Modify router app with below requirement

When employee name will be clicked it should pass respective employee id as a param to Employee compopnent and Employee component should call get method from the API to fetch details of respective employee

HTTP Get in Vue app using Axios

Asynchronous calls –

Promises – promises allows you defer next actions till previous actions are getting completed. If request

Is successful it returns the result i.e data else it give you error.

Axios is used to handle this promise based response which is coming from ajax calls i..e asynchronous calls. This avoids reloading of page and you can update UI with updated data by this call.

Promise is an object that represents state of your operations, if operation is successful it returns data in form of response else it return error.

To make asynchronous calls and to handle promise based response we should use Axios in vue. It is flexible HTTP client library that uses promises by default and run on both client and server side. IT makes appropriate fetching of data during server side rendering.

To fetch data from server

npm install -g json-server

create Products.json and run the following command

json-server --watch -p 8000 Products.json

**Assignment**

Provide Add Product button in the product.vue template, on click of Add Product should display a form

To accept information of new product and on click on submit should add new product to api.

Use axios.post()

Vuex –

Vue component role is to represent the information.

If we try to store the state in vue component, component is performing 2 roles i.e it is holding state as well as it is representing it.

Why do you need any architectural pattern ?

It has a principal i.e separation of concern.

In case if Vue if will make use of component they will be used only for presenting information.

And to store the information or to maintain state of your application you should have something else.

We will use Vuex as state management patten or library from vue to store and handle the state.

Vuex should have store to store the state and it should also be responsible for providing the logic to get and update the state depending on user action.

So if we want to store employee array as a state we will store in the store of vuex and vuex will allow us to write a logic for getting all employee details, single employee details, to update employee and to delete employee.

Vuex has centralized store to manage the state and all components from your application will be able to access the state from this store.

When you make use of this centralized store of Vuex and its composition API it looks like below

createStore() need to be imported from Vuex for creating centralized store for storing and managing state of you app.

createStore({

state{

//store the state

},

mutations{

//write methods for updating state synchronously

},

actions{

//write methods to update state asynchronously depending on user action

}

modules{

//if you want to maintain multiple state for your application

}

})

You have to use Vue component for presenting the state and for providing payload to store for updating the state.

So if you want the component should be provided with updated state always then access the state from store in component using its computed property.

To send the payload from component to store component has to use store.commit with mutations method name

**Vuex Assignment**

Maintain products array as state in Vuex store

In ProductComponent access array of products from store and in template provide a form to accept new product details and a button addProduct. On click of addProduct state in store should be update with new product.

**Reference Links**

**Module 10 : Http Request:**

* Http Requests & Http Methods (Verbs)
* Using Axios Instead Of "fetch()"
* Getting Data (GET Request) & Transforming Response Data

<https://www.techiediaries.com/vue-3-fetch-data-rest-api/>

<https://codesource.io/getting-started-with-vue-3-composition-api/>

* Videos :

<https://www.youtube.com/watch?v=NHfyF0B4y8E>

**Module 12 : Vuex**

* What & Why?

<https://next.vuex.vuejs.org/>

* Installation

<https://next.vuex.vuejs.org/installation.html>

* Creating & Using a Store

<https://next.vuex.vuejs.org/guide/#the-simplest-store>

* Connecting Components to State

<https://next.vuex.vuejs.org/guide/state.html>

* Introducing Mutations - A Better Way of Changing Data
* Passing Data to Mutations with Payloads

<https://next.vuex.vuejs.org/guide/mutations.html>

* Introducing Getters - A Better Way Of Getting Data

<https://next.vuex.vuejs.org/guide/getters.html>

* Running Async Code with Actions

<https://next.vuex.vuejs.org/guide/actions.html>

* Using Mapper Helpers

<https://itnext.io/mapping-for-vuex-beginners-43656ac41de9>

* Videos :

<https://www.youtube.com/watch?v=ksDmbBqpw2A>

<https://morioh.com/p/6d0136e4a2ff>