

# **Spring Data JPA : The Data Tier**

---

## **Spring Boot Quick Start**



## Unit 4

### Spring Data JPA: The Data Tier

### Spring Boot Quick Start



# Spring Data JPA

# Java Persistence API

# Object - Relational Mapping

# Relational Database

# Object - Relational Mapping

Class <-> Table

# Java Persistence API

# Spring Data JPA

## Package Explorer

course-api [boot]  
 > src/main/java  
 > src/main/resources  
 > src/test/java  
 > src/test/resources  
 > JRE System Library [JavaSE-1.8]  
 > Maven Dependencies  
 > src  
 > target  
 pom.xml  
 Servers

## course-api/pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  modelVersion="4.0.0"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <groupId>io.javabrains.springbootquickstart</groupId>
  <artifactId>course-api</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>Java Brains Course API</name>

  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>1.4.2.RELEASE</version>
  </parent>

  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-web</artifactId>
    </dependency>
  </dependencies>

  <properties>
    <java.version>1.8</java.version>
  </properties>

</project>
```

## Servers Boot Dashboard

Type tags to match:

&gt; local

Overview Dependencies Dependency Hierarchy Effective POM pom.xml

**Package Explorer**

- > course-api [boot]
- > Servers

**New**

- Spring Starter Project** (highlighted)
- Import Spring Getting Started Content
- Spring Project
- Java Project
- Static Web Project
- Dynamic Web Project
- Maven Project
- Project...
- Aspect
- Package
- Class
- Interface
- Enum
- Annotation
- JUnit Test Case
- Source Folder
- Java Working Set
- Spring Bean Configuration File
- Spring Web Flow Definition File
- Folder
- File
- Untitled Text File
- Example...
- Other...

**course-api/pom.xml**

```

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  modelVersion="4.0.0"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <groupId>io.javabrains.springbootquickstart</groupId>
  <artifactId>course-api</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>Java Brains Course API</name>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>1.4.2.RELEASE</version>
  </parent>

  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-web</artifactId>
    </dependency>
  </dependencies>

  <properties>
    <java.version>1.8</java.version>
  </properties>

  <object>
  
```

**Servers**

Boot Dashboard

Type tags to match:

> local

Overview Dependencies Dependency Hierarchy Effective POM pom.xml

### New Spring Starter Project

Name

Use default location

Location

Type:  Packaging:

Java Version:  Language:

Group

Artifact

Version

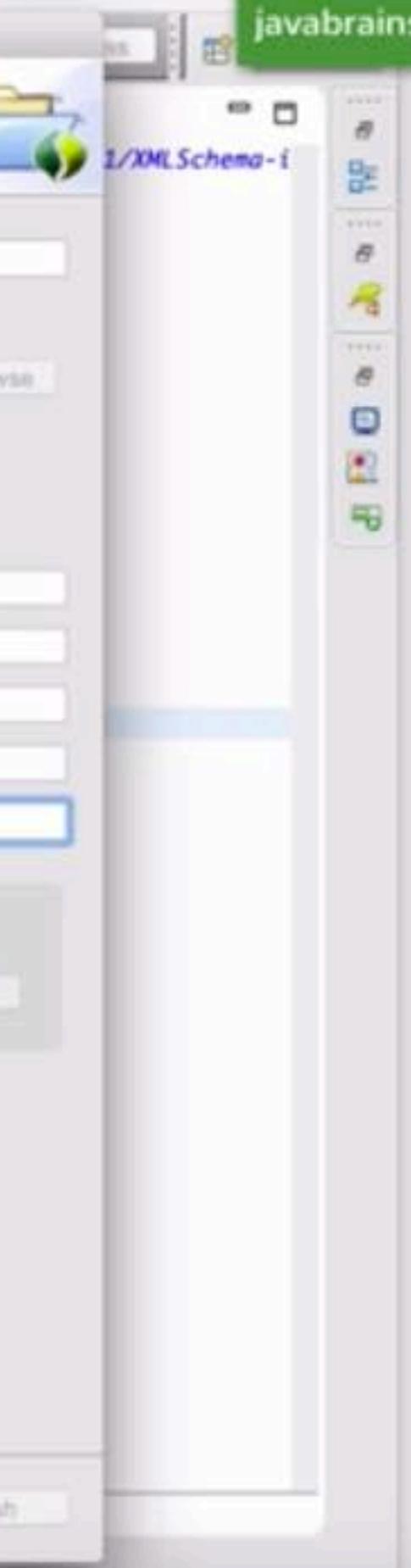
Description

Package

Working sets

Add project to working sets

Working sets:



New Spring Starter Project

Dependencies:

- ▶ Frequently Used
- ▶ Cloud AWS
- ▶ Cloud Circuit Breaker
- ▶ Cloud Cluster
- ▶ Cloud Config
- ▶ Cloud Contract
- ▶ Cloud Core
- ▶ Cloud Data Flow
- ▶ Cloud Discovery
- ▶ Cloud Messaging
- ▶ Cloud Routing
- ▶ Cloud Tracing
- ▶ Core
- ▶ Experimental
- ▶ I/O
- ▶ NoSQL
- ▶ Ops
- ▶ Pivotal Cloud Foundry
- ▶ SQL
- ▶ Social
- ▶ Template Engines
- ▶ Web

Web       Websocket       Web Services       Jersey (JAX-RS)       Ratpack  
 Vaadin       Rest Repositories       HATEOAS       Rest Repositories HAL Browser       Mobile  
 REST Docs

< Back      Next >      Cancel      **Finish**



New Spring Starter Project

Dependencies:

- ▶ Frequently Used
- ▶ Cloud AWS
- ▶ Cloud Circuit Breaker
- ▶ Cloud Cluster
- ▶ Cloud Config
- ▶ Cloud Contract
- ▶ Cloud Core
- ▶ Cloud Data Flow
- ▶ Cloud Discovery
- ▶ Cloud Messaging
- ▶ Cloud Routing
- ▶ Cloud Tracing
- ▶ Core
- ▶ Experimental
- ▶ I/O
- ▶ NoSQL
- ▶ Ops
- ▶ Pivotal Cloud Foundry
- SQL
  - JPA
  - JOOQ
  - MyBatis
  - JDBC
  - H2
  - HSQLDB
  - Apache Derby
  - MySQL
  - PostgreSQL
  - Flyway
  - Liquibase
- ▶ Social
- ▶ Template Engines
- ▶ Web

Servers: local

Type tags to match:

?

< Back Next > Cancel Finish



New Spring Starter Project

Dependencies:

- ▶ Frequently Used
- ▶ Cloud AWS
- ▶ Cloud Circuit Breaker
- ▶ Cloud Cluster
- ▶ Cloud Config
- ▶ Cloud Contract
- ▶ Cloud Core
- ▶ Cloud Data Flow
- ▶ Cloud Discovery
- ▶ Cloud Messaging
- ▶ Cloud Routing
- ▶ Cloud Tracing
- ▶ Core
- ▶ Experimental
- ▶ I/O
- ▶ NoSQL
- ▶ Ops
- ▶ Pivotal Cloud Foundry
- SQL
  - JPA
  - JOOQ
  - HSQLDB
  - Flyway
  - Apache Derby
  - Liquibase
  - MyBatis
  - MySQL
  - JDBC
  - PostgreSQL
  - H2
  - SQL Server
- ▶ Social
- ▶ Template Engines
- ▶ Web

?

< Back Next > Cancel Finish



1/XMLSchema-i



New Spring Starter Project

Dependencies:

- ▶ Frequently Used
- ▶ Cloud AWS
- ▶ Cloud Circuit Breaker
- ▶ Cloud Cluster
- ▶ Cloud Config
- ▶ Cloud Contract
- ▶ Cloud Core
- ▶ Cloud Data Flow
- ▶ Cloud Discovery
- ▶ Cloud Messaging
- ▶ Cloud Routing
- ▶ Cloud Tracing
- ▶ Core
- ▶ Experimental
- ▶ I/O
- ▶ NoSQL
- ▶ Ops
- ▶ Pivotal Cloud Foundry
- SQL
  - JPA
  - JOOQ
  - MyBatis
  - JDBC
  - H2
  - HSQLDB
  - Apache Derby
  - MySQL
  - PostgreSQL
  - SQL Server
  - Flyway
  - Liquibase
- ▶ Social
- ▶ Template Engines
- ▶ Web

?

< Back    Next >    Cancel    Finish



1/XMLSchema-i



New Spring Starter Project

Site Info

Base Url

Full Url

Servers Boot Dash

Type tags to match

> local

?

< Back Next > Cancel Finish



New Spring Starter Project

Site Info

Base Url

Full Url

Servers Boot Dash

Type tags to match

> local

?

< Back Next > Cancel Finish



Package Explorer course-api [boot] course-api-data Servers

course-api/pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  modelVersion="4.0.0"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <groupId>io.javabrains.springbootquickstart</groupId>
  <artifactId>course-api</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>Java Brains Course API</name>

  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>1.4.2.RELEASE</version>
  
```

Import Getting Started Content

! Importing http://start.spring.io/starter.zip

project course-api-data

Always run in background

Import Getting Started Content

Importing http://start.spring.io/starter.zip: project course-api-data

Building workspace (Waiting)

OK

Cancel << Details Run in Background

Servers Boot Dashboard

Type tags to match

local

Overview Dependencies Dependency Hierarchy Effective POM pom.xml

Import Getting Started Content: (24%)

Package Explorer

- course-api [boot]
- course-api-data [boot]
  - src/main/java
  - src/main/resources
  - src/test/java
  - JRE System Library [JavaSE-1.8]
  - Maven Dependencies
  - src
    - target
    - mvnw
    - mvnw.cmd
  - pom.xml**
- Servers

course-api-data/pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"
  modelVersion="4.0.0">

  <groupId>io.javabrains</groupId>
  <artifactId>course-api-data</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>jar</packaging>

  <name>course-api-data</name>
  <description>Course API with data for Spring Boot QuickStart Java Brains course</description>

  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>1.4.2.RELEASE</version>
    <relativePath/> 
  </parent>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
    <java.version>1.8</java.version>
  </properties>
```

]

```
<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-jpa</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>

  <dependency>
    <groupId>org.apache.derby</groupId>
    <artifactId>derby</artifactId>
    <scope>runtime</scope>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
```

Overview Dependencies Dependency Hierarchy Effective POM pom.xml

Package Explorer

- course-api [boot]
- course-api-data [boot]
  - src/main/java
    - io.javabrain
- src/main/resources
- src/test/java
- JRE System Library [JavaSE-1.8]
- Maven Dependencies
- src
  - target
  - mvnw
  - mvnw.cmd
  - pom.xml
- Servers

course-api-data/pom.xml CourseApiDataApplication.java

```
package io.javabrain;  
  
import org.springframework.boot.SpringApplication;  
  
@SpringBootApplication  
public class CourseApiDataApplication {  
  
    public static void main(String[] args) {  
        SpringApplication.run(CourseApiDataApplication.class, args);  
    }  
}
```

Servers Boot Dashboard

Type tags to match

local

Writable Smart Insert 9 : 18

Package Explorer    course-api [boot]    course-api-data/pom.xml    CourseApiDataApplication.java

```

package io.javabrains;
import org.springframework.boot.SpringApplication;
@SpringBootApplication
public class CourseApiDataApplication {
    public static void main(String[] args) {
        SpringApplication.run(CourseApiDataApplication.class, args);
    }
}

```

New    Go Into    Open in New Window    Open Type Hierarchy    Show In    Copy    Copy Qualified Name    Paste    Delete    Remove from Context    Build Path    Source    Refactor    Import...    Export...    References    Declarations    Refresh    Assign Working Sets...    Run As    Debug As    Profile As    Validate    Restore from Local History...    Team    Compare With    GitHub

Servers    Boot Dash

Type tags to match

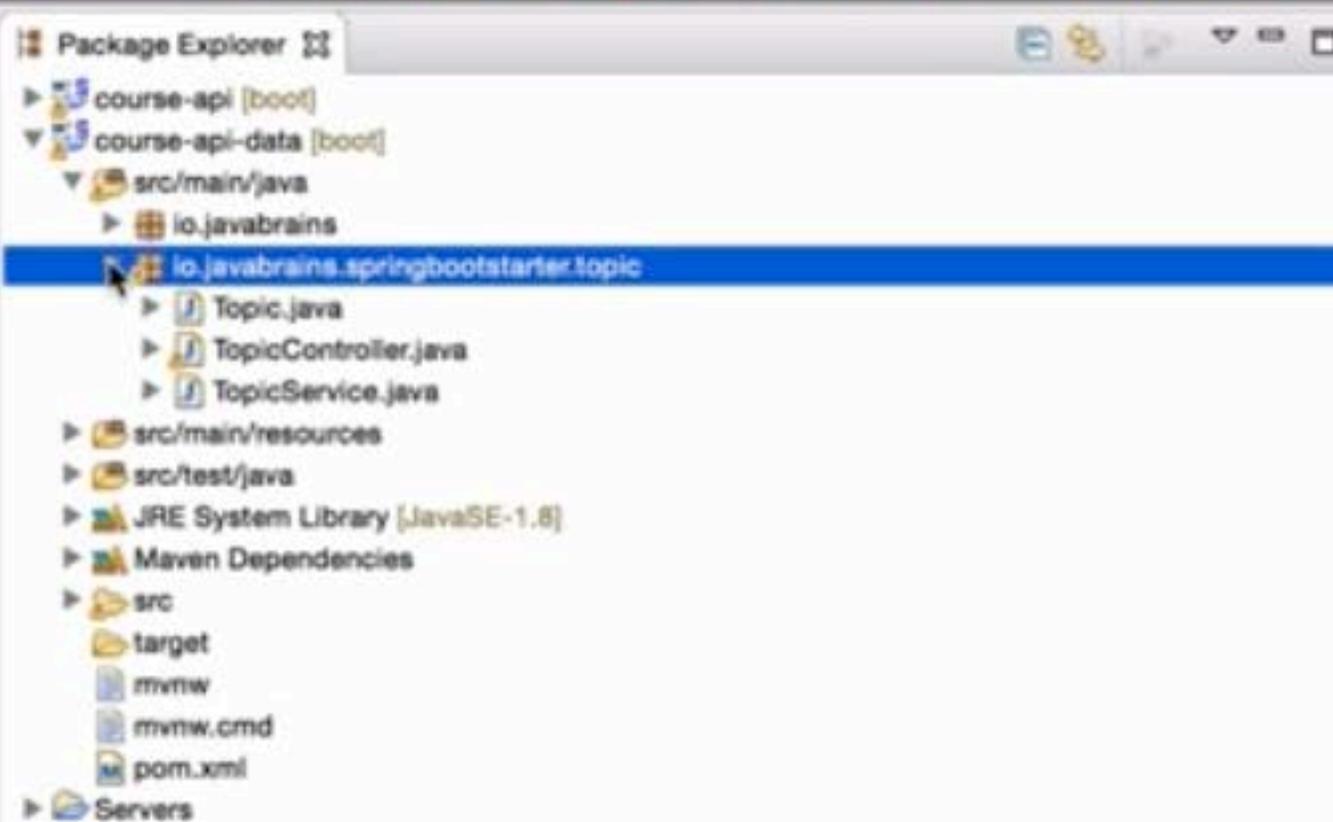
local

io.javabrains.springbootstar Properties

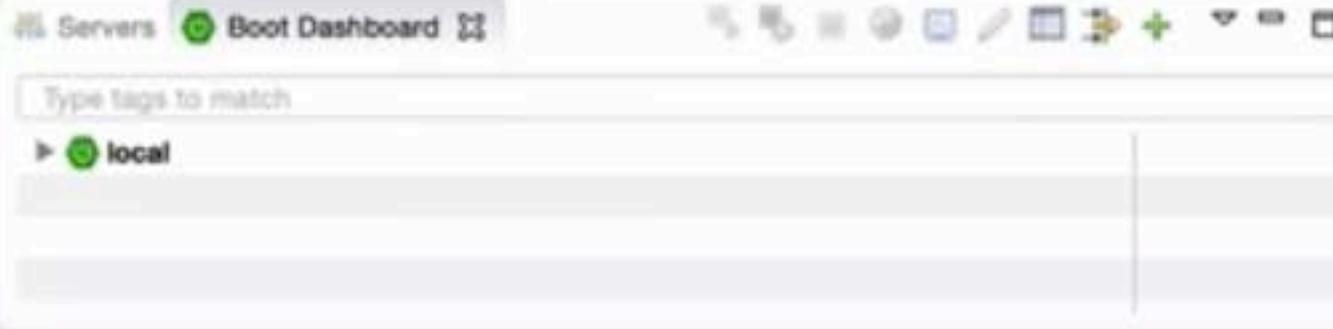
Package Explorer course-api [boot] course-api-data [boot] src/main/java New Go Into Open in New Window Open Type Hierarchy Show In ⌘W Copy ⌘C Copy Qualified Name ⌘V Paste ⌘V Delete Remove from Context ⌘O ⌘S Build Path Source ⌘S Refactor ⌘T Import... Export... Refresh F5 Assign Working Sets... Run As Debug As Profile As Validate Restore from Local History... Team Compare With GitHub Properties ⌘I

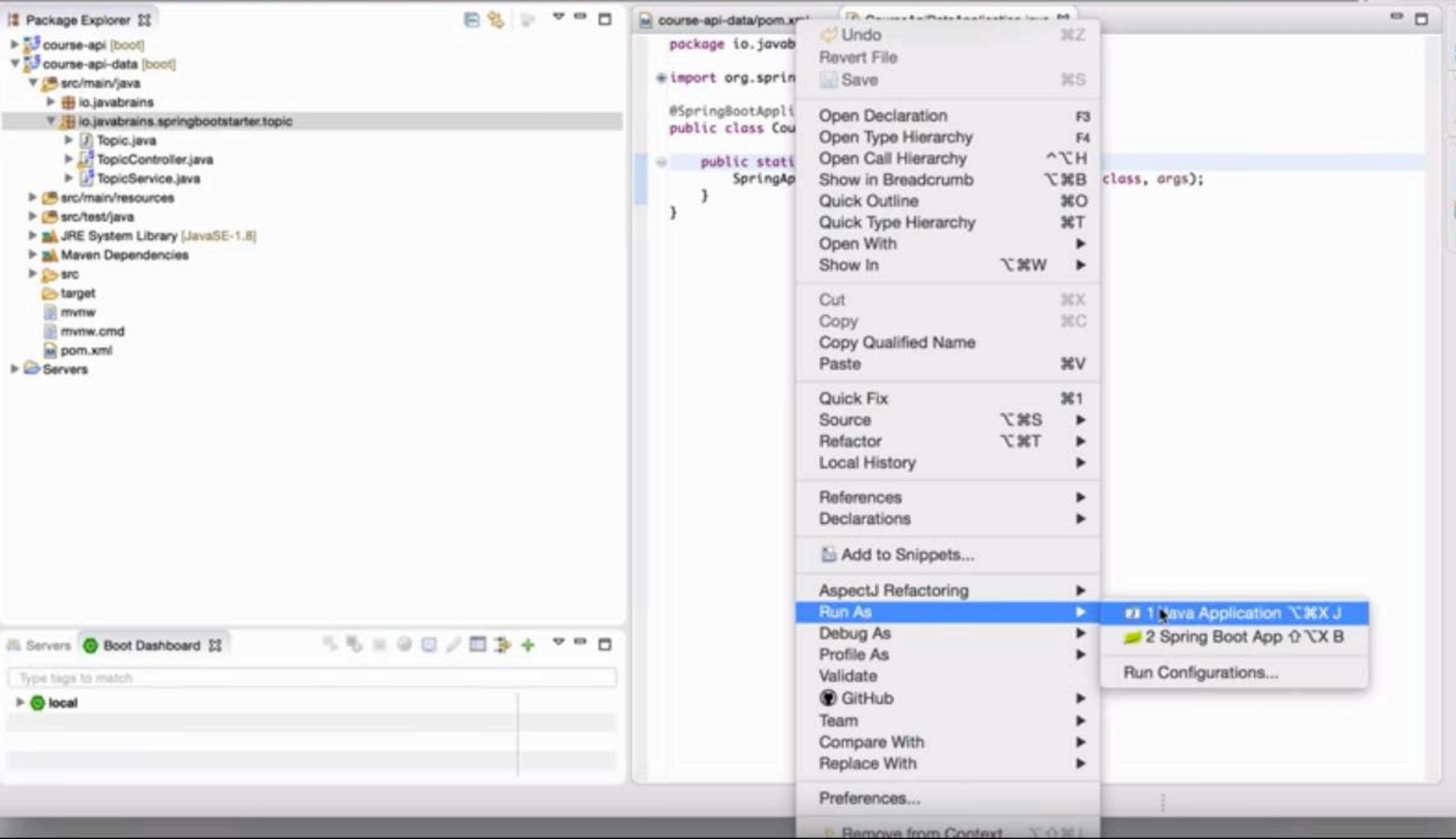
course-api-data/pom.xml CourseApiDataApplication.java

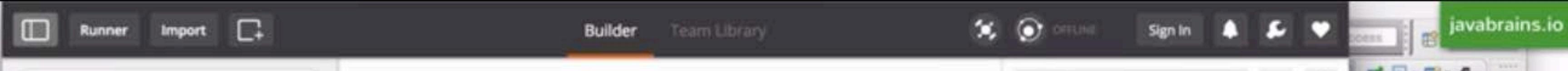
```
package io.javabrains;
import org.springframework.boot.SpringApplication;
@SpringBootApplication
public class CourseApiDataApplication {
    public static void main(String[] args) {
        SpringApplication.run(CourseApiDataApplication.class, args);
    }
}
```



```
course-api-data/pom.xml CourseApiDataApplication.java
package io.javabrain;
import org.springframework.boot.SpringApplication;
@SpringBootApplication
public class CourseApiDataApplication {
    public static void main(String[] args) {
        SpringApplication.run(CourseApiDataApplication.class, args);
    }
}
```







```
1 - []
2 - {
3 -     "id": "spring",
4 -     "name": "Spring Framework",
5 -     "description": "Spring Framework Description"
6 - },
7 - {
8 -     "id": "java",
9 -     "name": "Core Java",
10 -    "description": "Core Java Description"
11 - },
12 - {
13 -     "id": "javascript",
14 -     "name": "JavaScript",
15 -     "description": "JavaScript Description"
16 - }
17 - ]
```

Type tags to match:

> local

The screenshot shows the Spring Tool Suite interface with the following details:

- Package Explorer:** Shows the project structure with the following packages:
  - course-api [boot]
  - course-api-data [boot]
    - src/main/java
      - io.javabrain
      - io.javabrain.springbootstarter.topic
        - Topic.java
        - TopicController.java
        - TopicService.java
  - src/main/resources
  - src/test/java
  - JRE System Library [JavaSE-1.8]
  - Maven Dependencies
  - src
  - target
  - mvnw
  - mvnw.cmd
  - pom.xml
  - Servers
- Editor:** Displays the `CourseApiDataApplication.java` file content:

```
package io.javabrain;  
import org.springframework.boot.SpringApplication;  
@SpringBootApplication  
public class CourseApiDataApplication {  
    public static void main(String[] args) {  
        SpringApplication.run(CourseApiDataApplication.class, args);  
    }  
}
```
- Bottom Left:** Shows the `Boot Dashboard` with a single entry: `local`.
- Bottom Right:** Status bar showing `Writable`, `Smart Insert`, and `13 : 1`.

Package Explorer    course-api-data/pom.xml    CourseApiDataApplication.java

```

<version>0.0.1-SNAPSHOT</version>
<packaging>jar</packaging>

<name>course-api-data</name>
<description>Course API with data for Spring Boot QuickStart Java Brains course</description>

<parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>1.4.2.RELEASE</version>
    <relativePath/> <!-- lookup parent from repository -->
</parent>

<properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
    <java.version>1.8</java.version>
</properties>

<dependencies>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-data-jpa</artifactId>
    </dependency>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-web</artifactId>
    </dependency>

    <dependency>
        <groupId>org.apache.derby</groupId>
        <artifactId>derby</artifactId>
        <scope>runtime</scope>
    </dependency>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-test</artifactId>
        <scope>test</scope>
    </dependency>
</dependencies>

<build>
    <plugins>

```

Servers    Boot Dashboard

Type tags to match:

local

project/dependencies/dependency/artifactId/#text

Writable    Smart Insert    30 : 53

## Package Explorer

```
course-api [boot]
course-api-data [boot]
src/main/java
  io.javabrains
    io.javabrains.springbootstarter.topic
      Topic.java
      TopicController.java
      TopicService.java
src/main/resources
src/test/java
JRE System Library [JavaSE-1.8]
Maven Dependencies
src
  target
  mvnw
  mvnw.cmd
  pom.xml
Servers
```

## TopicService.java

## course-api-data/pom.xml

## Topic.java

```
package io.javabrains.springbootstarter.topic;
```

## @Entity

P Entity cannot be resolved to a type

6 quick fixes available:

- Import 'Entity' (javax.persistence)
- Import 'Entity' (org.hibernate.annotations)
- ⊕ Create annotation 'Entity'
- ⊖ Change to 'EntityScan' (org.springframework.boot.autoconfigure.domain)
- ⊖ Change to 'EntityScan' (org.springframework.boot.orm.jpa)
- ⊖ Fix project setup...

```
super();
this.id = id;
this.name = name;
this.description = description;
}
```

```
public String getId() {
    return id;
}
public void setId(String id) {
    this.id = id;
}
public String getName() {
    return name;
}
public void setName(String name) {
    this.name = name;
}
public String getDescription() {
    return description;
}
public void setDescription(String description) {
    this.description = description;
}
}
```

## Servers

## Boot Dashboard

Type tags to match

local

## Package Explorer

```
course-api [boot]
course-api-data [boot]
src/main/java
  io.javabrains
    io.javabrains.springbootstarter.topic
      Topic.java
      TopicController.java
      TopicService.java
src/main/resources
src/test/java
JRE System Library [JavaSE-1.8]
Maven Dependencies
src
  target
  mvnw
  mvnw.cmd
  pom.xml
Servers
```

## TopicService.java

## course-api-data/pom.xml

## Topic.java

```
package io.javabrains.springbootstarter.topic;

@Entity
public class Topic {
    Entity cannot be resolved to a type
    6 quick fixes available:
    - Import 'Entity' (javax.persistence)
    - Import 'Entity' (org.hibernate.annotations)
    - Create annotation 'Entity'
    - Change to 'EntityScan' (org.springframework.boot.autoconfigure.domain)
    - Change to 'EntityScan' (org.springframework.boot.orm.jpa)

    public Topic(String id, String name, String description) {
        super();
        this.id = id;
        this.name = name;
        this.description = description;
    }

    public String getId() {
        return id;
    }
    public void setId(String id) {
        this.id = id;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getDescription() {
        return description;
    }
    public void setDescription(String description) {
        this.description = description;
    }
}
```

## Servers

## Boot Dashboard

Type tags to match

&gt; local

## Package Explorer

- > course-api [boot]
- > course-api-data [boot]
  - > src/main/java
    - > io.javabrains
    - > io.javabrains.springbootstarter.topic
      - > Topic.java
      - > TopicController.java
      - > TopicService.java
- > src/main/resources
- > src/test/java
- > JRE System Library [JavaSE-1.8]
- > Maven Dependencies
- > src
  - > target
    - > mvnw
    - > mvnw.cmd
    - > pom.xml
- > Servers

## Servers

## Boot Dashboard

Type tags to match

&gt; local

## TopicService.java course-api-data/pom.xml Topic.java

```
import javax.persistence.Entity;  
  
@Entity  
public class Topic {  
  
    private String id;  
    private String name;  
    private String description;  
  
    public Topic() {  
    }  
  
    public Topic(String id, String name, String description) {  
        super();  
        this.id = id;  
        this.name = name;  
        this.description = description;  
    }  
  
    public String getId() {  
        return id;  
    }  
    public void setId(String id) {  
        this.id = id;  
    }  
    public String getName() {  
        return name;  
    }  
    public void setName(String name) {  
        this.name = name;  
    }  
    public String getDescription() {  
        return description;  
    }  
    public void setDescription(String description) {  
        this.description = description;  
    }  
}
```

## Package Explorer

- > course-api [boot]
- > course-api-data [boot]
  - > src/main/java
    - > io.javabrains
    - > io.javabrains.springbootstarter.topic
      - > Topic.java
      - > TopicController.java
      - > TopicService.java
- > src/main/resources
- > src/test/java
- > JRE System Library [JavaSE-1.8]
- > Maven Dependencies
- > src
  - > target
    - > mvnw
    - > mvnw.cmd
    - > pom.xml
- > Servers

## TopicService.java

## course-api-data/pom.xml

## Topic.java

```
import javax.persistence.Entity;
import javax.persistence.Id;

@Entity
public class Topic {

    @Id
    private String id;
    private String name;
    private String description;

    public Topic() {
    }

    public Topic(String id, String name, String description) {
        super();
        this.id = id;
        this.name = name;
        this.description = description;
    }

    public String getId() {
        return id;
    }
    public void setId(String id) {
        this.id = id;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getDescription() {
        return description;
    }
    public void setDescription(String description) {
        this.description = description;
    }
}
```

## Servers

## Boot Dashboard

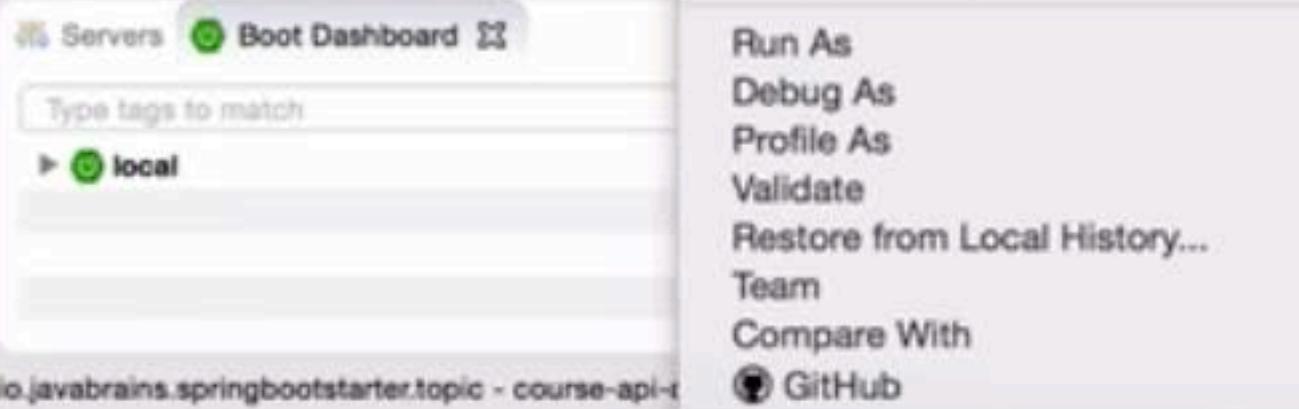
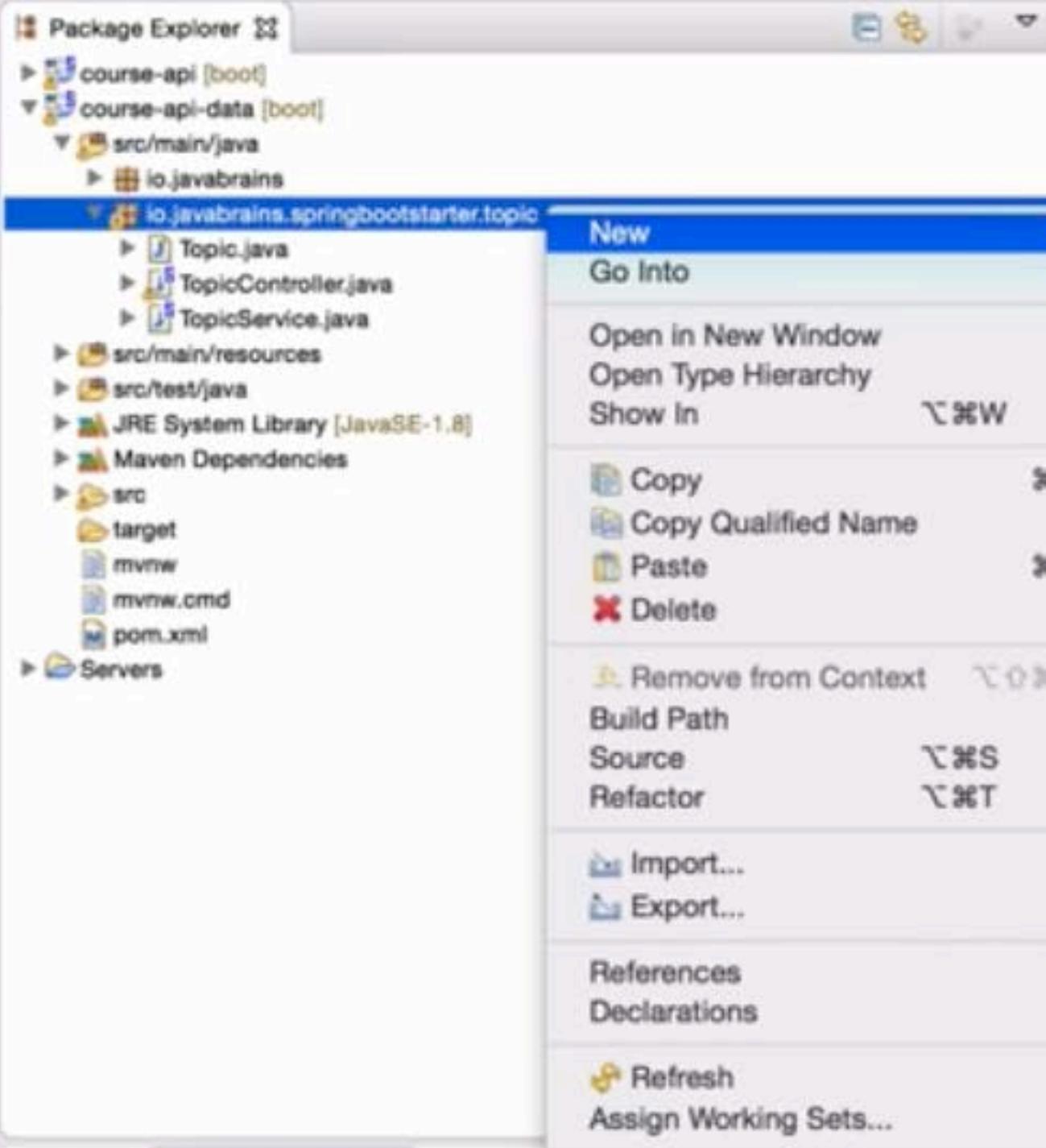
Type tags to match

&gt; local

## TopicService.java

## course-api-data/pom.xml

## Topic.java



TopicService.java

```

package io.javabrains.springbootstarter.topic;

import java.util.ArrayList;

@Service
public class TopicService {
    private ArrayList<Topic> topics = new ArrayList<Topic>();

    public void addTopic(Topic topic) {
        topics.add(topic);
    }

    public ArrayList<Topic> getAllTopics() {
        return topics;
    }

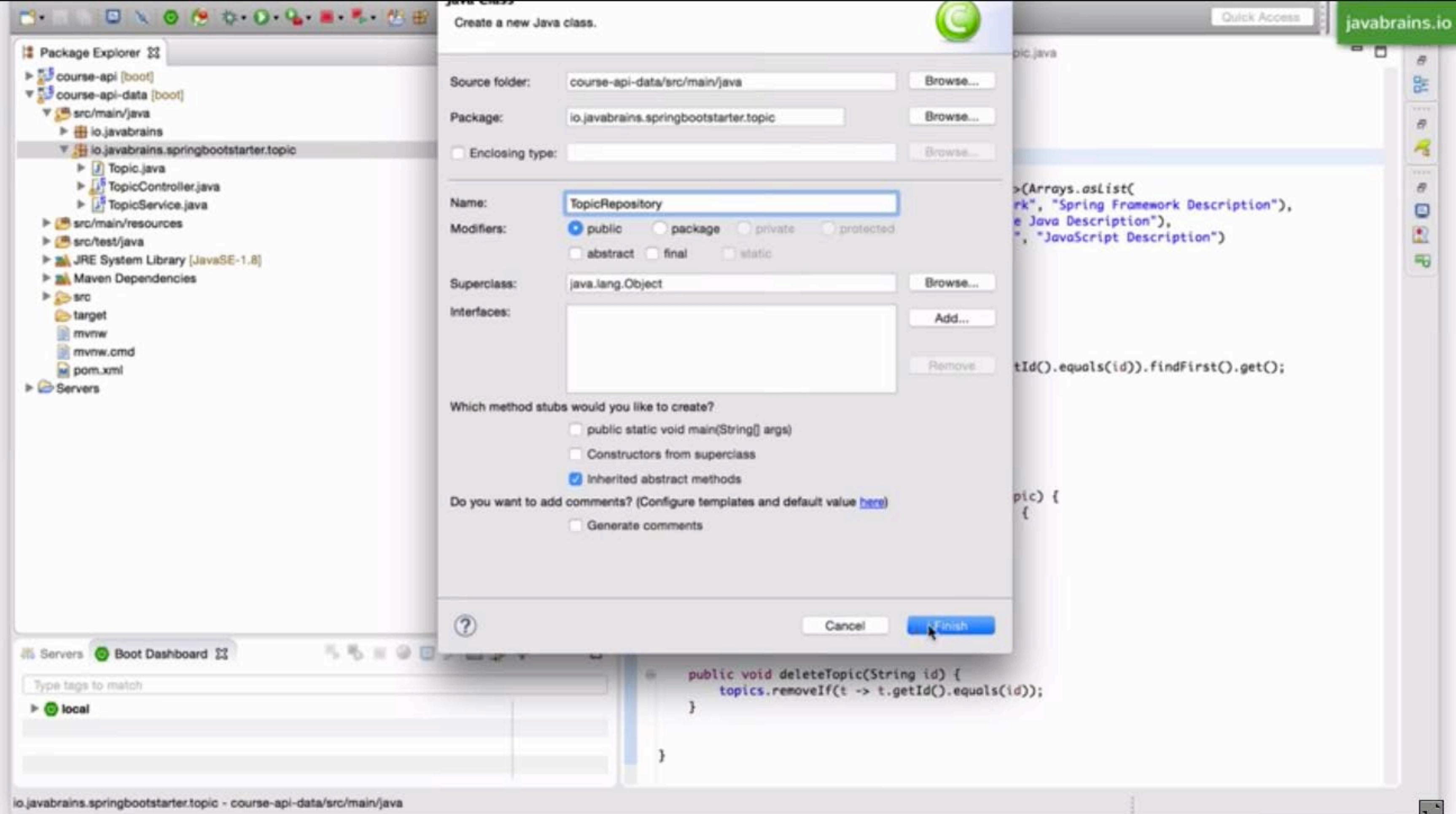
    public Topic getTopic(String id) {
        return topics.stream().filter(t -> t.getId().equals(id)).findFirst().get();
    }

    public void updateTopic(Topic topic) {
        topics.set(
            topics.indexOf(getTopic(topic.getId())),
            topic
        );
    }

    public void deleteTopic(String id) {
        topics.removeIf(t -> t.getId().equals(id));
    }
}

```





## Package Explorer

- > course-api [boot]
- > course-api-data [boot]
  - > src/main/java
    - > io.javabrains
    - > io.javabrains.springbootstarter.topic
      - > Topic.java
      - > TopicController.java
      - > TopicRepository.java
      - > TopicService.java
- > src/main/resources
- > src/test/java
- > JRE System Library [JavaSE-1.8]
- > Maven Dependencies
- > src
  - > target
  - > mvnw
  - > mvnw.cmd
  - > pom.xml
- > Servers

## TopicService.java

## course-api-data/pom.xml

## Topic.java

## TopicRepository.java

```
package io.javabrains.springbootstarter.topic;

public class TopicRepository {

    // getAllTopics()
    // getTopic(String id)
    // updateTopic(Topic t)
    // deleteTopic(String id)

}
```

## Servers

## Boot Dashboard

Type tags to match:

&gt; local

## Package Explorer

- > course-api [boot]
- > course-api-data [boot]
  - > src/main/java
    - > io.javabrains
    - > io.javabrains.springbootstarter.topic
      - > Topic.java
      - > TopicController.java
      - > TopicRepository.java
      - > TopicService.java
- > src/main/resources
- > src/test/java
- > JRE System Library [JavaSE-1.8]
- > Maven Dependencies
- > src
  - > target
  - > mvnw
  - > mvnw.cmd
  - > pom.xml
- > Servers

## TopicService.java

## course-api-data/pom.xml

## Topic.java

## TopicRepository.java

```
package io.javabrains.springbootstarter.topic;

public class TopicRepository {
```

## Servers

## Boot Dashboard

Type tags to match:

&gt; local

## Package Explorer

- > course-api [boot]
- > course-api-data [boot]
  - > src/main/java
    - > io.javabrains
    - > io.javabrains.springbootstarter.topic
      - > Topic.java
      - > TopicController.java
      - > TopicRepository.java
      - > TopicService.java
- > src/main/resources
- > src/test/java
- > JRE System Library [JavaSE-1.8]
- > Maven Dependencies
- > src
  - > target
  - > mvnw
  - > mvnw.cmd
  - > pom.xml
- > Servers

## TopicService.java

## course-api-data/pom.xml

## Topic.java

## TopicRepository.java

```
package io.javabrains.springbootstarter.topic;

public interface TopicRepository {
```

## Servers

## Boot Dashboard

Type tags to match:

&gt; local

## Package Explorer

- > course-api [boot]
- > course-api-data [boot]
  - > src/main/java
    - > io.javabrains
    - > io.javabrains.springbootstarter.topic
      - > Topic.java
      - > TopicController.java
      - > TopicRepository.java
      - > TopicService.java
- > src/main/resources
- > src/test/java
- > JRE System Library [JavaSE-1.8]
- > Maven Dependencies
- > src
  - > target
  - > mvnw
  - > mvnw.cmd
  - > pom.xml
- > Servers

## TopicService.java

## course-api-data/pom.xml

## Topic.java

## TopicRepository.java

```
package io.javabrains.springbootstarter.topic;

public interface TopicRepository extends CrudRepository {
    io.javabrains.springbootstarter.topic.TopicRepository
}
```

Press 'F2' for focus

## Servers

## Boot Dashboard



Type tags to match:

- > local

## Package Explorer

- > course-api [boot]
- > course-api-data [boot]
  - > src/main/java
    - > io.javabrains
    - > io.javabrains.springbootstarter.topic
      - > Topic.java
      - > TopicController.java
      - > TopicRepository.java
      - > TopicService.java
  - > src/main/resources
  - > src/test/java
  - > JRE System Library [JavaSE-1.8]
  - > Maven Dependencies
  - > src
    - > target
    - > mvnw
    - > mvnw.cmd
    - > pom.xml

## Servers

## TopicService.java

## course-api-data/pom.xml

## Topic.java

## TopicRepository.java

```
package io.javabrains.springbootstarter.topic;

public interface TopicRepository extends CrudRepository {
```

}

CrudRepository cannot be resolved to a type

4 quick fixes available:

- Import 'CrudRepository' (org.springframework.data.repository)
- Create interface 'CrudRepository'
- Change to 'ClassloaderRepository' (javax.management.loading)
- Fix project setup...

Press 'F2' for focus

## Servers

## Boot Dashboard

Type tags to match

> local

## Package Explorer

- > course-api [boot]
- > course-api-data [boot]
  - > src/main/java
    - > io.javabrains
    - > io.javabrains.springbootstarter.topic
      - > Topic.java
      - > TopicController.java
      - > TopicRepository.java
      - > TopicService.java
- > src/main/resources
- > src/test/java
- > JRE System Library [JavaSE-1.8]
- > Maven Dependencies
- > src
  - > target
  - > mvnw
  - > mvnw.cmd
  - > pom.xml
- > Servers

## TopicService.java

## course-api-data/pom.xml

## Topic.java

## TopicRepository.java

```
package io.javabrains.springbootstarter.topic;

import org.springframework.data.repository.CrudRepository;

public interface TopicRepository extends CrudRepository {
```

}

## Servers

## Boot Dashboard

Type tags to match

&gt; local

## Package Explorer

- > course-api [boot]
- > course-api-data [boot]
  - src/main/java
    - > io.javabrains
    - > io.javabrains.springbootstarter.topic
      - > Topic.java
      - > TopicController.java
      - > TopicRepository.java
      - > TopicService.java
- > src/main/resources
- > src/test/java
- > JRE System Library [JavaSE-1.8]
- > Maven Dependencies
- > src
  - target
  - mvnw
  - mvnw.cmd
  - pom.xml
- > Servers

## TopicService.java

```
package io.javabrains.springbootstarter.topic;

import org.springframework.data.repository.CrudRepository;

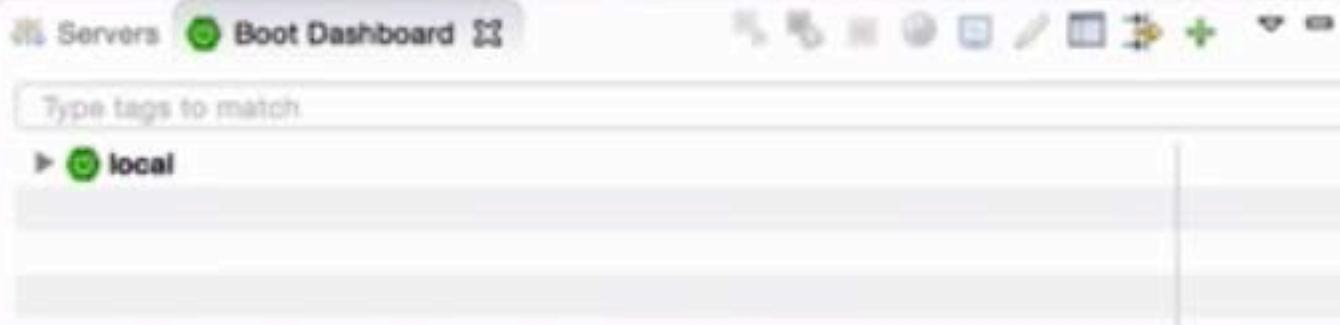
public interface TopicRepository extends CrudRepository {
```

CrudRepository is a raw type. References to generic type CrudRepository<T, ID> should be parameterized.

3 quick fixes available:

- Add type arguments to 'CrudRepository'
- Infer Generic Type Arguments...
- Add @SuppressWarnings 'rawtypes' to 'TopicRepository'

## Servers



## Package Explorer

- > course-api [boot]
- > course-api-data [boot]
  - > src/main/java
    - > io.javabrains
    - > io.javabrains.springbootstarter.topic
      - > Topic.java
      - > TopicController.java
      - > TopicRepository.java
      - > TopicService.java
- > src/main/resources
- > src/test/java
- > JRE System Library [JavaSE-1.8]
- > Maven Dependencies
- > src
  - > target
  - > mvnw
  - > mvnw.cmd
  - > pom.xml
- > Servers

## TopicService.java

## course-api-data/pom.xml

## Topic.java

## TopicRepository.java

```
package io.javabrains.springbootstarter.topic;

import org.springframework.data.repository.CrudRepository;

public interface TopicRepository extends CrudRepository<Topic, > {

}
```

## Servers

## Boot Dashboard

Type tags to match:

&gt; local

## Package Explorer

- > course-api [boot]
- > course-api-data [boot]
  - > src/main/java
    - > io.javabrains
    - > io.javabrains.springbootstarter.topic
      - > Topic.java
      - > TopicController.java
      - > TopicRepository.java
      - > TopicService.java
- > src/main/resources
- > src/test/java
- > JRE System Library [JavaSE-1.8]
- > Maven Dependencies
- > src
  - > target
  - > mvnw
  - > mvnw.cmd
  - > pom.xml
- > Servers

## TopicService.java

## course-api-data/pom.xml

## Topic.java

## TopicRepository.java

```
package io.javabrains.springbootstarter.topic;

import org.springframework.data.repository.CrudRepository;

public interface TopicRepository extends CrudRepository<Topic, String> {
```

## Servers

## Boot Dashboard

Type tags to match:

&gt; local

Package Explorer

- course-api [boot]
- course-api-data [boot]
  - src/main/java
    - io.javabrains
    - io.javabrains.springbootstarter.topic
      - Topic.java
      - TopicController.java
      - TopicRepository.java
      - TopicService.java
- src/main/resources
- src/test/java
- JRE System Library [JavaSE-1.8]
- Maven Dependencies
- src
  - target
  - mvnw
  - mvnw.cmd
- pom.xml

Servers

Boot Dashboard

Type tags to match

local

TopicService.java

```
package io.javabrains.springbootstarter.topic;

import java.util.ArrayList;

@Service
public class TopicService {

    private List<Topic> topics = new ArrayList<>(Arrays.asList(
        new Topic("spring", "Spring Framework", "Spring Framework Description"),
        new Topic("java", "Core Java", "Core Java Description"),
        new Topic("javascript", "JavaScript", "JavaScript Description")
    ));

    public List<Topic> getAllTopics() {
        return topics;
    }

    public Topic getTopic(String id) {
        return topics.stream().filter(t -> t.getId().equals(id)).findFirst().get();
    }

    public void addTopic(Topic topic) {
        topics.add(topic);
    }

    public void updateTopic(String id, Topic topic) {
        for (int i = 0; i < topics.size(); i++) {
            Topic t = topics.get(i);
            if (t.getId().equals(id)) {
                topics.set(i, topic);
                return;
            }
        }
    }

    public void deleteTopic(String id) {
        topics.removeIf(t -> t.getId().equals(id));
    }
}
```

course-api-data/pom.xml

Topic.java

TopicRepository.java

Writable

Smart Insert

12 : 11

Package Explorer    TopicService.java    course-api-data/pom.xml    Topic.java    TopicRepository.java

```
package io.javabrains.springbootstarter.topic;

import java.util.ArrayList;

@Service
public class TopicService {

    @Autowired
    private TopicRepository repository;

    public List<Topic> getAllTopics() {
        return repository.findAll();
    }

    public Topic getTopic(String id) {
        return repository.findById(id).orElse(null);
    }

    public void addTopic(Topic topic) {
        repository.save(topic);
    }

    public void updateTopic(String id, Topic topic) {
        Topic existingTopic = repository.findById(id).orElse(null);
        if (existingTopic != null) {
            existingTopic.setTitle(topic.getTitle());
            existingTopic.setDescription(topic.getDescription());
            repository.save(existingTopic);
        }
    }

    public void deleteTopic(String id) {
        repository.deleteById(id);
    }
}
```

TopicService.java

course-api-data/pom.xml

Topic.java

TopicRepository.java

Package Explorer

course-api [boot]

course-api-data [boot]

src/main/java

- io.javabrains
- io.javabrains.springbootstarter.topic
  - Topic.java
  - TopicController.java
  - TopicRepository.java
  - TopicService.java

src/main/resources

src/test/java

JRE System Library [JavaSE-1.8]

Maven Dependencies

src

target

mvnw

mvnw.cmd

pom.xml

Servers

Servers    Boot Dashboard

Type tags to match

local

Writable    Smart Insert    12 : 15

A code editor window showing the `TopicService.java` file. A tooltip is open at the `@Autowired` annotation, indicating that the annotation cannot be resolved to a type. Three quick fix options are listed: "Import 'Autowired' (org.springframework.beans.factory.annotation)", "Create annotation 'Autowired'", and "Fix project setup...". The "Import 'Autowired'" option is currently selected.

Package Explorer

- course-api [boot]
- course-api-data [boot]
  - src/main/java
    - io.javabrains
    - io.javabrains.springbootstarter.topic
      - Topic.java
      - TopicController.java
      - TopicRepository.java
      - TopicService.java
  - src/main/resources
  - src/test/java
  - JRE System Library [JavaSE-1.8]
  - Maven Dependencies
  - src
    - target
    - mvnw
    - mvnw.cmd
    - pom.xml
- Servers

TopicService.java

```
@Service
public class TopicService {

    @Autowired
    private TopicRepository topicRepository;

    private List<Topic> topics = new ArrayList<>(Arrays.asList(
        new Topic("spring", "Spring Framework", "Spring Framework Description"),
        new Topic("java", "Core Java", "Core Java Description"),
        new Topic("javascript", "JavaScript", "JavaScript Description")
    ));

    public List<Topic> getAllTopics() {
        // return topics;
        topicRepository.
    }

    public Topic getTopic(String id) {
        return topics.stream()
            .filter(t -> t.getId().equals(id))
            .findFirst().get();
    }

    public void addTopic(Topic topic) {
        topics.add(topic);
    }

    public void updateTopic(String id, Topic topic) {
        for (int i = 0; i < topics.size(); i++) {
            Topic t = topics.get(i);
            if (t.getId().equals(id)) {
                topics.set(i, topic);
                return;
            }
        }
    }

    public void deleteTopic(String id) {
        topics.removeIf(t -> t.getId().equals(id));
    }
}
```

Topic.java

TopicRepository.java

Boot Dashboard

Type tags to match

local

Writable Smart Insert 25 : 25

Package Explorer

- course-api [boot]
- course-api-data [boot]
  - src/main/java
    - io.javabrains
    - io.javabrains.springbootstarter.topic
      - Topic.java
      - TopicController.java
      - TopicRepository.java
      - TopicService.java
  - src/main/resources
  - src/test/java
  - JRE System Library [JavaSE-1.8]
  - Maven Dependencies
  - src
    - target
    - mvnw
    - mvnw.cmd
    - pom.xml
- Servers

TopicService.java

```
@Service
public class TopicService {

    @Autowired
    private TopicRepository topicRepository;

    private List<Topic> topics = new ArrayList<>(Arrays.asList(
        new Topic("spring", "Spring Framework", "Spring Framework Description"),
        new Topic("java", "Core Java", "Core Java Description"),
        new Topic("javascript", "JavaScript", "JavaScript Description")
    ));

    public List<Topic> getAllTopics() {
        // return topics;
        topicRepository.
    }

    public Topic getTopic(String id) {
        return topics.stream()
            .filter(t -> t.getId().equals(id))
            .findFirst().get();
    }

    public void addTopic(Topic topic) {
        topics.add(topic);
    }

    public void updateTopic(String id, Topic topic) {
        for (int i = 0; i < topics.size(); i++) {
            Topic t = topics.get(i);
            if (t.getId().equals(id)) {
                topics.set(i, topic);
                return;
            }
        }
    }

    public void deleteTopic(String id) {
        topics.removeIf(t -> t.getId().equals(id));
    }
}
```

course-api-data/pom.xml

Topic.java

TopicRepository.java

Boot Dashboard

Type tags to match

local

Writable Smart Insert 25 : 25

Package Explorer

- course-api [boot]
- course-api-data [boot]
  - src/main/java
    - io.javabrains
    - io.javabrains.springbootstarter.topic
      - Topic.java
      - TopicController.java
      - TopicRepository.java
      - TopicService.java
  - src/main/resources
  - src/test/java
  - JRE System Library [JavaSE-1.8]
  - Maven Dependencies
  - src
    - target
    - mvnw
    - mvnw.cmd
  - pom.xml
- Servers

TopicService.java

```
public class TopicService {  
    @Autowired  
    private TopicRepository topicRepository;  
  
    private List<Topic> topics = new ArrayList<>(Arrays.asList(  
        new Topic("spring", "Spring Framework", "Spring Framework Description"),  
        new Topic("java", "Core Java", "Core Java Description"),  
        new Topic("javascript", "JavaScript", "JavaScript Description")  
    ));  
  
    public List<Topic> getAllTopics() {  
        // return topics; |  
        List<Topic> topics = new ArrayList<>();  
        topicRepository.findAll()  
            .forEach(topics::add);  
        return topics;  
    }  
  
    public Topic getTopic(String id) {  
        return topics.stream().filter(t -> t.getId().equals(id)).findFirst().get();  
    }  
  
    public void addTopic(Topic topic) {  
        topics.add(topic);  
    }  
  
    public void updateTopic(String id, Topic topic) {  
        for (int i = 0; i < topics.size(); i++) {  
            Topic t = topics.get(i);  
            if (t.getId().equals(id)) {  
                topics.set(i, topic);  
                return;  
            }  
        }  
    }  
  
    public void deleteTopic(String id) {  
        topics.removeIf(t -> t.getId().equals(id));  
    }  
}
```

Servers

Boot Dashboard

Type tags to match

local

Writable Smart Insert 27 : 31

Package Explorer

- course-api [boot]
- course-api-data [boot]
  - src/main/java
    - io.javabrains
    - io.javabrains.springbootstarter.topic
      - Topic.java
      - TopicController.java
      - TopicRepository.java
      - TopicService.java
  - src/main/resources
  - src/test/java
  - JRE System Library [JavaSE-1.8]
  - Maven Dependencies
  - src
  - target
  - mvnw
  - mvnw.cmd
  - pom.xml

Servers

Boot Dashboard

Type tags to match

local

TopicService.java

```
public class TopicService {  
    @Autowired  
    private TopicRepository topicRepository;  
  
    private List<Topic> topics = new ArrayList<>(Arrays.asList(  
        new Topic("spring", "Spring Framework", "Spring Framework Description"),  
        new Topic("java", "Core Java", "Core Java Description"),  
        new Topic("javascript", "JavaScript", "JavaScript Description")  
    ));  
  
    public List<Topic> getAllTopics() {  
        //  
        List<Topic> topics = new ArrayList<>();  
        topicRepository.findAll()  
            .forEach(topics::add);  
        return topics;  
    }  
  
    public Topic getTopic(String id) {  
        return topics.stream().filter(t -> t.getId().equals(id)).findFirst().get();  
    }  
  
    public void addTopic(Topic topic) {  
        topicRepository.  
    }  
  
    public void updateTopic(Topic topic) {  
        for (int i = 0; i < topics.size(); i++) {  
            Topic t = topics.get(i);  
            if (t.getId().equals(topic.getId())) {  
                topics.set(i, topic);  
                return;  
            }  
        }  
    }  
  
    public void deleteTopic(String id) {  
        topics.removeIf(t -> t.getId().equals(id));  
    }  
}
```

topicRepository.

  - findOne(String arg0) : Topic - CrudRepository
  - getClass() : Class<?> - Object
  - hashCode() : int - Object
  - notify() : void - Object
  - notifyAll() : void - Object
  - save(Iterable<S> arg0) : Iterable<S> - CrudRepository
  - save(S arg0) : S - CrudRepository
  - toString() : String - Object
  - wait() : void - Object
  - wait(long timeout) : void - Object
  - wait(long timeout, int nanos) : void - Object

Press '^Space' to show Template Proposals

Writable Smart Insert 36 : 25

Package Explorer

- course-api [boot]
- course-api-data [boot]
  - src/main/java
    - io.javabrains
    - io.javabrains.springbootstarter.topic
      - Topic.java
      - TopicController.java
      - TopicRepository.java
      - TopicService.java
  - src/main/resources
  - src/test/java
  - JRE System Library [JavaSE-1.8]
  - Maven Dependencies
  - src
  - target
  - mvnw
  - mvnw.cmd
  - pom.xml

Servers

Boot Dashboard

Type tags to match

local

TopicService.java

```
public class TopicService {  
    @Autowired  
    private TopicRepository topicRepository;  
  
    private List<Topic> topics = new ArrayList<>(Arrays.asList(  
        new Topic("spring", "Spring Framework", "Spring Framework Description"),  
        new Topic("java", "Core Java", "Core Java Description"),  
        new Topic("javascript", "JavaScript", "JavaScript Description")  
    ));  
  
    public List<Topic> getAllTopics() {  
        //  
        List<Topic> topics = new ArrayList<>();  
        topicRepository.findAll()  
            .forEach(topics::add);  
        return topics;  
    }  
  
    public Topic getTopic(String id) {  
        return topics.stream().filter(t -> t.getId().equals(id)).findFirst().get();  
    }  
  
    public void addTopic(Topic topic) {  
        topicRepository.save(topic);  
    }  
  
    public void updateTopic(String id, Topic topic) {  
        for (int i = 0; i < topics.size(); i++) {  
            Topic t = topics.get(i);  
            if (t.getId().equals(id)) {  
                topics.set(i, topic);  
                return;  
            }  
        }  
    }  
  
    public void deleteTopic(String id) {  
        topics.removeIf(t -> t.getId().equals(id));  
    }  
}
```

Writable Smart Insert 36 : 37

The screenshot shows the Eclipse IDE interface with the following details:

- Package Explorer:** Shows the project structure with two modules: "course-api [boot]" and "course-api-data [boot]". The "src/main/java" folder under "course-api-data" contains packages "io.javabrains" and "io.javabrains.springbootstarter.topic". Inside "topic", there are files: Topic.java, TopicController.java, TopicRepository.java, and TopicService.java.
- TopicService.java Content:** The code defines a service class TopicService that interacts with a TopicRepository. It includes methods to get all topics, get a specific topic by ID, add a new topic, and update an existing topic. The code uses Java 8 features like streams and lambdas.

```
public class TopicService {  
    @Autowired  
    private TopicRepository topicRepository;  
  
    private List<Topic> topics = new ArrayList<>(Arrays.asList(  
        new Topic("spring", "Spring Framework", "Spring Framework Description"),  
        new Topic("java", "Core Java", "Core Java Description"),  
        new Topic("javascript", "JavaScript", "JavaScript Description")  
    ));  
  
    public List<Topic> getAllTopics() {  
        List<Topic> topics = new ArrayList<>();  
        topicRepository.findAll()  
            .forEach(topics::add);  
        return topics;  
    }  
  
    public Topic getTopic(String id) {  
        return topics.stream().filter(t -> t.getId().equals(id)).findFirst().get();  
    }  
  
    public void addTopic(Topic topic) {  
        topicRepository.save(topic);  
    }  
  
    public void updateTopic(String id, Topic topic) {  
        for (int i = 0; i < topics.size(); i++) {  
            Topic t = topics.get(i);  
            if (t.getId().equals(id)) {  
                topics.set(i, topic);  
                return;  
            }  
        }  
    }  
}
```



The framework sees the embedded Derby database in the classpath and assumes that to be the database to connect to. No connection information necessary!

Package Explorer

- course-api [boot]
- course-api-data [boot]
  - src/main/java
    - io.javabrains

TopicService.java

```
package io.javabrains;  
  
import org.springframework.boot.Spr  
  
@SpringBootApplication  
public class CourseApiDataApplicati  
    public static void main(String[] args)  
        SpringApplication.run(Cours  
    }  
}
```

CourseApiDataAp

File

- Undo
- Revert File
- Save
- Open Declaration F3
- Open Type Hierarchy F4
- Open Call Hierarchy ⌘⌥H
- Show in Breadcrumb ⌘⌘B
- Quick Outline ⌘O
- Quick Type Hierarchy ⌘T
- Open With ▶
- Show In ⌘⌘W ▶
- Cut ⌘X
- Copy ⌘C
- Copy Qualified Name
- Paste ⌘V
- Quick Fix ⌘1
- Source ⌘⌘S ▶
- Refactor ⌘⌘T ▶
- Local History ▶
- References ▶
- Declarations ▶
- Add to Snippets... ▶
- AspectJ Refactoring ▶
- Run As ▶
- Debug As ▶
- Profile As ▶
- Validate ▶
- GitHub ▶
- Team ▶
- Compare With ▶
- Replace With ▶
- Preferences...

Servers

Boot Dashboard

Type tags to match

local

Writable

1 Java Application ⌘⌘X J

2 Spring Boot App ⌘X B

Run Configurations...

Runner Import L+ Builder Team Library OFFLINE Sign In process javabrain

Filter History Collections http://localhost:8080/ + No Environment

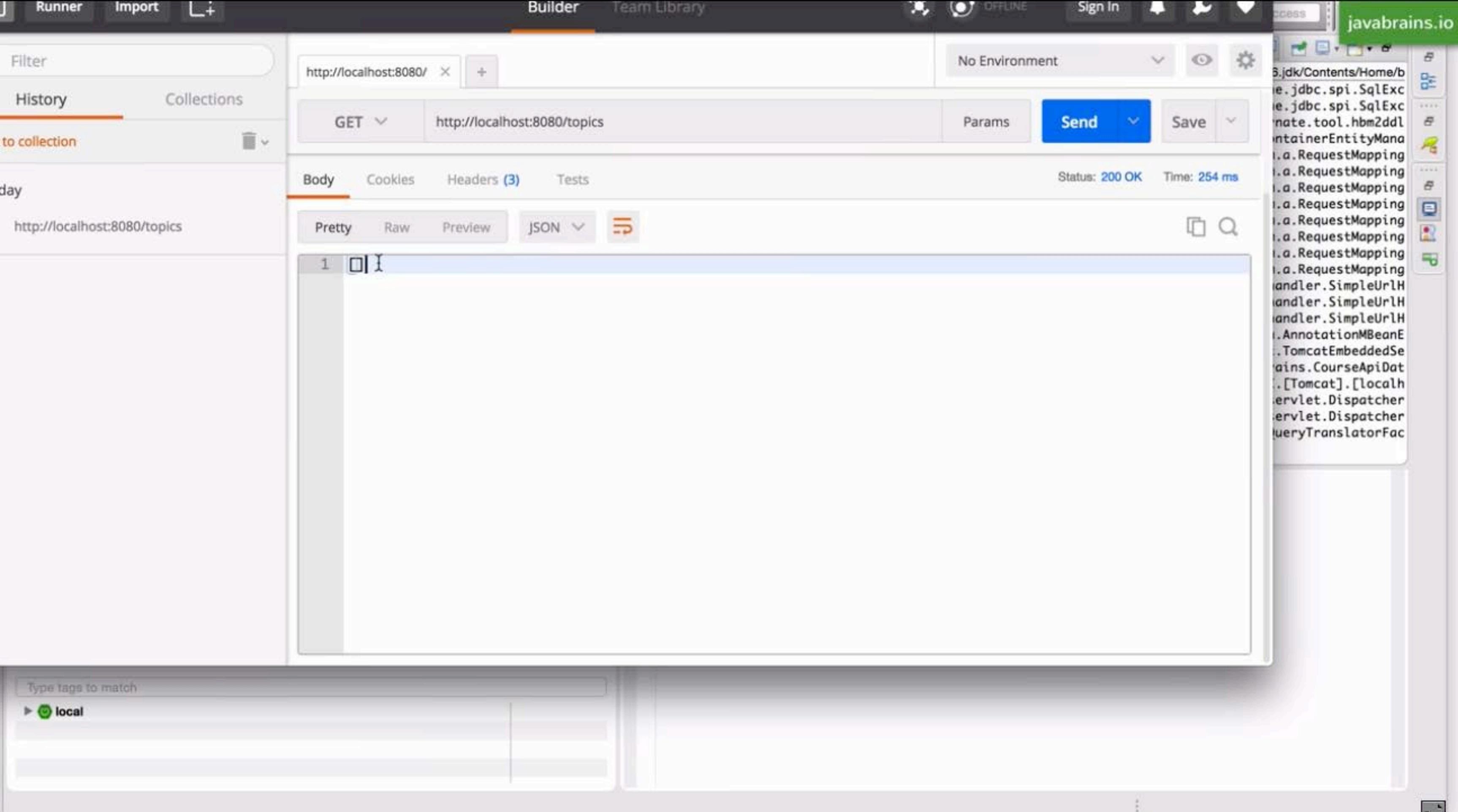
GET http://localhost:8080/topics Params Send Save

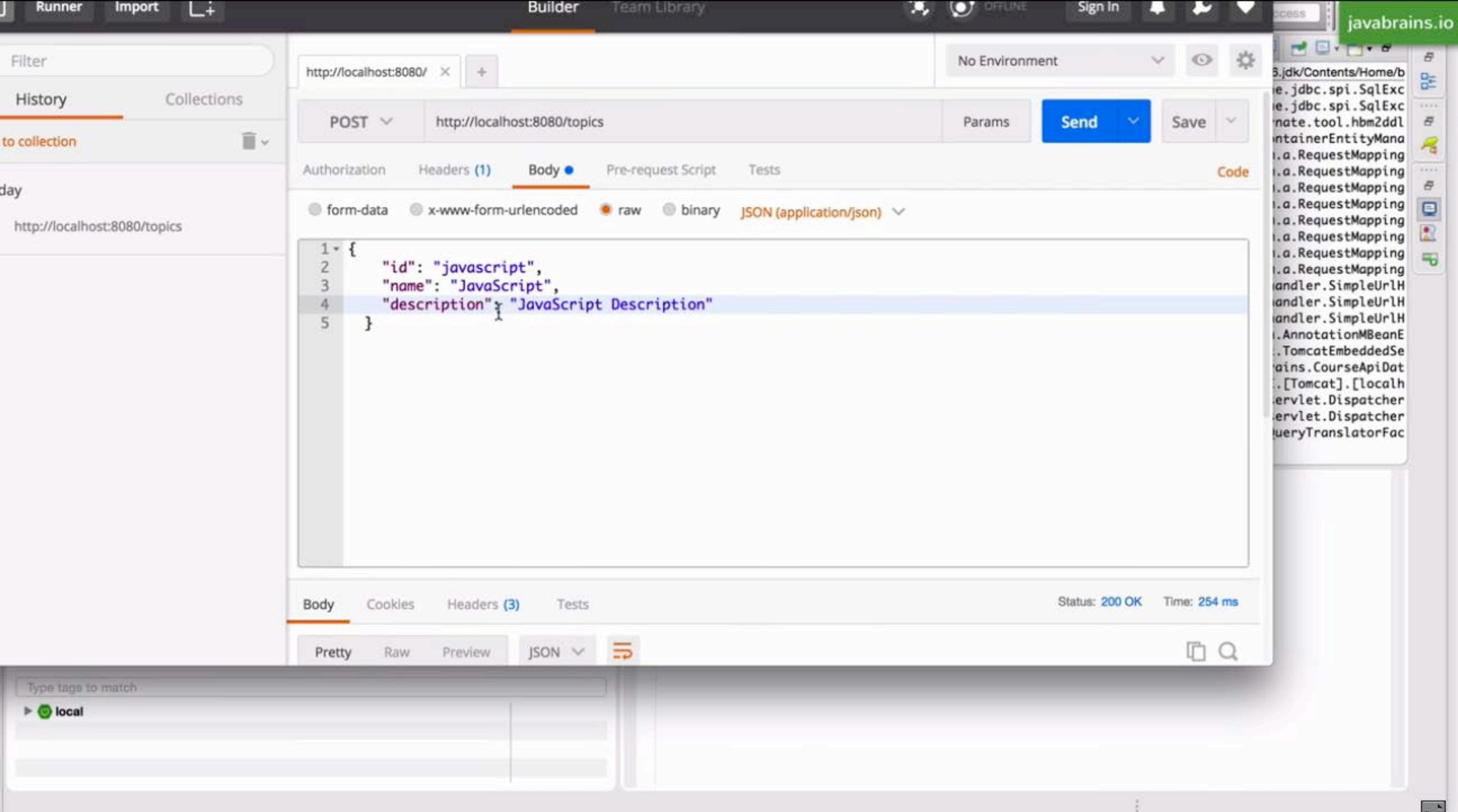
Body Cookies Headers (3) Tests Status: 200 OK Time: 191 ms

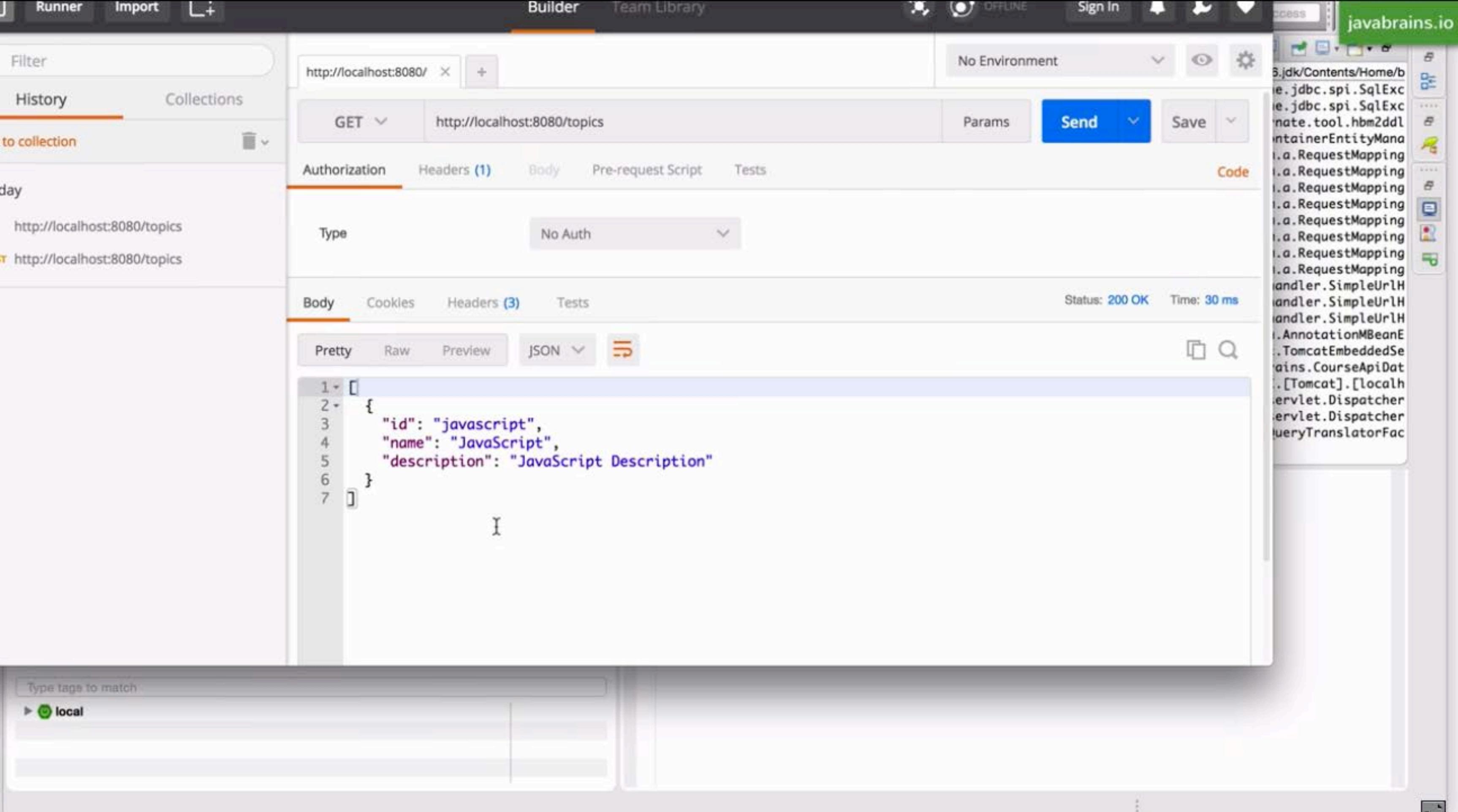
http://localhost:8080/topics Pretty Raw Preview JSON

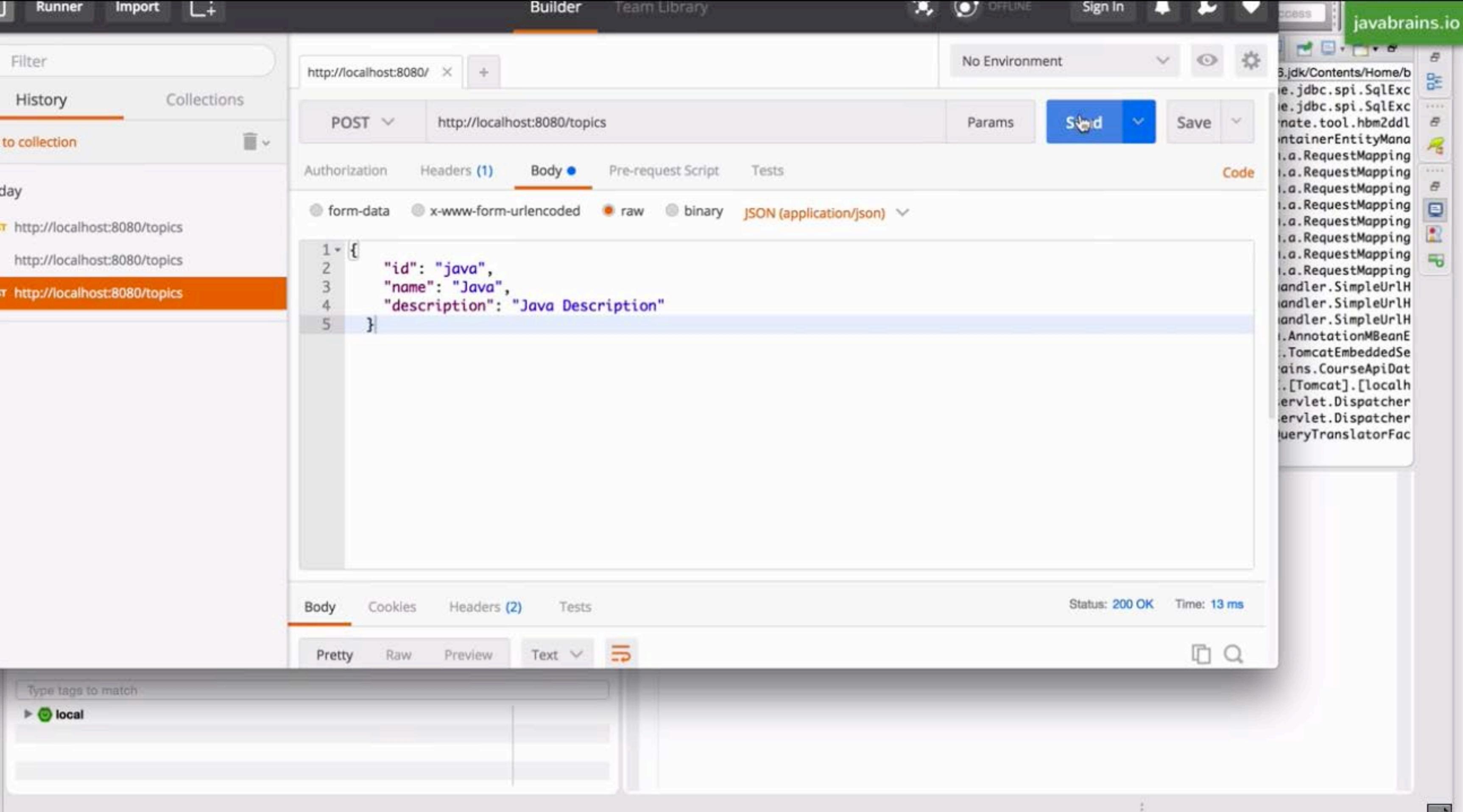
```
1 [
2   {
3     "id": "spring",
4     "name": "Spring Framework",
5     "description": "Spring Framework Description"
6   },
7   {
8     "id": "java",
9     "name": "Core Java",
10    "description": "Core Java Description"
11  },
12  {
13    "id": "javascript",
14    "name": "JavaScript",
15    "description": "JavaScript Description"
16  }
17 ]
```

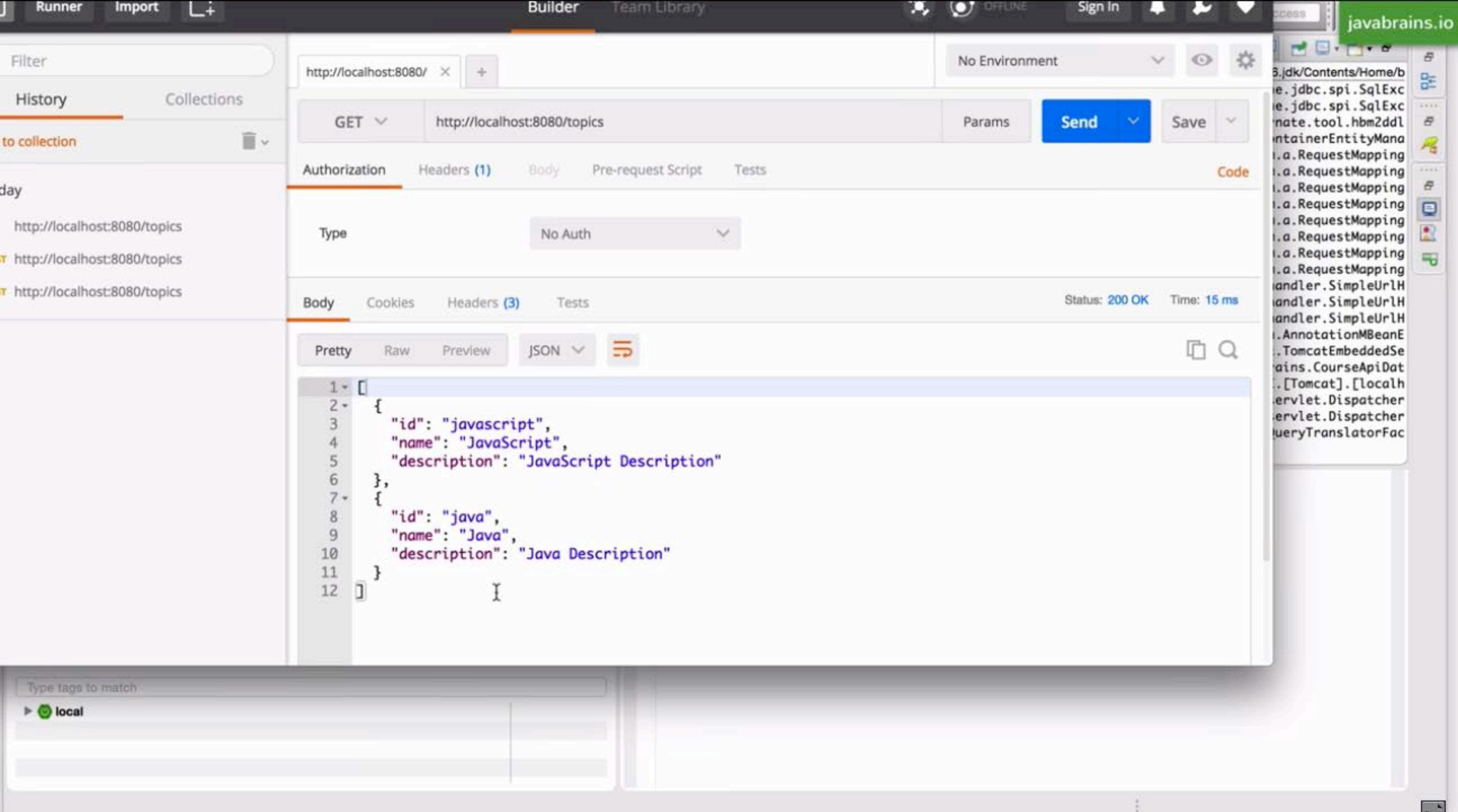
Type tags to match local Writable Smart Insert 13 : 1











Package Explorer

- course-api [boot]
- course-api-data [boot]
  - src/main/java
    - io.javabrains
- src/main/resources
- src/test/java
- JRE System Library [JavaSE-1.8]
- Maven Dependencies
- src
- target
- mvnw
- mvnw.cmd
- pom.xml

Servers

Boot Dashboard

Type tags to match

local

TopicService.j course-api-data/ Topic.java TopicRepository.j CourseApiDataAp

```
@Autowired
private TopicRepository topicRepository;

private List<Topic> topics = new ArrayList<>(Arrays.asList(
    new Topic("spring", "Spring Framework", "Spring Framework Description"),
    new Topic("java", "Core Java", "Core Java Description"),
    new Topic("javascript", "JavaScript", "JavaScript Description")
));

public List<Topic> getAllTopics() {
    //
    List<Topic> topics = new ArrayList<>();
    topicRepository.findAll()
        .forEach(topics::add);
    return topics;
}

public Topic getTopic(String id) {
    // return topics.stream().filter(t -> t.getId().equals(id)).findFirst().get();
    topicRepository.
}

public void addTopic
topicRepository.

}

public void updateTo
for (int i = 0;
Topic t = to
if (t.getId()
topics.s
return;
}

}

public void deleteTopic(String id) {
    topics.removeIf(t -> t.getId().equals(id));
}
```

Press '^Space' to show Template Proposals

Package Explorer

- course-api [boot]
- course-api-data [boot]
  - src/main/java
    - io.javabrains
- src/main/resources
- src/test/java
- JRE System Library [JavaSE-1.8]
- Maven Dependencies
- src
- target
  - mvnw
  - mvnw.cmd
  - pom.xml
- Servers

TopicService.j course-api-data/ Topic.java TopicRepository.j CourseApiDataAp

```
@Autowired
private TopicRepository topicRepository;

private List<Topic> topics = new ArrayList<>(Arrays.asList(
    new Topic("spring", "Spring Framework", "Spring Framework Description"),
    new Topic("java", "Core Java", "Core Java Description"),
    new Topic("javascript", "JavaScript", "JavaScript Description")
));

public List<Topic> getAllTopics() {
    //
    List<Topic> topics = new ArrayList<>();
    topicRepository.findAll()
        .forEach(topics::add);
    return topics;
}

public Topic getTopic(String id) {
    // return topics.stream().filter(t -> t.getId().equals(id)).findFirst().get();
    return topicRepository.findOne(id);
}

public void addTopic(Topic topic) {
    topicRepository.save(topic);
}

public void updateTopic(String id, Topic topic) {
    for (int i = 0; i < topics.size(); i++) {
        Topic t = topics.get(i);
        if (t.getId().equals(id)) {
            topics.set(i, topic);
            return;
        }
    }
}

public void deleteTopic(String id) {
    topics.removeIf(t -> t.getId().equals(id));
}
```

Servers Boot Dashboard

Type tags to match

local

Writable Smart Insert 33 : 15

Package Explorer

- course-api [boot]
- course-api-data [boot]
  - src/main/java
    - io.javabrains

CourseApiDataApplication.java

io.javabrains.springbootstarter.topic

- Topic.java
- TopicController.java
- TopicRepository.java
- TopicService.java

src/main/resources

src/test/java

JRE System Library [JavaSE-1.8]

Maven Dependencies

src

- target
- mvnw
- mvnw.cmd
- pom.xml

Servers

Boot Dashboard

Type tags to match

local

TopicService.j course-api-data/ Topic.java TopicRepository.j CourseApiDataAp

```
import java.util.ArrayList;

@Service
public class TopicService {

    @Autowired
    private TopicRepository topicRepository;

    private List<Topic> topics = new ArrayList<>(Arrays.asList(
        new Topic("spring", "Spring Framework", "Spring Framework Description"),
        new Topic("java", "Core Java", "Core Java Description"),
        new Topic("javascript", "JavaScript", "JavaScript Description")
    ));

    public List<Topic> getAllTopics() {
        List<Topic> topics = new ArrayList<>();
        topicRepository.findAll()
            .forEach(topics::add);
        return topics;
    }

    public Topic getTopic(String id) {
        return topicRepository.findOne(id);
    }

    public void addTopic(Topic topic) {
        topicRepository.save(topic);
    }

    public void updateTopic(String id, Topic topic) {
        topicRepository.save(topic);
    }

    public void deleteTopic(String id) {
        topics.removeIf(t -> t.getId().equals(id));
    }
}
```

Writable Smart Insert 41 : 31

Package Explorer

- course-api [boot]
- course-api-data [boot]
  - src/main/java
    - io.javabrains

src/main/java

- CourseApiDataApplication.java
- io.javabrains.springbootstarter.topic
  - Topic.java
  - TopicController.java
  - TopicRepository.java
  - TopicService.java

src/main/resources

src/test/java

JRE System Library [JavaSE-1.8]

Maven Dependencies

src

- target
- mvnw
- mvnw.cmd
- pom.xml

Servers

Boot Dashboard

Type tags to match

local

TopicService.j course-api-data/ Topic.java TopicRepository.j CourseApiDataAp

```
import java.util.ArrayList;

@Service
public class TopicService {

    @Autowired
    private TopicRepository topicRepository;

    private List<Topic> topics = new ArrayList<>(Arrays.asList(
        new Topic("spring", "Spring Framework", "Spring Framework Description"),
        new Topic("java", "Core Java", "Core Java Description"),
        new Topic("javascript", "JavaScript", "JavaScript Description")
    ));

    public List<Topic> getAllTopics() {
        //
        List<Topic> topics = new ArrayList<>();
        topicRepository.findAll()
            .forEach(topics::add);
        return topics;
    }

    public Topic getTopic(String id) {
        return topicRepository.findOne(id);
    }

    public void addTopic(Topic topic) {
        topicRepository.save(topic);
    }

    public void updateTopic(Topic topic) {
        topicRepository.save(topic);
    }

    public void deleteTopic(Topic topic) {
        topicRepository.delete(topic);
    }
}
```

count() : long - CrudRepository  
delete(Iterable<? extends Topic> arg0) : void - CrudRepository  
**delete(String arg0) : void - CrudRepository**  
delete(Topic arg0) : void - CrudRepository  
deleteAll() : void - CrudRepository  
equals(Object obj) : boolean - Object  
exists(String arg0) : boolean - CrudRepository  
findAll() : Iterable<Topic> - CrudRepository  
findAll(Iterable<String> arg0) : Iterable<Topic> - CrudRepository  
findOne(String arg0) : Topic - CrudRepository  
getClassName() : Class<?> - Object

Press '^Space' to show Template Proposals

Writable Smart Insert 46 : 25

Package Explorer

- course-api [boot]
- course-api-data [boot]
  - src/main/java
    - io.javabrains
- CourseApiDataApplication.java
- io.javabrains.springbootstarter.topic
  - Topic.java
  - TopicController.java
  - TopicRepository.java
  - TopicService.java
- src/main/resources
- src/test/java
- JRE System Library [JavaSE-1.8]
- Maven Dependencies
- src
  - target
  - mvnw
  - mvnw.cmd
  - pom.xml
- Servers

TopicService.j course-api-data/ Topic.java TopicRepository.j CourseApiDataAp

```
import java.util.ArrayList;

@Service
public class TopicService {

    @Autowired
    private TopicRepository topicRepository;

    private List<Topic> topics = new ArrayList<>(Arrays.asList(
        new Topic("spring", "Spring Framework", "Spring Framework Description"),
        new Topic("java", "Core Java", "Core Java Description"),
        new Topic("javascript", "JavaScript", "JavaScript Description")
    ));

    public List<Topic> getAllTopics() {
        List<Topic> topics = new ArrayList<>();
        topicRepository.findAll()
            .forEach(topics::add);
        return topics;
    }

    public Topic getTopic(String id) {
        return topicRepository.findOne(id);
    }

    public void addTopic(Topic topic) {
        topicRepository.save(topic);
    }

    public void updateTopic(String id, Topic topic) {
        topicRepository.save(topic);
    }

    public void deleteTopic(String id){
        topicRepository.delete(id);
    }
}
```

Servers Boot Dashboard

Type tags to match

local

Writable Smart Insert 46 : 36

Package Explorer

- course-api [boot]
- course-api-data [boot]
  - src/main/java
    - io.javabrains
- CourseApiDataApplication.java
- io.javabrains.springbootstarter.topic
  - Topic.java
  - TopicController.java
  - TopicRepository.java
  - TopicService.java
- src/main/resources
- src/test/java
- JRE System Library [JavaSE-1.8]
- Maven Dependencies
- src
- target
  - mvnw
  - mvnw.cmd
  - pom.xml
- Servers

TopicService.java

```
package io.javabrains.springbootstarter.topic;

import java.util.ArrayList;

@Service
public class TopicService {

    @Autowired
    private TopicRepository topicRepository;

    public List<Topic> getAllTopics() {
        List<Topic> topics = new ArrayList<>();
        topicRepository.findAll()
            .forEach(topics::add);
        return topics;
    }

    public Topic getTopic(String id) {
        return topicRepository.findOne(id);
    }

    public void addTopic(Topic topic) {
        topicRepository.save(topic);
    }

    public void updateTopic(String id, Topic topic) {
        topicRepository.save(topic);
    }

    public void deleteTopic(String id) {
        topicRepository.delete(id);
    }
}
```

Servers

Boot Dashboard

Type tags to match

local

Writable Smart Insert 17 : 1

Package Explorer    course-api [boot]    course-api-data [boot]    io.javabrains.springbootstarter.topic

src/main/java  
io.javabrains  
CourseApiDataApplication.java  
io.javabrains.springbootstarter.topic  
Topic.java  
TopicController.java  
TopicRepository.java  
TopicService.java  
src/main/resources  
src/test/java  
JRE System Library [JavaSE-1.8]  
Maven Dependencies  
src  
target  
mvnw  
mvnw.cmd  
pom.xml  
Servers

TopicService.java    course-api-data/    Topic.java    TopicRepository.java    CourseApiDataAp

```
package io.javabrains.springbootstarter.topic;

import java.util.ArrayList;

@Service
public class TopicService {

    @Autowired
    private TopicRepository topicRepository;

    public List<Topic> getAllTopics() {
        List<Topic> topics = new ArrayList<>();
        topicRepository.findAll()
            .forEach(topics::add);
        return topics;
    }

    public Topic getTopic(String id) {
        return topicRepository.findOne(id);
    }

    public void addTopic(Topic topic) {
        topicRepository.save(topic);
    }

    public void updateTopic(String id, Topic topic) {
        topicRepository.save(topic);
    }

    public void deleteTopic(String id) {
        topicRepository.delete(id);
    }
}
```

Servers    Boot Dashboard

Type tags to match

local

Writable    Smart Insert    21 : 23

Builder Team Library OFFLINE Sign In

History Collections

Save to collection

Today

GET http://localhost:8080/topics

POST http://localhost:8080/topics

POST http://localhost:8080/topics

Authorization Headers (1) Body Pre-request Script Tests Code

Type No Auth

Body Cookies Headers (3) Tests Status: 200 OK Time: 15 ms

Pretty Raw Preview JSON

```
1 [  
2 {  
3   "id": "javascript",  
4   "name": "JavaScript",  
5   "description": "JavaScript Description"  
6 },  
7 {  
8   "id": "java",  
9   "name": "Java",  
10  "description": "Java Description"  
11 }  
12 ]
```

Builder Team Library OFFLINE Sign In

Runner Import +

Filter History Collections

Save to collection

Today

GET http://localhost:8080/topics

POST http://localhost:8080/topics

POST http://localhost:8080/topics

No Environment

GET http://localhost:8080/topics Params Send Save

Authorization Headers (1) Body Pre-request Script Tests Code

Type No Auth

Status: 200 OK Time: 267 ms

Body Cookies Headers (3) Tests

Pretty Raw Preview JSON

1

Server local

javabrains.io

me/b LExc lExc 2ddl Mana ping ping ping ping ping ping ping JrlH JrlH JrlH eanE edSe iDat calh cher cher rFac

Runner Import +

Builder Team Library OFFLINE Sign In

Filter History Collections

Save to collection

Today

POST http://localhost:8080/topics

GET http://localhost:8080/topics

**POST http://localhost:8080/topics**

POST http://localhost:8080/topics X

No Environment

POST http://localhost:8080/topics

http://localhost:8080/ +

Params Send Save

Authorization Headers (1) Body Pre-request Script Tests Code

form-data x-www-form-urlencoded raw binary JSON (application/json)

```
1 {  
2   "id": "java",  
3   "name": "Java",  
4   "description": "Java Description"  
5 }
```

Body Cookies Headers (2) Tests Status: 200 OK Time: 114 ms

Pretty Raw Preview Text

Type local

me/b LExc LExc 2ddl Mana ping ping ping ping ping ping ping JrlH JrlH SeanE edSe iDat calh cher cher rFac

Runner Import +

Builder Team Library OFFLINE Sign In

Filter History Collections

Save to collection

Today

POST http://localhost:8080/topics

POST http://localhost:8080/topics

GET http://localhost:8080/topics

POST http://localhost:8080/topics

POST http://localhost:8080/topics

POST http://localhost:8080/topics

Server Type local

http://localhost:8080/ + No Environment

POST http://localhost:8080/topics Params Send Save

Authorization Headers (1) Body Pre-request Script Tests Code

form-data x-www-form-urlencoded raw binary JSON (application/json)

```
1 {  
2   "id": "javascript",  
3   "name": "JavaScript",  
4   "description": "JavaScript Description"  
5 }
```

Body Cookies Headers (2) Tests Status: 200 OK Time: 11 ms

Pretty Raw Preview Text

javabrains.io

Builder Team Library OFFLINE Sign In Filter History Collections Save to collection Today GET http://localhost:8080/topics Params Send Save Authorization Headers (1) Body Pre-request Script Tests Code

Type No Auth

Status: 200 OK Time: 27 ms

Body Cookies Headers (3) Tests

Pretty Raw Preview JSON

```
1 - []
2 - [
3 - {
4 -   "id": "java",
5 -   "name": "Java",
6 -   "description": "Java Description"
7 - },
8 - {
9 -   "id": "javascript",
10 -  "name": "JavaScript",
11 -  "description": "JavaScript Description"
12 - }
```

Builder Team Library OFFLINE Sign In

Runner Import +

Filter History Collections

Save to collection

Today

GET http://localhost:8080/topics/java  
GET http://localhost:8080/topics  
POST http://localhost:8080/topics  
POST http://localhost:8080/topics  
POST http://localhost:8080/topics  
POST http://localhost:8080/topics

http://localhost:8080/ + No Environment

GET http://localhost:8080/topics/java Params Send Save

Authorization Headers (1) Body Pre-request Script Tests Code

Type No Auth

Body Cookies Headers (3) Tests Status: 200 OK Time: 23 ms

Pretty Raw Preview JSON ↻

```
1 - {  
2   "id": "java",  
3   "name": "Java",  
4   "description": "Java Description"  
5 }
```

Builder Team Library OFFLINE Sign In

Filter History Collections Save to collection

PUT http://localhost:8080/topics/java Params Send Save

Authorization Headers (1) Body Pre-request Script Tests Code

form-data x-www-form-urlencoded raw binary JSON (application/json)

```
1 {  
2   "id": "javascript",  
3   "name": "Updated JavaScript",  
4   "description": "Updated JavaScript Description"  
5 }
```

Status: 200 OK Time: 23 ms

Body Cookies Headers (3) Tests

Pretty Raw Preview JSON

Type local

Runner Import +

Builder Team Library OFFLINE Sign In

Filter History Collections Save to collection

No Environment

PUT http://localhost:8080/topics/java Params Send Save

Authorization Headers (1) Body Pre-request Script Tests Code

form-data x-www-form-urlencoded raw binary JSON (application/json)

```
1 * {  
2   "id": "java",  
3   "name": "Updated Java",  
4   "description": "Updated Java Description"  
5 }
```

Status: 200 OK Time: 23 ms

Body Cookies Headers (3) Tests

Pretty Raw Preview JSON

Type local

Builder Team Library OFFLINE Sign In Filter History Collections Save to collection Today GET http://localhost:8080/topics Authorization Headers (1) Body Pre-request Script Tests Code

Type No Auth

Body Cookies Headers (3) Tests Status: 200 OK Time: 13 ms

Pretty Raw Preview JSON

```
1 [  
2 {  
3   "id": "java",  
4   "name": "Updated Java",  
5   "description": "Updated Java Description"  
6 },  
7 {  
8   "id": "javascript",  
9   "name": "JavaScript",  
10  "description": "JavaScript Description"  
11 }  
12 ]
```

Builder Team Library OFFLINE Sign In

Runner Import

Filter History Collections

Save to collection

Today

GET http://localhost:8080/topics

PUT http://localhost:8080/topics/java

GET http://localhost:8080/topics/java

POST http://localhost:8080/topics

POST http://localhost:8080/topics

POST http://localhost:8080/topics

POST http://localhost:8080/topics

Runner Import

Builder Team Library OFFLINE Sign In

No Environment

DELETE http://localhost:8080/topics/javascript

Params Send Save

Authorization Headers (1) Body ● Pre-request Script Tests Code

Type No Auth

Body Cookies Headers (3) Tests Status: 200 OK Time: 13 ms

Pretty Raw Preview JSON

```
1 [ ]  
2 {  
3   "id": "java",  
4   "name": "Updated Java",  
5   "description": "Updated Java Description"  
6 },  
7 {  
8   "id": "javascript",  
9   "name": "JavaScript",  
10  "description": "JavaScript Description"  
11 }  
12 ]
```

Builder Team Library OFFLINE Sign In Filter History Collections Save to collection Today GET http://localhost:8080/topics Params Send Save Authorization Headers (1) Body Pre-request Script Tests Code

Type No Auth

Body Cookies Headers (3) Tests Status: 200 OK Time: 12 ms

Pretty Raw Preview JSON

```
[{"id": "java", "name": "Updated Java", "description": "Updated Java Description"}]
```

Package Explorer    TopicService.java    course-api-data/    Topic.java    TopicRepository.    CourseApiDataAp

course-api [boot]  
course-api-data [boot]  
src/main/java  
  io.javabrains  
    CourseApiDataApplication.java  
  io.javabrains.springbootstarter.topic  
    Topic.java  
    TopicController.java  
    TopicRepository.java  
    TopicService.java  
src/main/resources  
src/test/java  
JRE System Library [JavaSE-1.8]  
Maven Dependencies  
src  
  target  
    mvnw  
    mvnw.cmd  
pom.xml  
Servers

```
package io.javabrains.springbootstarter.topic;

import org.springframework.data.repository.CrudRepository;

public interface TopicRepository extends CrudRepository<Topic, String> {
```

Servers    Boot Dashboard

Type tags to match

local

Writable    Smart Insert    7 : 2

Package Explorer    TopicService.java    course-api-data/    Topic.java    TopicRepository.java    CourseApiDataAp

```
TopicService.java
package io.javabrains.springbootstarter.topic;

import java.util.ArrayList;

@Service
public class TopicService {

    @Autowired
    private TopicRepository topicRepository;

    public List<Topic> getAllTopics() {
        List<Topic> topics = new ArrayList<>();
        topicRepository.findAll()
            .forEach(topics::add);
        return topics;
    }

    public Topic getTopic(String id) {
        return topicRepository.findOne(id);
    }

    public void addTopic(Topic topic) {
        topicRepository.save(topic);
    }

    public void updateTopic(String id, Topic topic) {
        topicRepository.save(topic);
    }

    public void deleteTopic(String id) {
        topicRepository.delete(id);
    }
}
```

Servers    Boot Dashboard

Type tags to match

local

Writable    Smart Insert    39 : 31

Package Explorer    TopicService.java    course-api-data/    Topic.java    TopicRepository.java    CourseApiDataAp

```
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-parent</artifactId>
<version>1.4.2.RELEASE</version>
<relativePath/> <!-- lookup parent from repository -->
</parent>

<properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
    <java.version>1.8</java.version>
</properties>

<dependencies>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-data-jpa</artifactId>
    </dependency>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-web</artifactId>
    </dependency>

    <dependency>
        <groupId>org.apache.derby</groupId>
        <artifactId>derby</artifactId>
        <scope>runtime</scope>
    </dependency>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-test</artifactId>
        <scope>test</scope>
    </dependency>
</dependencies>

<build>
    <plugins>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
        </plugin>
    </plugins>
</build>
```

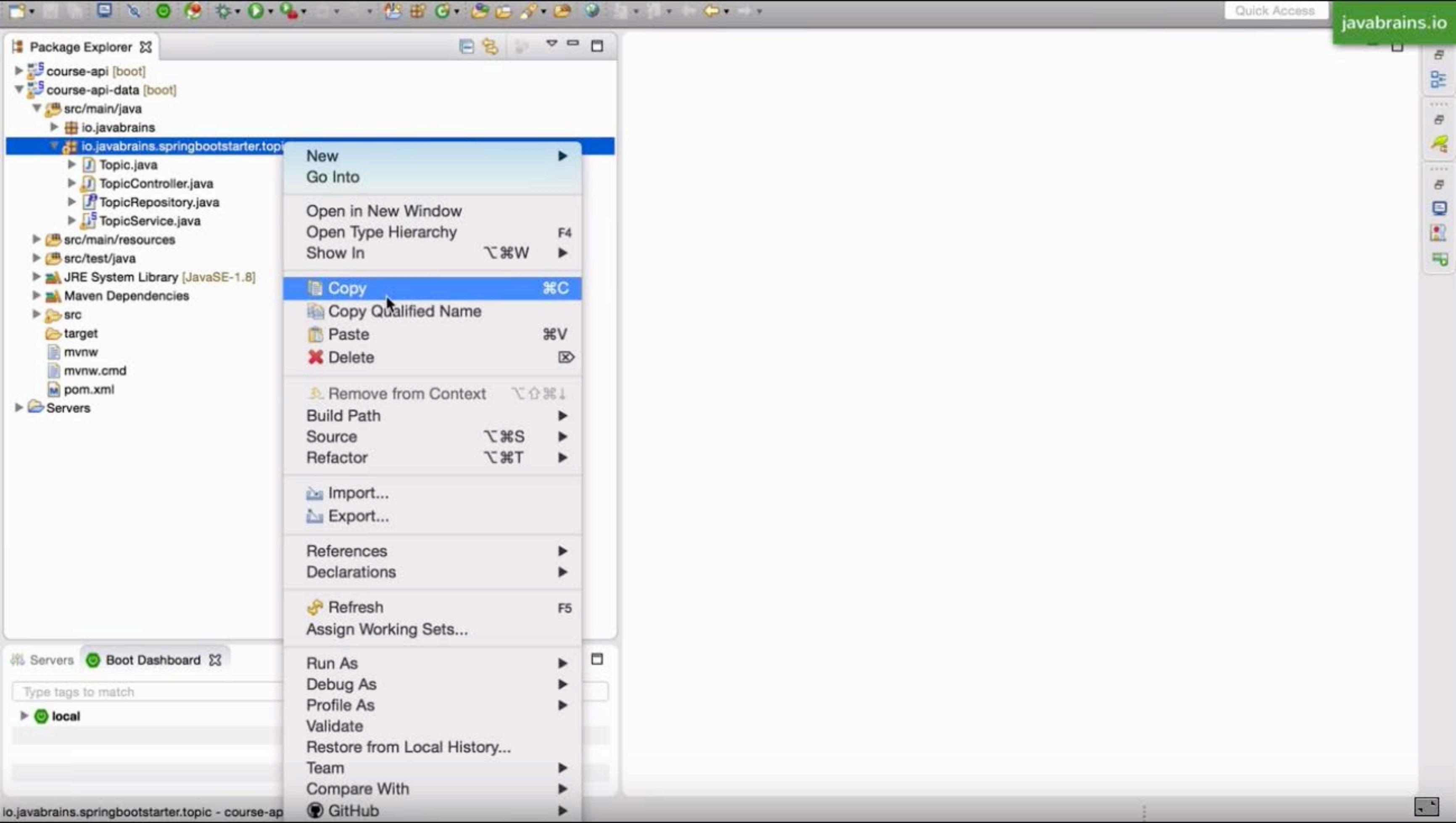
Servers    Boot Dashboard

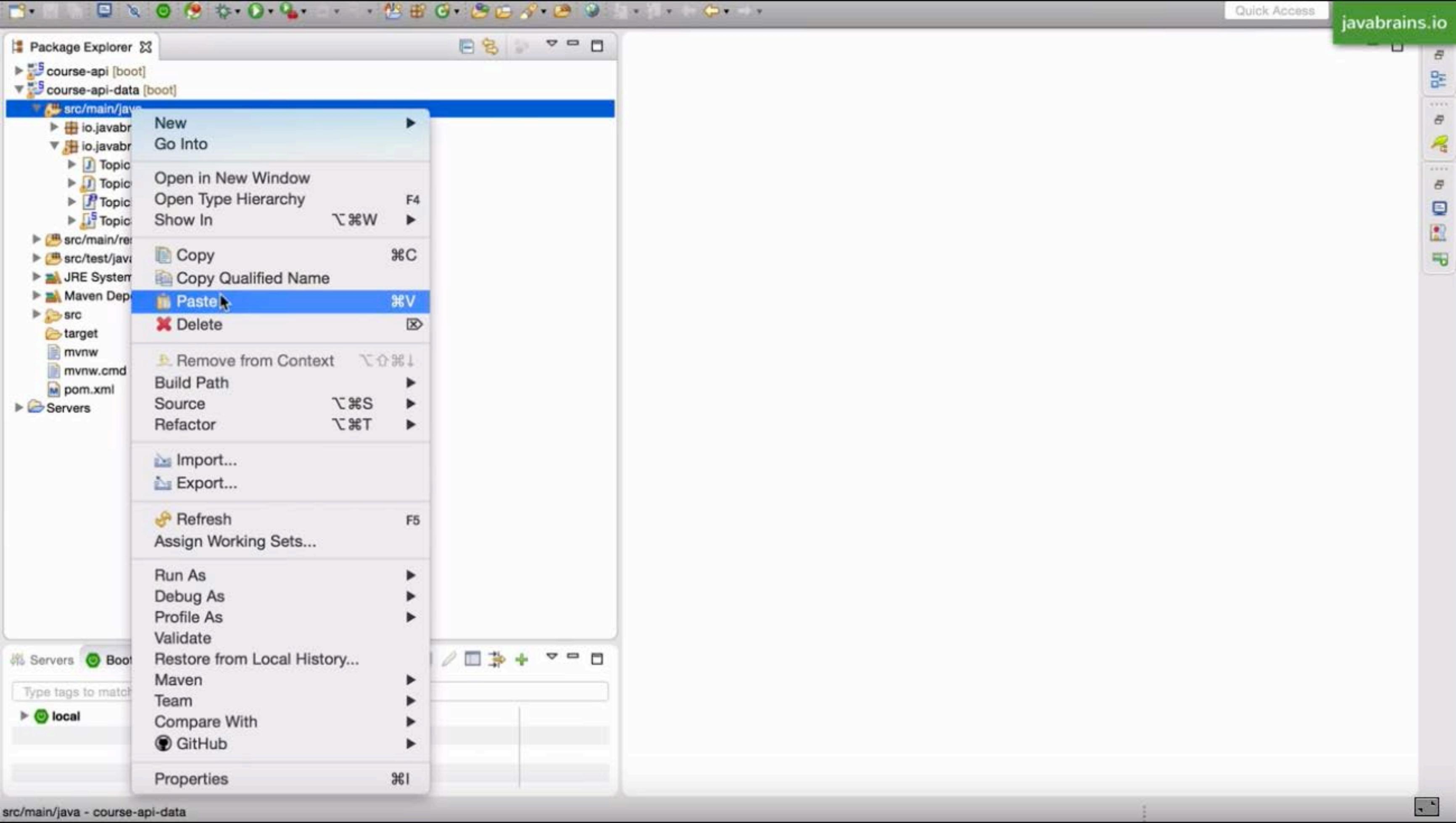
Type tags to match

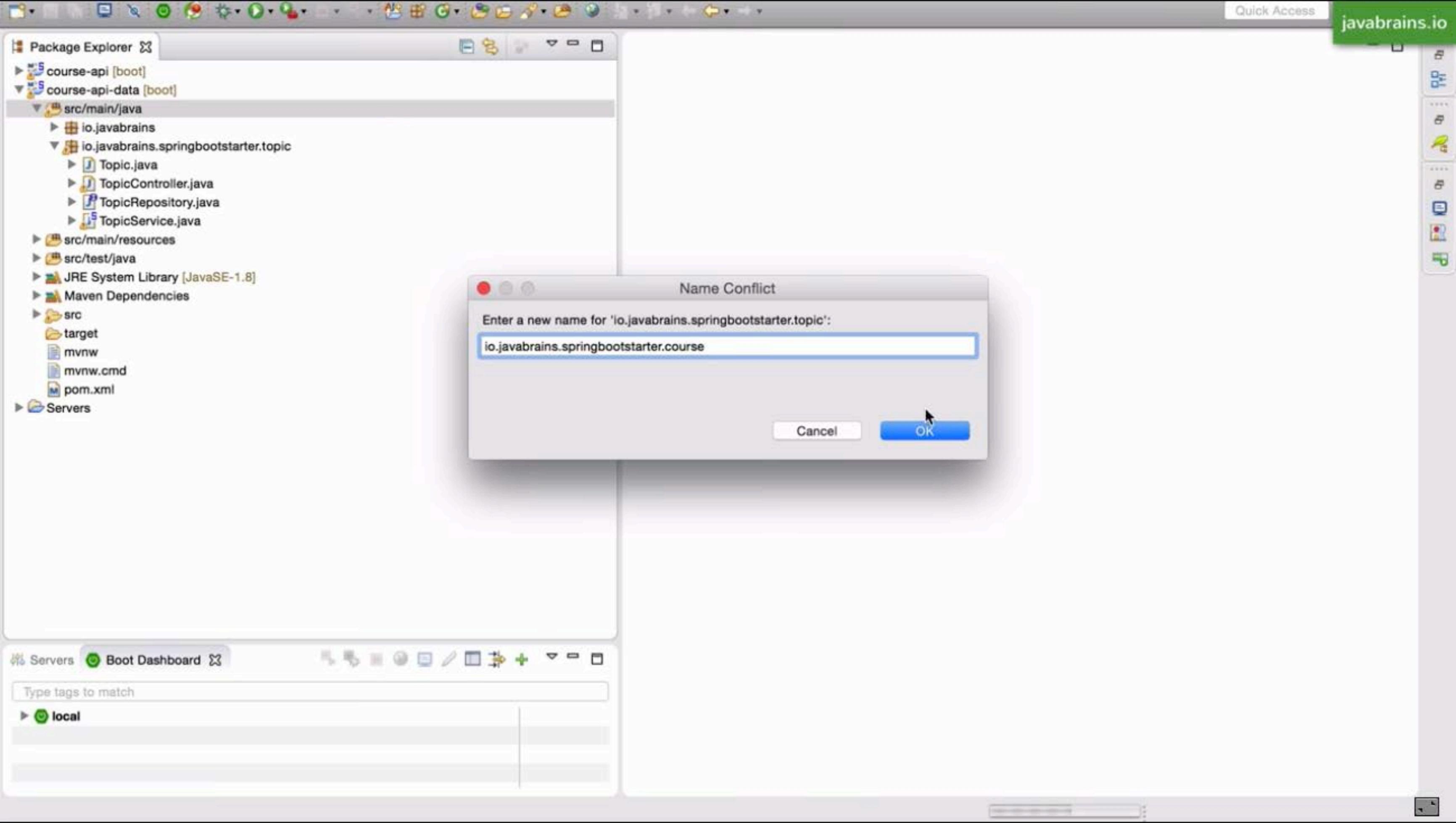
local

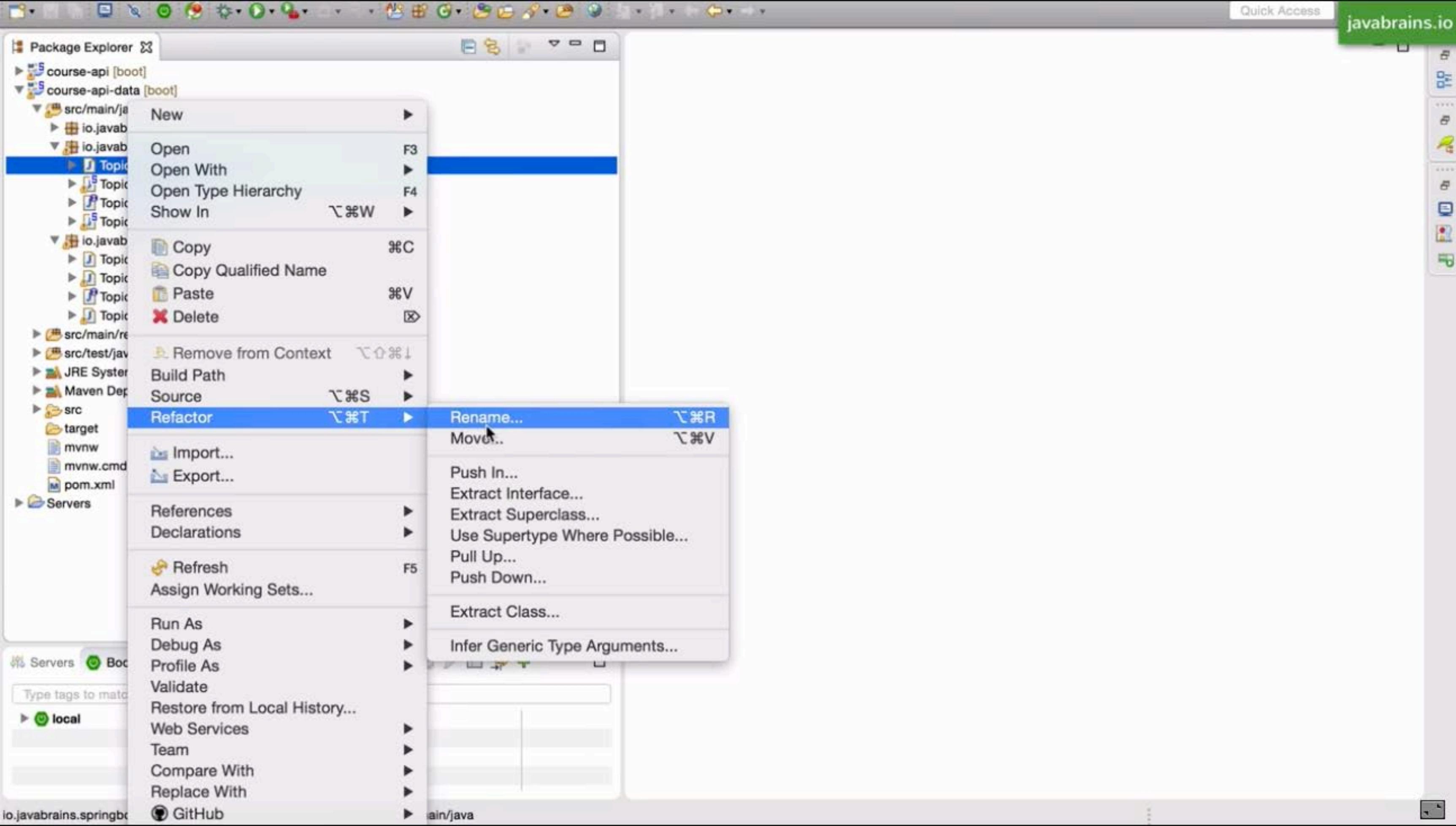
project/dependencies/dependency

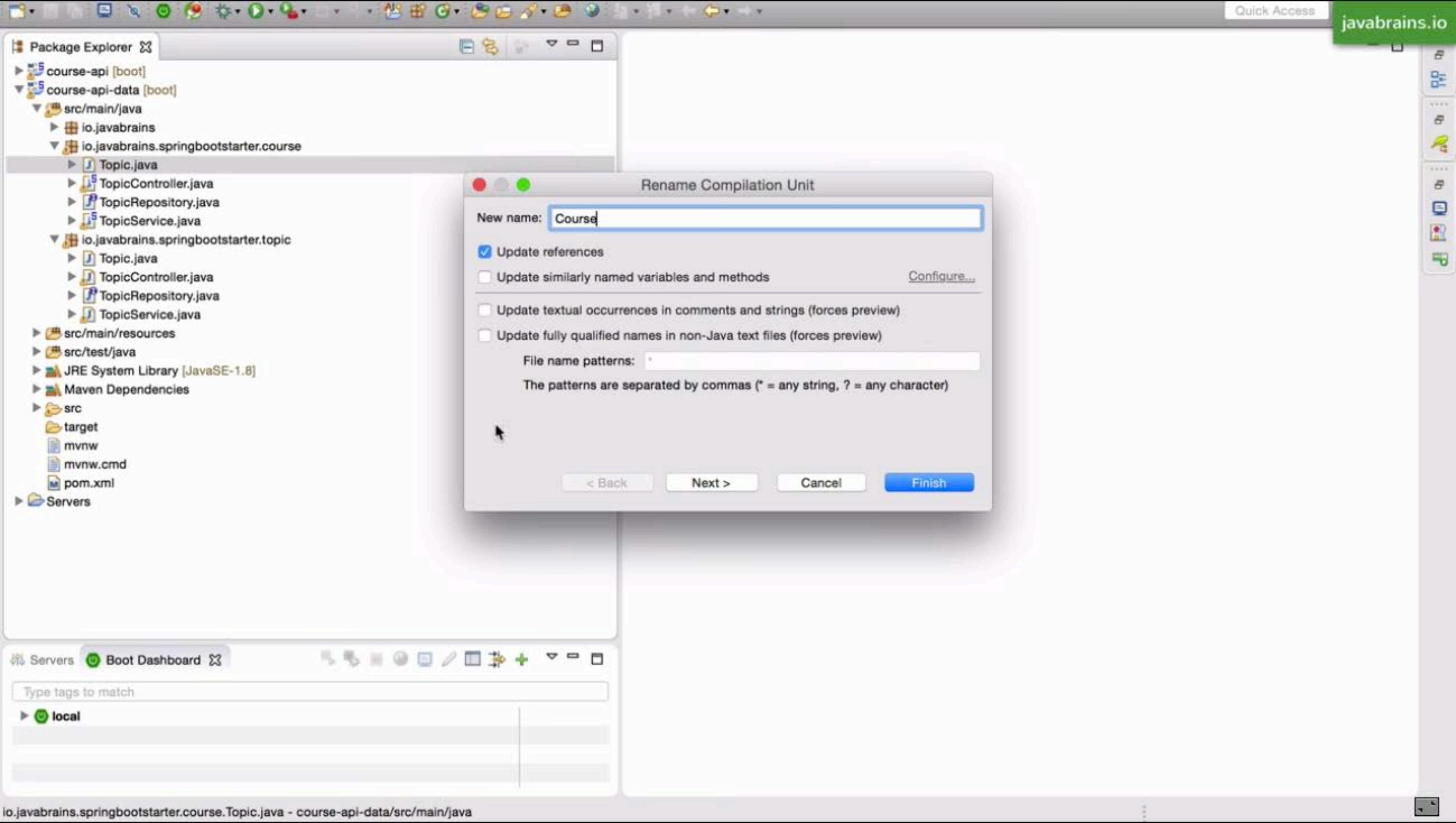
Writable    Smart Insert    41 : 22

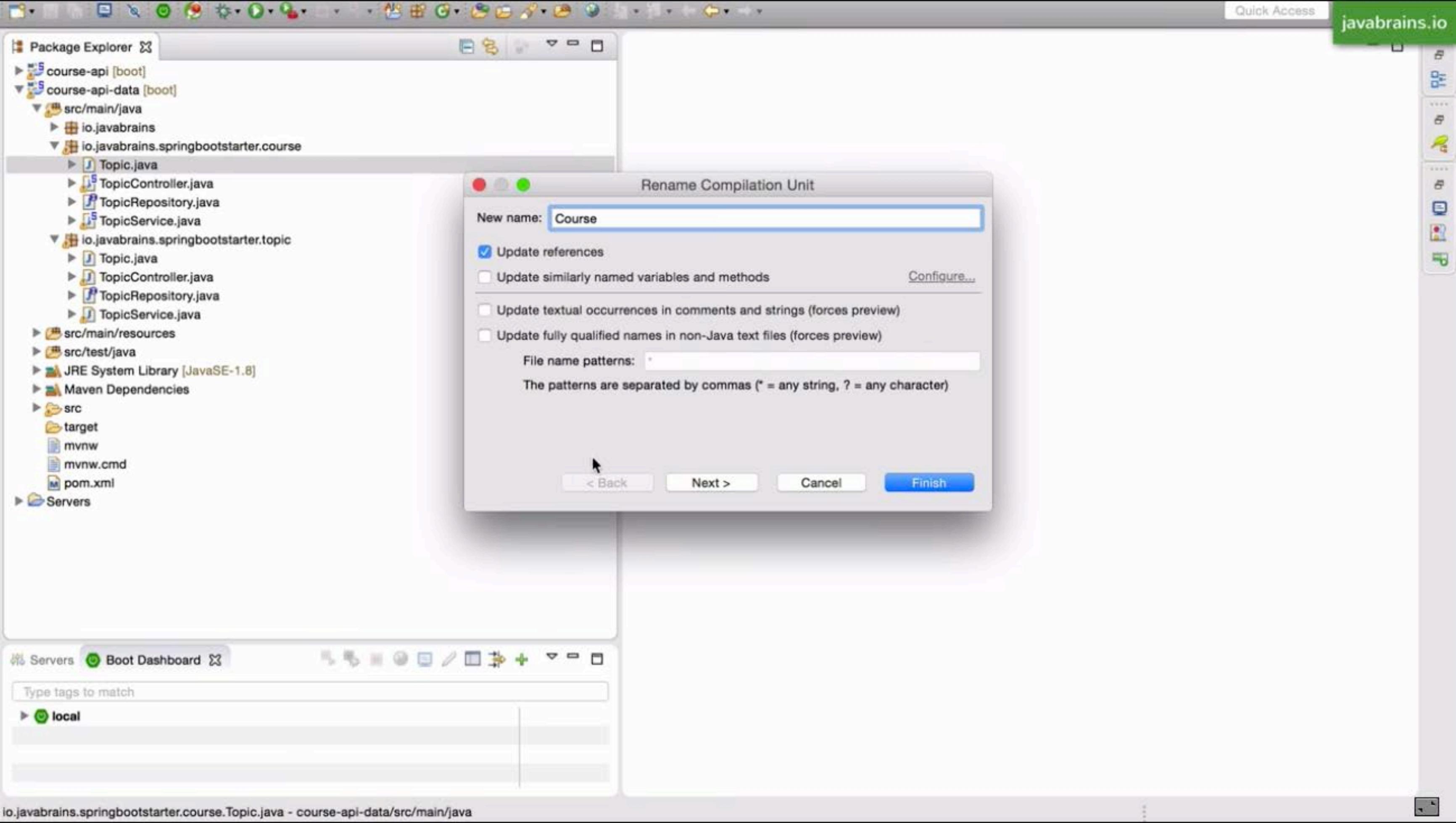












Package Explorer Course.java

```
course-api [boot]
course-api-data [boot]
src/main/java
  io.javabrains
  io.javabrains.springbootstarter.course
    Course.java
    CourseController.java
    CourseRepository.java
    CourseService.java
  io.javabrains.springbootstarter.topic
    Topic.java
    TopicController.java
    TopicRepository.java
    TopicService.java
src/main/resources
src/test/java
JRE System Library [JavaSE-1.8]
Maven Dependencies
src
  target
  mvnw
  mvnw.cmd
  pom.xml
Servers
```

Servers Boot Dashboard

Type tags to match

local

Course.java

```
package io.javabrains.springbootstarter.course;

import javax.persistence.Entity;

@Entity
public class Course {

    @Id
    private String id;
    private String name;
    private String description;

    public Course() {
    }

    public Course(String id, String name, String description) {
        super();
        this.id = id;
        this.name = name;
        this.description = description;
    }

    public String getId() {
        return id;
    }
    public void setId(String id) {
        this.id = id;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getDescription() {
        return description;
    }
    public void setDescription(String description) {
        this.description = description;
    }
}
```

Writable Smart Insert 13 : 5

Package Explorer Course.java CourseRepository.java CourseService.java

```
package io.javabrains.springbootstarter.course;

import java.util.ArrayList;

@Service
public class CourseService {

    @Autowired
    private CourseRepository courseRepository;

    public List<Course> getAllTopics() {
        List<Course> topics = new ArrayList<>();
        courseRepository.findAll()
            .forEach(topics::add);
        return topics;
    }

    public Course getTopic(String id) {
        return courseRepository.findOne(id);
    }

    public void addTopic(Course topic) {
        courseRepository.save(topic);
    }

    public void updateTopic(String id, Course topic) {
        courseRepository.save(topic);
    }

    public void deleteTopic(String id) {
        courseRepository.delete(id);
    }
}
```

Servers Boot Dashboard

Type tags to match

local

Writable Smart Insert 17 : 37

Package Explorer Course.java CourseRepository.java CourseService.java

```
package io.javabrains.springbootstarter.course;

import java.util.ArrayList;

@Service
public class CourseService {

    @Autowired
    private CourseRepository courseRepository;

    public List<Course> getAllCourses() {
        List<Course> courses = new ArrayList<>();
        courseRepository.findAll()
            .forEach(courses::add);
        return courses;
    }

    public Course getCourse(String id) {
        return courseRepository.findOne(id);
    }

    public void addCourse(Course topic) {
        courseRepository.save(topic);
    }

    public void updateCourse(String id, Course course) {
        courseRepository.save(course);
    }

    public void deleteCourse(String id) {
        courseRepository.delete(id);
    }
}
```

Servers Boot Dashboard

Type tags to match

local

Writable Smart Insert 37 : 1

Package Explorer Course.java CourseRepository.java CourseService.java CourseController.java

```
import javax.persistence.Entity;

@Entity
public class Course {

    @Id
    private String id;
    private String name;
    private String description;

    private Topic topic;

    public Course() {
    }

    public Course(String id, String name, String description, String topicId) {
        super();
        this.id = id;
        this.name = name;
        this.description = description;
        this.topic = new Topic(topicId, "", "");
    }

    public String getId() {
        return id;
    }
    public void setId(String id) {
        this.id = id;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getDescription() {
        return description;
    }
    public void setDescription(String description) {
        this.description = description;
    }
}
```

Servers Boot Dashboard

Type tags to match

local

Writable Smart Insert 26 : 23

Package Explorer Course.java CourseRepository.java CourseService.java CourseController.java

```
package io.javabrains.springbootstarter.course;

import java.util.Arrays;

@RestController
public class CourseController {

    @Autowired
    private CourseService courseService;

    @RequestMapping("/topics/{id}/courses")
    public List<Course> getAllCourses(@PathVariable String id) {
        return courseService.getAllCourses(id);
    }

    @RequestMapping("/topics/{topicId}/courses/{id}")
    public Course getCourse(@PathVariable String id) {
        return courseService.getCourse(id);
    }

    @RequestMapping(method=RequestMethod.POST, value="/topics/{topicId}/courses")
    public void addCourse(@RequestBody Course course, @PathVariable String topicId) {
        course.setTopic(new Topic(topicId, "", ""));
        courseService.addCourse(course);
    }

    @RequestMapping(method=RequestMethod.PUT, value="/topics/{topicId}/courses/{id}")
    public void updateCourse(@RequestBody Course course, @PathVariable String topicId, @PathVariable String id) {
        course.setTopic(new Topic(topicId, "", ""));
        courseService.updateCourse(course);
    }

    @RequestMapping(method=RequestMethod.DELETE, value="/topics/{topicId}/courses/{id}")
    public void deleteTopic(@PathVariable String id) {
        courseService.deleteCourse(id);
    }
}
```

Servers Boot Dashboard

Type tags to match

local

Writable Smart Insert 30 : 5

Package Explorer Course.java CourseRepository.java CourseService.java CourseController.java

```
package io.javabrains.springbootstarter.course;

import java.util.Arrays;

@RestController
public class CourseController {

    @Autowired
    private CourseService courseService;

    @RequestMapping("/topics/{id}/courses")
    public List<Course> getAllCourses(@PathVariable String id) {
        return courseService.getAllCourses(id);
    }

    @RequestMapping("/topics/{topicId}/courses/{id}")
    public Course getCourse(@PathVariable String id) {
        return courseService.getCourse(id);
    }

    @RequestMapping(method=RequestMethod.POST, value="/topics/{topicId}/courses")
    public void addCourse(@RequestBody Course course, @PathVariable String topicId) {
        course.setTopic(new Topic(topicId, "", ""));
        courseService.addCourse(course);
    }

    @RequestMapping(method=RequestMethod.PUT, value="/topics/{topicId}/courses/{id}")
    public void updateCourse(@RequestBody Course course, @PathVariable String topicId, @PathVariable String id) {
        course.setTopic(new Topic(topicId, "", ""));
        courseService.updateCourse(course);
    }

    @RequestMapping(method=RequestMethod.DELETE, value="/topics/{topicId}/courses/{id}")
    public void deleteTopic(@PathVariable String id) {
        courseService.deleteCourse(id);
    }
}
```

Servers Boot Dashboard

Type tags to match

local

Writable Smart Insert 46 : 6

Package Explorer Course.java CourseRepository.java CourseService.java CourseController.java

```
package io.javabrains.springbootstarter.course;

import java.util.Arrays;

@RestController
public class CourseController {

    @Autowired
    private CourseService courseService;

    @RequestMapping("/topics/{id}/courses")
    public List<Course> getAllCourses(@PathVariable String id) {
        return courseService.getAllCourses(id);
    }

    @RequestMapping("/topics/{topicId}/courses/{id}")
    public Course getCourse(@PathVariable String id) {
        return courseService.getCourse(id);
    }

    @RequestMapping(method=RequestMethod.POST, value="/topics/{topicId}/courses")
    public void addCourse(@RequestBody Course course, @PathVariable String topicId) {
        course.setTopic(new Topic(topicId, "", ""));
        courseService.addCourse(course);
    }

    @RequestMapping(method=RequestMethod.PUT, value="/topics/{topicId}/courses/{id}")
    public void updateCourse(@RequestBody Course course, @PathVariable String topicId, @PathVariable String id) {
        course.setTopic(new Topic(topicId, "", ""));
        courseService.updateCourse(course);
    }

    @RequestMapping(method=RequestMethod.DELETE, value="/topics/{topicId}/courses/{id}")
    public void deleteTopic(@PathVariable String id) {
        courseService.deleteCourse(id);
    }
}
```

Servers Boot Dashboard

Type tags to match

local

The method getAllCourses() in the type CourseService is not applicable for the arguments (String)

Writable Smart Insert 25 : 5

Package Explorer Course.java CourseRepository.java CourseService.java CourseController.java

```
+import javax.persistence.Entity;||

@Entity
public class Course {

    @Id
    private String id;
    private String name;
    private String description;

    private Topic topic;

    public Course() {
    }

    public Course(String id, String name, String description, String topicId) {
        super();
        this.id = id;
        this.name = name;
        this.description = description;
        this.topic = new Topic(topicId, "", "");
    }

    public String getId() {
        return id;
    }
    public void setId(String id) {
        this.id = id;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getDescription() {
        return description;
    }
    public void setDescription(String description) {
        this.description = description;
    }
}
```

Servers Boot Dashboard

Type tags to match

local

Writable Smart Insert 16 : 5

Package Explorer Course.java CourseRepository.java CourseService.java CourseController.java Topic.java

```
+import javax.persistence.Entity;■
@Entity
public class Course {
    @Id
    private String id;
    private String name;
    private String description;
    @ManyToOne
    private Topic topic;

    public Course() {
    }

    public Course(String id, String name, String description, String topicId) {
        super();
        this.id = id;
        this.name = name;
        this.description = description;
        this.topic = new Topic(topicId, "", "");
    }

    public String getId() {
        return id;
    }
    public void setId(String id) {
        this.id = id;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getDescription() {
        return description;
    }
    public void setDescription(String description) {
        this.description = description;
    }
}
```

Servers Boot Dashboard

Type tags to match

local

Package Explorer Course.java CourseRepository.java CourseService.java CourseController.java Topic.java

```
+import javax.persistence.Entity;■
@Entity
public class Course {
    @Id
    private String id;
    private String name;
    private String description;
    @ManyToOne
    P ManyToOne cannot be resolved to a type
    P 5 quick fixes available:
    } ← Import 'ManyToOne' (javax.persistence)
    @ Create annotation 'ManyToOne'
    P ↗ Change to 'ManyToMany' (org.hibernate.annotations)
    ↗ Change to 'ManyToMany' (javax.persistence)
    ↗ Fix project setup...
    description, String topicId) {
    this.topic = new Topic(topicId, "", "");
}

    public String getId() {
        return id;
    }
    public void setId(String id) {
        this.id = id;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getDescription() {
        return description;
    }
    public void setDescription(String description) {
        this.description = description;
    }
}
```

Servers Boot Dashboard

Type tags to match

local

Writable Smart Insert 15 : 15

Package Explorer Course.java CourseRepository.java CourseService.java CourseController.java Topic.java

```
package io.javabrains.springbootstarter.course;

import java.util.ArrayList;

@Service
public class CourseService {

    @Autowired
    private CourseRepository courseRepository;

    public List<Course> getAllCourses() {
        List<Course> courses = new ArrayList<>();
        courseRepository.findAll()
            .forEach(courses::add);
        return courses;
    }

    public Course getCourse(String id) {
        return courseRepository.findOne(id);
    }

    public void addCourse(Course course) {
        courseRepository.save(course);
    }

    public void updateCourse(Course course) {
        courseRepository.save(course);
    }

    public void deleteCourse(String id) {
        courseRepository.delete(id);
    }
}
```

src/main/java  
io.javabrains  
io.javabrains.springbootstarter.course  
Course.java  
CourseController.java  
CourseRepository.java  
CourseService.java  
io.javabrains.springbootstarter.topic  
Topic.java  
TopicController.java  
TopicRepository.java  
TopicService.java

src/main/resources

src/test/java

JRE System Library [JavaSE-1.8]

Maven Dependencies

src  
target  
mvnw  
mvnw.cmd  
pom.xml

Servers

Servers Boot Dashboard

Type tags to match

local

Writable Smart Insert 19 : 25

Package Explorer Course.java CourseRepository.java \*CourseService.java \*CourseController.java Topic.java

```
package io.javabrains.springbootstarter.course;

import java.util.List;

public interface CourseRepository extends CrudRepository<Course, String> {

    public List<Course> getCourseByTopic(String topicId);
}
```

course-api [boot] course-api-data [boot]

src/main/java

- io.javabrains
- io.javabrains.springbootstarter.course
  - Course.java
  - CourseController.java
  - CourseRepository.java
  - CourseService.java
- io.javabrains.springbootstarter.topic
  - Topic.java
  - TopicController.java
  - TopicRepository.java
  - TopicService.java

src/main/resources

src/test/java

JRE System Library [JavaSE-1.8]

Maven Dependencies

src

- target
- mvnw
- mvnw.cmd
- pom.xml

Servers

Servers Boot Dashboard

Type tags to match

local

Writable Smart Insert 9 : 42

Package Explorer Course.java \*CourseRepository.java \*CourseService.java \*CourseController.java Topic.java

```
package io.javabrains.springbootstarter.course;

import java.util.List;

public interface CourseRepository extends CrudRepository<Course, String> {
    public List<Course> findByTopicId(String topicId);
}
```

course-api [boot] course-api-data [boot]

src/main/java

- io.javabrains
- io.javabrains.springbootstarter.course
  - Course.java
  - CourseController.java
  - CourseRepository.java
  - CourseService.java
- io.javabrains.springbootstarter.topic
  - Topic.java
  - TopicController.java
  - TopicRepository.java
  - TopicService.java

src/main/resources

src/test/java

JRE System Library [JavaSE-1.8]

Maven Dependencies

src

- target
- mvnw
- mvnw.cmd
- pom.xml

Servers

Servers Boot Dashboard

Type tags to match

local

Writable Smart Insert 9 : 25

Package Explorer Course.java \*CourseRepository.java \*CourseService.java \*CourseController.java Topic.java

```
package io.javabrains.springbootstarter.course;

import java.util.List;

public interface CourseRepository extends CrudRepository<Course, String> {
    public List<Course> getCourseByName(String name);
}
```

course-api [boot] course-api-data [boot]

src/main/java

- io.javabrains
- io.javabrains.springbootstarter.course
- Course.java
- CourseController.java
- CourseRepository.java
- CourseService.java

io.javabrains.springbootstarter.topic

- Topic.java
- TopicController.java
- TopicRepository.java
- TopicService.java

src/main/resources

src/test/java

JRE System Library [JavaSE-1.8]

Maven Dependencies

src

- target
- mvnw
- mvnw.cmd

pom.xml

Servers

Servers Boot Dashboard

Type tags to match

local

Writable Smart Insert 11 : 1

Package Explorer Course.java \*CourseRepository.java \*CourseService.java \*CourseController.java Topic.java

```
package io.javabrains.springbootstarter.course;
import java.util.List;
public interface CourseRepository extends CrudRepository<Course, String> {
    public List<Course> findByName(String name);
}
```

course-api [boot] course-api-data [boot]

src/main/java

- io.javabrains
- io.javabrains.springbootstarter.course
- Course.java
- CourseController.java
- CourseRepository.java
- CourseService.java

io.javabrains.springbootstarter.topic

- Topic.java
- TopicController.java
- TopicRepository.java
- TopicService.java

src/main/resources

src/test/java

JRE System Library [JavaSE-1.8]

Maven Dependencies

src

- target
- mvnw
- mvnw.cmd
- pom.xml

Servers

Servers Boot Dashboard

Type tags to match

local

Writable Smart Insert 9 : 7

The screenshot shows the Eclipse IDE interface with the following details:

- Package Explorer:** Shows the project structure with packages like `course-api [boot]`, `course-api-data [boot]`, and `io.javabrains.springbootstarter.course` and `io.javabrains.springbootstarter.topic`.
- Editor:** Displays the `CourseService.java` file content. The code defines a service class `CourseService` that interacts with a `CourseRepository` to perform operations on `Course` objects.

```
package io.javabrains.springbootstarter.course;
import java.util.ArrayList;
@Service
public class CourseService {
    @Autowired
    private CourseRepository courseRepository;
    public List<Course> getAllCourses() {
        List<Course> courses = new ArrayList<>();
        courseRepository.
        .forEach(courses::add);
        return courses;
    }
    public Course getCourse(String id) {
        courseReposi|
        return courseRepository.findOne(id);
    }
    public void addCourse(Course course) {
        courseRepository.save(course);
    }
    public void updateCourse(Course course) {
        courseRepository.save(course);
    }
    public void deleteCourse(String id) {
        courseRepository.delete(id);
    }
}
```




You don't have to implement the method! Just declare the method with the `findByProperty` name format, and Spring Data JPA will implement the method for you!

Package Explorer Course.java \*CourseRepository.java \*CourseService.java \*CourseController.java Topic.java

```
package io.javabrains.springbootstarter.course;

import java.util.List;

public interface CourseRepository extends CrudRepository<Course, String> {

    public List<Course> findByName(String name);
    public List<Course> findByDescription(String name);
}
```

List<Course> io.javabrains.springbootstarter.course.CourseRepository.findByDescription(String name)

Press 'F2' for focus

Course.java \*CourseRepository.java \*CourseService.java \*CourseController.java Topic.java

course-api [boot] course-api-data [boot]

src/main/java io.javabrains io.javabrains.springbootstarter.course Course.java CourseController.java CourseRepository.java CourseService.java

io.javabrains.springbootstarter.topic Topic.java TopicController.java TopicRepository.java TopicService.java

src/main/resources src/test/java JRE System Library [JavaSE-1.8] Maven Dependencies

src target mvnw mvnw.cmd pom.xml

Servers

Servers Boot Dashboard

Type tags to match

local

Writable Smart Insert 10 : 31

Package Explorer Course.java \*CourseRepository.java \*CourseService.java \*CourseController.java Topic.java

```
package io.javabrains.springbootstarter.course;

import java.util.List;

public interface CourseRepository extends CrudRepository<Course, String> {

    public List<Course> findByName(String name);
    public List<Course> findByDescription(String foo);
}
```

src/main/java  
io.javabrains.springbootstarter.course  
Course.java  
CourseController.java  
CourseRepository.java  
CourseService.java  
io.javabrains.springbootstarter.topic  
Topic.java  
TopicController.java  
TopicRepository.java  
TopicService.java  
src/main/resources  
src/test/java  
JRE System Library [JavaSE-1.8]  
Maven Dependencies  
src  
target  
mvnw  
mvnw.cmd  
pom.xml  
Servers

Servers Boot Dashboard

Type tags to match

local

Writable Smart Insert 12 : 1

Package Explorer Course.java \*CourseRepository.java \*CourseService.java \*CourseController.java Topic.java

```
package io.javabrains.springbootstarter.course;
}
import java.util.List;

public interface CourseRepository extends CrudRepository<Course, String> {
    public List<Course> findByTopic(String topicId);
}
```

course-api [boot] course-api-data [boot]

src/main/java

- io.javabrains
- io.javabrains.springbootstarter.course
- Course.java
- CourseController.java
- CourseRepository.java
- CourseService.java

io.javabrains.springbootstarter.topic

- Topic.java
- TopicController.java
- TopicRepository.java
- TopicService.java

src/main/resources

src/test/java

JRE System Library [JavaSE-1.8]

Maven Dependencies

src

- target
- mvnw
- mvnw.cmd
- pom.xml

Servers

Servers Boot Dashboard

Type tags to match

local

Writable Smart Insert 9 : 50

Package Explorer Course.java \*CourseRepository.java \*CourseService.java \*CourseController.java Topic.java

```
+import javax.persistence.Entity; 

@Entity
public class Course {

    @Id
    private String id;
    private String name;
    private String description;
    @ManyToOne
    private Topic topic;

    public Course() {
    }

    public Course(String id, String name, String description, String topicId) {
        super();
        this.id = id;
        this.name = name;
        this.description = description;
        this.topic = new Topic(topicId, "", "");
    }

    public String getId() {
        return id;
    }
    public void setId(String id) {
        this.id = id;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getDescription() {
        return description;
    }
    public void setDescription(String description) {
        this.description = description;
    }
}
```

Servers Boot Dashboard

Type tags to match

local

The screenshot shows the Eclipse IDE interface with the following details:

- Package Explorer:** Shows the project structure under "course-api [boot]".
  - src/main/java:** Contains packages `io.javabrains` and `io.javabrains.springbootstarter.course`.
    - `Course.java`
    - `CourseController.java`
    - `CourseRepository.java`
    - `CourseService.java`
  - `io.javabrains.springbootstarter.topic` package contains:
    - `Topic.java`
    - `TopicController.java`
    - `TopicRepository.java`
    - `TopicService.java`
  - `src/main/resources`
  - `src/test/java`
  - `JRE System Library [JavaSE-1.8]`
  - `Maven Dependencies`
  - `src`, `target`, `mvnw`, `mvnw.cmd`, and `pom.xml`
  - `Servers`
- Editor:** Displays the `Course.java` file content.

```
package io.javabrains.springbootstarter.course;  
import java.util.List;  
  
public interface CourseRepository extends CrudRepository<Course, String> {  
    public List<Course> findByTopicId(String topicId);  
}
```

Make sure you always write the method name in camel-case!

Package Explorer Course.java CourseRepository.java \*CourseService.java \*CourseController.java Topic.java

```
package io.javabrains.springbootstarter.course;

import java.util.ArrayList;

@Service
public class CourseService {

    @Autowired
    private CourseRepository courseRepository;

    public List<Course> getAllCourses(String topicId) {
        List<Course> courses = new ArrayList<>();
        courseRepository.findByTopicId(topicId)
            .forEach(courses::add);
        return courses;
    }

    public Course getCourse(String id) {
        return courseRepository.findOne(id);
    }

    public void addCourse(Course course) {
        courseRepository.save(course);
    }

    public void updateCourse(Course course) {
        courseRepository.save(course);
    }

    public void deleteCourse(String id) {
        courseRepository.delete(id);
    }
}
```

Servers Boot Dashboard

Type tags to match

local

Writable Smart Insert 19 : 47

Package Explorer Course.java CourseRepository.java \*CourseService.java \*CourseController.java Topic.java

```
package io.javabrains.springbootstarter.course;
import java.util.List;
public interface CourseRepository extends CrudRepository<Course, String> {
    public List<Course> findByTopicId(String topicId);
}
```

course-api [boot] course-api-data [boot]

src/main/java

- io.javabrains
- io.javabrains.springbootstarter.course
- Course.java
- CourseController.java
- CourseRepository.java
- CourseService.java

io.javabrains.springbootstarter.topic

- Topic.java
- TopicController.java
- TopicRepository.java
- TopicService.java

src/main/resources

src/test/java

JRE System Library [JavaSE-1.8]

Maven Dependencies

src

- target
- mvnw
- mvnw.cmd
- pom.xml

Servers

Servers Boot Dashboard

Type tags to match

local

Writable Smart Insert 9 : 5

Package Explorer Course.java CourseRepository.java CourseService.java CourseController.java Topic.java

```
package io.javabrains.springbootstarter.course;

import java.util.Arrays;

@RestController
public class CourseController {

    @Autowired
    private CourseService courseService;

    @RequestMapping("/topics/{id}/courses")
    public List<Course> getAllCourses(@PathVariable String id) {
        return courseService.getAllCourses(id);
    }

    @RequestMapping("/topics/{topicId}/courses/{id}")
    public Course getCourse(@PathVariable String id) {
        return courseService.getCourse(id);
    }

    @RequestMapping(method=RequestMethod.POST, value="/topics/{topicId}/courses")
    public void addCourse(@RequestBody Course course, @PathVariable String topicId) {
        course.setTopic(new Topic(topicId, "", ""));
        courseService.addCourse(course);
    }

    @RequestMapping(method=RequestMethod.PUT, value="/topics/{topicId}/courses/{id}")
    public void updateCourse(@RequestBody Course course, @PathVariable String topicId, @PathVariable String id) {
        course.setTopic(new Topic(topicId, "", ""));
        courseService.updateCourse(course);
    }

    @RequestMapping(method=RequestMethod.DELETE, value="/topics/{topicId}/courses/{id}")
    public void deleteTopic(@PathVariable String id) {
        courseService.deleteCourse(id);
    }
}
```

src/main/java  
io.javabrains  
io.javabrains.springbootstarter.course  
Course.java  
CourseController.java  
CourseRepository.java  
CourseService.java  
io.javabrains.springbootstarter.topic  
Topic.java  
TopicController.java  
TopicRepository.java  
TopicService.java  
src/main/resources  
src/test/java  
JRE System Library [JavaSE-1.8]  
Maven Dependencies  
src  
target  
mvnw  
mvnw.cmd  
pom.xml  
Servers

Servers Boot Dashboard

Type tags to match

local

Writable Smart Insert 14 : 1

CourseApiDataApplication

2 demo (1)  
3 demo  
4 Copy of course-api (1)  
5 Copy of course-api  
6 course-api - CourseApiApp  
7 CourseApiApp

Run As  
Run Configurations...  
Organize Favorites...

```
io.javabrains.springbootstarter.course;  
java.util.Arrays;  
  
controller  
class CourseController {  
  
    towired  
    @Value CourseService courseService;  
  
    @GetMapping("/topics/{id}/courses")  
    public List<Course> getAllCourses(@PathVariable String id) {  
        return courseService.getAllCourses(id);  
    }  
  
    @RequestMapping("/topics/{topicId}/courses/{id}")  
    public Course getCourse(@PathVariable String id) {  
        return courseService.getCourse(id);  
    }  
  
    @RequestMapping(method=RequestMethod.POST, value="/topics/{topicId}/courses")  
    public void addCourse(@RequestBody Course course, @PathVariable String topicId) {  
        course.setTopic(new Topic(topicId, "", ""));  
        courseService.addCourse(course);  
    }  
  
    @RequestMapping(method=RequestMethod.PUT, value="/topics/{topicId}/courses/{id}")  
    public void updateCourse(@RequestBody Course course, @PathVariable String topicId, @PathVariable String id) {  
        course.setTopic(new Topic(topicId, "", ""));  
        courseService.updateCourse(course);  
    }  
  
    @RequestMapping(method=RequestMethod.DELETE, value="/topics/{topicId}/courses/{id}")  
    public void deleteTopic(@PathVariable String id) {  
        courseService.deleteCourse(id);  
    }  
}
```

Servers Boot Dashboard

Type tags to match

local

Writable Smart Insert 14 : 1

Builder Team Library OFFLINE Sign In

http://localhost:8080/ No Environment

GET http://localhost:8080/topics Params Send Save

Authorization Headers (1) Body Pre-request Script Tests Code

Type No Auth

Response

Hit the Send button to get a response.

GET http://localhost:8080/topics

POST http://localhost:8080/topics

PUT http://localhost:8080/topics/java

GET http://localhost:8080/topics/java

POST http://localhost:8080/topics

Type tags to match:

local

Builder Team Library OFFLINE Sign In

Filter History Collections Save to collection Today GET http://localhost:8080/topics Authorization Headers (1) Body Pre-request Script Tests Code

GET http://localhost:8080/topics Status: 200 OK Time: 302 ms

Body Cookies Headers (3) Tests

Pretty Raw Preview JSON

1

Type No Auth

Authorization

POST http://localhost:8080/topics

DEL http://localhost:8080/topics/javascript

PUT http://localhost:8080/topics/java

GET http://localhost:8080/topics/java

Type tags to match

local

Builder Team Library OFFLINE Sign In

http://localhost:8080/ No Environment

POST http://localhost:8080/topics Params Send Save

Authorization Headers (1) Body Pre-request Script Tests Code

form-data x-www-form-urlencoded raw binary JSON (application/json)

```
1 {  
2   "id": "javascript",  
3   "name": "JavaScript",  
4   "description": "JavaScript Description"  
5 }
```

Status: 200 OK Time: 152 ms

Type tags to match:

History Collections

Save to collection

Today

POST http://localhost:8080/topics

GET http://localhost:8080/topics

December 17

POST http://localhost:8080/topics

DEL http://localhost:8080/topics/javascript

PUT http://localhost:8080/topics/javascript

local

Builder Team Library OFFLINE Sign In

Filter History Collections Save to collection

POST http://localhost:8080/topics

Authorization Headers (1) Body ● Pre-request Script Tests Code

form-data x-www-form-urlencoded raw binary JSON (application/json)

```
1 {  
2   "id": "java",  
3   "name": "Java",  
4   "description": "Java Description"  
5 }
```

Response

Type tags to match

local

Runner Import +

Builder Team Library OFFLINE Sign In

Filter History Collections

Save to collection

Today

- GET http://localhost:8080/topics
- POST http://localhost:8080/topics
- POST http://localhost:8080/topics

December 17

- POST http://localhost:8080/topics
- DEL http://localhost:8080/topics/javascript

Type No Auth

Body Headers (3) Tests Status: 200 OK Time: 22 ms

Pretty Raw Preview JSON

```
1 [ ]  
2 {  
3   "id": "javascript",  
4   "name": "JavaScript",  
5   "description": "JavaScript Description"  
6 },  
7 {  
8   "id": "java",  
9   "name": "Java",  
10  "description": "Java Description"  
11 }  
12 ]
```

Builder Team Library OFFLINE Sign In

Filter History Collections

Save to collection

Today

- GET http://localhost:8080/topics/java/courses
- GET http://localhost:8080/topics
- POST http://localhost:8080/topics
- POST http://localhost:8080/topics

December 17

- POST http://localhost:8080/topics

Type No Auth

Body Headers (3) Tests Status: 200 OK Time: 50 ms

Pretty Raw Preview JSON

1

OFFLINE

javabrains.io

me/b ping JrlH JrlH JrlH beanE edSe iDat calh cher cher 'Fac

Builder Team Library OFFLINE Sign In

http://localhost:8080/ No Environment

POST http://localhost:8080/topics/java/courses Params Send Save

Authorization Headers (1) Body Pre-request Script Tests Code

form-data x-www-form-urlencoded raw binary JSON (application/json)

```
1 {  
2   "id": "java-streams",  
3   "name": "Java Streams",  
4   "description": "Java Streams Description"  
5 }
```

Status: 200 OK Time: 50 ms

Type tags to match

History Collections

Save to collection

Today

- GET http://localhost:8080/topics/java/courses
- GET http://localhost:8080/topics
- POST http://localhost:8080/topics
- POST http://localhost:8080/topics

December 17

- POST http://localhost:8080/topics

Pretty Raw Preview JSON

local

Runner Import +

Builder Team Library OFFLINE Sign In

Filter History Collections

Save to collection

Today

- POST http://localhost:8080/topics/java/courses
- GET http://localhost:8080/topics/java/courses
- GET http://localhost:8080/topics
- POST http://localhost:8080/topics
- POST http://localhost:8080/topics

December 17

- POST http://localhost:8080/topics

Type tags to match

local

http://localhost:8080/ + No Environment

POST http://localhost:8080/topics/java/courses Params Send Save

Authorization Headers (1) Body Pre-request Script Tests Code

form-data x-www-form-urlencoded raw binary JSON (application/json)

```
1 {  
2   "id": "java-streams",  
3   "name": "Java Streams",  
4   "description": "Java Streams Description"  
5 }
```

Body Cookies Headers (2) Tests Status: 200 OK Time: 32 ms

Pretty Raw Preview Text ⌂

javabrains.io

Runner Import +

Builder Team Library OFFLINE Sign In

Filter History Collections

Save to collection

Today

- GET http://localhost:8080/topics/java/courses
- POST http://localhost:8080/topics/java/courses
- GET http://localhost:8080/topics
- POST http://localhost:8080/topics
- POST http://localhost:8080/topics

December 17

- POST http://localhost:8080/topics

Type No Auth

Body Headers (3) Tests Status: 200 OK Time: 20 ms

Pretty Raw Preview JSON

```
1 [ {  
2   "id": "java-streams",  
3   "name": "Java Streams",  
4   "description": "Java Streams Description",  
5   "topic": {  
6     "id": "java",  
7     "name": "Java",  
8     "description": "Java Description"  
9   }  
10 }]  
11 ]
```

javabrains.io

Package Explorer Course.java CourseRepository.java CourseService.java CourseController.java Topic.java

```
+import javax.persistence.Entity; \n\n@Entity\npublic class Course {\n    @Id\n    private String id;\n    private String name;\n    private String description;\n    @ManyToOne\n    private Topic topic;\n\n    public Course() {\n    }\n\n    public Course(String id, String name, String description, String topicId) {\n        super();\n        this.id = id;\n        this.name = name;\n        this.description = description;\n        this.topic = new Topic(topicId, "", "");\n    }\n\n    public String getId() {\n        return id;\n    }\n    public void setId(String id) {\n        this.id = id;\n    }\n    public String getName() {\n        return name;\n    }\n    public void setName(String name) {\n        this.name = name;\n    }\n    public String getDescription() {\n        return description;\n    }\n    public void setDescription(String description) {\n        this.description = description;\n    }\n}
```

Servers Boot Dashboard

Type tags to match

local

Writable Smart Insert 17 : 25

Package Explorer Course.java CourseRepository.java CourseService.java CourseController.java Topic.java

```
+import javax.persistence.Entity; \n\n@Entity\npublic class Course {\n    @Id\n    private String id;\n    private String name;\n    private String description;\n    @ManyToOne\n    private Topic topic;\n\n    public Course() {\n    }\n\n    public Course(String id, String name, String description, String topicId) {\n        super();\n        this.id = id;\n        this.name = name;\n        this.description = description;\n        this.topic = new Topic(topicId, "", "");\n    }\n\n    public String getId() {\n        return id;\n    }\n    public void setId(String id) {\n        this.id = id;\n    }\n    public String getName() {\n        return name;\n    }\n    public void setName(String name) {\n        this.name = name;\n    }\n    public String getDescription() {\n        return description;\n    }\n    public void setDescription(String description) {\n        this.description = description;\n    }\n}
```

Servers Boot Dashboard

Type tags to match

local

Writable Smart Insert 16 : 5

Builder Team Library OFFLINE Sign In Filter History Collections Save to collection

No Environment

GET http://localhost:8080/topics/java/courses Params Send Save

Authorization Headers (1) Body Pre-request Script Tests Code

Type No Auth

Status: 200 OK Time: 20 ms

Body Cookies Headers (3) Tests

Pretty Raw Preview JSON

```

1 [ 
2   {
3     "id": "java-streams",
4     "name": "Java Streams",
5     "description": "Java Streams Description",
6     "topic": [
7       {
8         "id": "java",
9         "name": "Java",
10        "description": "Java Description"
11      }
12    ]
13  }
14 }
```

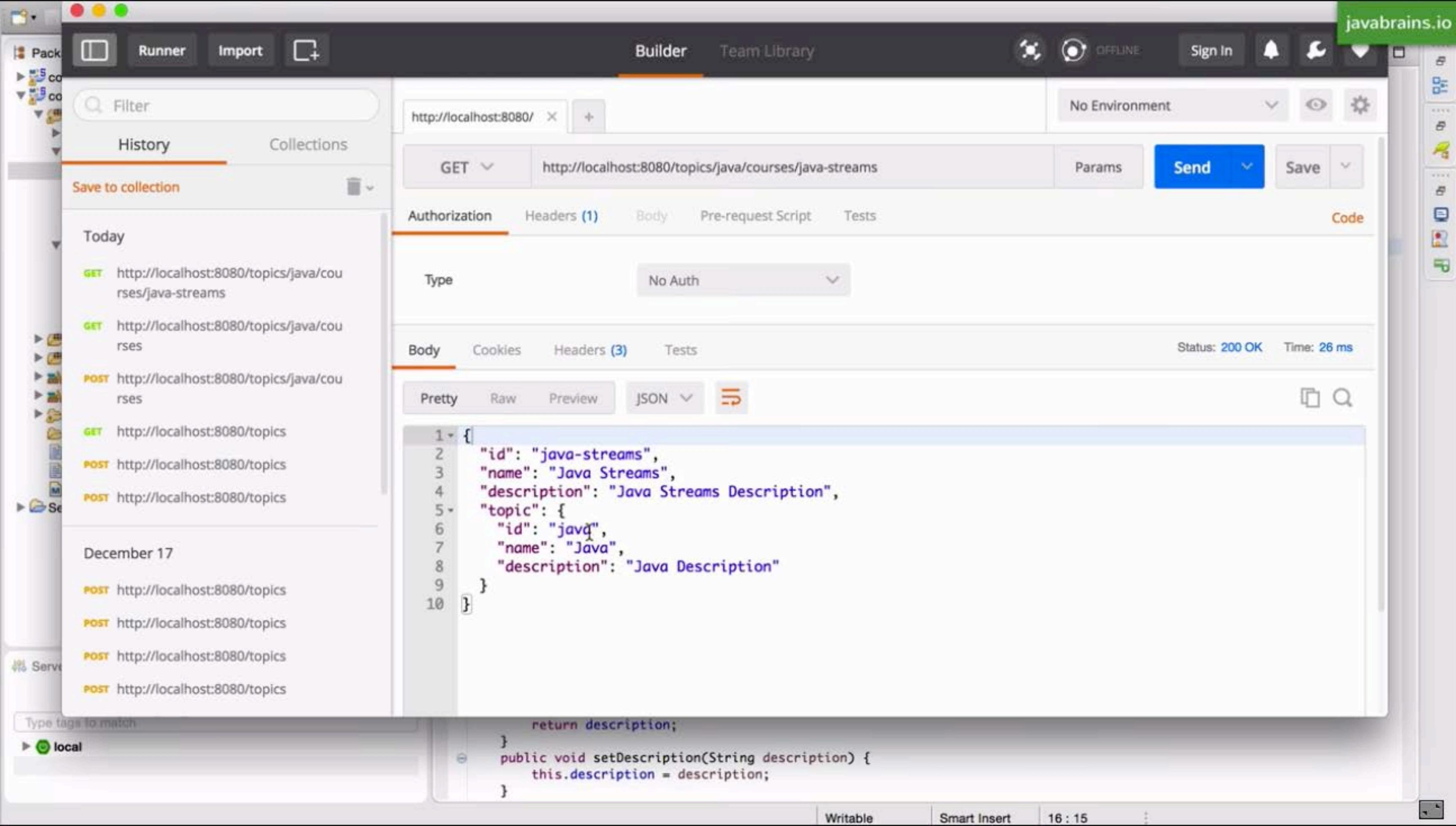
Type tags to match

local

```

    return description;
}
public void setDescription(String description) {
    this.description = description;
}
```

Writable Smart Insert 16 : 15



Builder Team Library OFFLINE Sign In Filter History Collections Save to collection

No Environment

PUT http://localhost:8080/topics/java/courses/java-streams Params Send Save

Authorization Headers (1) Body Pre-request Script Tests Code

form-data x-www-form-urlencoded raw binary JSON (application/json)

```

1  {
2    "id": "java-streams",
3    "name": "Java 8 Streams",
4    "description": "Java 8 Streams Description"
5
6 }
```

Status: 200 OK Time: 27 ms

Type tags to match

Body Cookies Headers (2) Tests

Pretty Raw Preview Text

```

        return description;
    }
    public void setDescription(String description) {
        this.description = description;
    }
}
```

Writable Smart Insert 16 : 15

Builder Team Library OFFLINE Sign In

Filter History Collections

No Environment

GET http://localhost:8080/topics/java/courses Params Send Save

Authorization Headers (1) Body Pre-request Script Tests Code

Type No Auth

Status: 200 OK Time: 17 ms

Body Cookies Headers (3) Tests

Pretty Raw Preview JSON

```

1  [
2   {
3     "id": "java-streams",
4     "name": "Java 8 Streams",
5     "description": "Java 8 Streams Description",
6     "topic": {
7       "id": "java",
8       "name": "Java",
9       "description": "Java Description"
10    }
11  }
12 ]

```

December 17

POST http://localhost:8080/topics

POST http://localhost:8080/topics

POST http://localhost:8080/topics

Type tags to match

local

```

        return description;
    }
    public void setDescription(String description) {
        this.description = description;
    }
}

```

Writable Smart Insert 16 : 15

Package Explorer Course.java CourseRepository.java CourseService.java CourseController.java Topic.java

```
+import javax.persistence.Entity; \n\n@Entity\npublic class Course {\n    @Id\n    private String id;\n    private String name;\n    private String description;\n    @ManyToOne\n    private Topic topic;\n\n    public Course() {\n    }\n\n    public Course(String id, String name, String description, String topicId) {\n        super();\n        this.id = id;\n        this.name = name;\n        this.description = description;\n        this.topic = new Topic(topicId, "", "");\n    }\n\n    public String getId() {\n        return id;\n    }\n    public void setId(String id) {\n        this.id = id;\n    }\n    public String getName() {\n        return name;\n    }\n    public void setName(String name) {\n        this.name = name;\n    }\n    public String getDescription() {\n        return description;\n    }\n    public void setDescription(String description) {\n        this.description = description;\n    }\n}
```

Servers Boot Dashboard

Type tags to match

local

Writable Smart Insert 16 : 15

Package Explorer Course.java CourseRepository.java CourseService.java CourseController.java Topic.java

```
package io.javabrains.springbootstarter.course;
import java.util.List;
public interface CourseRepository extends CrudRepository<Course, String> {
    public List<Course> findByTopicId(String topicId);
}
```

course-api [boot] course-api-data [boot]

src/main/java

- io.javabrains
- io.javabrains.springbootstarter.course
- Course.java
- CourseController.java
- CourseRepository.java
- CourseService.java

io.javabrains.springbootstarter.topic

- Topic.java
- TopicController.java
- TopicRepository.java
- TopicService.java

src/main/resources

src/test/java

JRE System Library [JavaSE-1.8]

Maven Dependencies

src

target

mvnw

mvnw.cmd

pom.xml

Servers

Servers Boot Dashboard

Type tags to match

local

Writable Smart Insert 12 : 1