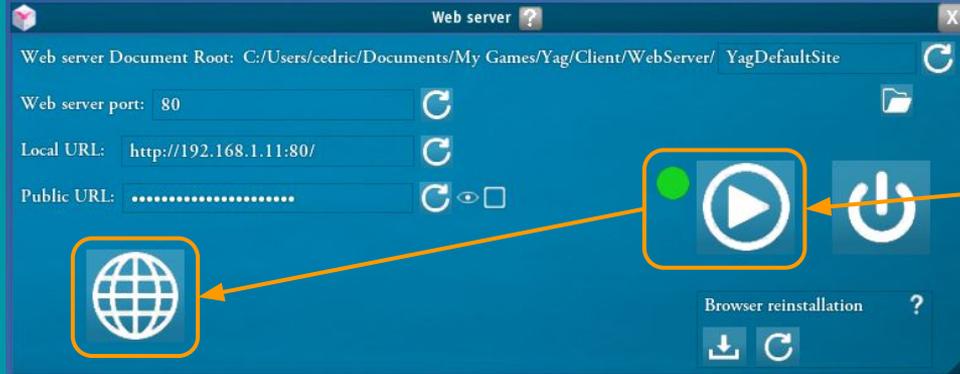


The quick way

Here is how to start the server and the browser quickly and with no explanation.

1. Open the web server window
2. Start the server
 - a. The light should change from red to green
3. Launch the browser
 - a. It opens on the default site page



The rest of this document explains how the tools in this window work.

Generalities

Yag has a small web server that can be started and stopped manually.

This allows 2 things:

- Display web content like any web server
 - This allows for example to easily prepare a mini site containing the material of your campaign: maps, images, documentation, pdf, game aids, rules, etc.
 - This site can be distributed to all players in a module or be publicly accessible
- Respond to special requests able to interact with Yag (REST API)
 - This allows you to control some Yag functions from the web such as rolling the dice or triggering a pawn animation.
 - All of these requests form a library called an API (Application Programming Interface)
 - The API used for Yag obeys a standard called REST (REpresentational State Transfer).
 - We will therefore speak of the REST API in the following to mean the collection of these requests used to drive Yag.
- The combination of these 2 possibilities makes it possible to create a website capable of interacting with Yag.
 - For example: a character sheet using web languages (html / css / javascript).

The web server window

The Yag web server is managed from its window, accessible from the main menu:



- Server port
- Local URL
- Public URL

- Launching the browser on the local address

- Site Directory
- Opening the directory
- Default Directory

- Server shutdown / restart
- Start indicator
 - Red / Green

- Reinstallation of components
 - Browser application
 - Browser data

- Warning: **no confirmation**

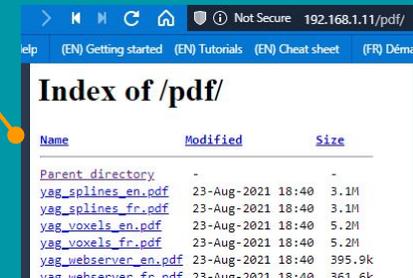
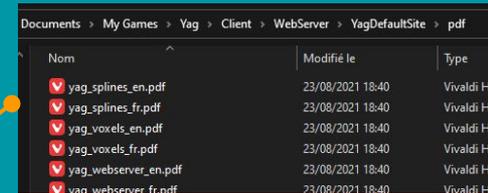
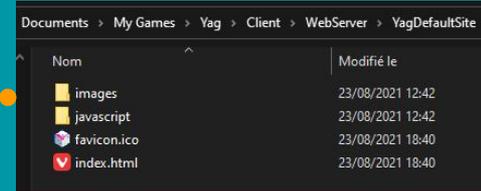
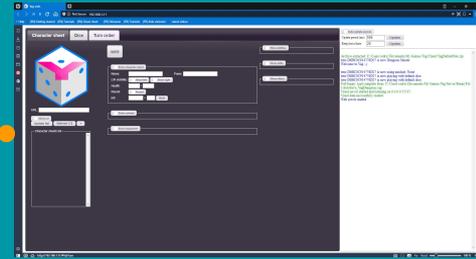
Web server properties

Some precisions on the different properties of the web server

- **Site Directory**
 - You can change its name, which allows you to manage multiple sites
 - Its location must remain the module directory (automatically managed by Yag)
- **Web server port**
 - By default it is the standard http port (80)
 - You can change it if you already have another server listening on 80, or if you want to use a non-standard port
- **Local address**
 - This is the address of your computer on your router (internet box)
 - You can use it as long as you are on your local network (wifi): from a computer, a phone, a tablet, etc.
 - This is the address that Yag opens by default when you launch the onboard browser
- **Public address**
 - This is the address that your internet service provider gives you, the one through which you are visible on the internet
 - It is hidden to avoid showing your IP if you are broadcasting your game publicly.
 - You can use it to access your site from anywhere on the net
 - In this case, check that your computer can receive a stream on this port: NAT on your router, presence of a firewall, etc.

Web content

- Yag offers by default a small website located in the following directory:
 - <MyDocuments> \ My Games \ Yag \ Client \ WebServer \ YagDefaultSite
- It contains an html file and a javascript script providing a small, simple interface for managing and using Yag character sheets, as well as dice and turn management.
 - This is just an example, you can modify it or make an entirely different site as you wish.
- It is very easy to modify it to add your own data.
 - For example, you can create a “pdf” directory and upload your pdf:
 - <MyDocuments> \ My Games \ Yag \ Client \ WebServer \ YagDefaultSite \ pdf
 - You can then visit the “pdf” page of the site to see your files.



Working with the web server

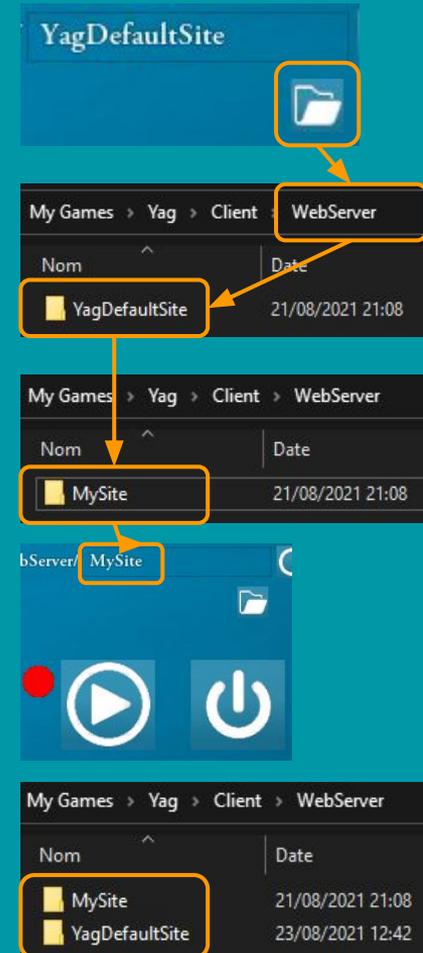
Some details on the functioning of the web server

- The default website is reinstalled each time you start Yag
 - This makes sure that the user always has the latest official version by default
- Therefore it is important to NEVER work directly in the “YagDefaultSite” directory, which is overwritten on each restart.

To quickly duplicate a new site:

- Start Yag to be sure the YagDefaultSite is up to date
- Click on the “folder” button to open the site directory in an explorer
- Go up a notch to go to the "WebServer" directory
- Rename “YagDefaultSite” to your new site (here: “MySite”)
- Enter the name of your site in the interface for the web server to start
- You can now start and work in “MySite” without fear of losing your work
- You can verify that on the next restart, YagDefaultSite is recreated without touching your site.

You can thus duplicate as many sites as you want.



The embedded web browser

Yag comes with a full embedded browser.

- This is a portable version of the Vivaldi browser.
- Many thanks to Vivaldi, who gave their permission to embark and distribute their browser for free.

The advantages of an on-board browser are:

- Yag is not invasive on your personal browser (no bookmarks, plugins, etc.)
- Yag can use Vivaldi to launch its pdf documentation (there is no guarantee that your default browser can read pdf)
- Vivaldi can be configured for Yag (look & feel, bookmarks ...)

You can disable the use of Vivaldi by unchecking the following option in the player options:

- In this case, Yag will launch its links in your default browser.



You can update the browser if it proposes it.

You can reinstall it:

- Yag will overwrite the browser with the version it embeds without touching your data

You can reinstall user data:

- You will lose the changes you made on the onboard browser (bookmarks, etc.)



REST API

Special API requests take the form of a normal address:

- <http://MyWebServer.com/yagapi/dice/roll/init/current>

But instead of returning an html file, Yag will perform an action and return some information. For example, the above address asks to roll the initiative of the currently selected pawn

/ yagapi / dice / roll / init / current

Talking to the rest api ...

... it's about dice ...

... we ask to roll ...

... the initiative ...

... for the current pawn

After rolling the dice, Yag will send the result back to the browser for processing and displaying if needed.

The operation and the list of currently available requests are described in the dedicated documentation which is accessible in two ways:

- Online on the documentation page of the Yag site: <https://yagame.fr/documentation/>
- Locally in the pdf directory, accessible from the home window



To learn a bit more

This little tutorial just scratches the subject.

If you want to understand a little better how it all works you can consult the Yag REST API documentation.

As a reminder, it is accessible:

- Online on the documentation page of the Yag site: <https://yagame.fr/documentation/>
- Locally in the pdf directory, accessible from the home window

It describes in detail the entire REST API, but more importantly, it explains in a long introduction the architecture of Yag and how it works in a network.

So this can be an interesting read even if you are not a developer but just want to understand a little better how the web browser, Yag client, web server, and Yag server interact.