

React.js and Spring Data REST

This tutorial shows a collection of apps that use Spring Data REST and its powerful backend functionality combined with React's sophisticated features to build an easy-to-grok UI.

- Spring Data REST (<https://www.youtube.com/watch?v=TgCr7v9tdKM>) provides a fast way to build hypermedia-powered repositories.
- React (<https://facebook.github.io/react/index.html>) is Facebook's solution to efficient, fast, and easy-to-use views in the land of JavaScript.

Part 1 - Basic Features

Welcome Spring community,

In this section, you will see how to get a bare-bones Spring Data REST application up and running quickly. Then you will build a simple UI on top of it using Facebook's React.js toolset.

Step 0 - Setting up your environment

Feel free to grab the code (<https://github.com/spring-guides/tut-react-and-spring-data-rest/tree/master/basic>) from this repository and follow along.

If you want to do it yourself, visit <https://start.spring.io> (<https://start.spring.io>) and pick these items:

- Rest Repositories
- Thymeleaf
- JPA
- H2

This demo uses Java 8, Maven Project, and the latest stable release of Spring Boot. It also uses React.js coded in ES6 (<http://es6-features.org/>). This will give you a clean, empty project. From there, you can add the various files shown explicitly in this section, and/or borrow from the repository listed above.

In the beginning...

In the beginning there was data. And it was good. But then people wanted to access the data through various means. Over the years, people cobbled together lots of MVC controllers, many using Spring's powerful REST support. But doing over and over cost a lot of time.

Spring Data REST addresses how simple this problem can be if some assumptions are made:

- The developer uses a Spring Data project that supports the repository model.