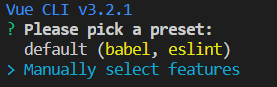
# Simple Workshop: Vuex – To Do List

## Creating the Project

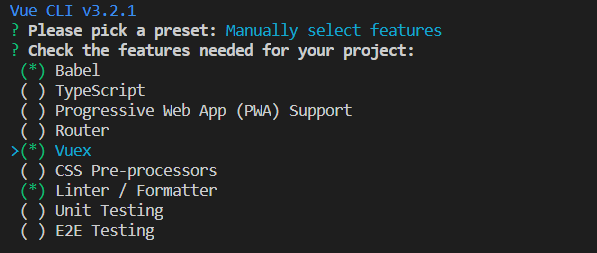
Open the terminal in your project folder and and run the following command to start a new Vue.js project.



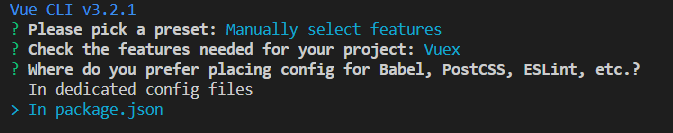
After that, you will be presented with the following options: Choose the “**Manually select features**” one.



Upon clicking the selected option, you will be presented with the following prompt. Babel and Linter are selected by default. Choose “**Vuex**” (with keyboard up and down keys and space to select).



Then you will be asked where you want the **config files** to be placed. Choose “**In package.json**”.



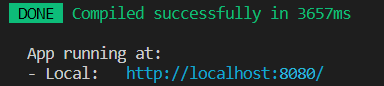
After successfully having created a project, to run it, type the following command:



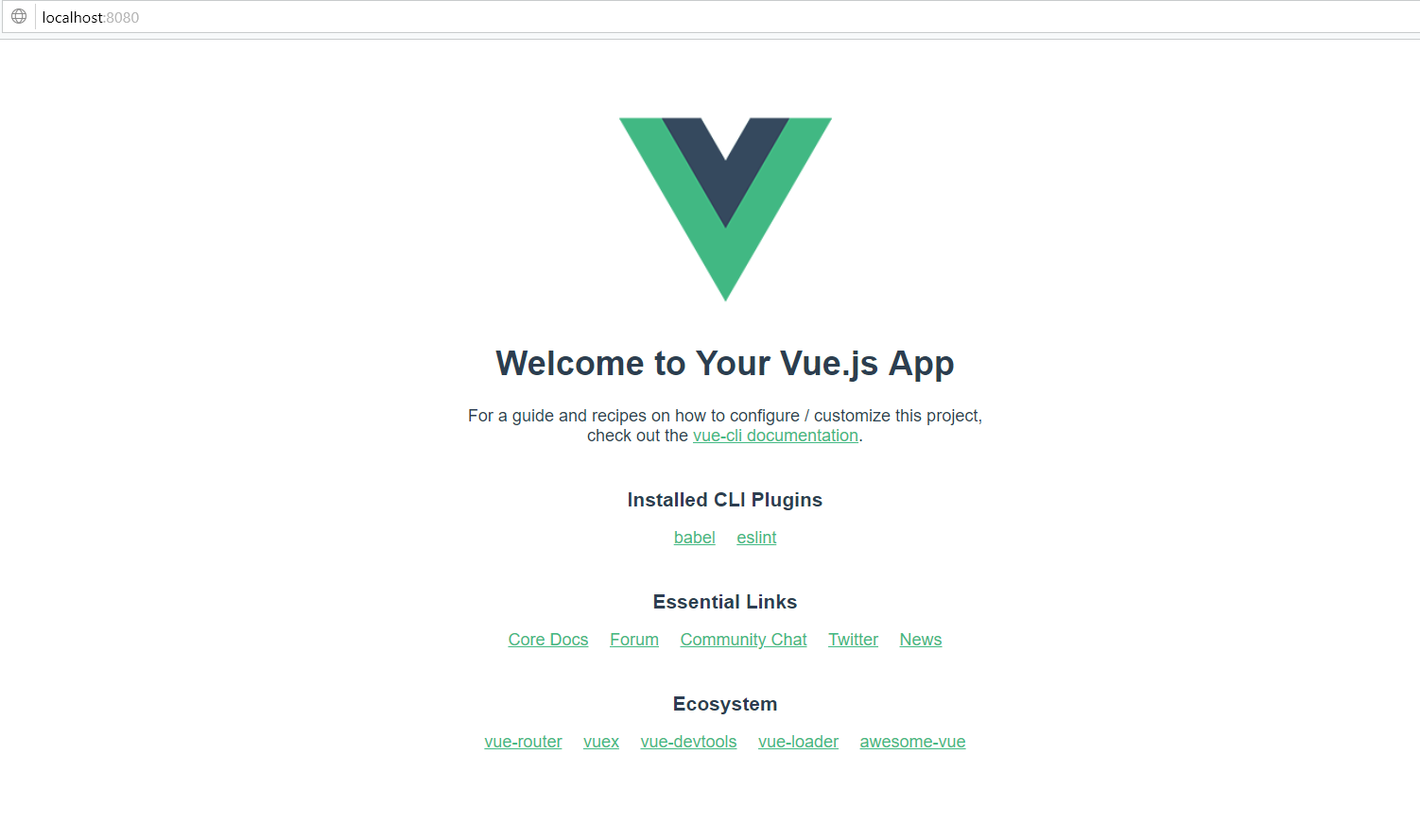
And then:



Something like that should appear:



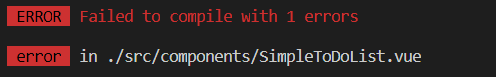
If we go to **http://localhost:8080**, we will see that a default app has been created.



## Creating our Vue

In the **src\components** folder that has been created for us, we have to create the template for our app. Just go to the components folder and rename **HelloWorld.vue** to **SimpleToDoList.vue** and delete the content in it, as we will create our own very simple template there.

Because the HelloWorld.vue is connected to the App.vue, you will receive an error after renaming the file.

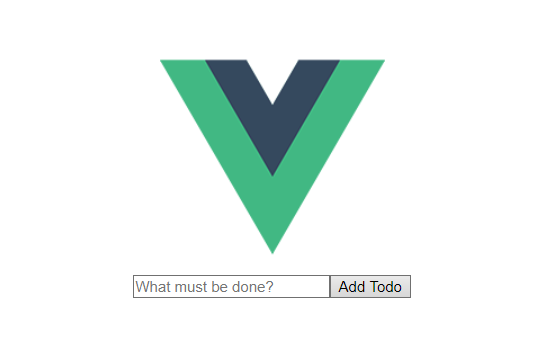


Go to **App.vue** and make sure that you have **replaced every HelloWorld with SimpleToDoList**.

Then, to create our own template, we will need to use a template tag which will contain a form where we will write our tasks and a list to append them. Use the following template:



Now, if we go to **http://localhost:8080,** this should appear:

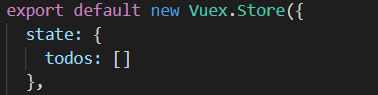


However, it has **no functionality**.

## Creating the Functionality

To create the functionality, it is time to use the **Vuex store**. Note that the **store variable names and functions have to be with the same name as the ones in the template**, otherwise it won’ t work.

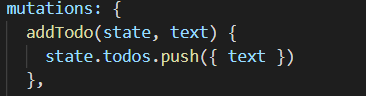
To begin with, we need to create an empty array for our To Do List in the state. Go to **store.js** file and **create an empty array in the state**:



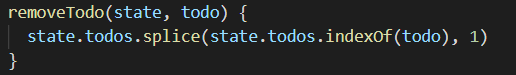
Then, it is time for the **mutations**. There, we need to create **two functions**:

One that **adds our tasks** to the list and one that **removes them**.

Let’ s start with adding the tasks. It is very simple. We just need to **add the input we receive** in the form **to the array** we just created. We do that by using the following code:



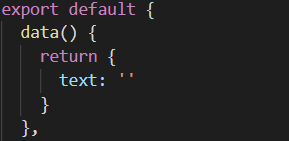
After that, it is time to think of a way that can **remove our tasks** from the array. We do that by doing the following operation:



However, if you go to **http://localhost:8080**, we will see that it is **still not working**. That’ s because we **haven’ t linked the store to the template** yet. Go to **SimpleToDoLost.vue** again and below the template tag, open a script one. In it, we should **import the store** we have just created.



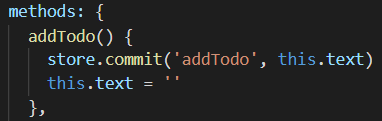
Now that we have access to the store, we should export our data by doing the following:



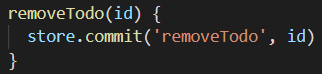
Then, we need to use **computed** to **get our array**:



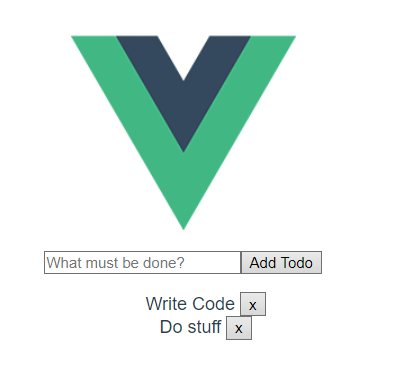
Finally, we use the methods to create the main functionality. In order to add to our array, we need to use the **store.commit** function. To reset the value of the input field, we use this.text = ‘’.



Our remove function will receive one parameter – the id of the task we want to remove.



Finally, if we go to **http://localhost:8080**, we should have a simple working app. First, we can test the add function.



And then the remove one:

