

Steps:

Step1: Go to <https://start.spring.io/>

The screenshot shows the Spring Initializr web application. At the top, it says "SPRING INITIALIZR bootstrap your application now". Below this, there's a form to generate a project. The form has three main sections: "Generate a", "with", and "and Spring Boot". The "Generate a" section has a dropdown menu set to "Maven Project". The "with" section has a dropdown menu set to "Java". The "and Spring Boot" section has a dropdown menu set to "2.0.4". Below these sections, there are two columns. The left column is titled "Project Metadata" and contains "Artifact coordinates" with a "Group" field set to "com.university.crud" and an "Artifact" field set to "spring-boot-crud-rest". The right column is titled "Dependencies" and contains "Add Spring Boot Starters and dependencies to your application" with a "Search for dependencies" field set to "Web, Security, JPA, Actuator, Devtools...". Below the search field, there's a "Selected Dependencies" section with five buttons: "Web", "H2", "Actuator", "DevTools", and "JPA", each with a close icon. At the bottom of the form, there's a green button labeled "Generate Project" with a keyboard shortcut "alt + ⌘". Below the button, there's a link that says "Don't know what to look for? Want more options? Switch to the full version."

Step2 : Make it as a git repo and push it to SCM

- Git init
- Git add remote
- Git push

Step 3: Build ,run and import as Maven Project

- mvnw.bat clean install
- mvnw.bat spring-boot:run
- Import to eclipse/STS

Step4: Understand the Effective dependencies

- transitive dependencies
- how starters are working
- How Spring boot is opinionated

Step5: Generate autoconfig report

- `debug=true` in `application.properties`
- `logging.file=application.log`

Step6: Added Data source

- Create bean of type Data source

Step7: Add a model class

Step8: Changed context-path

```
server.servlet.context-path=/demoapp
```