



pokedex\_vue

Create a website with vue.js

*Summary: This document is the subject for vue.js @42Tokyo.*

# Contents

<b>I</b>	<b>Instructions</b>	<b>2</b>
<b>II</b>	<b>Foreword</b>	<b>3</b>
<b>III</b>	<b>Exercise 00 : NPM</b>	<b>4</b>
<b>IV</b>	<b>Exercise 01 : Pokedex</b>	<b>5</b>
<b>V</b>	<b>Exercise 02 : Pokemon</b>	<b>7</b>
<b>VI</b>	<b>Bonus part</b>	<b>9</b>

# Chapter I

## Instructions

- Only this page will serve as reference; do not trust rumors.
- Watch out! This document could potentially change up to an hour before submission.
- Make sure you have the appropriate permissions on your files and directories.
- You have to follow the submission procedures for every exercise.
- Your exercises will be checked and graded by your fellow classmates.
- You cannot leave any additional file in your directory than those specified in the subject.
- Got a question? Ask your peer on the right. Otherwise, try your peer on the left.
- Your reference guide is called `Google / man / the Internet / ....`
- Examine the examples thoroughly. They could very well call for details that are not explicitly mentioned in the subject...
- your vue version should be  $\geq 2.5.16$ .
- You are allowed to develop this site on your own environnement.
- Evaluation will be done on the reviewees host.
- You are allowed to cutomize the visual for this website.


# Chapter II

## Foreword

Vue is a progressive framework for building user interfaces. Unlike other monolithic frameworks, Vue is designed from the ground up to be incrementally adoptable. The core library is focused on the view layer only, and is easy to pick up and integrate with other libraries or existing projects. On the other hand, Vue is also perfectly capable of powering sophisticated Single-Page Applications when used in combination with modern tooling and supporting libraries.

# Chapter III

## Exercise 00 : NPM

	Exercise 00
NPM	
Turn-in directory : <i>ex00/</i>	
Files to turn in : *	
Allowed functions : *	


- Install vue/cli by using npm.
- Execute 'vue create pokedex' and make sure you chose 'Default ([Vue 2] babel, eslint)'.

```
? Please pick a preset: (Use arrow keys)
> Default ([Vue 2] babel, eslint)
  Default (Vue 3 Preview) ([Vue 3] babel, eslint)
  Manually select features
```

- Execute 'cd pokedex && npm run serve' and then go to the browser and type 'http://localhost:8080/'.
- If you see the Welcome page then You are done with this first exercise.

# Chapter IV

## Exercise 01 : Pokedex

	Exercise 01
pokedex	
Turn-in directory : <i>ex01/</i>	
Files to turn in : *	
Allowed functions : *	

- Inside the components folder, delete the `HelloWorld.vue` component and create a file called `Pokedex.vue`.
- Replace the `App.vue` file with this content below.

```
<template>
  <div id="app">
    <Pokedex/>
  </div>
</template>

<script>
import Pokedex from './components/Pokedex.vue'

export default {
  name: 'App',
  components: {
    Pokedex
  }
}
</script>
```

- `Pokedex.vue` component will list all the pokemon available on
- <https://pokeapi.co/api/v2/pokemon>.
- This component should display the pokemon id, name and the image.
- While it's fetching the data from that endpoint, it should display 'Loading...' on the screen.
- If Any axios related error occurs, this component should display the axios related error message on the screen.




axios



lifecycle

# Chapter V

## Exercise 02 : Pokemon

	Exercise 02
Pokemon	
Turn-in directory : <i>ex02/</i>	
Files to turn in : *	
Allowed functions : *	

- Execute 'npm install vue-router'.
- Inside the src folder, create a `router.js` file and copy paste the content below.

```
import Vue from 'vue'
import Router from 'vue-router'
import Pokedex from './components/Pokedex'
import Pokemon from './components/Pokemon'

Vue.use(Router)

export default new Router({
  routes: [
    {
      path: '/',
      name: 'pokedex',
      component: Pokedex
    },
    {
      path: '/pokemon/:id',
      name: 'pokemon',
      component: Pokemon
    }
  ],
  mode: 'history'
})
```



- Replace the `main.js` file with this content below.

```
import Vue from 'vue'
import App from './App.vue'
import router from './router.js'

Vue.config.productionTip = false

new Vue({
  router,
  render: h => h(App),
}).$mount('#app')
```

- Replace the `App.vue` file with this content below.

```
<template>
<div id="app">
  <router-view/>
</div>
</template>

<script>

export default {
  name: 'App',
}
</script>
```

- Create a new component called `Pokemon.vue` Inside the component folder.
- This component should display the pokemon id, name, image, type(grass, poison, etc...) and base stats(hp attack defense special-attack special-defense speed) of a specific pokemon.
- While it's fetching the data from that endpoint, it should display 'Loading...' on the screen.
- If Any axios related error occurs, this component should display the axios related error message on the screen.
- whenever a user click on pokemon's name on the Pokedex Component it should render the Pokemon component of that pokemon on the screen.
- On the top left of the Pokemon component make sure there is a link to go back to the Pokedex Component.

# Chapter VI

## Bonus part

- If the Mandatory part is not perfect don't even think about bonuses
- Add a search bar.
- Add a filter bar.
- Feel free to add more feature which might be useful.