Chapter 01: Core Features of Spring Boot

No-images…

Chapter 02: Creating a Web Application with Spring Boot

spring initializr

Project
- ○ Gradle - Groovy
- ○ Gradle - Kotlin
- ● Maven

Language
- ● Java
- ○ Kotlin
- ○ Groovy

Spring Boot
- ○ 3.0.1 (SNAPSHOT)
- ● 3.0.0
- ○ 2.7.7 (SNAPSHOT)
- ○ 2.7.6

Project Metadata

Group	com.springbootlearning.learningspringboot3
Artifact	ch2
Name	Chapter 2
Description	Creating a Web Application with Spring Boot
Package name	com.springbootlearning.learningspringboot3.ch2
Packaging	● Jar ○ War
Java	● 17 ○ 11 ○ 8

web| Press ⌘ for multiple adds

Spring Web `WEB`
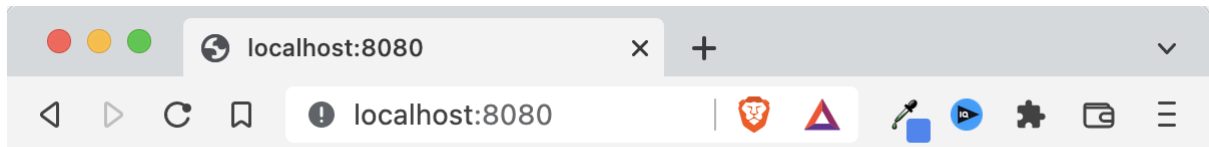Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container. ↵
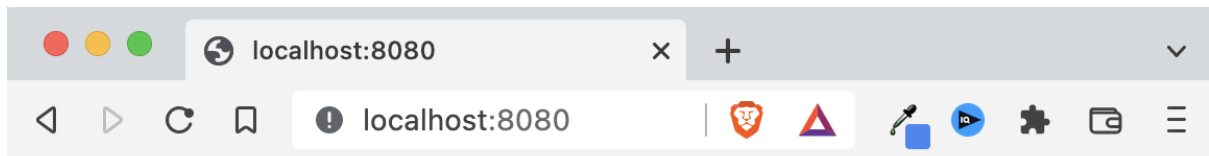
GENERATE

mustache

Mustache **TEMPLATE ENGINES**

Logic-less Templates. There are no if stat

EXPLORE CTRL + SPACE

localhost:8080

localhost:8080

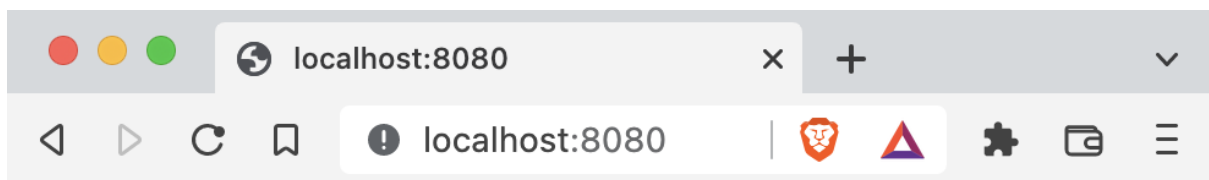# Greetings Learning Spring Boot 3.0 fans!

In this chapter, we are learning how to make a web app using Spring Boot 3.0

Greetings Learning Spring Boot 3.0 fans!

In this chapter, we are learning how to make a web app using Spring Boot 3.0

- Need HELP with your SPRING BOOT 3 App?
- Don't do THIS to your own CODE!
- SECRETS to fix BROKEN CODE!

Greetings Learning Spring Boot 3.0 fans!

In this chapter, we are learning how to make a web app using Spring Boot 3.0

- Need HELP with your SPRING BOOT 3 App?
- Don't do THIS to your own CODE!
- SECRETS to fix BROKEN CODE!
- Learning Spring Boot 3
  [   ] Submit

Chapter 03: Querying for Data with Spring Boot

No-images…

Chapter 04: Securing an Application with Spring Boot

Please sign in

Username	
Password	

Sign in

Please sign in

alice	
••••••••	

Sign in

Greetings Learning Spring Boot 3.0 fans!

In this chapter, we are learning how to make a web app using Spring Boot 3.0

User Profile

- Username: alice
- Authorities: [ROLE_USER]

Logout

Access to localhost was denied

You don't have authorization to view this page.

HTTP ERROR 403

Choose an account

to continue to **YouTube Manager**

▇▇▇▇▇▇▇▇▇▇
▇▇▇▇▇▇▇▇▇▇

G ▇▇▇▇▇▇▇
▇▇▇▇▇▇▇▇▇

Greg L. Turnquist
greg.l.turnquist@gmail.com

Choose your account or a brand account

to continue to **YouTube Manager**
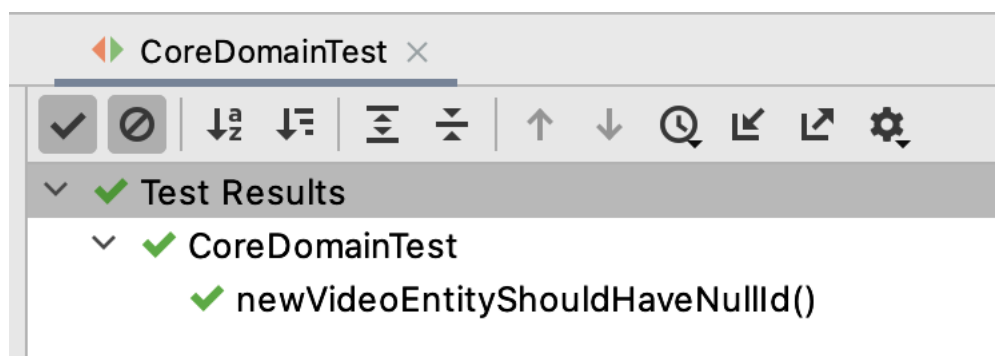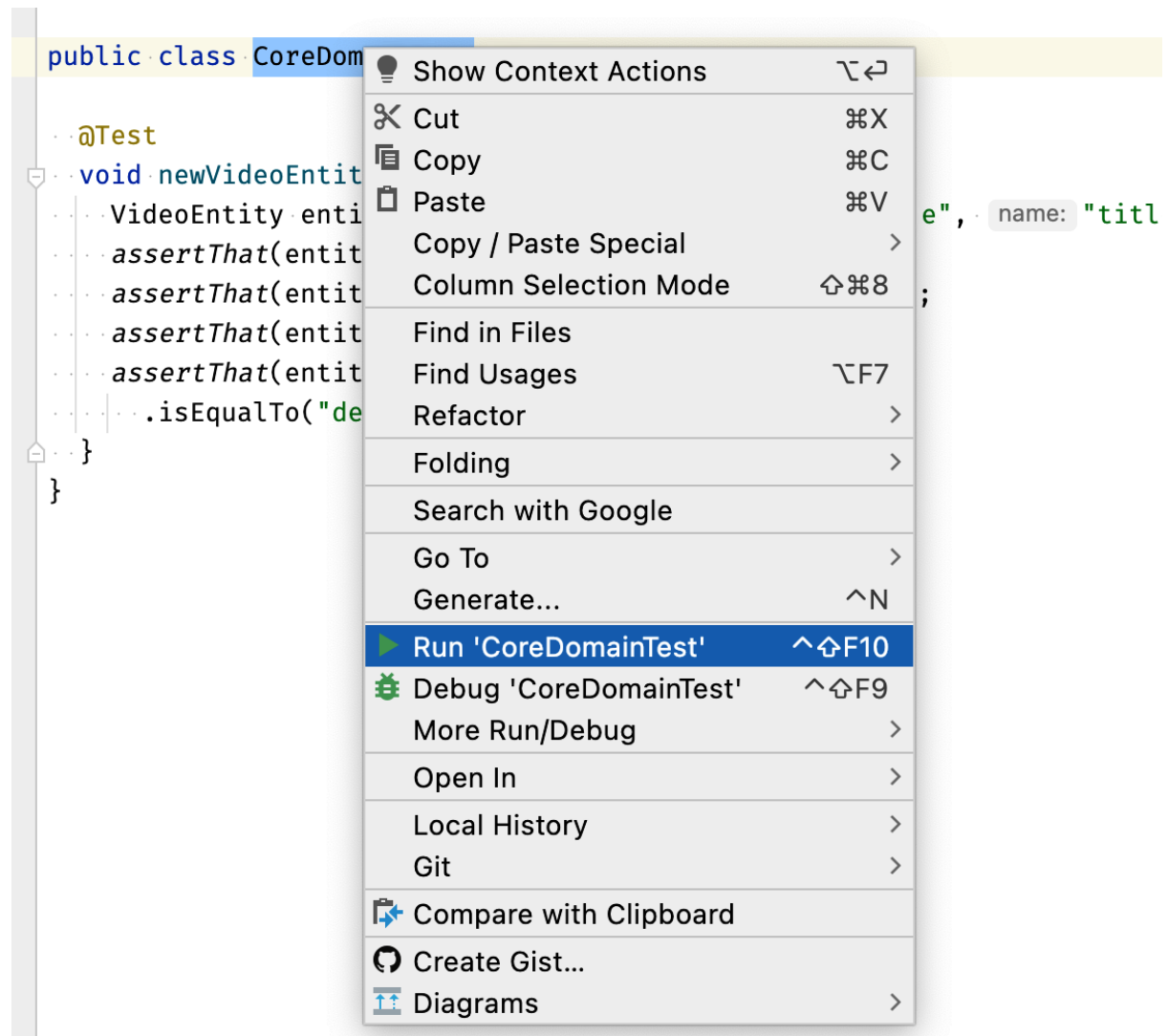
G

Spring Boot Learning
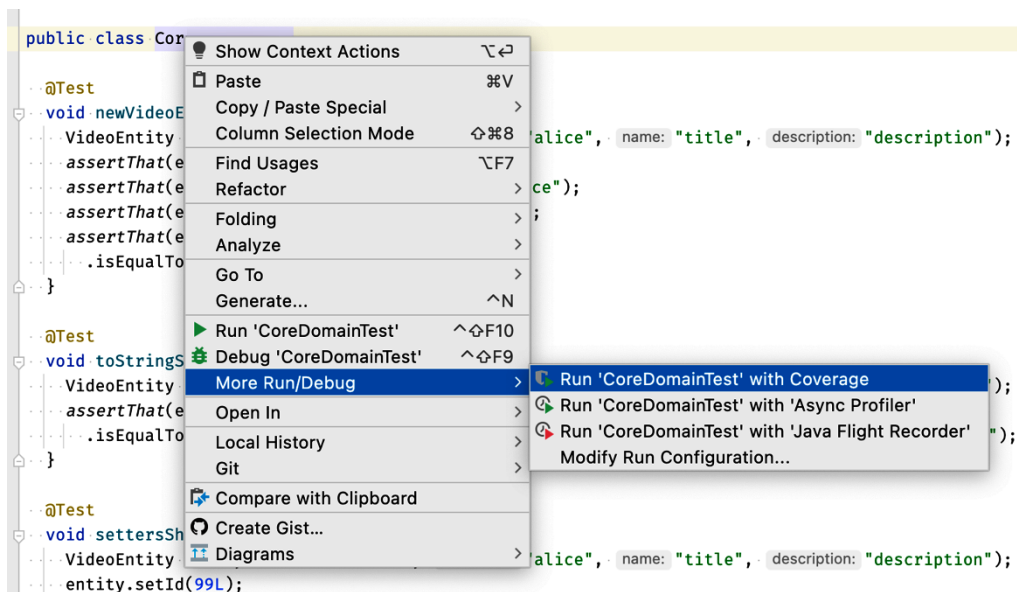Youtube

Greetings Learning Spring Boot 3.0 fans!

In this section, we are learning how to make a web app using Spring Boot 3.0 + OAuth 2.0

Your Videos

Id	Published	Thumbnail	Title	Description
KNcemsIbKcw	2021-03-12T12:00:12Z		SPRING BOOT and SPRING NATIVE	SPRING BOOT and SPRING NATIVE: Creating apps that start fast has never been easier! With the Sprin
v7xX6FxBhLs	2021-07-22T17:55:16Z		QUERY for DATA with SPRING BOOT	No web app is complete without fetching data. Find out why Spring Data is THE WAY to not only write
DD_Q4jGJsZ0	2021-04-16T18:42:36Z		5 WAYS to go to PRODUCTION with SPRING BOOT (ft. Josh Long)!	5 WAYS to go to PRODUCTION with SPRING BOOT: How are you going to get your app out to your users?

Chapter 05: Testing with Spring Boot

```
public class CoreDom
                        💡  Show Context Actions         ⌥↵

  @Test                 ✂  Cut                           ⌘X
  void newVideoEntit    ▣  Copy                          ⌘C
    VideoEntity enti    📋  Paste                         ⌘V    e",  name: "titl
    assertThat(entit       Copy / Paste Special          >
    assertThat(entit       Column Selection Mode       ⇧⌘8     ;
    assertThat(entit       Find in Files
    assertThat(entit       Find Usages                  ⌥F7
       .isEqualTo("de       Refactor                     >
  }                         Folding                       >
}
                           Search with Google

                           Go To                         >
                           Generate...                  ^N
                        ▶  Run 'CoreDomainTest'      ^⇧F10
                        🐞  Debug 'CoreDomainTest'     ^⇧F9
                           More Run/Debug                >

                           Open In                       >

                           Local History                 >
                           Git                           >
                        📥  Compare with Clipboard
                        ◯  Create Gist...
                        ↕  Diagrams                       >
```

CoreDomainTest ✕

✔ ⊘ | ↓ᵃ ↓≡ | ⤓ ⤒ | ↑ ↓ Q ↙ ↗ ⚙

∨ ✔ Test Results
   ∨ ✔ CoreDomainTest
        ✔ newVideoEntityShouldHaveNullId()

```
public class Cor
    @Test
    void newVideoE
        VideoEntity                                    alice", name: "title", description: "description");
        assertThat(e
        assertThat(e                               ce");
        assertThat(e                           ;
        assertThat(e
             .isEqualTo
    }

    @Test
    void toStringS
        VideoEntity                                              );
        assertThat(e
             .isEqualTo                                          ");
    }

    @Test
    void settersSh
        VideoEntity                                    alice", name: "title", description: "description");
        entity.setId(99L);
```

Context menu:
- 💡 Show Context Actions   ⌥↵
- 📋 Paste   ⌘V
- Copy / Paste Special   >
- Column Selection Mode   ⇧⌘8
- Find Usages   ⌥F7
- Refactor   >
- Folding   >
- Analyze   >
- Go To   >
- Generate...   ^N
- ▶ Run 'CoreDomainTest'   ^⇧F10
- 🐞 Debug 'CoreDomainTest'   ^⇧F9
- More Run/Debug   >
- Open In   >
- Local History   >
- Git   >
- Compare with Clipboard
- Create Gist...
- Diagrams   >

Submenu:
- 🟢 Run 'CoreDomainTest' with Coverage
- 🔵 Run 'CoreDomainTest' with 'Async Profiler'
- 🔵 Run 'CoreDomainTest' with 'Java Flight Recorder'
- Modify Run Configuration...

✔ Test Results	195 ms
✔ HomeControllerTest	195 ms
✔ postNewVideoShouldWork()	126 ms
✔ indexPageHasSeveralHtmlForms()	69 ms

✔ Test Results	460 ms
✔ VideoRepositoryTestcontainersTest	460 ms
✔ findAllShouldProduceAllVideos()	401 ms
✔ findByName()	35 ms
✔ findByNameOrDescription()	24 ms

Chapter 06: Configuring an Application with Spring Boot

Chapter 07: Releasing an Application with Spring Boot

Chapter 08: Going Native with Spring Boot

```
[2/7] Performing analysis...  [*********]                                    (50.5s @ 5.54GB)
  32,496 (94.29%) of 34,465 classes reachable
  55,941 (71.35%) of 78,401 fields reachable
 156,744 (62.52%) of 250,720 methods reachable
   1,416 classes, 1,822 fields, and 10,190 methods registered for reflection
      67 classes,    75 fields, and    57 methods registered for JNI access
       5 native libraries: -framework CoreServices, -framework Foundation, dl, pthread, z
[3/7] Building universe...                                                   (6.6s @ 4.37GB)
[4/7] Parsing methods...       [**]                                          (3.3s @ 4.85GB)
[5/7] Inlining methods...      [***]                                         (2.0s @ 3.33GB)
[6/7] Compiling methods...     [****]                                        (20.6s @ 5.15GB)
[7/7] Creating image...                                                      (7.3s @ 4.08GB)
  67.76MB (51.26%) for code area:   104,582 compilation units
  63.14MB (47.76%) for image heap:  634,397 objects and 526 resources
   1.29MB ( 0.98%) for other data
 132.20MB in total
-------------------------------------------------------------------------------------------
Top 10 packages in code area:              Top 10 object types in image heap:
   2.29MB jdk.proxy4                          14.76MB byte[] for code metadata
   1.66MB sun.security.ssl                     7.93MB java.lang.Class
   1.16MB java.util                            7.35MB byte[] for embedded resources
 914.47KB java.lang.invoke                     6.13MB java.lang.String
 717.60KB com.sun.crypto.provider              5.63MB byte[] for java.lang.String
 695.62KB org.hibernate.dialect                4.49MB byte[] for general heap data
 640.51KB org.h2.command                       2.73MB com.oracle.svm.core.hub.DynamicHubCompanion
 638.91KB org.apache.tomcat.util.net           2.02MB byte[] for reflection metadata
 540.62KB org.h2.table                         1.17MB java.lang.String[]
 537.87KB org.apache.catalina.core             1.11MB c.o.svm.core.hub.DynamicHub$ReflectionMetadata
  57.47MB for 1248 more packages               8.81MB for 5562 more object types
-------------------------------------------------------------------------------------------
               7.6s (7.2% of total time) in 65 GCs | Peak RSS: 8.19GB | CPU load: 5.31
-------------------------------------------------------------------------------------------
Produced artifacts:
 /Users/gturnquist/src/learning-spring-boot-3rd-edition-code/ch8/target/ch8 (executable)
 /Users/gturnquist/src/learning-spring-boot-3rd-edition-code/ch8/target/ch8.build_artifacts.txt (txt)
```
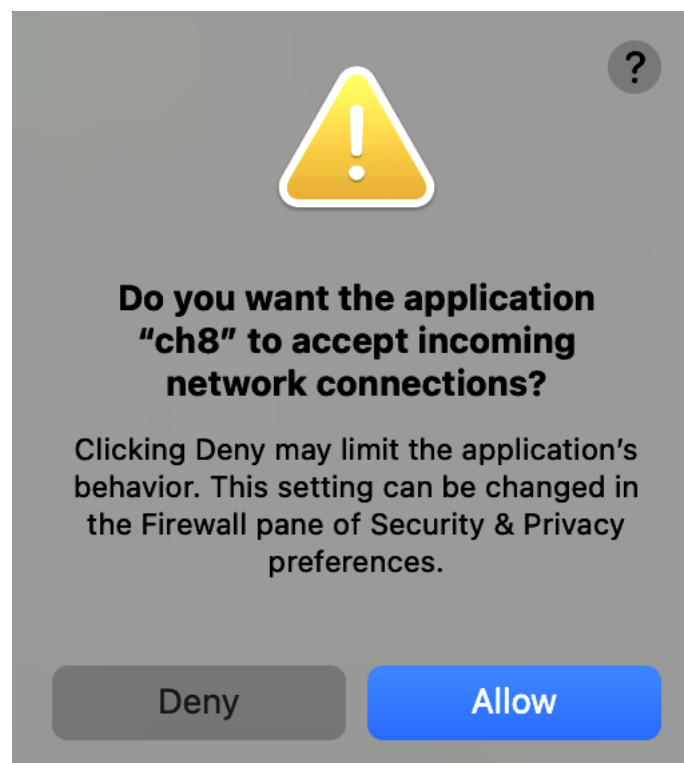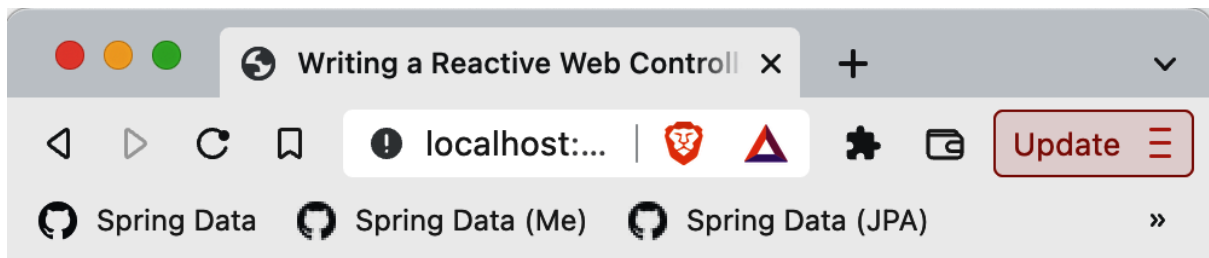
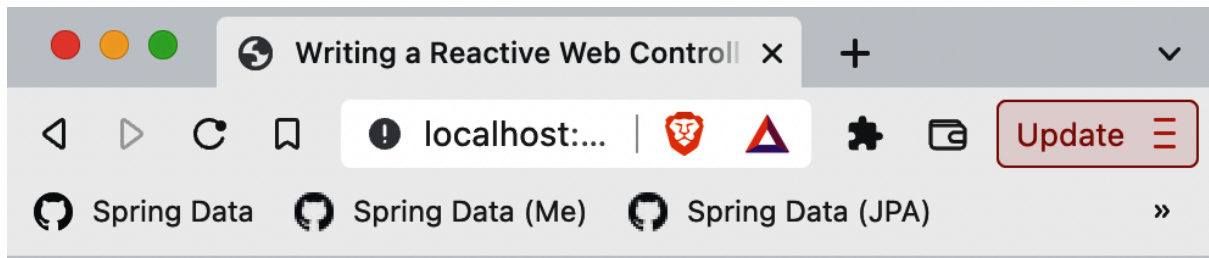# Chapter 09: Writing Reactive Web Controllers
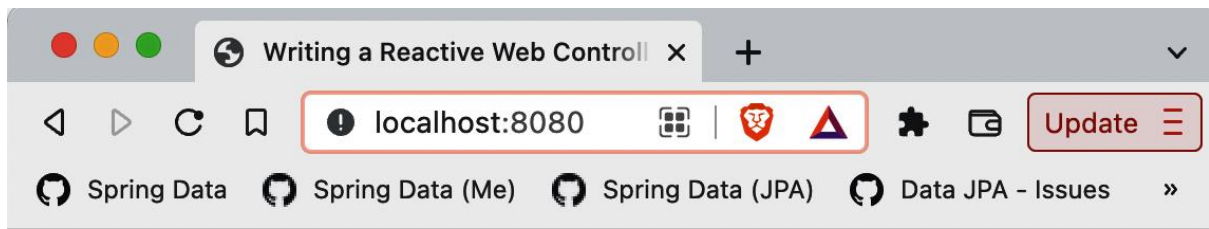
Employees

- Frodo Baggins (ring bearer)
- Samwise Gamgee (gardener)
- Bilbo Baggins (burglar)

Employees

- Frodo Baggins (ring bearer)
- Samwise Gamgee (gardener)
- Bilbo Baggins (burglar)

| Gandalf | wizard | Submit |

localhost:...

Spring Data   Spring Data (Me)   Spring Data (JPA)

Employees

- Frodo Baggins (ring bearer)
- Samwise Gamgee (gardener)
- Bilbo Baggins (burglar)
- Gandalf (wizard)

Submit

Chapter 10: Working with Data Reactively

No-images…