root' user:

```bash
su xrf
cd /opt/xrf-explorer
git pull
cd xrf_explorer/client
npm build
exit
systemctl restart xrf-explorer

```

After an update the configuration options may have been adjusted, in this case it is necessary to check whether `/etc/xrf-explorer/config.yml` and `/opt/xrf-explorer/config/backend.yml` contain the same options. If not, the new options will need to be added to `/etc/xrf-explorer/config.yml` before restarting XRF Explorer.

GitHub Actions Runner

On the webserver also GitHub Actions Runner is installed, and used by the XRF Explorer repository (https://github.com/olive-groves/xrf-explorer) to do checking.

This runner is installed as the property of the user `runner` in `/home/runner/actions-runner`.

It can be started by the command `/home/runner/actions-runner/svc.sh start'
Or stopped '/home/runner/actions-runner/svc.sh stop'
Other useful command options are: 'status' 'install runner' 'uninstall'.

[3] How to change a menu entry in the XRF-Explorer WindowMenu?

If I would like to add an entry in that menu, or change the URL of the Github menu item, where would I change this?

The team responded with: The links can be reconfigured under root/xrf-explorer/xrf-explorer/client/src/components/menus/MainMenu.vue (etc…)