**Vue.js + Node.js + Express + PostgreSQL example: Build a full-stack CRUD Application**

[Last modified: April 24, 2020](https://bezkoder.com/vue-node-express-postgresql/) [bezkoder](https://bezkoder.com/author/bezkoder/) [Full Stack](https://bezkoder.com/category/full-stack/), [Node.js](https://bezkoder.com/category/node-js/), [Vue.js](https://bezkoder.com/category/vue/)

In this tutorial, I will show you how to build full-stack (Vue.js + Node.js + Express + PostgreSQL) example with a CRUD Application. The back-end server uses Node.js + Express for REST APIs, front-end side is a Vue client with Vue Router and axios.

More Practice: [Node.js Express + Vue.js: JWT Authentication & Authorization example](https://bezkoder.com/node-express-vue-jwt-auth/)

Contents [[hide](https://bezkoder.com/vue-node-express-postgresql/)]

* [Vue.js + Node.js + Express + PostgreSQL example Overview](https://bezkoder.com/vue-node-express-postgresql/#Vuejs_Nodejs_Express_PostgreSQL_example_Overview)
* [Full-stack CRUD App Architecture](https://bezkoder.com/vue-node-express-postgresql/#Full-stack_CRUD_App_Architecture)
* [Node.js Express Back-end](https://bezkoder.com/vue-node-express-postgresql/#Nodejs_Express_Back-end)
  + [Overview](https://bezkoder.com/vue-node-express-postgresql/#Overview)
  + [Project Structure](https://bezkoder.com/vue-node-express-postgresql/#Project_Structure)
  + [Implementation](https://bezkoder.com/vue-node-express-postgresql/#Implementation)
* [Vue.js Front-end](https://bezkoder.com/vue-node-express-postgresql/#Vuejs_Front-end)
  + [Overview](https://bezkoder.com/vue-node-express-postgresql/#Overview-2)
  + [Technology](https://bezkoder.com/vue-node-express-postgresql/#Technology)
  + [Project Structure](https://bezkoder.com/vue-node-express-postgresql/#Project_Structure-2)
  + [Implementation](https://bezkoder.com/vue-node-express-postgresql/#Implementation-2)
* [Conclusion](https://bezkoder.com/vue-node-express-postgresql/#Conclusion)

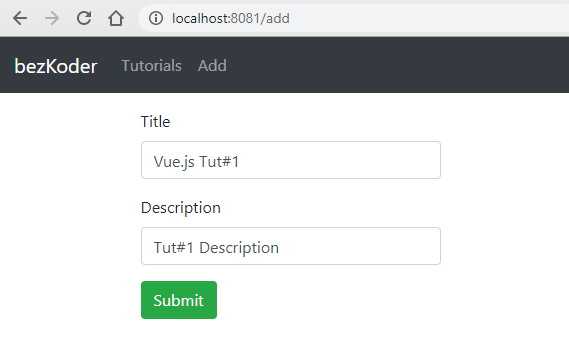
**Vue.js + Node.js + Express + PostgreSQL example Overview**

We will build a full-stack Tutorial Application in that:

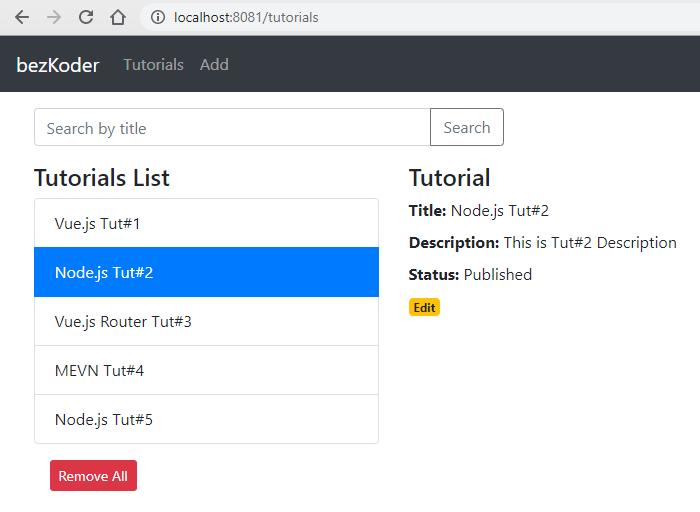
* Tutorial has id, title, description, published status.
* User can create, retrieve, update, delete Tutorials.
* There is a search box for finding Tutorials by title.

Here are screenshots of the example.

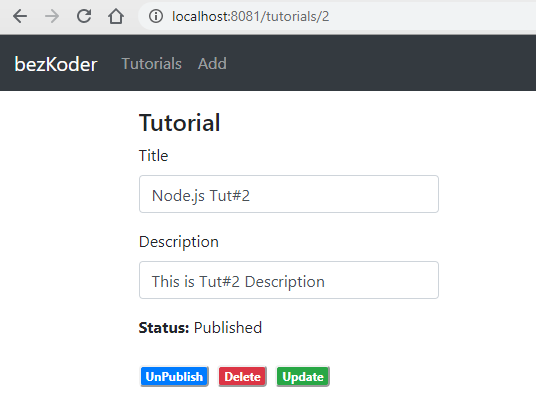
– Add an object:



– Show all objects:



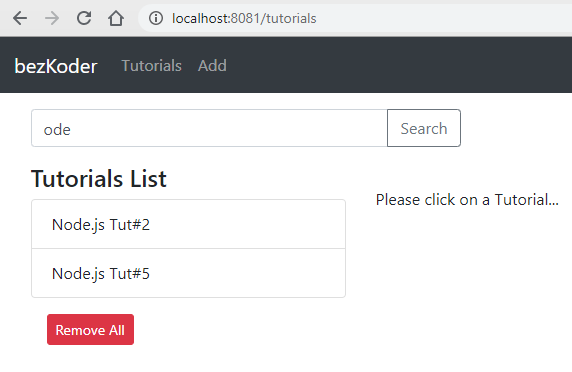
– Click on **Edit** button to update an object:



On this Page, you can:

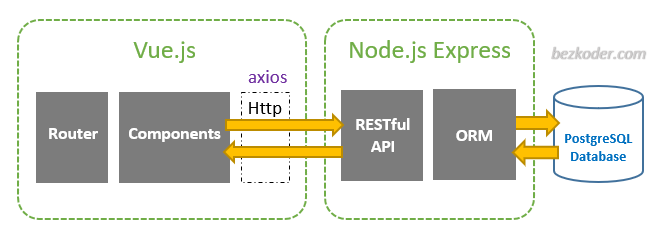
* change status to **Published**/**Pending** using **Publish**/**UnPublished** button
* remove the object from PostgreSQL Database using **Delete** button
* update this object’s details on Database with **Update** button

– Search objects by field ‘title’:



**Full-stack CRUD App Architecture**

We’re gonna build the application with following architecture:



– Node.js Express exports REST APIs & interacts with PostgreSQL Database using Sequelize ORM.  
– Vue Client sends HTTP Requests and retrieves HTTP Responses using *axios*, consume data on the components. Vue Router is used for navigating to pages.

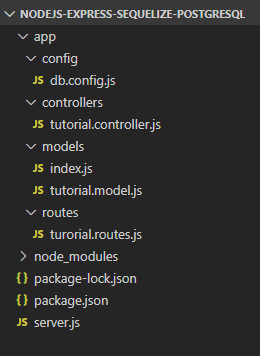
**Node.js Express Back-end**

**Overview**

These are APIs that Node.js Express App will export:

| **Methods** | **Urls** | **Actions** |
| --- | --- | --- |
| GET | api/tutorials | get all Tutorials |
| GET | api/tutorials/:id | get Tutorial by id |
| POST | api/tutorials | add new Tutorial |
| PUT | api/tutorials/:id | update Tutorial by id |
| DELETE | api/tutorials/:id | remove Tutorial by id |
| DELETE | api/tutorials | remove all Tutorials |
| GET | api/tutorials?title=[kw] | find all Tutorials which title contains 'kw' |

**Project Structure**



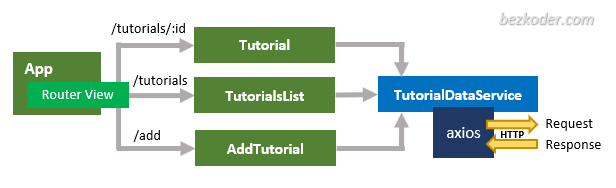
– *db.config.js* exports configuring parameters for PostgreSQL connection & Sequelize.  
– **Express** web server in *server.js* where we configure CORS, initialize & run Express REST APIs.  
– Next, we add configuration for PostgreSQL database in **models**/*index.js*, create **Sequelize** data model in **models**/*tutorial.model.js*.  
– Tutorial controller in **controllers**.  
– Routes for handling all CRUD operations (including custom finder) in *tutorial.routes.js*.

**Implementation**

You can find step by step to implement this Node.js Express App in the post:  
[Node.js CRUD Rest APIs with Express, Sequelize & PostgreSQL example](https://bezkoder.com/node-express-sequelize-postgresql/)

**Vue.js Front-end**

**Overview**



– The App component is a container with router-view. It has navbar that links to routes paths.

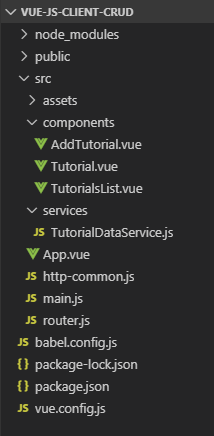
– TutorialsList component gets and displays Tutorials.  
– Tutorial component has form for editing Tutorial’s details based on :id.  
– AddTutorial component has form for submission new Tutorial.

– These Components call TutorialDataService methods which use axios to make HTTP requests and receive responses.

**Technology**

* vue: 2.6.10
* vue-router: 3.1.3
* axios: 0.19.0

**Project Structure**



– **package.json** contains 3 main modules: vue, vue-router, axios.  
– There are 3 components: TutorialsList, Tutorial, AddTutorial.  
– **router.js** defines routes for each component.  
– **http-common.js** 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.

**Implementation**

You can find step by step to implement this Vue App in the post:  
[Vue.js CRUD App with Vue Router & Axios](https://bezkoder.com/vue-js-crud-app/)

**Conclusion**

Now we have an overview of Vue.js + Node.js Express + PostgreSQL example when building a full-stack CRUD App.

We also take a look at client-server architecture for REST API using Express & Sequelize ORM, as well as Vue.js project structure for building a front-end app to make HTTP requests and consume responses.

Next tutorials show you more details about how to implement the system:

* [Back-end](https://bezkoder.com/node-express-sequelize-postgresql/)
* [Front-end](https://bezkoder.com/vue-js-crud-app/)

If you want a Typescript version for the Vue App, it is here:  
[Vue Typescript CRUD Application to consume Web API example](https://bezkoder.com/vue-typescript-crud/)

Happy learning, see you again!