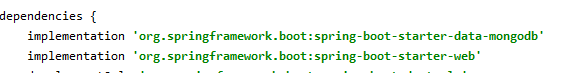
**Spring Boot MongoDB**

**Step 1**: Create a spring boot project from a starter project using spring initializer.

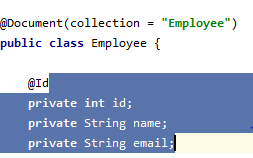
**Step 2**: Install and run a MongoDB instance locally. This tutorial uses a MongoDB database for persisting the data. Download the community edition from here and extract the zip to a folder of your choice. Add the bin folder in the operating system path. Then create a folder where you want MongoDB to keep its data file. Use the following command to run MongoDB from console.

*Install the Mongo DB using this official website :* <https://www.mongodb.com/try/download/community>

**Step 3**: Replace the build.gradle with the following file. This file has two additional dependencies. The library **spring-boot-starter-data-rest** is used for enabling REST endpoints for data repositories and **spring-boot-starter-data-mongodb** is used for MongoDB connectivity. Spring boot auto configuration wires them all together and assumes that you are connecting to a local Mongo instance at the default port. No further configuration is required!



**Step 4**: Create a Java object representation for MongoDB table. In this tutorial we will create a simple crud for customer data. Following is a Java bean representing a single customer record in the mongo customer table. Please note that you don't need to create this collection/table in mongo, it will be automatically created during the first insert.



**Step 5**: Create a repository class for customer object. A spring repository is an interface which allows us to access our database tables. By extending our repository from MongoRepository.



**Step 7**: Access the crud features from the following links. To POST data, you can use browser extensions such as [postman](https://www.getpostman.com/) or use the [curl command line tool](https://en.wikipedia.org/wiki/CURL).