

Subscribe to DeepL Pro to edit this document.  
Visit [www.DeepL.com/pro](https://www.deepl.com/pro?cta=edit-document) for more information.

**Published January 27, 2024**

**Laravel API Documentation Guide with Insomnia and Swagger**

**Authors:**

Ntandou Sacré [**(**](mailto:sacretnandou2.0@gmail.com)**sacretnandou2.0@gmail.com)**

Ba Magid [**(**](mailto:bamagid60@gmail.com)**bamagid60@gmail.com)**

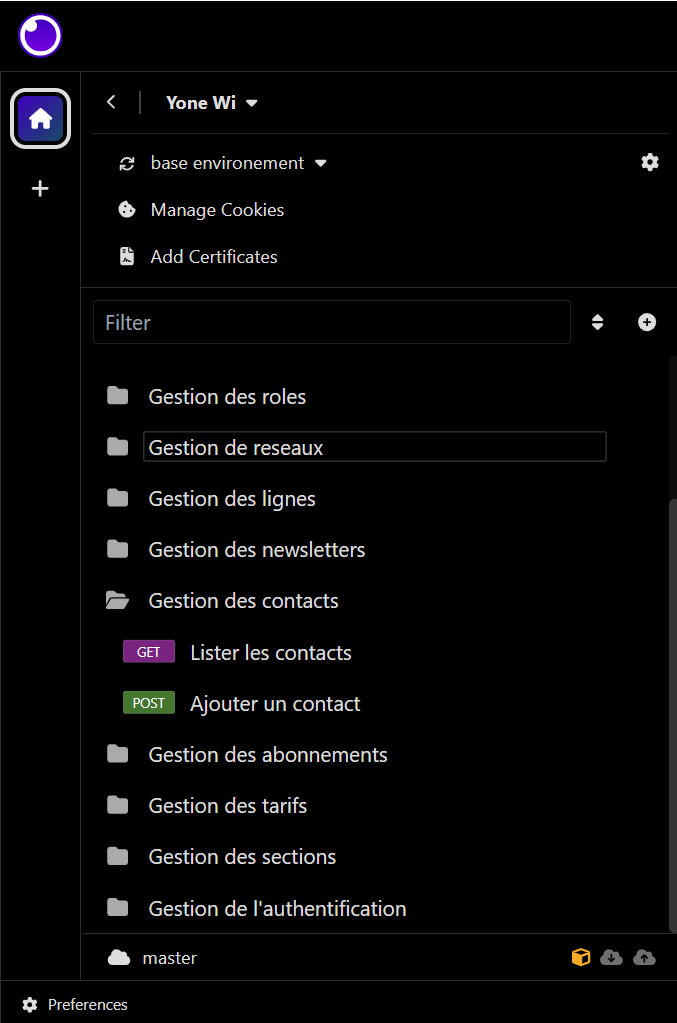
**Introduction :**

This document aims to provide a detailed guide to documenting a Laravel API using Insomnia and Swagger through the **Insomswagger** package . This approach was developed by back-end developer Ba Magid and shared after extensive discussions to validate it.

We hope you find this guide useful.

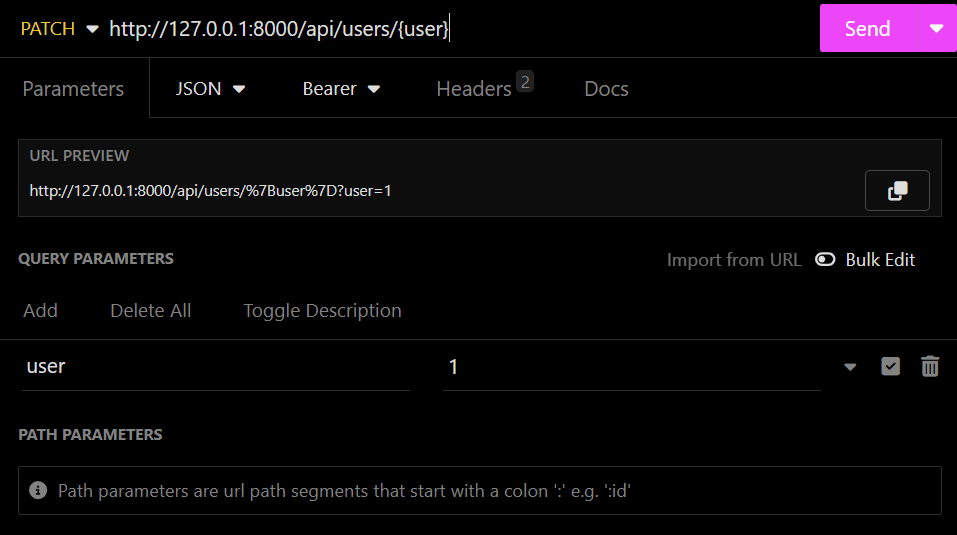
**Step 1:** Organizing requests in Insomnia

Organize your queries in separate folders on Insomnia. For example, each folder could correspond to a module in your application.



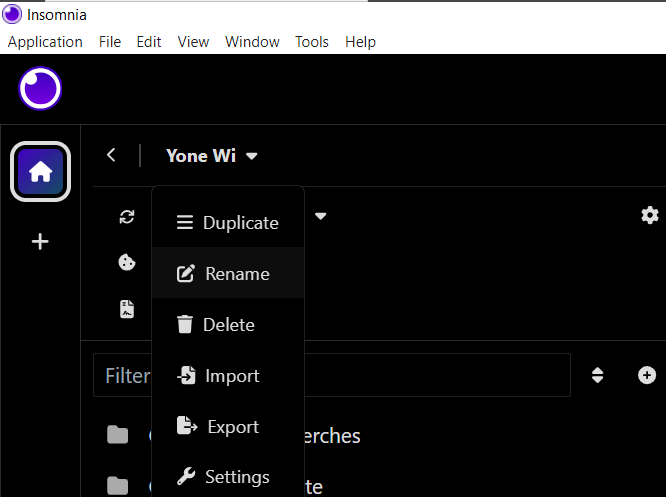
Note: Do not include subfolders.

**Step 2:** Route preparation

Before exporting the file to be documented, make sure you organize your routes correctly in the queries. Use the route defined in the Api.php file (green zone) and for routes that take parameters, specify them in the **Parameters** or **Query** section using the name specified in the route (red zone).



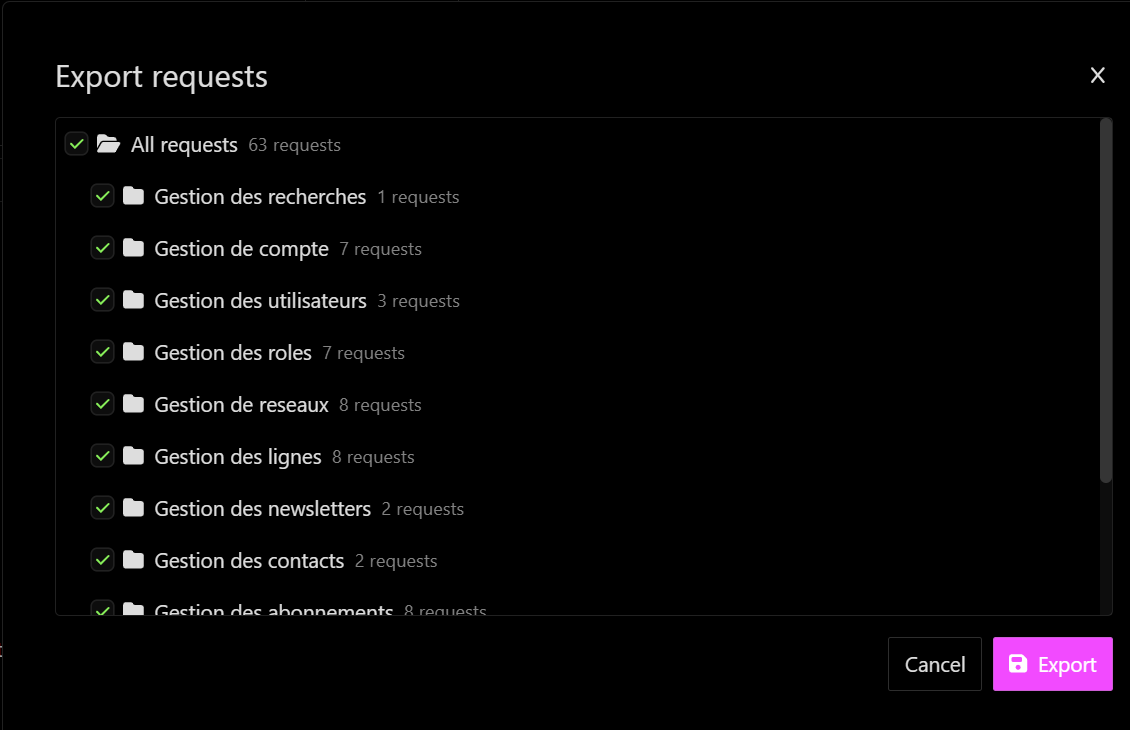
**Once you've finished, click on the tab at the top**, where you'll find the name of your collection, scroll down the list and choose the "**Export**" option. Make sure your code is working properly and that the query body is in the appropriate format( JSON or multipart/form-data) before exporting.

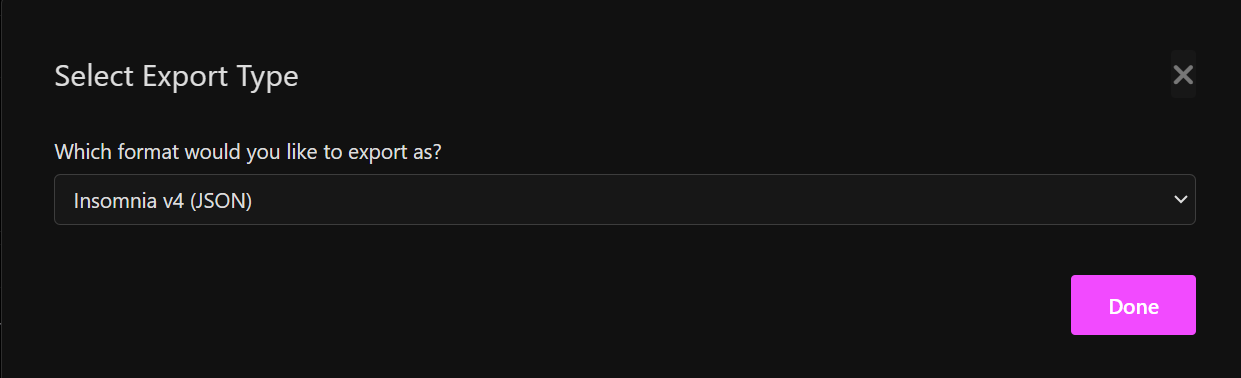


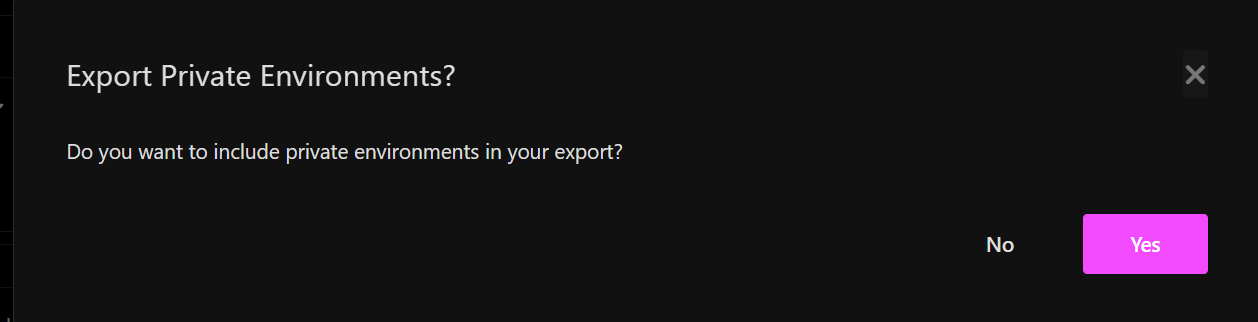




Uncheck "**All requests**", check the folder you wish to export, then click on **Export, Done,** and **YES**.







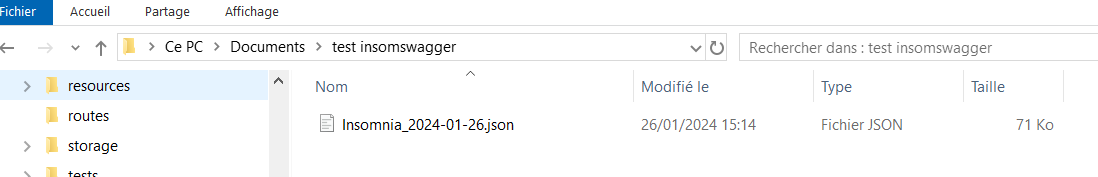
**Step 3:** Installing Insomswagger

Once the file has been exported, run the following command to install Insomswagger globally:

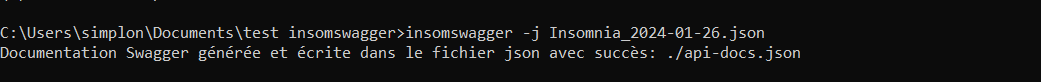
**npm install -g insomswagger**

This installation will be valid for all your Laravel projects, so you won't need to repeat it. Make sure you have Node.js installed on your machine.

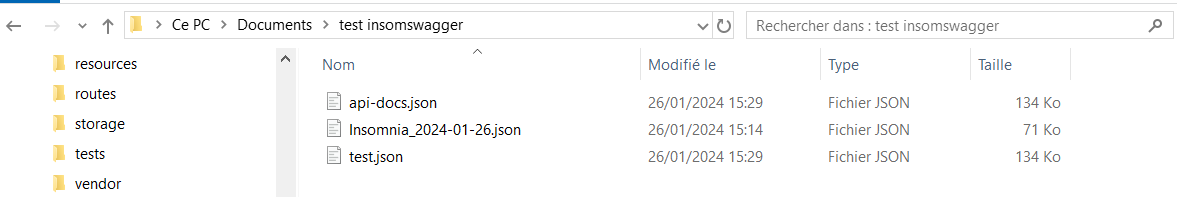
Then, in the terminal, access the folder containing the exported file. I exported it to my own folder. **C:\Users\simplon\Documents\test** insomswagger

****

run the command:

insomswagger -j [Name of exported file]]. 

This command creates a default **api-docs.json** file.



Create a **folder** named "**api-docs**" in the Laravel project in the "storage" directory. Paste the file you've created into the "**storage/api-docs/api-docs.json**" folder.

Once you've done that, customize the information linked to your API in the "info" section of the api-docs.json file:

"info": {

"title": "Your API Title",

"description": "Your API Description",

"version": "1.0.0"

},

Replace the title, description and version with your API information.

**View Swagger documentation with Darkaonline/L5-Swagger :**

To view the Swagger documentation for your Laravel project, use tools such as Darkaonline/L5-Swagger.

If L5-Swagger is not yet used in your Laravel project, you have two options:

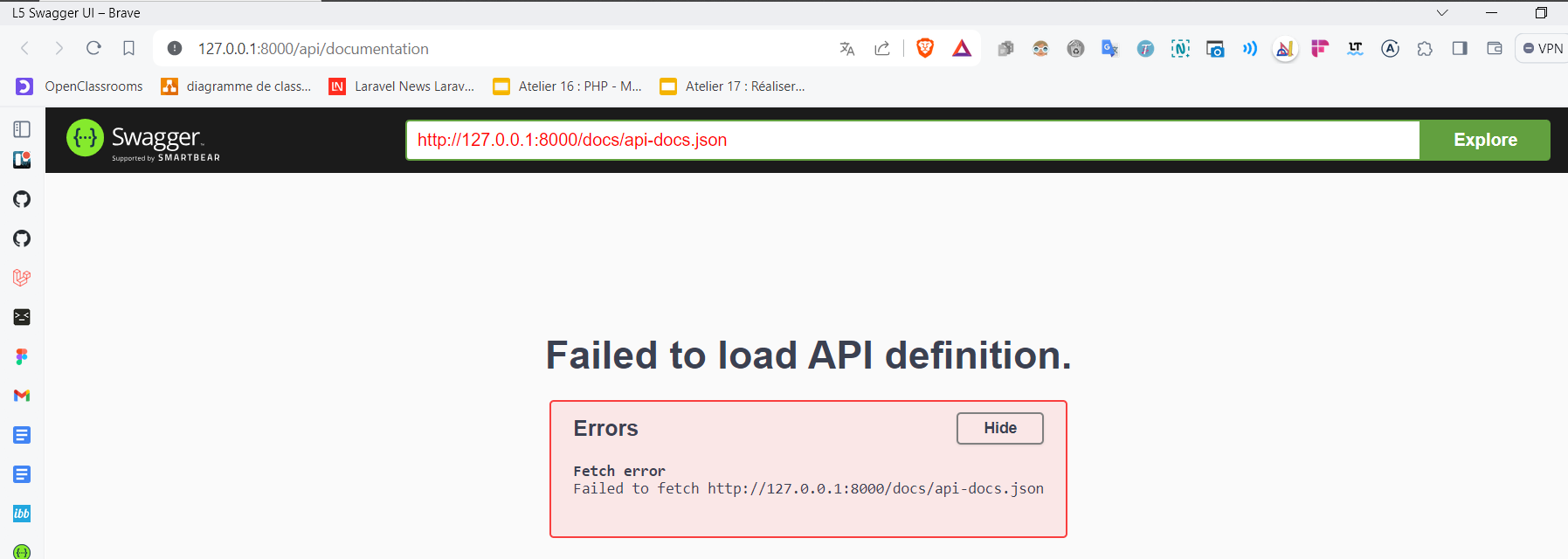
a. Follow the manual installation instructions in the [L5-Swagger](https://github.com/DarkaOnLine/L5-Swagger) repository.

b. Alternatively, run the following command to automate the installation of Swagger documentation with Darkaonline/L5-Swagger:

**insomswagger install**

Once the installation is complete, you'll see an interface in your browser with all your routes.

**NB**:



If, however, you see this error on your browser interface, don't panic, it simply means that the **api-docs** folder has not yet been created in the storage directory. You'll need to create it manually, and don't forget to include the **api-docs.json** file.

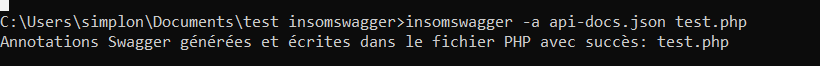
Now you can test your API on the documentation. To fill in the token, click on the green "**Authorize"** button, then paste your token in the appropriate field. All routes requiring authentication will automatically have the token, eliminating the need to copy and paste it each time.

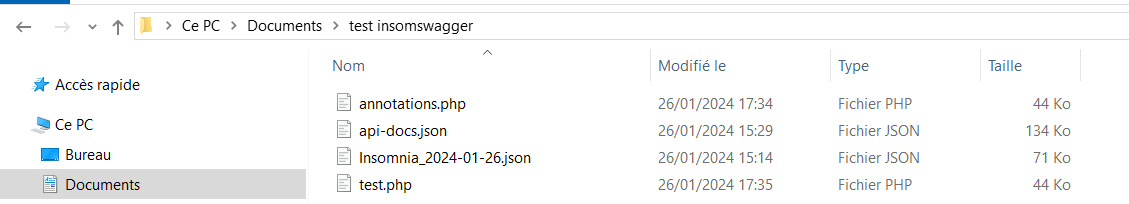
**PHP Swagger annotation generation :**

To generate PHP Swagger annotations from your created file, use the following command. Replace [File created] with the file you originally created and [New file to create] with the file you wish to create. If no output path is specified, the default name **./annotations.php** will be used.

**insomswagger -a [File created] [New file to be created]**







This command will generate PHP Swagger annotations. You can then take these annotations and manually add them to the various methods of your controller.

**YAML annotation generation :**

Similarly, to generate YAML annotations, use the following command:

**insomswagger -y [File created] [New file to be created]**

This will generate YAML annotations that you can use as required.

**Package update :**

If you wish to update the Insomswagger package, use the following command:

**insomswagger update**

This will ensure that you are using the most recent version of the package with the latest features.

**To Contribute and Report Problems :**

We encourage contributions from the community to improve this documentation and report any problems. Here's how you can get involved:

To suggest features, report bugs or problems, please visit our [**GitHub**](https://github.com/bamagid/insom-swagger.git) page.

You can also submit pull-requests to suggest improvements directly in the code.

For package documentation on npm visit our [NPM](https://www.npmjs.com/package/insomswagger?activeTab=readme) package

Your commitment is invaluable in helping us maintain and improve this documentation. We appreciate all your contributions!

©2024 We are developers