

## Section 3

# *Creating a File Upload Controller*



In this Section, we are going to take a look at...

- Creating a service to store images
- Creating a Spring MVC controller to handle incoming requests
- Uploading an image using cURL

Greg L. Turnquist

Video 3.1

## *Creating a Service to Store Images*



In this Video, we are going to take a look at...

- Creating an auto-wired Spring service
- Adding methods to lookup, store, and delete images
- Pre-load some test data

LearningSpringBootApplication

learning-spring-boot-video src main java com gregturnquist learningspringboot ImageRepository

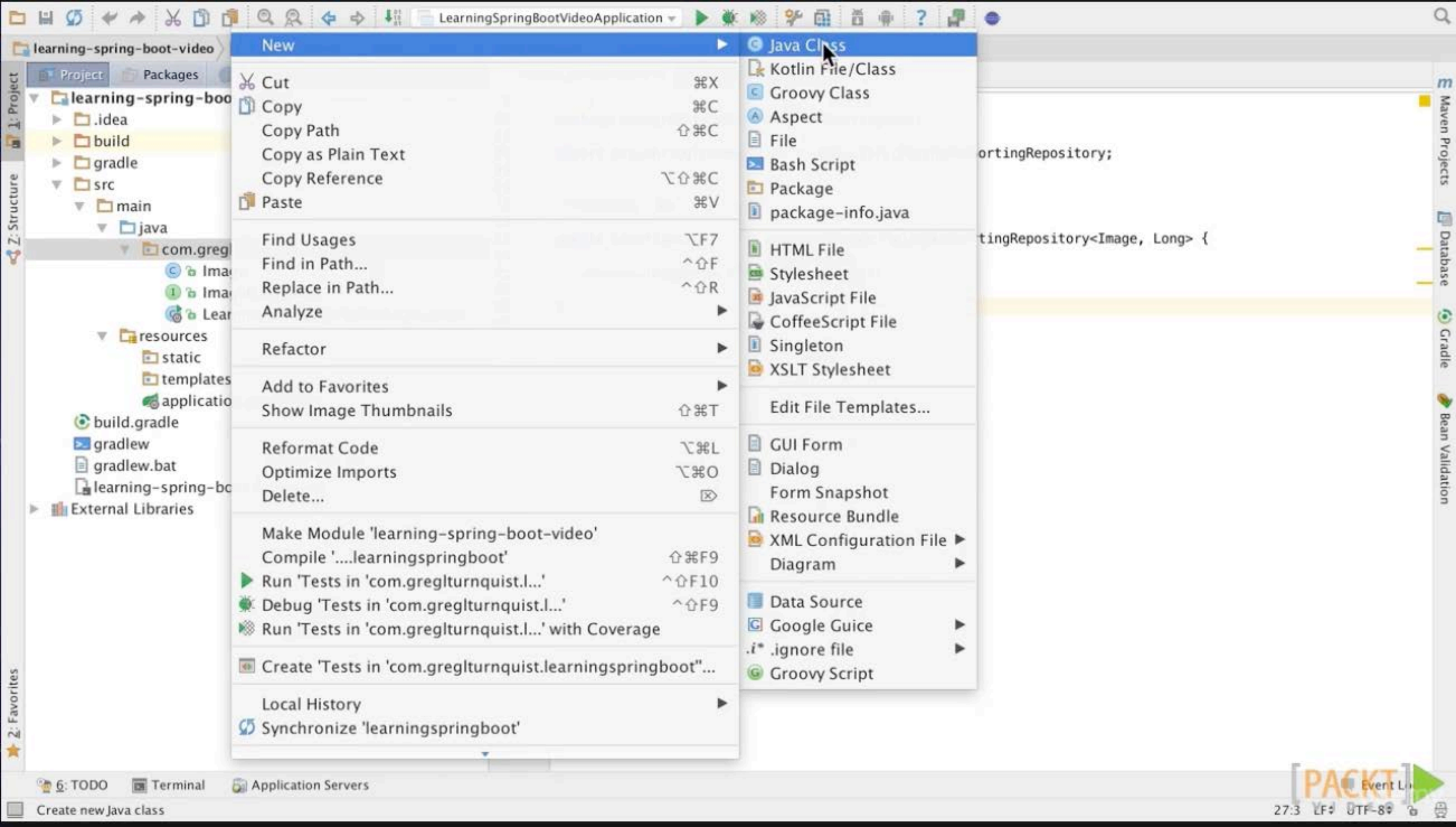
Project Packages Project Files ImageRepository.java x Image.java x

learning-spring-boot-video (~/.Downloads/learning-sp

- .idea
- build
- gradle
- src
  - main
    - java
      - com.gregturnquist.learningspringboot
        - Image
        - ImageRepository
        - LearningSpringBootApplication
      - resources
        - static
        - templates
        - application.properties
    - build.gradle
    - gradlew
    - gradlew.bat
    - learning-spring-boot-video.iml
  - External Libraries

```
1  ../../
16 package com.gregturnquist.learningspringboot;
17
18 import org.springframework.data.repository.PagingAndSortingRepository;
19
20 /**
21  * @author Greg Turnquist
22  */
23 public interface ImageRepository extends PagingAndSortingRepository<Image, Long> {
24
25     public Image findByName(String name);
26
27 }
28
```





New

- Cut ⌘X
- Copy ⌘C
- Copy Path ⇧⌘C
- Copy as Plain Text ⌘⇧C
- Copy Reference ⌘⇧C
- Paste ⌘V

- Find Usages ⌘F7
- Find in Path... ⇧⌘F
- Replace in Path... ⇧⌘R
- Analyze ▶

- Refactor ▶

- Add to Favorites ▶
- Show Image Thumbnails ⇧⌘T

- Reformat Code ⌘⇧L
- Optimize Imports ⌘⇧O
- Delete... ⌘⇧X

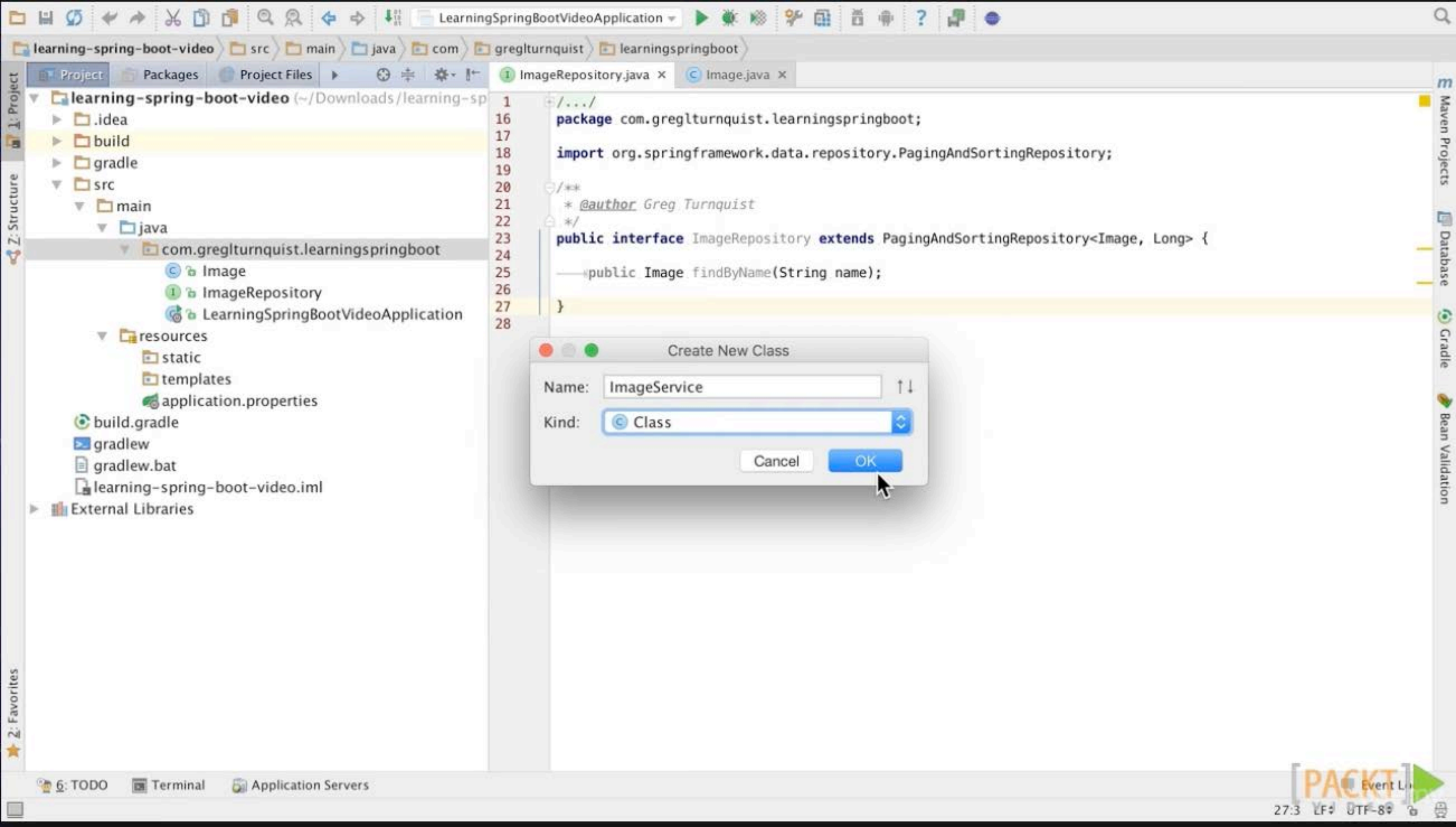
- Make Module 'learning-spring-boot-video'
- Compile '....learningspringboot' ⇧⌘F9
- Run 'Tests in 'com.greglturnquist.l...' ⇧⌘F10
- Debug 'Tests in 'com.greglturnquist.l...' ⇧⌘F9
- Run 'Tests in 'com.greglturnquist.l...' with Coverage

- Create 'Tests in 'com.greglturnquist.learningspringboot'...

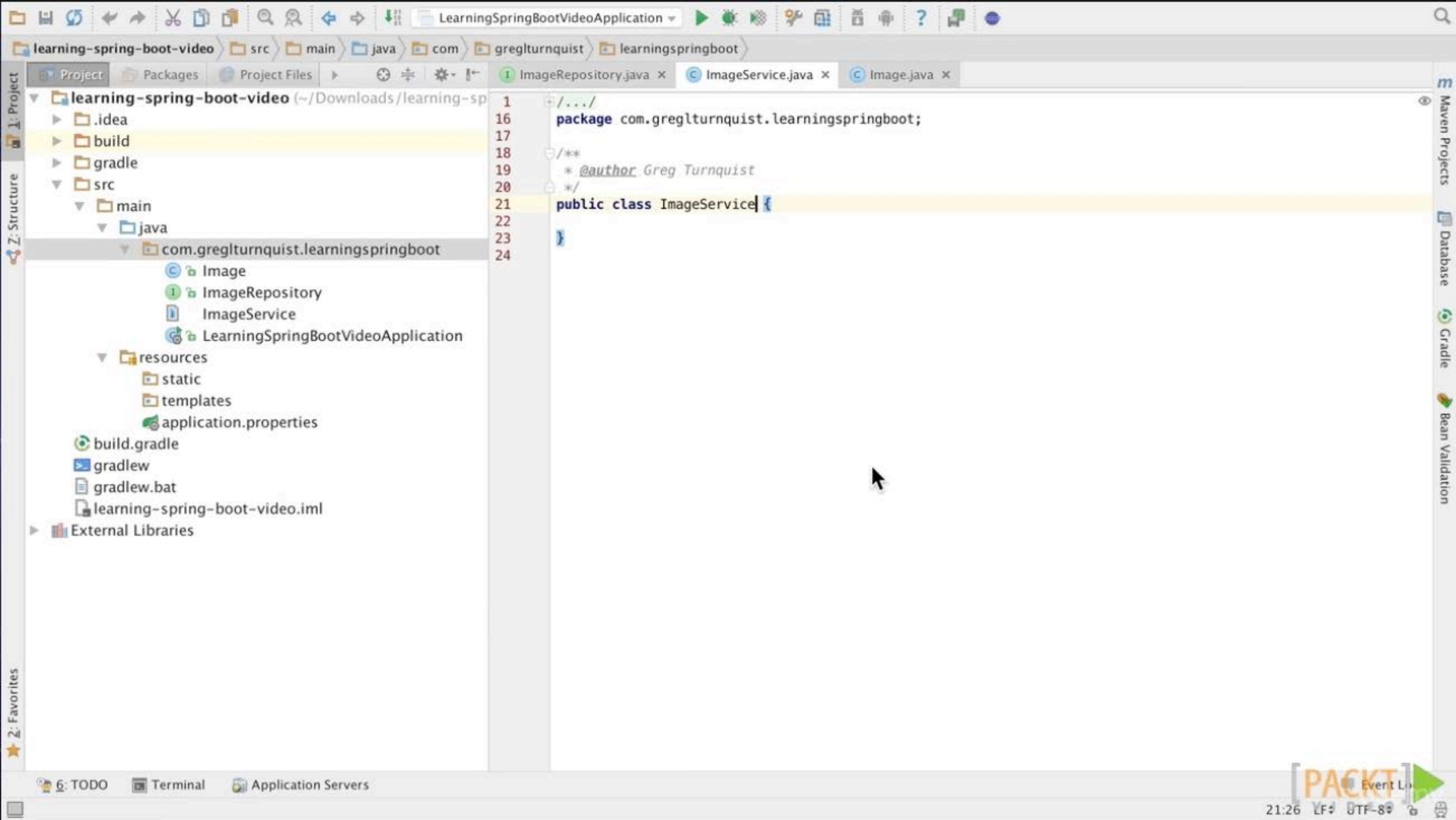
- Local History ▶
- Synchronize 'learningspringboot'

Java Class

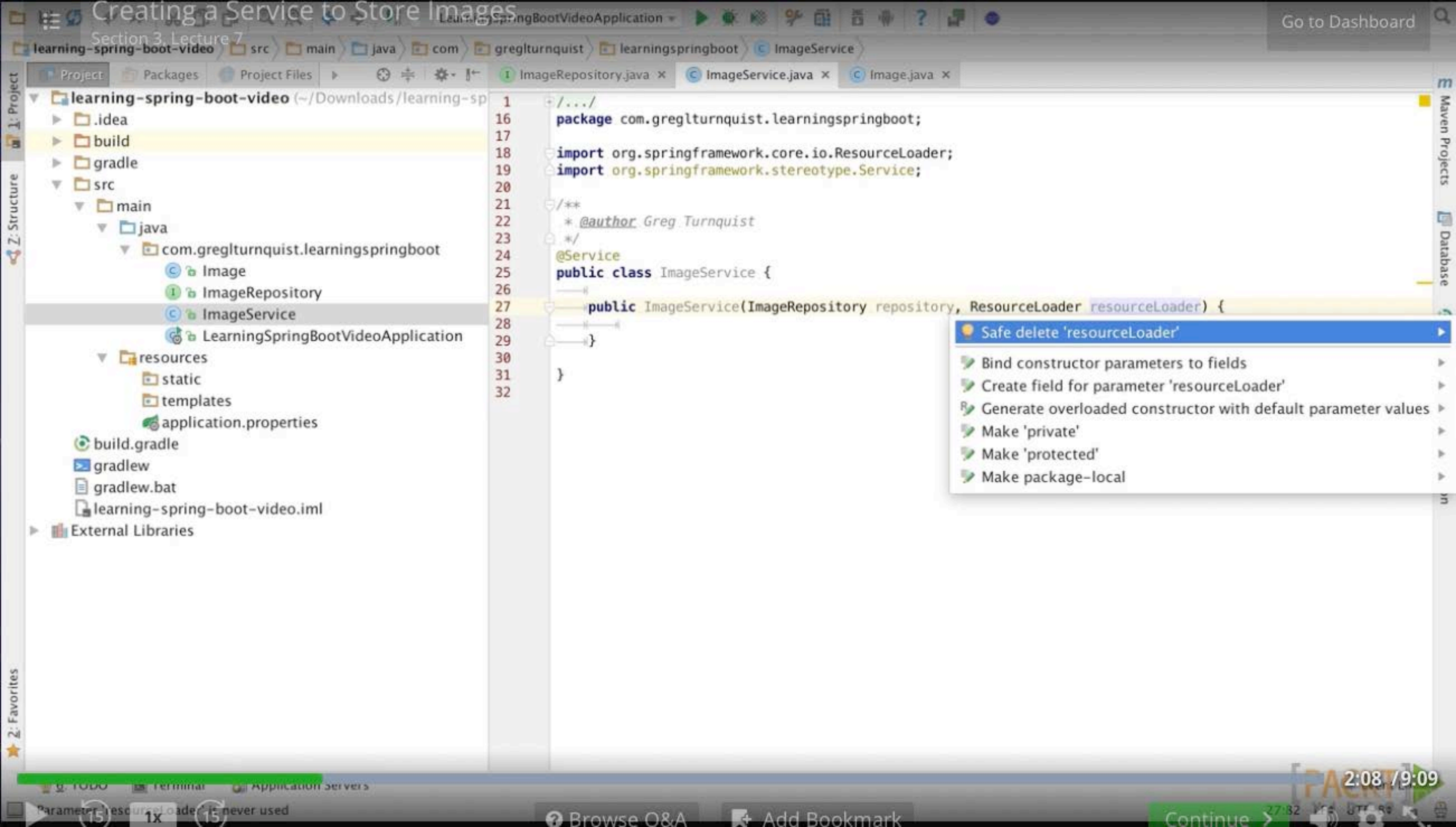
- Kotlin File/Class
- Groovy Class
- Aspect
- File
- Bash Script
- Package
- package-info.java
- HTML File
- Stylesheet
- JavaScript File
- CoffeeScript File
- Singleton
- XSLT Stylesheet
- Edit File Templates...
- GUI Form
- Dialog
- Form Snapshot
- Resource Bundle
- XML Configuration File ▶
- Diagram ▶
- Data Source
- Google Guice ▶
- .i\* .ignore file ▶
- Groovy Script







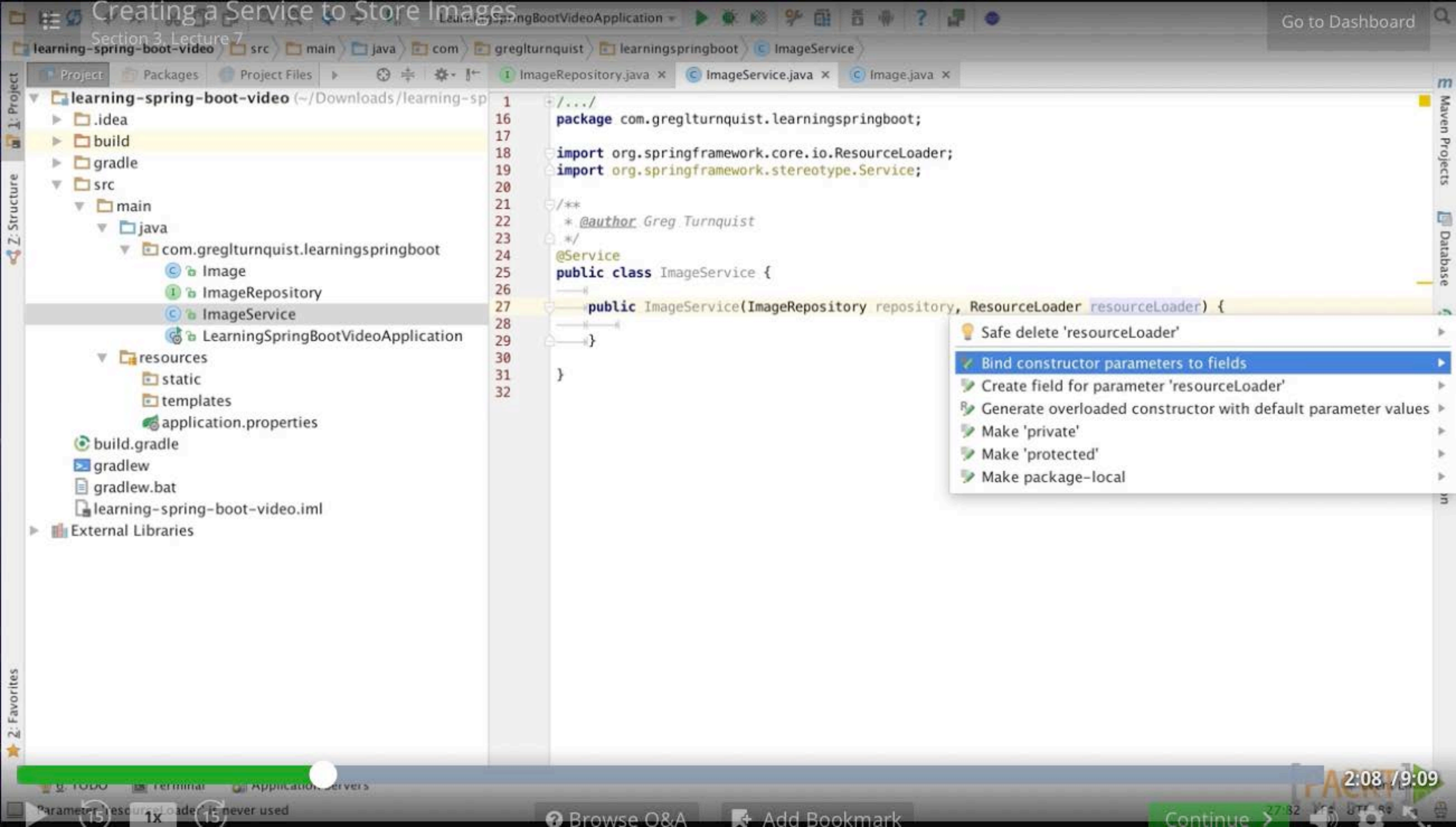




learning-spring-boot-video (~/.Downloads/learning-sp  
└─ .idea  
└─ build  
└─ gradle  
└─ src  
└─ main  
└─ java  
└─ com.gregturnquist.learningspringboot  
└─ Image  
└─ ImageRepository  
└─ ImageService  
└─ LearningSpringBootApplication  
└─ resources  
└─ static  
└─ templates  
└─ application.properties  
└─ build.gradle  
└─ gradlew  
└─ gradlew.bat  
└─ learning-spring-boot-video.iml  
└─ External Libraries

```
1  +/.../  
16 package com.gregturnquist.learningspringboot;  
17  
18 import org.springframework.core.io.ResourceLoader;  
19 import org.springframework.stereotype.Service;  
20  
21 /**  
22  * @author Greg Turnquist  
23  */  
24 @Service  
25 public class ImageService {  
26  
27     public ImageService(ImageRepository repository, ResourceLoader resourceLoader) {  
28  
29     }  
30  
31 }  
32
```

- Safe delete 'resourceLoader'
- Bind constructor parameters to fields
- Create field for parameter 'resourceLoader'
- Generate overloaded constructor with default parameter values
- Make 'private'
- Make 'protected'
- Make package-local



# Creating a Service to Store Images

Section 3, Lecture 7

Go to Dashboard

learning-spring-boot-video > src > main > java > com > gregturnquist > learningspringboot > ImageService

Project Packages Project Files ImageRepository.java x ImageService.java x Image.java x

learning-spring-boot-video (~/.Downloads/learning-sp  
└─ .idea  
└─ build  
└─ gradle  
└─ src  
└─ main  
└─ java  
└─ com.gregturnquist.learningspringboot  
└─ Image  
└─ ImageRepository  
└─ ImageService  
└─ LearningSpringBootVideoApplication  
└─ resources  
└─ static  
└─ templates  
└─ application.properties  
└─ build.gradle  
└─ gradlew  
└─ gradlew.bat  
└─ learning-spring-boot-video.iml  
└─ External Libraries

```
1  +/.../  
16 package com.gregturnquist.learningspringboot;  
17  
18 import org.springframework.core.io.ResourceLoader;  
19 import org.springframework.stereotype.Service;  
20  
21 /**  
22  * @author Greg Turnquist  
23  */  
24 @Service  
25 public class ImageService {  
26  
27     public ImageService(ImageRepository repository, ResourceLoader resourceLoader) {  
28  
29     }  
30  
31 }  
32
```

- Safe delete 'resourceLoader'
- Bind constructor parameters to fields
- Create field for parameter 'resourceLoader'
- Generate overloaded constructor with default parameter values
- Make 'private'
- Make 'protected'
- Make package-local

2: Favorites

g. TODO Terminal Application Servers

Parameter 'resourceLoader' is never used

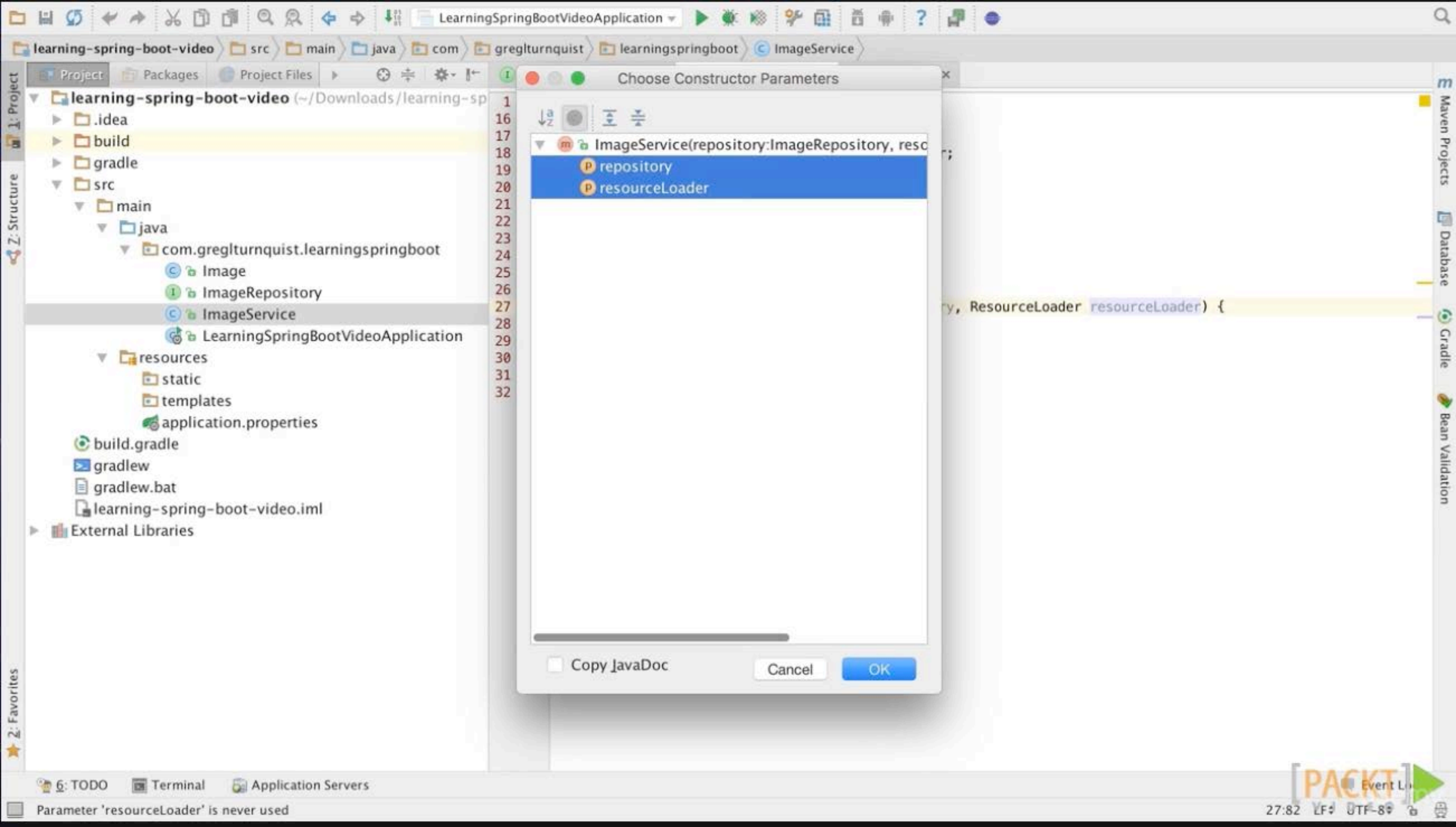
Browse O&A

Add Bookmark

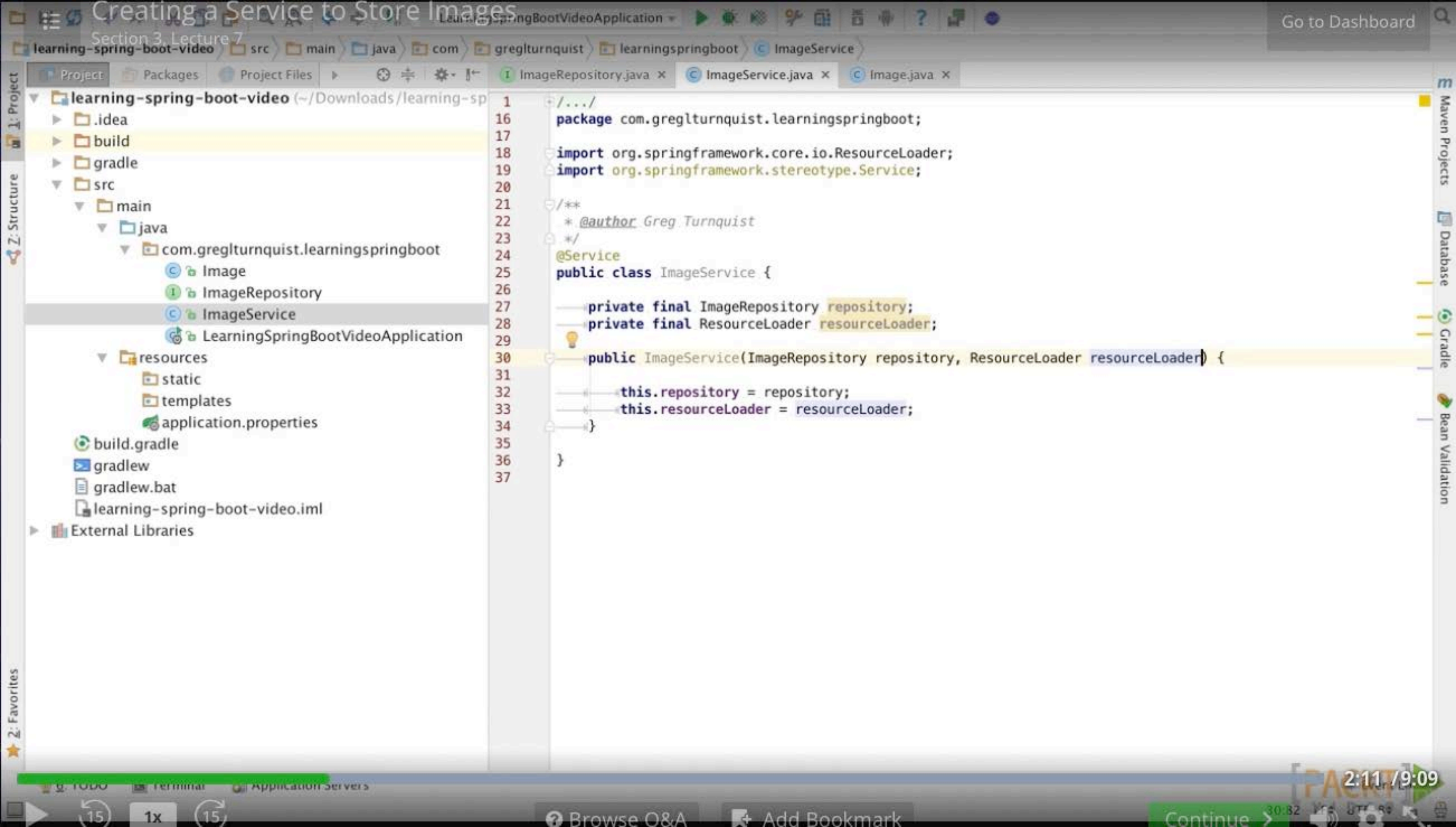
Continue

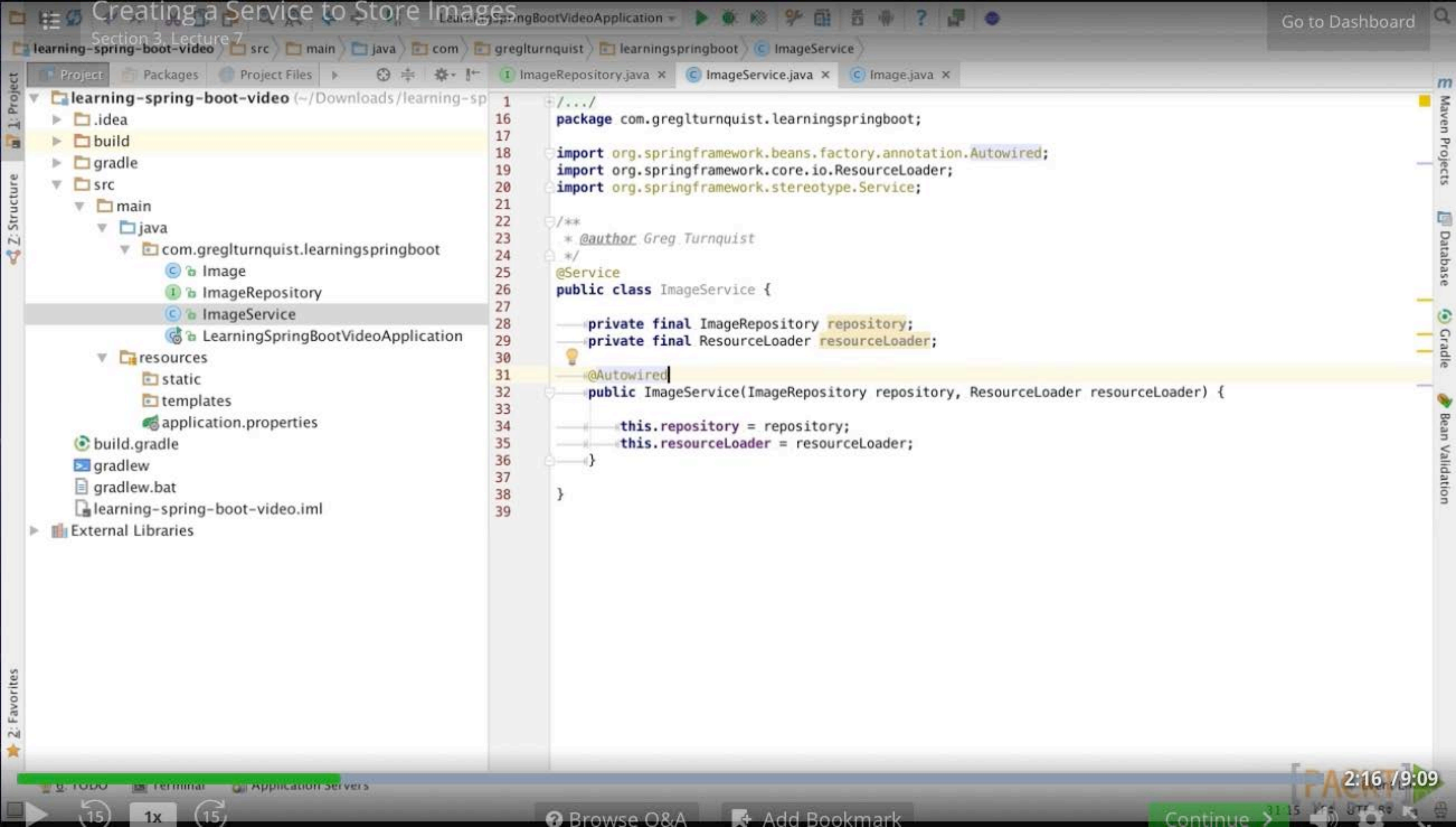
2:08 / 9:09











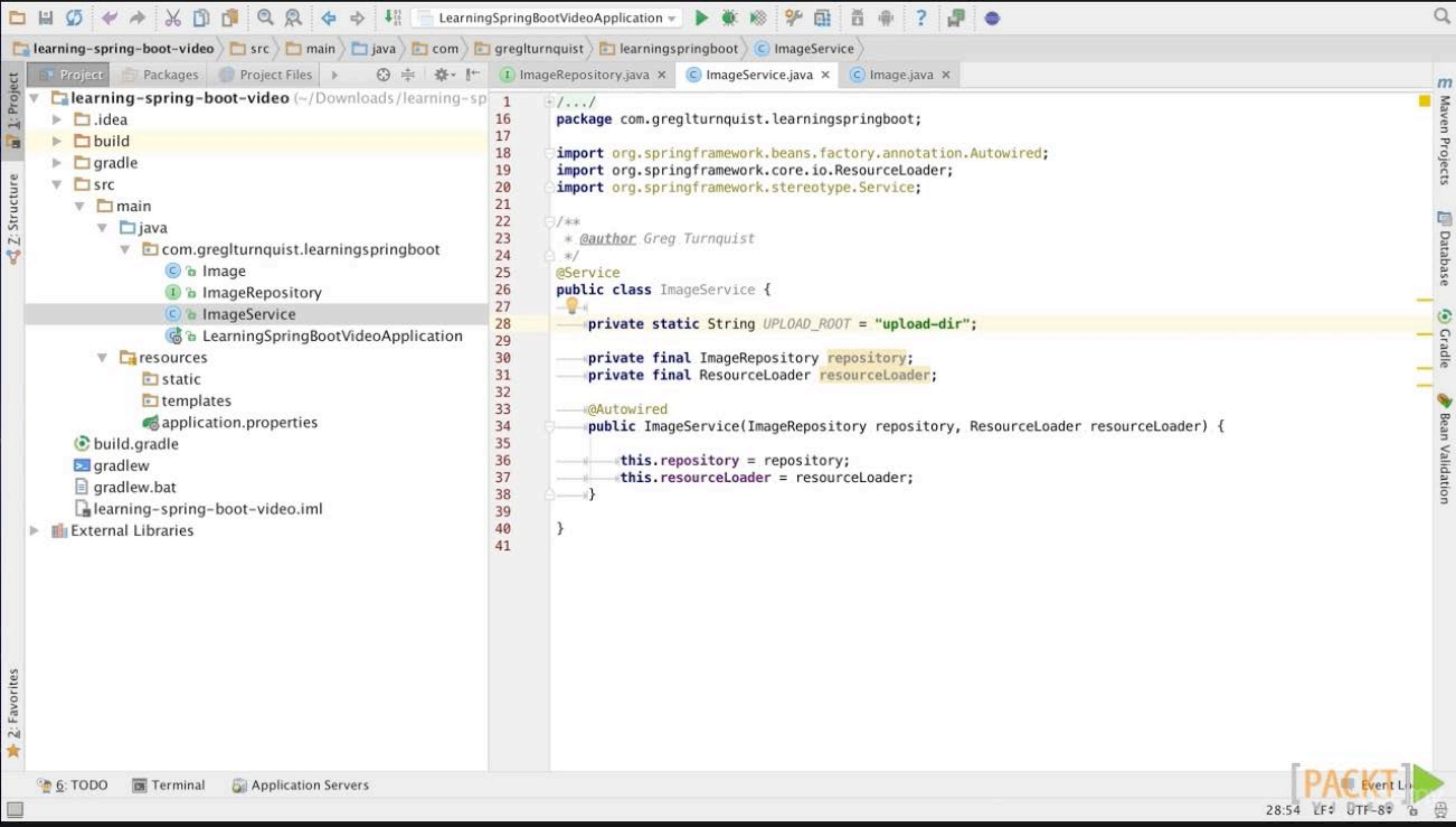
learning-spring-boot-video (~/.Downloads/learning-sp)

- .idea
- build
- gradle
- src
  - main
    - java
      - com.gregturnquist.learningspringboot
        - Image
        - ImageRepository
        - ImageService
        - LearningSpringBootVideoApplication
  - resources
    - static
    - templates
    - application.properties
- build.gradle
- gradlew
- gradlew.bat
- learning-spring-boot-video.iml

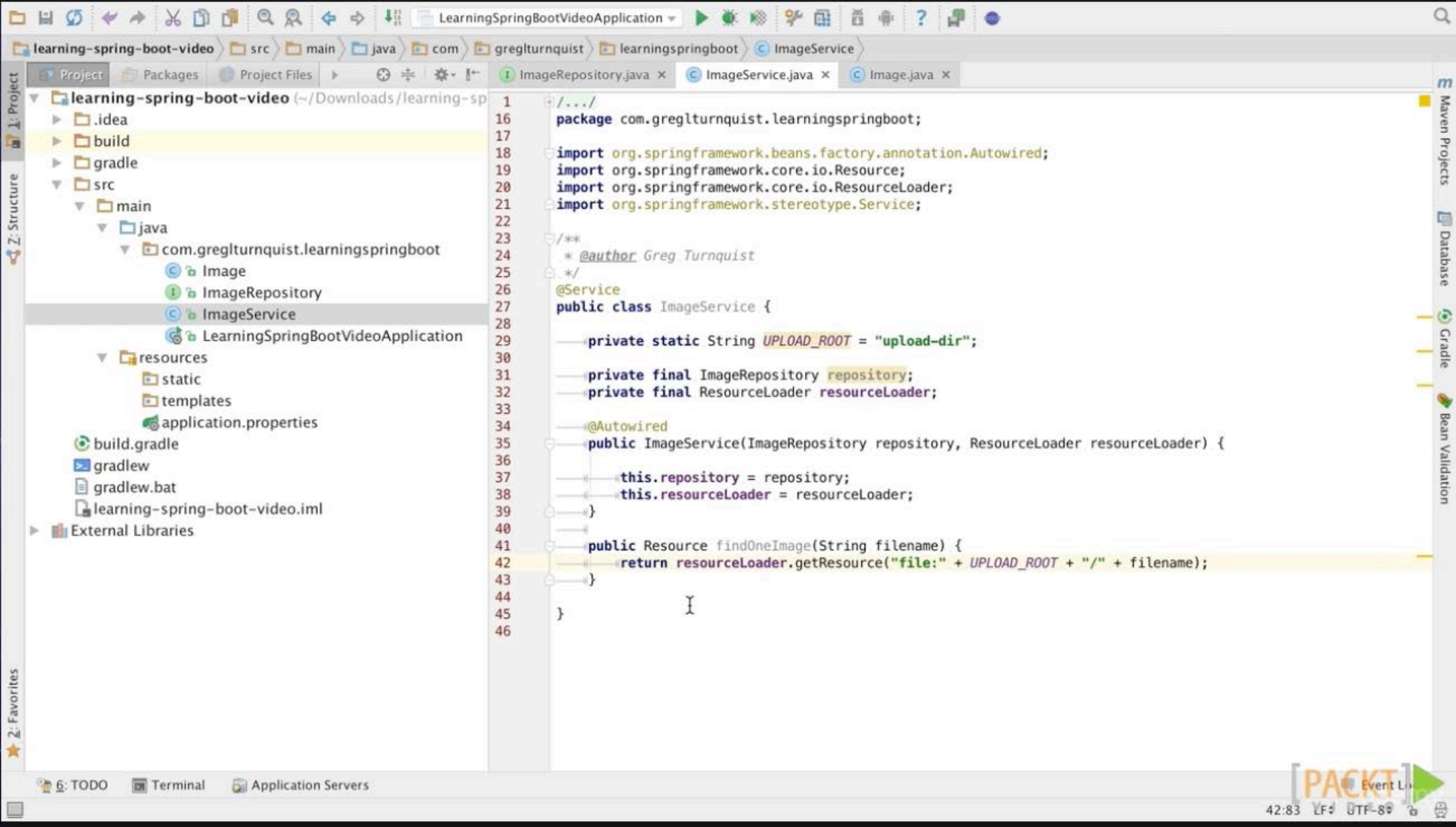
External Libraries

```
1  +/.../  
16 package com.gregturnquist.learningspringboot;  
17  
18 import org.springframework.beans.factory.annotation.Autowired;  
19 import org.springframework.core.io.ResourceLoader;  
20 import org.springframework.stereotype.Service;  
21  
22 /**  
23  * @author Greg Turnquist  
24  */  
25 @Service  
26 public class ImageService {  
27  
28     private final ImageRepository repository;  
29     private final ResourceLoader resourceLoader;  
30  
31     @Autowired  
32     public ImageService(ImageRepository repository, ResourceLoader resourceLoader) {  
33  
34         this.repository = repository;  
35         this.resourceLoader = resourceLoader;  
36     }  
37  
38 }  
39
```











learning-spring-boot-video

src

main

java

com

greghturnquist

learningspringboot

ImageService

Project

Packages

Project Files

learning-spring-boot-video

idea

build

gradle

src

main

java

com.greghturnquist.learningspringboot

Image

ImageRepository

ImageService

LearningSpringBootVideoApplication

resources

static

templates

application.properties

build.gradle

gradlew

gradlew.bat

learning-spring-boot-video.iml

External Libraries

ImageRepository.java

ImageService.java

Image.java

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

import java.nio.file.Files;

import java.nio.file.Paths;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.core.io.Resource;

import org.springframework.core.io.ResourceLoader;

import org.springframework.stereotype.Service;

import org.springframework.web.multipart.MultipartFile;

/\*\*

\* @author Greg Turnquist

\*/

@Service

public class ImageService {

private static String UPLOAD\_ROOT = "upload-dir";

private final ImageRepository repository;

private final ResourceLoader resourceLoader;

@Autowired

public ImageService(ImageRepository repository, ResourceLoader resourceLoader) {

this.repository = repository;

this.resourceLoader = resourceLoader;

}

public Resource findOneImage(String filename) {

return resourceLoader.getResource("file:" + UPLOAD\_ROOT + "/" + filename);

}

public void createImage(MultipartFile file) {

if (!file.isEmpty()) {

Files.copy(file.getInputStream(), Paths.get(UPLOAD\_ROOT, file.getOriginalFilename()));

repository.save(new Image(file.getOriginalFilename()));

}

}

Go to Dashboard

Maven Projects

Database

Gradle

Bean Validation

2: Favorites

Unhandled exception: java.io.IOException

4:27 / 9:09

Browse O&A

Add Bookmark

Continue



learning-spring-boot-video

src

main

java

com

gregturnquist

learningspringboot

ImageService

Project

Packages

Project Files

learning-spring-boot-video

.idea

build

gradle

src

main

java

com.gregturnquist.learningspringboot

Image

ImageRepository

ImageService

LearningSpringBootVideoApplication

resources

static

templates

application.properties

build.gradle

gradlew

gradlew.bat

learning-spring-boot-video.iml

External Libraries

ImageRepository.java

ImageService.java

Image.java

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

import java.nio.file.Files;

import java.nio.file.Paths;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.core.io.Resource;

import org.springframework.core.io.ResourceLoader;

import org.springframework.stereotype.Service;

import org.springframework.web.multipart.MultipartFile;

/\*\*

\* @author Greg Turnquist

\*/

@Service

public class ImageService {

private static String UPLOAD\_ROOT = "upload-dir";

private final ImageRepository repository;

private final ResourceLoader resourceLoader;

@Autowired

public ImageService(ImageRepository repository, ResourceLoader resourceLoader) {

this.repository = repository;

this.resourceLoader = resourceLoader;

}

public Resource findOneImage(String filename) {

return resourceLoader.getResource("file:" + UPLOAD\_ROOT + "/" + filename);

}

public void createImage(MultipartFile file) {

if (!file.isEmpty()) {

Files.copy(file.getInputStream(), Paths.get(UPLOAD\_ROOT, file.getOriginalFilename()));

repository.save(new Image(file.getOriginalFilename(), file.getBytes()));

}

}

Go to Dashboard

Maven Projects

Database

Gradle

Bean Validation

2: Favorites

Unhandled exception: java.io.IOException

4:28 / 9:09

Browse Q&A

Add Bookmark

Continue



learning-spring-boot-video

src

main

java

com

gregturnquist

learningspringboot

ImageService

Project

Packages

Project Files

learning-spring-boot-video

.idea

build

gradle

src

main

java

com.gregturnquist.learningspringboot

Image

ImageRepository

ImageService

LearningSpringBootVideoApplication

resources

static

templates

application.properties

build.gradle

gradlew

gradlew.bat

learning-spring-boot-video.iml

External Libraries

ImageRepository.java

ImageService.java

Image.java

```
24 import org.springframework.core.io.ResourceLoader;
25 import org.springframework.stereotype.Service;
26 import org.springframework.web.multipart.MultipartFile;
27
28 /**
29  * @author Greg Turnquist
30  */
31 @Service
32 public class ImageService {
33
34     private static String UPLOAD_ROOT = "upload-dir";
35
36     private final ImageRepository repository;
37     private final ResourceLoader resourceLoader;
38
39     @Autowired
40     public ImageService(ImageRepository repository, ResourceLoader resourceLoader) {
41
42         this.repository = repository;
43         this.resourceLoader = resourceLoader;
44     }
45
46     public Resource findOneImage(String filename) {
47         return resourceLoader.getResource("file:" + UPLOAD_ROOT + "/" + filename);
48     }
49
50     public void createImage(MultipartFile file) throws IOException {
51
52         if (!file.isEmpty()) {
53             Files.copy(file.getInputStream(), Paths.get(UPLOAD_ROOT, file.getOriginalFilename()));
54             repository.save(new Image(file.getOriginalFilename()));
55         }
56     }
57
58 }
59
```

Maven Projects

Database

Gradle

Bean Validation

6: TODO

Terminal

Application Servers

PACKT

Event Log

53:13 LF UTF-8



learning-spring-boot-video

src

main

java

com

greghturnquist

learningspringboot

ImageService

Project

Packages

Project Files

learning-spring-boot-video

src

main

java

com.greghturnquist.learningspringboot

Image

ImageRepository

ImageService

LearningSpringBootVideoApplication

resources

static

templates

application.properties

build.gradle

gradlew

gradlew.bat

learning-spring-boot-video.iml

External Libraries

ImageRepository.java

ImageService.java

Image.java

```
24 import org.springframework.core.io.ResourceLoader;
25 import org.springframework.stereotype.Service;
26 import org.springframework.web.multipart.MultipartFile;
27
28 /**
29  * @author Greg Turnquist
30  */
31 @Service
32 public class ImageService {
33
34     private static String UPLOAD_ROOT = "upload-dir";
35
36     private final ImageRepository repository;
37     private final ResourceLoader resourceLoader;
38
39     @Autowired
40     public ImageService(ImageRepository repository, ResourceLoader resourceLoader) {
41
42         this.repository = repository;
43         this.resourceLoader = resourceLoader;
44     }
45
46     public Resource findOneImage(String filename) {
47         return resourceLoader.getResource("file:" + UPLOAD_ROOT + "/" + filename);
48     }
49
50     public void createImage(MultipartFile file) throws IOException {
51
52         if (!file.isEmpty()) {
53             Files.copy(file.getInputStream(), Paths.get(UPLOAD_ROOT, file.getOriginalFilename()));
54             repository.save(new Image(file.getOriginalFilename()));
55         }
56     }
57
58     public void deleteImage(String filename) throws IOException {
59
60         final Image byName = repository.findByName(filename);
61         repository.delete(byName);
62         Files.deleteIfExists(Paths.get(UPLOAD_ROOT, filename));
63     }
64 }
```

Maven Projects

Database

Gradle

Bean Validation

6: TODO

Terminal

Application Servers

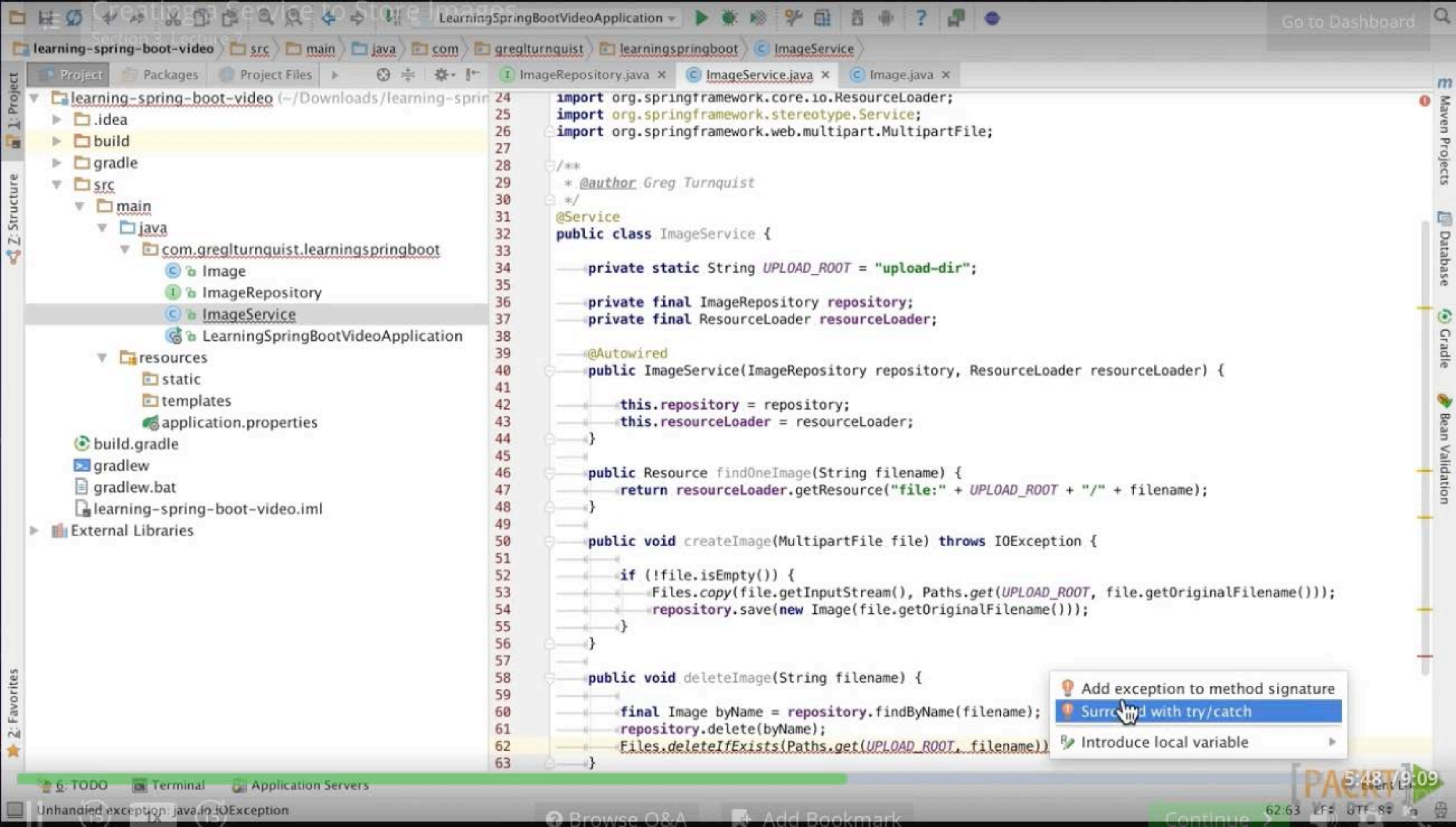
Unhandled exception: java.io.IOException

62:63 LF UTF-8

PACKT

Event L





learning-spring-boot-video src main java com gregturnquist learningspringboot ImageService

Project Packages Project Files ImageRepository.java x ImageService.java x Image.java x

learning-spring-boot-video (~/.Downloads/learning-sprin

- .idea
- build
- gradle
- src
  - main
    - java
      - com.gregturnquist.learningspringboot
        - Image
        - ImageRepository
        - ImageService
        - LearningSpringBootVideoApplication
  - resources
    - static
    - templates
    - application.properties
  - build.gradle
  - gradlew
  - gradlew.bat
  - learning-spring-boot-video.iml
- External Libraries

```
24 import org.springframework.core.io.ResourceLoader;
25 import org.springframework.stereotype.Service;
26 import org.springframework.web.multipart.MultipartFile;
27
28 /**
29  * @author Greg Turnquist
30  */
31 @Service
32 public class ImageService {
33
34     private static String UPLOAD_ROOT = "upload-dir";
35
36     private final ImageRepository repository;
37     private final ResourceLoader resourceLoader;
38
39     @Autowired
40     public ImageService(ImageRepository repository, ResourceLoader resourceLoader) {
41
42         this.repository = repository;
43         this.resourceLoader = resourceLoader;
44     }
45
46     public Resource findOneImage(String filename) {
47         return resourceLoader.getResource("file:" + UPLOAD_ROOT + "/" + filename);
48     }
49
50     public void createImage(MultipartFile file) throws IOException {
51
52         if (!file.isEmpty()) {
53             Files.copy(file.getInputStream(), Paths.get(UPLOAD_ROOT, file.getOriginalFilename()));
54             repository.save(new Image(file.getOriginalFilename()));
55         }
56     }
57
58     public void deleteImage(String filename) {
59
60         final Image byName = repository.findByName(filename);
61         repository.delete(byName);
62         Files.deleteIfExists(Paths.get(UPLOAD_ROOT, filename));
63     }
64 }
```

- ⚠ Add exception to method signature
- 👉 Surrender with try/catch
- 🔗 Introduce local variable



learning-spring-boot-video

src

main

java

com

gregturnquist

learningspringboot

ImageService

Project

Packages

Project Files

learning-spring-boot-video (~/.Downloads/learning-sp

.idea

build

gradle

src

main

java

com.gregturnquist.learningspringboot

Image

ImageRepository

ImageService

LearningSpringBootVideoApplication

resources

static

templates

application.properties

build.gradle

gradlew

gradlew.bat

learning-spring-boot-video.iml

External Libraries

ImageRepository.java x

ImageService.java x

Image.java x

```
32 public class ImageService {
33
34     private static String UPLOAD_ROOT = "upload-dir";
35
36     private final ImageRepository repository;
37     private final ResourceLoader resourceLoader;
38
39     @Autowired
40     public ImageService(ImageRepository repository, ResourceLoader resourceLoader) {
41
42         this.repository = repository;
43         this.resourceLoader = resourceLoader;
44     }
45
46     public Resource findOneImage(String filename) {
47         return resourceLoader.getResource("file:" + UPLOAD_ROOT + "/" + filename);
48     }
49
50     public void createImage(MultipartFile file) throws IOException {
51
52         if (!file.isEmpty()) {
53             Files.copy(file.getInputStream(), Paths.get(UPLOAD_ROOT, file.getOriginalFilename()));
54             repository.save(new Image(file.getOriginalFilename()));
55         }
56     }
57
58     public void deleteImage(String filename) throws IOException {
59
60         final Image byName = repository.findByName(filename);
61         repository.delete(byName);
62         Files.deleteIfExists(Paths.get(UPLOAD_ROOT, filename));
63     }
64 }
65
66
```

Maven Projects

Database

Gradle

Bean Validation

6: TODO

Terminal

Application Servers

PACKT

Event Log

62:63 LF UTF-8



LearningSpringBootVideoApplication

learning-spring-boot-video src main java com gregturnquist learningspringboot ImageService

Project Packages Project Files ImageRepository.java x ImageService.java x Image.java x

learning-spring-boot-video (~/.Downloads/learning-spring-boot-video)  
├── .idea  
├── build  
├── gradle  
├── src  
│ ├── main  
│ │ ├── java  
│ │ │ ├── com.gregturnquist.learningspringboot  
│ │ │ │ ├── Image  
│ │ │ │ ├── ImageRepository  
│ │ │ │ ├── ImageService  
│ │ │ │ └── LearningSpringBootVideoApplication  
│ │ └── resources  
│ │ ├── static  
│ │ ├── templates  
│ │ └── application.properties  
│ └── test  
├── build.gradle  
├── gradlew  
├── gradlew.bat  
└── learning-spring-boot-video.iml

External Libraries

```
52 public Resource findOneImage(String filename) {  
53     return resourceLoader.getResource("file:" + UPLOAD_ROOT + "/" + filename);  
54 }  
55  
56 public void createImage(MultipartFile file) throws IOException {  
57  
58     if (!file.isEmpty()) {  
59         Files.copy(file.getInputStream(), Paths.get(UPLOAD_ROOT, file.getOriginalFilename()));  
60         repository.save(new Image(file.getOriginalFilename()));  
61     }  
62 }  
63  
64 public void deleteImage(String filename) throws IOException {  
65  
66     final Image byName = repository.findByName(filename);  
67     repository.delete(byName);  
68     Files.deleteIfExists(Paths.get(UPLOAD_ROOT, filename));  
69 }  
70  
71 @Bean  
72 CommandLineRunner setUp(ImageRepository repository) throws IOException {  
73  
74     FileSystemUtils.deleteRecursively(new File(UPLOAD_ROOT));  
75  
76     Files.createDirectory(Paths.get(UPLOAD_ROOT));  
77  
78     FileCopyUtils.copy("Test file", new FileWriter(UPLOAD_ROOT + "/test"));  
79     repository.save(new Image("test"));  
80  
81     FileCopyUtils.copy("Test file2", new FileWriter(UPLOAD_ROOT + "/test2"));  
82     repository.save(new Image("test2"));  
83  
84     FileCopyUtils.copy("Test file3", new FileWriter(UPLOAD_ROOT + "/test3"));  
85     repository.save(new Image("test3"));  
86  
87 }  
88  
89 }  
90
```



learning-spring-boot-video

src

main

java

com

gregturnquist

learningspringboot

ImageService

Project

Packages

Project Files

learning-spring-boot-video

~/.Downloads/learning-sprin

.idea

build

gradle

src

main

java

com.gregturnquist.learningspringboot

Image

ImageRepository

ImageService

LearningSpringBootVideoApplication

resources

static

templates

application.properties

build.gradle

gradlew

gradlew.bat

learning-spring-boot-video.iml

External Libraries

ImageRepository.java

ImageService.java

Image.java

```
58         if (!file.isEmpty()) {
59             Files.copy(file.getInputStream(), Paths.get(UPLOAD_ROOT, file.getOriginalFilename()));
60             repository.save(new Image(file.getOriginalFilename()));
61         }
62     }
63
64     public void deleteImage(String filename) throws IOException {
65
66         final Image byName = repository.findByName(filename);
67         repository.delete(byName);
68         Files.deleteIfExists(Paths.get(UPLOAD_ROOT, filename));
69     }
70
71     @Bean
72     CommandLineRunner setUp(ImageRepository repository) throws IOException {
73
74         return (args) -> {
75             FileSystemUtils.deleteRecursively(new File(UPLOAD_ROOT));
76
77             Files.createDirectory(Paths.get(UPLOAD_ROOT));
78
79             FileCopyUtils.copy("Test file", new FileWriter(UPLOAD_ROOT + "/test"));
80             repository.save(new Image("test"));
81
82             FileCopyUtils.copy("Test file2", new FileWriter(UPLOAD_ROOT + "/test2"));
83             repository.save(new Image("test2"));
84
85             FileCopyUtils.copy("Test file3", new FileWriter(UPLOAD_ROOT + "/test3"));
86             repository.save(new Image("test3"));
87         }
88     }
89 }
90
91 }
92 }
```

Maven Projects

Database

Gradle

Bean Validation

6: TODO

Terminal

Application Servers

PACKT

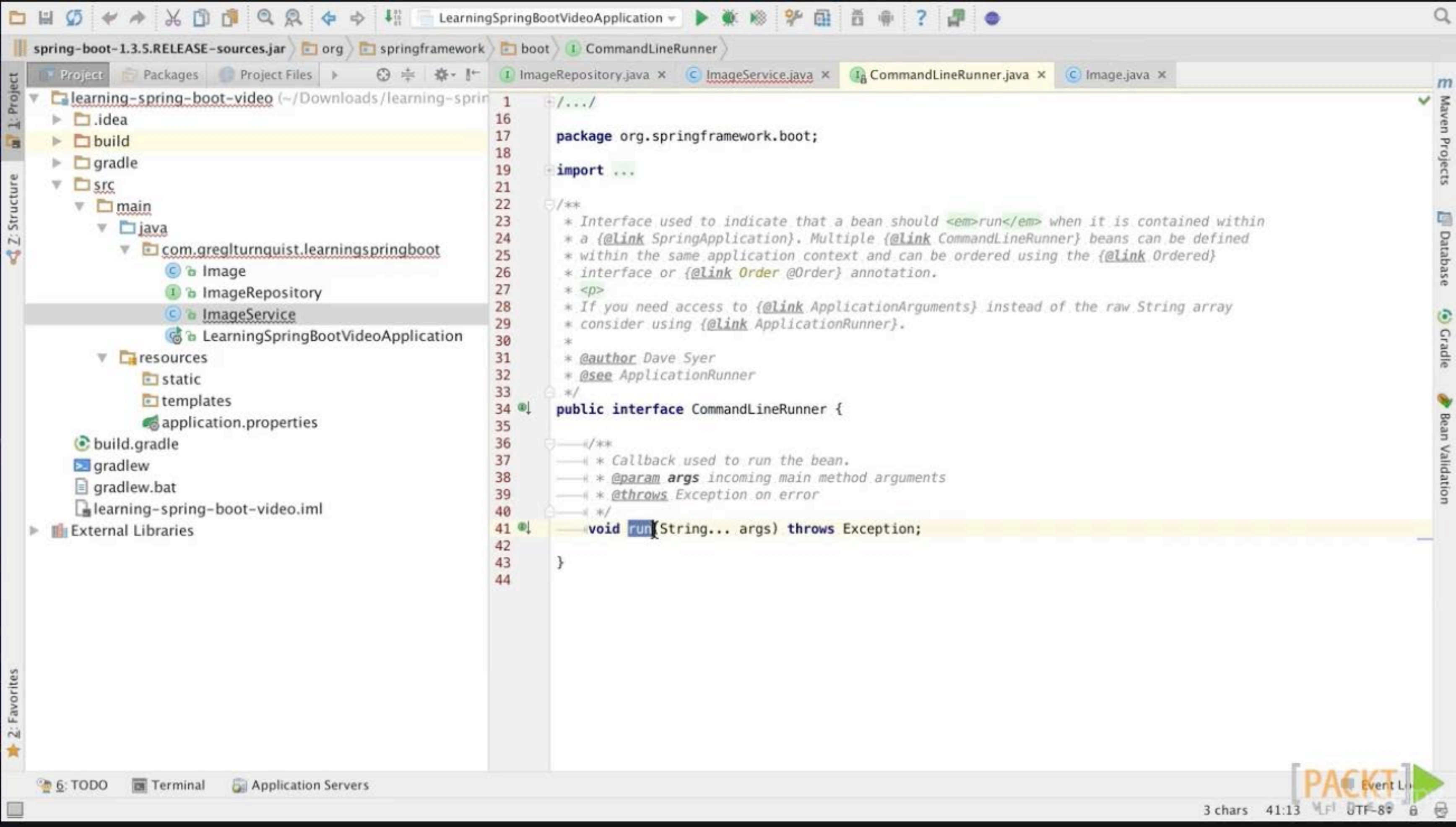
Event Li

74:9

LF

UTF-8





Next Video

Creating a Spring MVC Controller to  
Handle Incoming Requests