C1 - MVVM frameworks

C-DEV-121

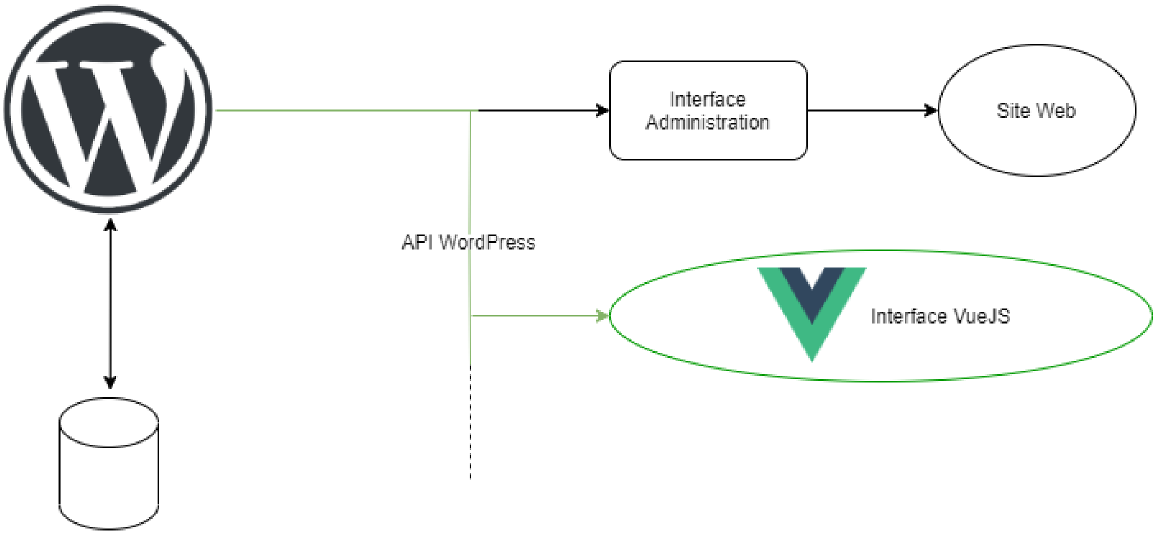
TWP

Bootstrap

Vue.js

1.2

You're going to set up a first functional Vue.js / WordPress stack.



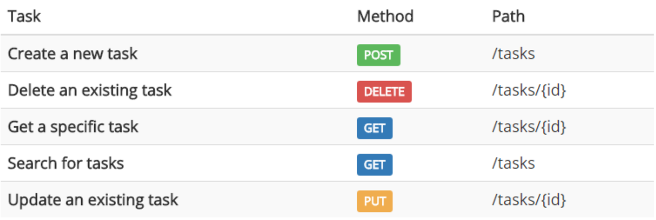
Node and npm are required dependencies for this bootstrap.

It is strongly advised to inquire about the npm files (package[-lock].json | node_modules) before the start of the project.

REMINDERS

CRUD

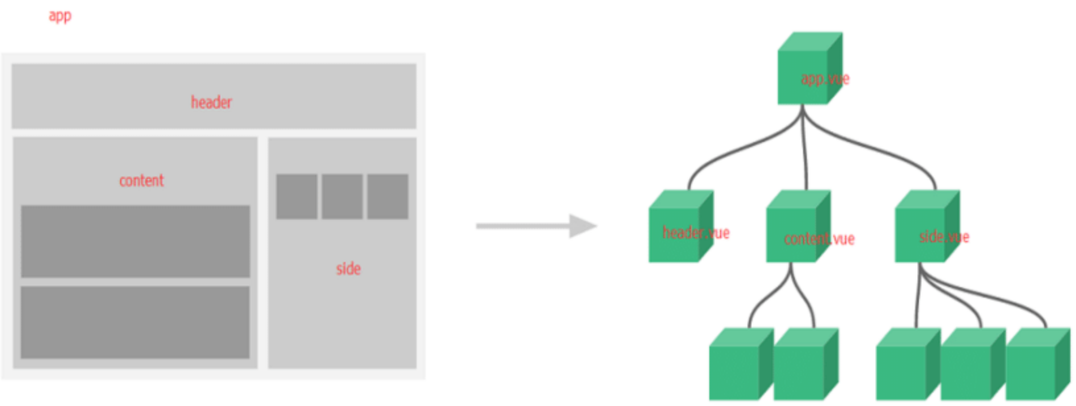
Create, Read, Update et Delete (CRUD) are the four basic functions of an entity.

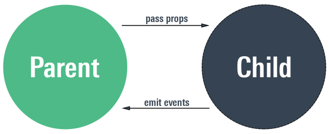


Task	Method	Path
Create a new task	POST	/tasks
Delete an existing task	DELETE	/tasks/{id}
Get a specific task	GET	/tasks/{id}
Search for tasks	GET	/tasks
Update an existing task	PUT	/tasks/{id}

In a REST environment, a CRUD corresponds to POST, GET, PUT, and DELETE requests.

VUE.JS COMPONENTS REMINDER

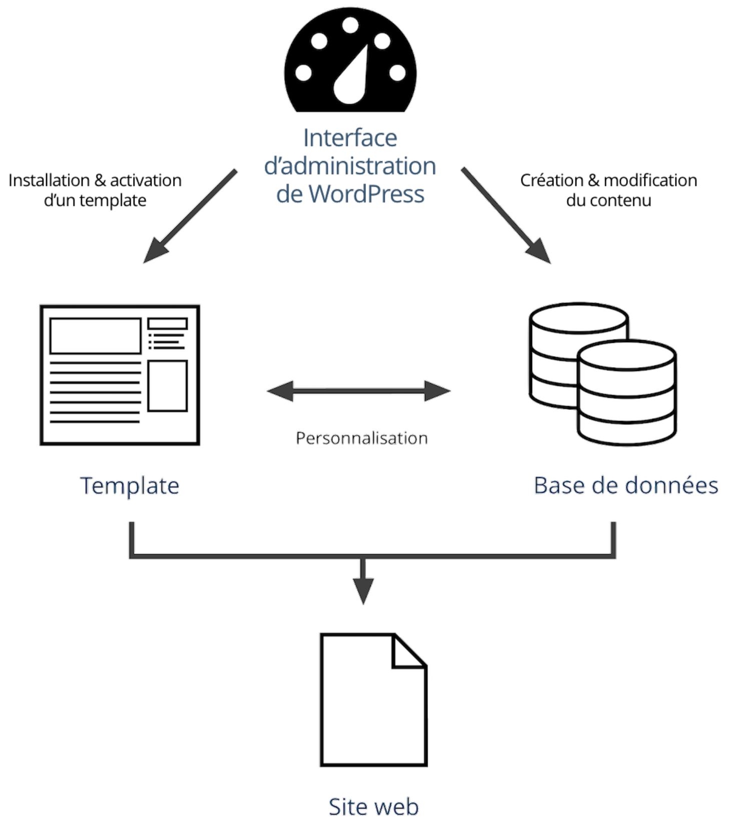




WordPress

Wordpress is a CMS that allows you to easily create sites with easily customizable interfaces.

The REST WordPress API is an interface that allows applications to interact with your WordPress site by sending and receiving data as JSON objects.


Interface d'administration de WordPress
Installation & activation d'un template
Création & modification du contenu
Template
Personnalisation
Base de données
Site web

The REST API is a WordPress feature for developers. It provides access to your site's data and allows you to interact with all of your site's content from a third-party client.

Installing WordPress

- First, make sure that the latest version of PHP is installed on your system and that you have a working database system (MySQL, for instance).

- Then, create a database and a user with full rights on this database.

- Finally, install WordPress.

- Once the installation is over, you should access the admin interface on the route `/wp-admin/`.

- In addition, you should access to the API routes, like `/wp-json/wp/v2/posts`, for instance.

Have a closer look to the JSON structure returned by the API.

Exploring WordPress API

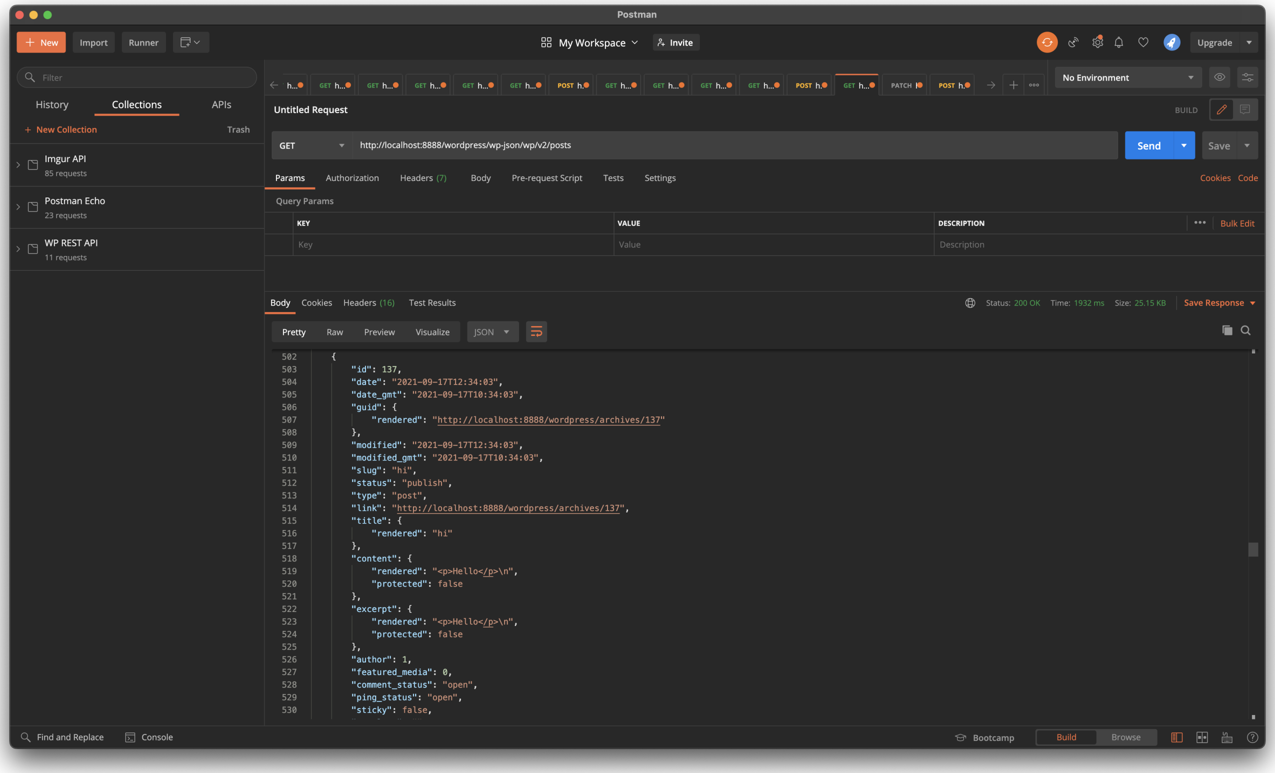
Use Postman to explore the API, you should be able, at least, to:

- See the `Posts`
- See a particular `Posts`
- Create a `Posts`
- See the `Comments` of a `Posts`
- Create a `Categories`
- ...

All API requests are detailed in the official documentation Wordpress API.

Get all of your posts

First make sure that your WordPress is running, so you are able to use the API provided by your website.
Then to try your API, open **Postman**. As explained before, the route to get all of your posts would be : `/wp-json/wp/v2/posts`
When sent, you should receive a JSON looking like this :



GET http://localhost:8888/wordpress/wp-json/wp/v2/posts

Status: 200 OK Time: 1932 ms Size: 25.15 KB

```
    {
        "id": 137,
        "date": "2021-09-17T12:34:03",
        "date_gmt": "2021-09-17T10:34:03",
        "guid": {
            "rendered": "http://localhost:8888/wordpress/archives/137"
        },
        "modified": "2021-09-17T12:34:03",
        "modified_gmt": "2021-09-17T10:34:03",
        "slug": "hi",
        "status": "publish",
        "type": "post",
        "link": "http://localhost:8888/wordpress/archives/137",
        "title": {
            "rendered": "hi"
        },
        "content": {
            "rendered": "<p>Hello</p>\n",
            "protected": false
        },
        "excerpt": {
            "rendered": "<p>Hello</p>\n",
            "protected": false
        },
        "author": 1,
        "featured_media": 0,
        "comment_status": "open",
        "ping_status": "open",
        "sticky": false,
```

You can see different elements such as :

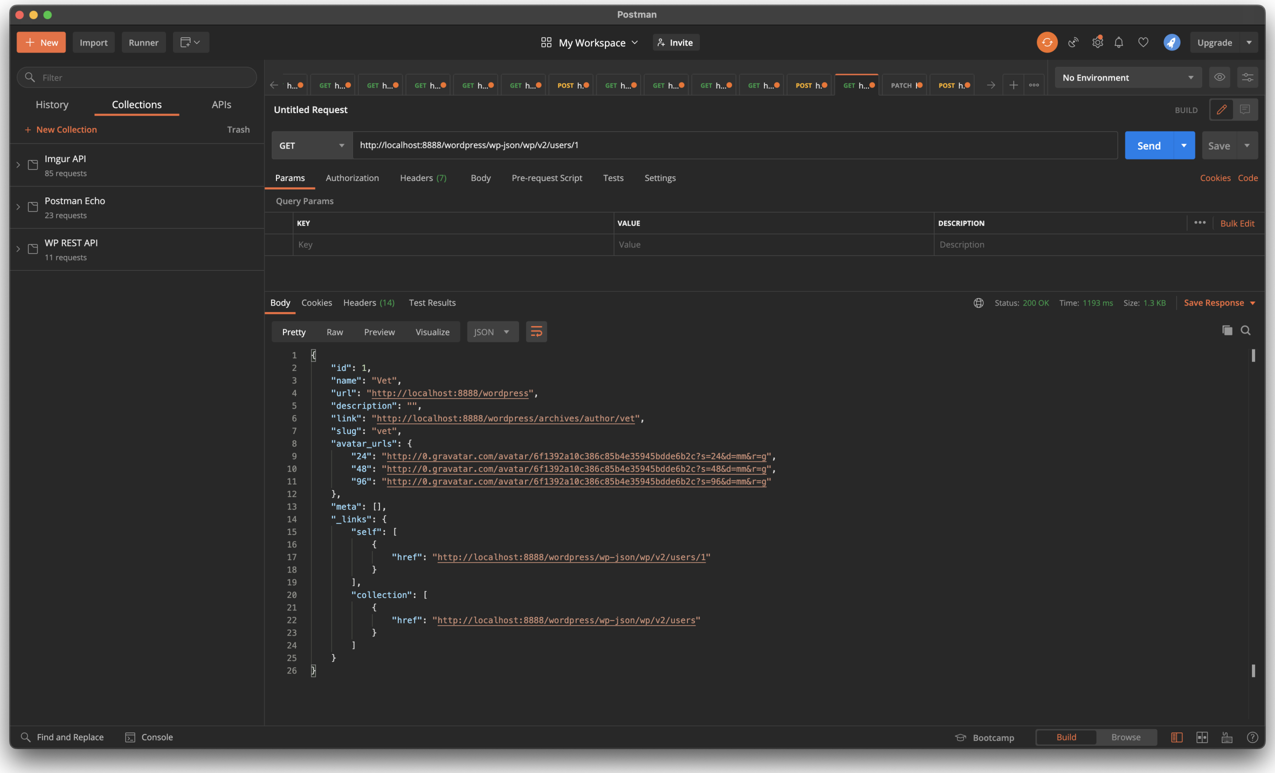
- `id` that would be your unique identifier to select a single post.
- `title` that contains `rendered` which is the title of your post.
- `content` that contains `rendered` which is all of the content of your post.
- `author` which is an id.

GET A SPECIFIC POST

Like the previous request the route of the API would be almost the same.
You will just need to specify the `id` of the post that you want.
The API call become `/wp-json/wp/v2/posts/id`
This method allows you to get a single post from the given `id`.

Get the author of a post

Once you retrieve the author id in a given post, you can then use another API call to get the full informations about the author.
The API call become `/wp-json/wp/v2/users/author`
Here from our previous call we had *author = 1* so the call would be :
`/wp-json/wp/v2/users/1`
The JSON you will receive contains the name of the author that can be useful to display later on.



GET http://localhost:8888/wordpress/wp-json/wp/v2/users/1

Status: 200 OK Time: 1193 ms Size: 1.3 KB

```
{
    "id": 1,
    "name": "Vet",
    "url": "http://localhost:8888/wordpress",
    "description": "",
    "link": "http://localhost:8888/wordpress/archives/author/vet",
    "slug": "vet",
    "avatar_urls": {
        "24": "http://0.gravatar.com/avatar/6f1392a10c386c85b4e35945bdde6b2c?s=24&d=mm&r=g",
        "48": "http://0.gravatar.com/avatar/6f1392a10c386c85b4e35945bdde6b2c?s=48&d=mm&r=g",
        "96": "http://0.gravatar.com/avatar/6f1392a10c386c85b4e35945bdde6b2c?s=96&d=mm&r=g"
    },
    "meta": [],
    "_links": {
        "self": [
            {
                "href": "http://localhost:8888/wordpress/wp-json/wp/v2/users/1"
            }
        ],
        "collection": [
            {
                "href": "http://localhost:8888/wordpress/wp-json/wp/v2/users"
            }
        ]
    }
}
```

Create a Vue.js application

- Create a new Vue.js application named “MVVM_TWP”.
  Everything that will be done afterwards will be done in this folder.
- Add the dependencies vue-router and vuex to your project.
- Perform API queries with your Vue.js application. Tools like Axios and Fetch are handful.

WordPress API requests need parameters and inputs as outputs are JSON.

Encapsulating the exchanges with the API can make things easier for you.
Reinventing the wheel will decrease your productivity: look for some npm packages which could simplify your work and/or your interactions with the WordPress API.

Achievements

Congratulations! You now have all the basics to interact with the most widely used web solution in the world. You should dig deeper now, because you will probably do it later!