

# Vue Typescript

## CRUD Application

### Vue Router - Axios

## Vue Typescript example: Build a CRUD Application

📅 Last modified: December 18, 2020 (<https://bezcoder.com/vue-typescript-crud/>) 👤 bezcoder (<https://bezcoder.com/author/bezkoder/>) 📁 Vue.js (<https://bezcoder.com/category/vue/>)

In this tutorial, I will show you how to build a Vue.js Typescript CRUD Application to consume REST APIs, display and modify data using Vue, Vue Router, Axios & Bootstrap.

More Practice:

- Vue/Vuex Typescript: JWT Authentication example (<https://bezcoder.com/vuex-typescript-jwt-auth/>)
- Vue File Upload example using Axios (<https://bezcoder.com/vue-axios-file-upload/>)

Fullstack:

- Vue.js + Node.js + Express + MySQL example (<https://bezcoder.com/vue-js-node-js-express-mysql-crud-example/>)

We use cookies to improve your experience with the site. To find out more, you can read the full [Privacy & Policy](https://bezcoder.com/privacy-policy/)

- Vue.js + Node.js + Express + MongoDB example (<https://bezcoder.com/vue-node-js-express-mongodb-crud-example/>)

Accept

[express-mongodb-mevn-crud/](#))

– Vue.js + Spring Boot + MySQL/PostgreSQL example (<https://bezkoder.com/spring-boot-vue-js-crud-example/>)

– Vue.js + Spring Boot + MongoDB example (<https://bezkoder.com/spring-boot-vue-mongodb/>)

– Vue.js + Django example (<https://bezkoder.com/django-vue-js-rest-framework/>)

## Contents [hide]

Overview of Vue Typescript CRUD Application

Vue Typescript App Component Diagram

Technology

Project Structure

Setup Vue.js Typescript Project

Add Vue Router to Vue Typescript App

Add Navbar and Router View to Vue Typescript App

Initialize Axios for Vue Typescript CRUD HTTP Client

Create Data Service

Create Vue Components

    Add item Component

    List of items Component

    Item details Component

Configure Port for Vue Typescript CRUD App

Run Vue Typescript App

Conclusion

Further Reading

Source Code

## Overview of Vue Typescript CRUD Application

We will build a Vue.js front-end Tutorial Application in that:

- Each Tutorial has id, title, description, published status.
- We can create, retrieve, update, delete Tutorials.
- There is a Search bar for finding Tutorials by title.

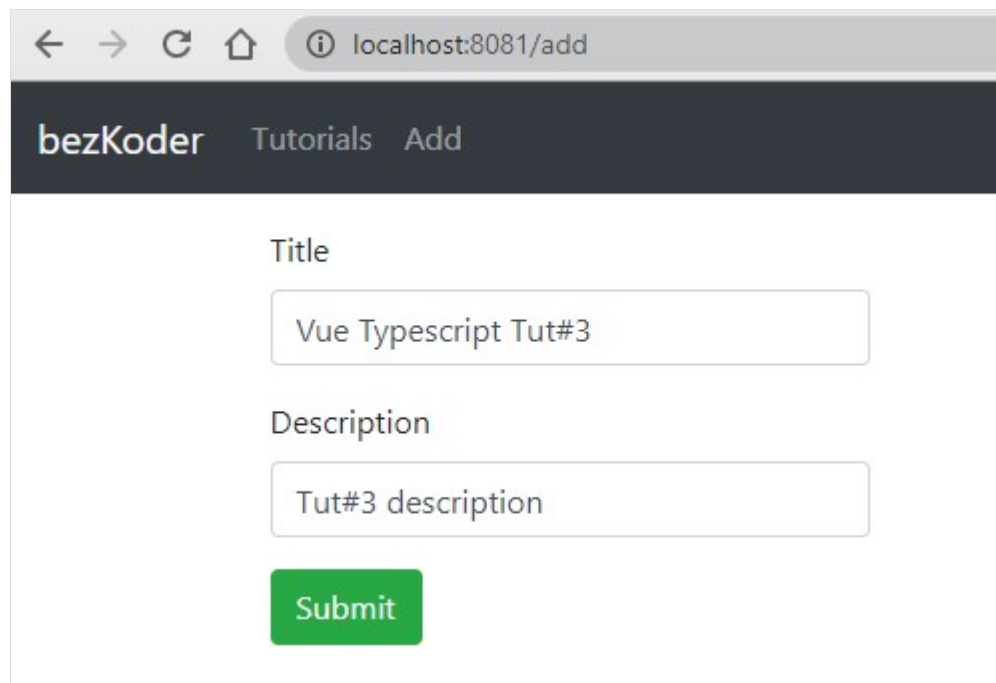
Here are screenshots of our Vue Typescript CRUD Application.

– Create an object:

We use cookies to improve your experience with the site. To find out more, you can read the full

[Privacy & Policy \(https://bezkoder.com/privacy-policy/\)](https://bezkoder.com/privacy-policy/)

Accept



← → ↻ 🏠 ⓘ localhost:8081/add

bezKoder Tutorials Add

Title

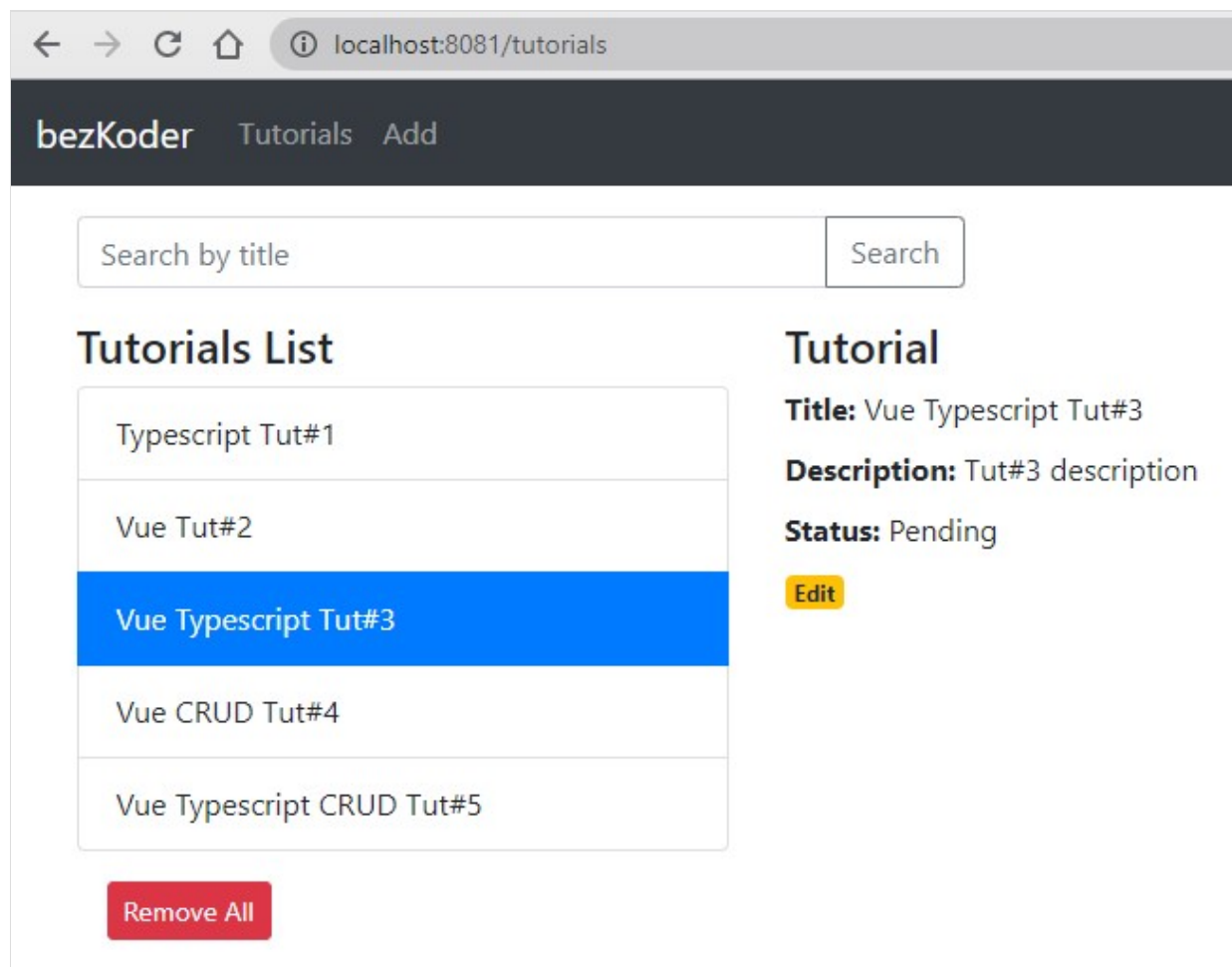
Vue Typescript Tut#3

Description

Tut#3 description

Submit

– Retrieve all objects:



← → ↻ 🏠 ⓘ localhost:8081/tutorials

bezKoder Tutorials Add

Search by title Search

### Tutorials List

Typescript Tut#1
Vue Tut#2
Vue Typescript Tut#3
Vue CRUD Tut#4
Vue Typescript CRUD Tut#5

Remove All

### Tutorial

**Title:** Vue Typescript Tut#3

**Description:** Tut#3 description

**Status:** Pending

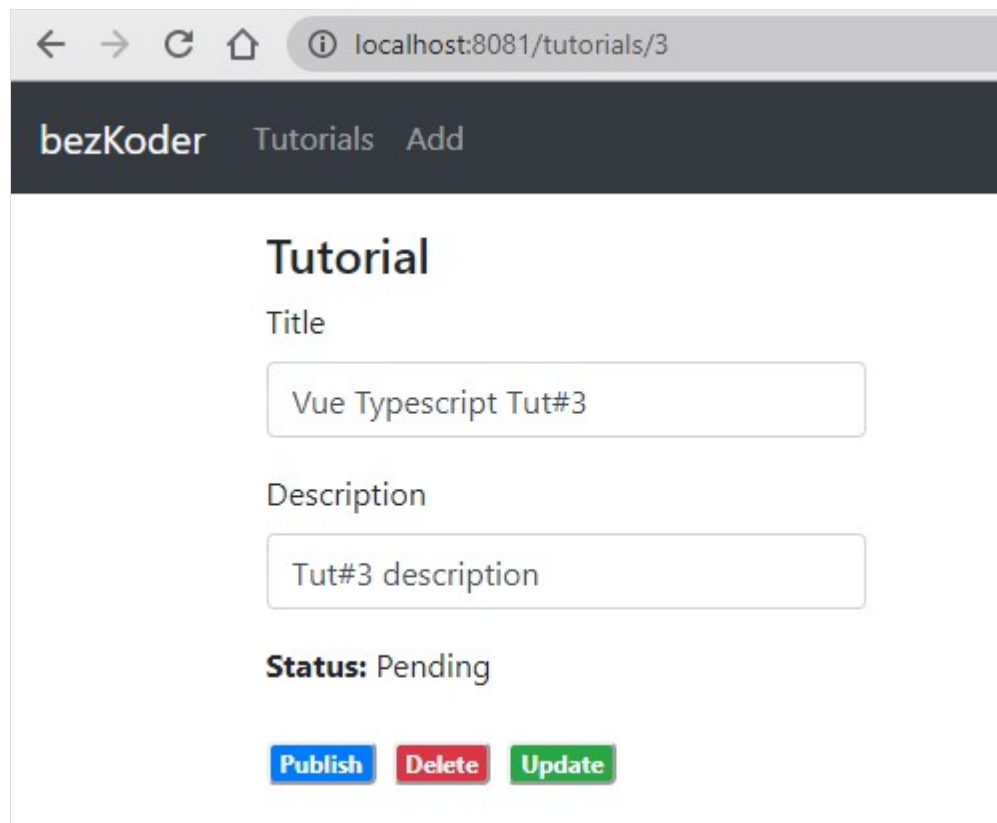
Edit

– Click on **Edit** button to update an object:

We use cookies to improve your experience with the site. To find out more, you can read the full

[Privacy & Policy \(https://bezcoder.com/privacy-policy/\)](https://bezcoder.com/privacy-policy/)

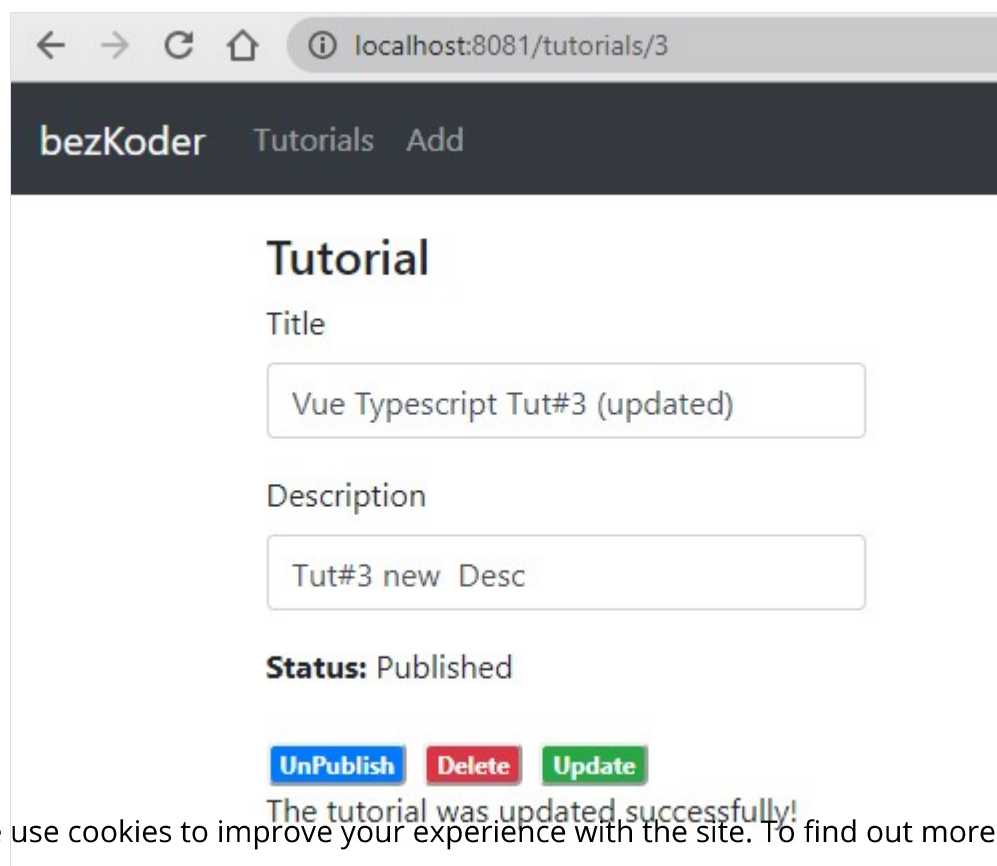
Accept



A screenshot of a web browser at localhost:8081/tutorials/3. The page has a dark header with 'bezKoder', 'Tutorials', and 'Add' links. The main content area is titled 'Tutorial'. It contains two text input fields: 'Title' with the value 'Vue Typescript Tut#3' and 'Description' with the value 'Tut#3 description'. Below these fields, the status is 'Status: Pending'. At the bottom, there are three buttons: 'Publish' (blue), 'Delete' (red), and 'Update' (green).

On this Page, you can:

- change status to **Published** using **Publish** button
- delete the Tutorial using **Delete** button
- update the Tutorial details with **Update** button



A screenshot of the same web browser, but the tutorial has been updated. The 'Title' field now contains 'Vue Typescript Tut#3 (updated)' and the 'Description' field contains 'Tut#3 new Desc'. The status has changed to 'Status: Published'. The buttons are now 'UnPublish' (blue), 'Delete' (red), and 'Update' (green). A green success message 'The tutorial was updated successfully!' is displayed at the bottom of the form area.

We use cookies to improve your experience with the site. To find out more, you can read the full

[Privacy & Policy \(https://bezcoder.com/privacy-policy/\)](https://bezcoder.com/privacy-policy/)

Accept

– Search Tutorials by title:

The introduction above is for Vue.js Client with assumption that we have a Server exporting REST APIs:

Methods	Urls	Actions
POST	/api/tutorials	create new Tutorial
GET	/api/tutorials	retrieve all Tutorials
GET	/api/tutorials/:id	retrieve a Tutorial by :id
PUT	/api/tutorials/:id	update a Tutorial by :id
DELETE	/api/tutorials/:id	delete a Tutorial by :id
DELETE	/api/tutorials	delete all Tutorials
GET	/api/tutorials?title=[keyword]	find all Tutorials which title contains keyword

You can find step by step to build a Server like this in one of these posts:

- Express, Sequelize & MySQL (<https://bezcoder.com/node-js-express-sequelize-mysql/>)
- Express, Sequelize & PostgreSQL (<https://bezcoder.com/node-express-sequelize-postgresql/>)

– Express & MongoDB (<https://bezcoder.com/node-express-mongodb-crud-rest-api/>)

We use cookies to improve your experience with the site. To find out more, you can read the full

– Spring Boot & MySQL (<https://bezcoder.com/spring-boot-jpa-crud-rest-api/>)

– Spring Boot & PostgreSQL (<https://bezcoder.com/spring-boot-postgresql-example/>)

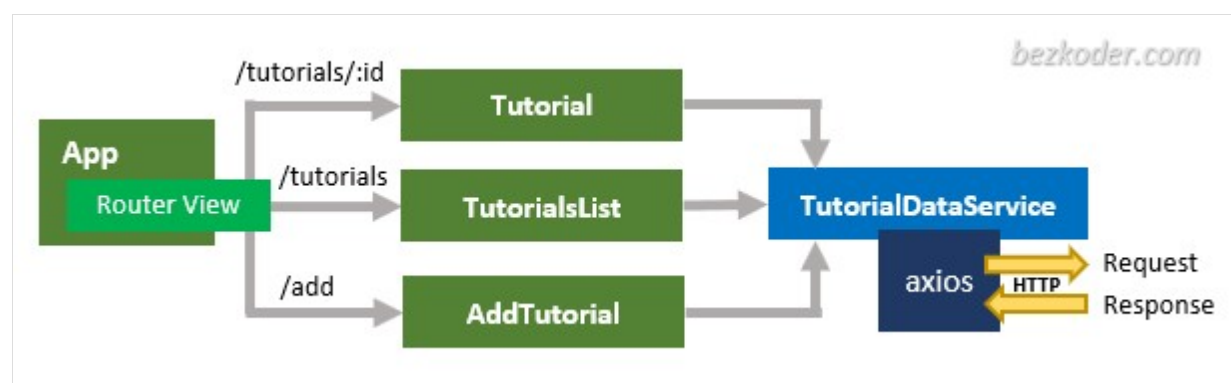
[Privacy & Policy](#) (<https://bezcoder.com/privacy-policy/>)

[Accept](#)

- Spring Boot & MongoDB (<https://bezcoder.com/angular-spring-boot-mongodb/>)
- Spring Boot & H2 (<https://bezcoder.com/spring-boot-jpa-h2-example/>)
- Spring Boot & Cassandra (<https://bezcoder.com/spring-boot-cassandra-crud/>)
- Django & MySQL (<https://bezcoder.com/django-crud-mysql-rest-framework/>)
- Django & PostgreSQL (<https://bezcoder.com/django-postgresql-crud-rest-framework/>)
- Django & MongoDB (<https://bezcoder.com/django-mongodb-crud-rest-framework/>)

All of them can work well with this Vue Typescript App.

## Vue Typescript App Component Diagram



- The **App** component is a container with **router-view**. It has navbar that links to routes paths.
- **TutorialsList** component gets and displays Tutorials.
- **Tutorial** component has form for editing Tutorial's details based on `:id`.
- **AddTutorial** component has form for submission new Tutorial.
- These Components call **TutorialDataService** methods which use **axios** to make HTTP requests and receive responses.

## Technology

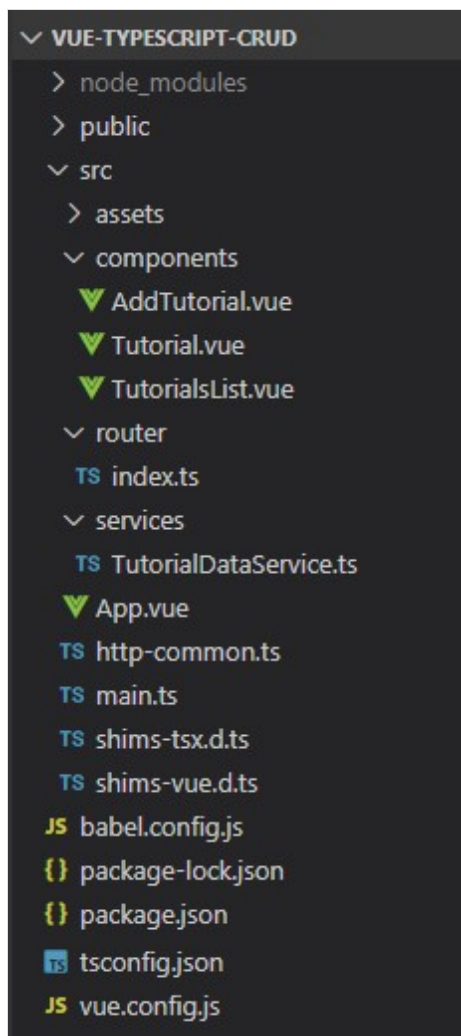
- vue: 2.6.11
- vue-router: 3.1.6
- axios: 0.19.2
- vue-class-component 7.2.3
- vue-property-decorator 8.4.1

## Project Structure

We use cookies to improve your experience with the site. To find out more, you can read the full

[Privacy & Policy \(https://bezcoder.com/privacy-policy/\)](https://bezcoder.com/privacy-policy/)

Accept



Let me explain it briefly.

- **package.json** contains main modules: `vue`, `vue-router`, `vue-class-component`, `axios`.
- There are 3 components: `TutorialsList`, `Tutorial`, `AddTutorial`.
- **router/index.ts** defines routes for each component.
- **http-common.ts** initializes axios with HTTP base Url and headers.
- `TutorialDataService` has methods for sending HTTP requests to the Apis.
- **vue.config.js** configures *port* for this Vue Client.

## Setup Vue.js Typescript Project

Open cmd at the folder you want to save Project folder, run command:

```
vue create vue-typescript-crud
```

You will see some options, choose **Manually select features**.

Select the following options:

We use cookies to improve your experience with the site. To find out more, you can read the full

[Privacy & Policy \(https://bezkoder.com/privacy-policy/\)](https://bezkoder.com/privacy-policy/)

Accept

```
? Please pick a preset: Manually select features
? Check the features needed for your project:
  (*) Babel
> (*) TypeScript
  ( ) Progressive Web App (PWA) Support
  ( ) Router
  ( ) Vuex
  ( ) CSS Pre-processors
  ( ) Linter / Formatter
  ( ) Unit Testing
  ( ) E2E Testing
```

Then configure the project like this:

```
? Please pick a preset: Manually select features
? Check the features needed for your project: Babel, TS
? Use class-style component syntax? Yes
? Use Babel alongside TypeScript (required for modern mode, auto-detected polyfills)? Yes
? Where do you prefer placing config for Babel, PostCSS, ESLint, etc.? In package.json
? Save this as a preset for future projects? (y/N) No
```

After the process is done. We create new folders and files like the following tree:

```
├── public
│   └── index.html
├── src
│   ├── components
│   │   ├── AddTutorial.vue
│   │   ├── Tutorial.vue
│   │   └── TutorialsList.vue
│   ├── services
│   │   └── TutorialDataService.ts
│   ├── App.vue
│   └── main.ts
└── package.json
```

Open **public/index.html**, add bootstrap inside `<head>` tag:

We use cookies to improve your experience with the site. To find out more, you can read the full

[Privacy & Policy \(https://bezcoder.com/privacy-policy/\)](https://bezcoder.com/privacy-policy/)

Accept



```
<!DOCTYPE html>
<html lang="en">
  <head>
    ...
    <title>Vue Typescript CRUD</title>
    <link type="text/css" rel="stylesheet" href="//unpkg.com/bootstrap/dist/css/
  </head>
  <body>
    ...
  </body>
</html>
```

## Add Vue Router to Vue Typescript App

- Run the command: `npm install vue-router` .
- In **src/router** folder, create *index.ts* and define `Router` as following code:

We use cookies to improve your experience with the site. To find out more, you can read the full

[Privacy & Policy \(https://bezcoder.com/privacy-policy/\)](https://bezcoder.com/privacy-policy/)

Accept

```
import Vue from "vue";
import VueRouter, { RouteConfig } from "vue-router";

Vue.use(VueRouter);

const routes: Array<RouteConfig> = [
  {
    path: "/",
    alias: "/tutorials",
    name: "tutorials",
    component: () => import("../components/TutorialsList.vue")
  },
  {
    path: "/tutorials/:id",
    name: "tutorial-details",
    component: () => import("../components/Tutorial.vue")
  },
  {
    path: "/add",
    name: "add",
    component: () => import("../components/AddTutorial.vue")
  }
];

const router = new VueRouter({
  mode: "history",
  base: process.env.BASE_URL,
  routes
});

export default router;
```

– Open **src/main.ts**, then import `router` :

```
import Vue from "vue";
import App from "./App.vue";
import router from "./router";

Vue.config.productionTip = false;

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

We use cookies to improve your experience with the site. To find out more, you can read the full

[Privacy & Policy \(https://bezcoder.com/privacy-policy/\)](https://bezcoder.com/privacy-policy/)

## Add Navbar and Router View to Vue Typescript App

Let's open **src/App.vue**, this **App** component is the root container for our application, it will contain a **navbar**.

```
<template>
  <div id="app">
    <nav class="navbar navbar-expand navbar-dark bg-dark">
      <a href="#" class="navbar-brand">bezKoder</a>
      <div class="navbar-nav mr-auto">
        <li class="nav-item">
          <a href="/tutorials" class="nav-link">Tutorials</a>
        </li>
        <li class="nav-item">
          <a href="/add" class="nav-link">Add</a>
        </li>
      </div>
    </nav>

    <div class="container mt-3">
      <router-view />
    </div>
  </div>
</template>

<script lang="ts">
import { Component, Vue } from "vue-property-decorator";

@Component
export default class App extends Vue {}
</script>
```

## Initialize Axios for Vue Typescript CRUD HTTP Client

Let's install *axios* with command: `npm install axios`.

Then, under **src** folder, we create *http-common.ts* file like this:

```
import axios from "axios";

export default axios.create({
  baseURL: "http://localhost:8080/api",
  headers: {
    "Content-type": "application/json"
  }
});
```

We use cookies to improve your experience with the site. To find out more, you can read the full

Remember to change the `baseURL`, it depends on REST APIs url that your Server  
[Privacy & Policy \(https://bezkoder.com/privacy-policy/\)](https://bezkoder.com/privacy-policy/)

configures.

## Create Data Service

Our service will use axios from HTTP client above to send HTTP requests.

**services**/TutorialDataService.ts

```
import http from "../http-common";

class TutorialDataService {
  getAll() {
    return http.get("/tutorials");
  }

  get(id: string) {
    return http.get(`/tutorials/${id}`);
  }

  create(data: any) {
    return http.post("/tutorials", data);
  }

  update(id: string, data: any) {
    return http.put(`/tutorials/${id}`, data);
  }

  delete(id: string) {
    return http.delete(`/tutorials/${id}`);
  }

  deleteAll() {
    return http.delete(`/tutorials`);
  }

  findByTitle(title: string) {
    return http.get(`/tutorials?title=${title}`);
  }
}

export default new TutorialDataService();
```

## Create Vue Components

As I've said before, there are 3 components corresponding to 3 routes defined in Vue Router.

We use cookies to improve your experience with the site. To find out more, you can read the full

### Add Item Component

[Privacy & Policy \(https://bezcoder.com/privacy-policy/\)](https://bezcoder.com/privacy-policy/)

Accept

This component has a Form to submit new Tutorial with 2 fields: title &

`description` . It calls `TutorialDataService.create()` method.

**components**/*AddTutorial.vue*

We use cookies to improve your experience with the site. To find out more, you can read the full

**[Privacy & Policy \(https://bezcoder.com/privacy-policy/\)](https://bezcoder.com/privacy-policy/)**

Accept

```

<template>
  <div class="submit-form">
    <div v-if="!submitted">
      <div class="form-group">
        <label for="title">Title</label>
        <input
          type="text"
          class="form-control"
          id="title"
          required
          v-model="tutorial.title"
          name="title"
        />
      </div>

      <div class="form-group">
        <label for="description">Description</label>
        <input
          class="form-control"
          id="description"
          required
          v-model="tutorial.description"
          name="description"
        />
      </div>

      <button @click="saveTutorial" class="btn btn-success">Submit</button>
    </div>

    <div v-else>
      <h4>You submitted successfully!</h4>
      <button class="btn btn-success" @click="newTutorial">Add</button>
    </div>
  </div>
</template>

<script lang="ts">
import { Component, Vue } from "vue-property-decorator";
import TutorialDataService from "../services/TutorialDataService";

@Component
export default class AddTutorial extends Vue {
  private tutorial: any = {
    id: null,
    title: "",
    description: "",
  };

  public saveTutorial() {
    TutorialDataService.create(this.tutorial);
  }

  public newTutorial() {
    this.tutorial = {
      id: null,
      title: "",
      description: "",
    };
  }
}

```

We use cookies to improve your experience with the site. To find out more, you can read the full

[Privacy & Policy \(https://bezcoder.com/privacy-policy/\)](https://bezcoder.com/privacy-policy/)

Accept

```
};

private submitted: boolean = false;

saveTutorial() {
  var data = {
    title: this.tutorial.title,
    description: this.tutorial.description,
  };

  TutorialDataService.create(data)
    .then((response) => {
      this.tutorial.id = response.data.id;
      console.log(response.data);
      this.submitted = true;
    })
    .catch((e) => {
      console.log(e);
    });
}

newTutorial() {
  this.submitted = false;
  this.tutorial = {};
}
}
</script>

<style scoped>
.submit-form {
  max-width: 300px;
  margin: auto;
}
</style>
```

## List of items Component

This component calls 3 `TutorialDataService` methods:

- `getAll()`
- `deleteAll()`
- `findByTitle()`

**components**/*TutorialsList.vue*

We use cookies to improve your experience with the site. To find out more, you can read the full

[Privacy & Policy \(https://bezcoder.com/privacy-policy/\)](https://bezcoder.com/privacy-policy/)

Accept

```

<template>
  <div class="list row">
    <div class="col-md-8">
      <div class="input-group mb-3">
        <input
          type="text"
          class="form-control"
          placeholder="Search by title"
          v-model="title"
        />
        <div class="input-group-append">
          <button
            class="btn btn-outline-secondary"
            type="button"
            @click="searchTitle"
          >
            Search
          </button>
        </div>
      </div>
    </div>
    <div class="col-md-6">
      <h4>Tutorials List</h4>
      <ul class="list-group">
        <li
          class="list-group-item"
          :class="{ active: index == currentIndex }"
          v-for="(tutorial, index) in tutorials"
          :key="index"
          @click="setActiveTutorial(tutorial, index)"
        >
          {{ tutorial.title }}
        </li>
      </ul>

      <button class="m-3 btn btn-sm btn-danger" @click="removeAllTutorials">
        Remove All
      </button>
    </div>
    <div class="col-md-6">
      <div v-if="currentTutorial">
        <h4>Tutorial</h4>
        <div>
          <label><strong>Title:</strong></label> {{ currentTutorial.title }}
        </div>
      </div>
    </div>
  </div>

```

We use cookies to improve your experience with the site. To find out more, you can read the full

[Privacy & Policy \(https://bezkoder.com/privacy-policy/\)](https://bezkoder.com/privacy-policy/)

Accept



```

        {{ currentTutorial.description }}
      </div>
      <div>
        <label><strong>Status:</strong></label>
        {{ currentTutorial.published ? "Published" : "Pending" }}
      </div>

      <a
        class="badge badge-warning"
        :href="'/tutorials/' + currentTutorial.id"
      >
        Edit
      </a>
    </div>
    <div v-else>
      <br />
      <p>Please click on a Tutorial...</p>
    </div>
  </div>
</div>
</template>

```

```

<script lang="ts">
import { Component, Vue } from "vue-property-decorator";
import TutorialDataService from "../services/TutorialDataService";

```

```

@Component
export default class TutorialsList extends Vue {
  private tutorials: any[] = [];
  private currentTutorial: any = null;
  private currentIndex: number = -1;
  private title: string = "";

```

```

  retrieveTutorials() {
    TutorialDataService.getAll()
      .then((response) => {
        this.tutorials = response.data;
        console.log(response.data);
      })
      .catch((e) => {
        console.log(e);
      });
  }

```

```

  refreshList() {

```

```

    this.retrieveTutorials();

```

```

    this.currentTutorial = null;

```

```

    this.currentIndex = -1;

```

We use cookies to improve your experience with the site. To find out more, you can read the full

[Privacy & Policy \(https://bezkoder.com/privacy-policy/\)](https://bezkoder.com/privacy-policy/)

Accept

```

    }

    setActiveTutorial(tutorial: any, index: number) {
      this.currentTutorial = tutorial;
      this.currentIndex = index;
    }

    removeAllTutorials() {
      TutorialDataService.deleteAll()
        .then((response) => {
          console.log(response.data);
          this.refreshList();
        })
        .catch((e) => {
          console.log(e);
        });
    }

    searchTitle() {
      TutorialDataService.findByTitle(this.title)
        .then((response) => {
          this.tutorials = response.data;
          console.log(response.data);
        })
        .catch((e) => {
          console.log(e);
        });
    }

    mounted() {
      this.retrieveTutorials();
    }
  }
</script>

<style scoped>
.list {
  text-align: left;
  max-width: 750px;
  margin: auto;
}
</style>

```

If you click on **Edit** button of any Tutorial, the app will direct you to *Tutorial* page with

url: `/tutorials/:tutorialId`

We use cookies to improve your experience with the site. To find out more, you can read the full

**Privacy Policy** (<https://bezcoder.com/privacy-policy/>)

Accept

**Item details Component**

This component will use 3 `TutorialDataService` methods for getting data & update, delete the Tutorial:

- `get()`
- `update()`
- `delete()`

**components**/*Tutorial.vue*

We use cookies to improve your experience with the site. To find out more, you can read the full

**[Privacy & Policy \(https://bezcoder.com/privacy-policy/\)](https://bezcoder.com/privacy-policy/)**

Accept

```

<template>
  <div v-if="currentTutorial" class="edit-form">
    <h4>Tutorial</h4>
    <form>
      <div class="form-group">
        <label for="title">Title</label>
        <input
          type="text"
          class="form-control"
          id="title"
          v-model="currentTutorial.title"
        />
      </div>
      <div class="form-group">
        <label for="description">Description</label>
        <input
          type="text"
          class="form-control"
          id="description"
          v-model="currentTutorial.description"
        />
      </div>

      <div class="form-group">
        <label><strong>Status:</strong></label>
        {{ currentTutorial.published ? "Published" : "Pending" }}
      </div>
    </form>

    <button
      class="badge badge-primary mr-2"
      v-if="currentTutorial.published"
      @click="updatePublished(false)"
    >
      UnPublish
    </button>
    <button
      v-else
      class="badge badge-primary mr-2"
      @click="updatePublished(true)"
    >
      Publish
    </button>

    <button class="badge badge-danger mr-2" @click="deleteTutorial">
      Delete
    </button>
  </div>

```

We use cookies to improve your experience with the site. To find out more, you can read the full

[Privacy & Policy \(https://bezcoder.com/privacy-policy/\)](https://bezcoder.com/privacy-policy/)

Accept

```

    <button type="submit" class="badge badge-success" @click="updateTutorial">
      Update
    </button>
    <p>{{ message }}</p>
  </div>

  <div v-else>
    <br />
    <p>Please click on a Tutorial...</p>
  </div>
</template>

<script lang="ts">
import { Component, Vue } from "vue-property-decorator";
import TutorialDataService from "../services/TutorialDataService";

@Component
export default class Tutorial extends Vue {
  private currentTutorial: any = null;
  private message: string = "";

  getTutorial(id: string) {
    TutorialDataService.get(id)
      .then((response) => {
        this.currentTutorial = response.data;
        console.log(response.data);
      })
      .catch((e) => {
        console.log(e);
      });
  }

  updatePublished(status: boolean) {
    var data = {
      id: this.currentTutorial.id,
      title: this.currentTutorial.title,
      description: this.currentTutorial.description,
      published: status,
    };

    TutorialDataService.update(this.currentTutorial.id, data)
      .then((response) => {
        this.currentTutorial.published = status;
        console.log(response.data);
      })
      .catch((e) => {
        console.log(e);
      });
  }
}

```

We use cookies to improve your experience with the site. To find out more, you can read the full

[Privacy & Policy \(https://bezcoder.com/privacy-policy/\)](https://bezcoder.com/privacy-policy/)

Accept

```

    });
  }

  updateTutorial() {
    TutorialDataService.update(this.currentTutorial.id, this.currentTutorial)
      .then((response) => {
        console.log(response.data);
        this.message = "The tutorial was updated successfully!";
      })
      .catch((e) => {
        console.log(e);
      });
  }

  deleteTutorial() {
    TutorialDataService.delete(this.currentTutorial.id)
      .then((response) => {
        console.log(response.data);
        this.$router.push({ name: "tutorials" });
      })
      .catch((e) => {
        console.log(e);
      });
  }

  mounted() {
    this.message = "";
    this.getTutorial(this.$route.params.id);
  }
}
</script>

<style scoped>
.edit-form {
  max-width: 300px;
  margin: auto;
}
</style>

```

## Configure Port for Vue Typescript CRUD App

Because most of HTTP Server use CORS configuration that accepts resource sharing retriected to some sites or ports, so we also need to configure port for our App.

In project root folder, create *vue.config.js* file with following content:

We use cookies to improve your experience with the site. To find out more, you can read the full

[Privacy & Policy \(https://bezkoder.com/privacy-policy/\)](https://bezkoder.com/privacy-policy/)

Accept

```
module.exports = {  
  devServer: {  
    port: 8081  
  }  
}
```

We've set our app running at port `8081`.

## Run Vue Typescript App

You can run our App with command: `npm run serve`.

If the process is successful, open Browser with Url: `http://localhost:8081/` and check it.

This Vue Client will work well with following back-end Rest APIs:

- Express, Sequelize & MySQL (<https://bezcoder.com/node-js-express-sequelize-mysql/>)
- Express, Sequelize & PostgreSQL (<https://bezcoder.com/node-express-sequelize-postgresql/>)
- Express & MongoDB (<https://bezcoder.com/node-express-mongodb-crud-rest-api/>)
- Spring Boot & MySQL (<https://bezcoder.com/spring-boot-jpa-crud-rest-api/>)
- Spring Boot & PostgreSQL (<https://bezcoder.com/spring-boot-postgresql-example/>)
- Spring Boot & MongoDB (<https://bezcoder.com/angular-spring-boot-mongodb/>)
- Spring Boot & H2 (<https://bezcoder.com/spring-boot-jpa-h2-example/>)
- Spring Boot & Cassandra (<https://bezcoder.com/spring-boot-cassandra-crud/>)
- Django & MySQL (<https://bezcoder.com/django-crud-mysql-rest-framework/>)
- Django & PostgreSQL (<https://bezcoder.com/django-postgresql-crud-rest-framework/>)
- Django & MongoDB (<https://bezcoder.com/django-mongodb-crud-rest-framework/>)

## Conclusion

Today we've built a Vue Typescript example – CRUD Application successfully with Vue Router, Axios, Bootstrap. Now we can consume REST APIs, display and modify data in a clean way. I hope you apply it in your project at ease.

You can also know how to add Token based Authentication into Vue/Vuex Typescript App with this post:

Vue/Vuex Typescript: JWT Authentication example (<https://bezcoder.com/vuex-typescript-jwt-auth/>)

Happy learning, see you again!

We use cookies to improve your experience with the site. To find out more, you can read the full

## Further Reading

[Privacy Policy](https://bezcoder.com/privacy-policy/) (<https://bezcoder.com/privacy-policy/>)

Accept

- Vue Class Component (<https://class-component.vuejs.org/>)
- Vue Router guide (<https://router.vuejs.org/>)
- <https://www.npmjs.com/package/axios> (<https://www.npmjs.com/package/axios>)

## Source Code

You can find the complete source code for this tutorial on Github (<https://github.com/bezkoder/vue-typescript-crud>).

(<https://bezcoder.com/tag/axios/>)

(<https://bezcoder.com/tag/crud/>)

(<https://bezcoder.com/tag/rest-api/>)

(<https://bezcoder.com/tag/vue/>)

(<https://bezcoder.com/tag/vue-class-component/>)

(<https://bezcoder.com/tag/vue-router/>)

(<https://bezcoder.com/tag/vue-typescript/>)

## 2 thoughts to “Vue Typescript example: Build a CRUD Application”



**ooouuiii**

September 6, 2020 at 10:27 pm (<https://bezcoder.com/vue-typescript-crud/#comment-4618>)

Sir, this saved my project. Thank You!

REPLY



**st**

September 7, 2020 at 8:59 pm (<https://bezcoder.com/vue-typescript-crud/#comment-4626>)

One of the best tutorials I have ever seen! Thanks!

REPLY

### Leave a Reply

Your email address will not be published. Required fields are marked \*

Comment

We use cookies to improve your experience with the site. To find out more, you can read the full

[Privacy & Policy \(https://bezcoder.com/privacy-policy/\)](https://bezcoder.com/privacy-policy/)

Accept



Name \*

Email \*

Website

☐

Save my name, email, and website in this browser for the next time I comment.

[◀ Spring Boot + React: JWT Authentication with Spring Security \(https://bezcoder.com/spring-boot-react-jwt-auth/\)](https://bezcoder.com/spring-boot-react-jwt-auth/)[Angular 8 Firebase CRUD Realtime DB | AngularFireDatabase ▶ \(https://bezcoder.com/angular-8-firebase-crud/\)](https://bezcoder.com/angular-8-firebase-crud/)

We use cookies to improve your experience with the site. To find out more, you can read the full **FOLLOW US**

[Privacy & Policy \(https://bezcoder.com/privacy-policy/\)](https://bezcoder.com/privacy-policy/)



(htt

ps://

ww

w.yo

utub

e.co

m

/cha



nnel

(htt /UC



ps:// p0m (htt

face x9R ps://

boo H0Jx gith

k.co aFs ub.c

m mM om

/bez vKX /bez

kod A86 kod

er) Q) er)

## TOOLS

Json Formatter (<https://bezkoder.com/json-formatter/>)



We use cookies to improve your experience with the site. To find out more, you can read the full [Privacy & Policy \(https://bezkoder.com/privacy-policy/\)](https://www.dmca.com/Protection/Status.aspx?ID=3f543dd5-c6d8-4208-9a6b-0e92057fd597&refurl=https://bezkoder.com/vue-typescript-crud/)

Accept

---

[Home \(https://bezkoder.com/\)](https://bezkoder.com/)

[Privacy Policy \(https://bezkoder.com/privacy-policy/\)](https://bezkoder.com/privacy-policy/)

[Contact Us \(https://bezkoder.com/contact-us/\)](https://bezkoder.com/contact-us/)

[About Us \(https://bezkoder.com/about/\)](https://bezkoder.com/about/)

BezKoder 2019

We use cookies to improve your experience with the site. To find out more, you can read the full

[Privacy & Policy \(https://bezkoder.com/privacy-policy/\)](https://bezkoder.com/privacy-policy/)

Accept