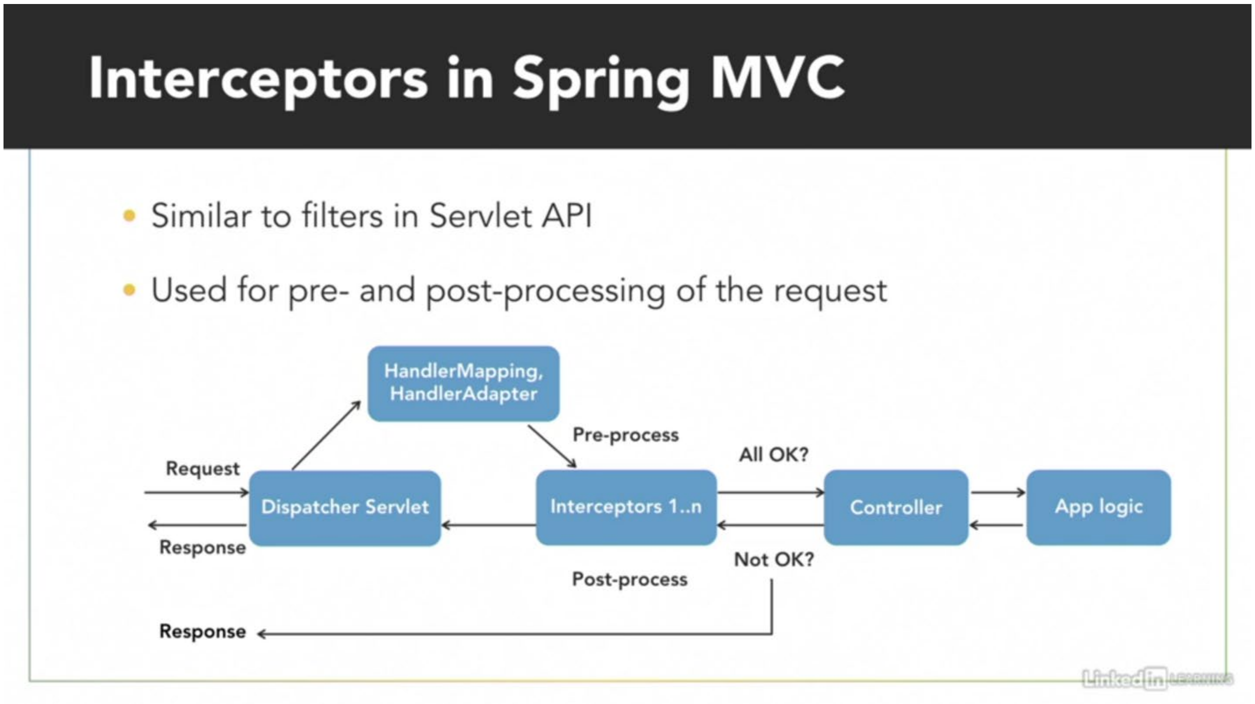
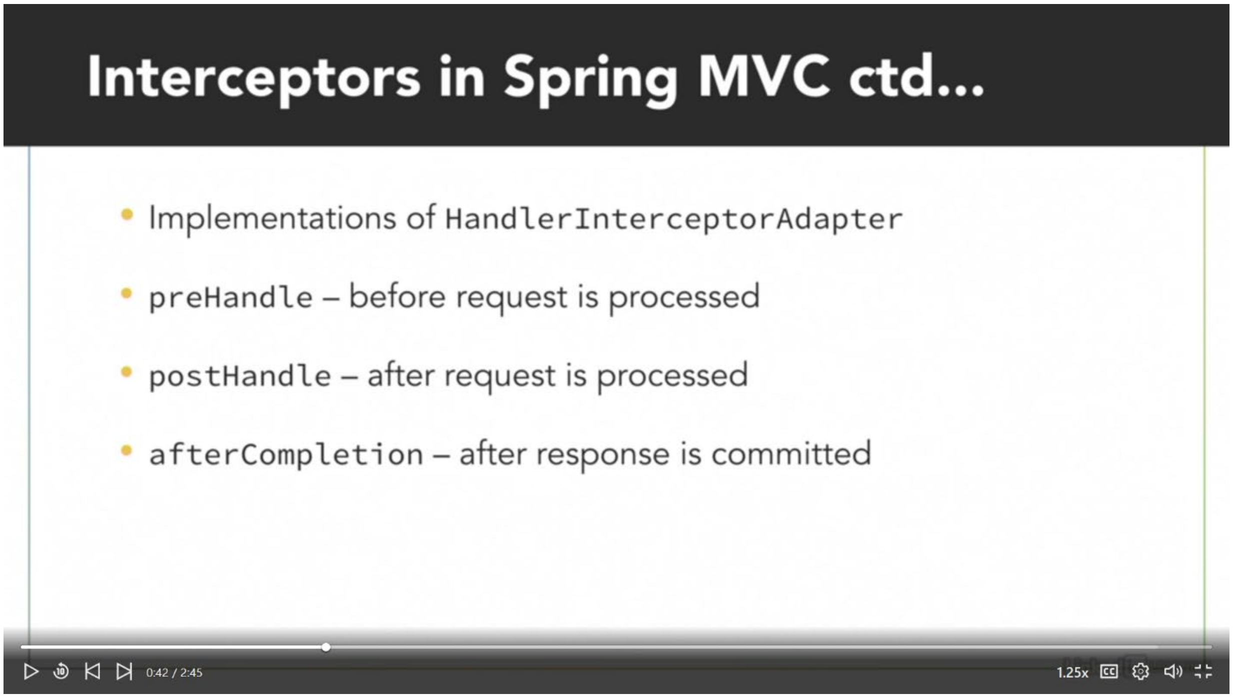
Interceptors and in Spring MVC



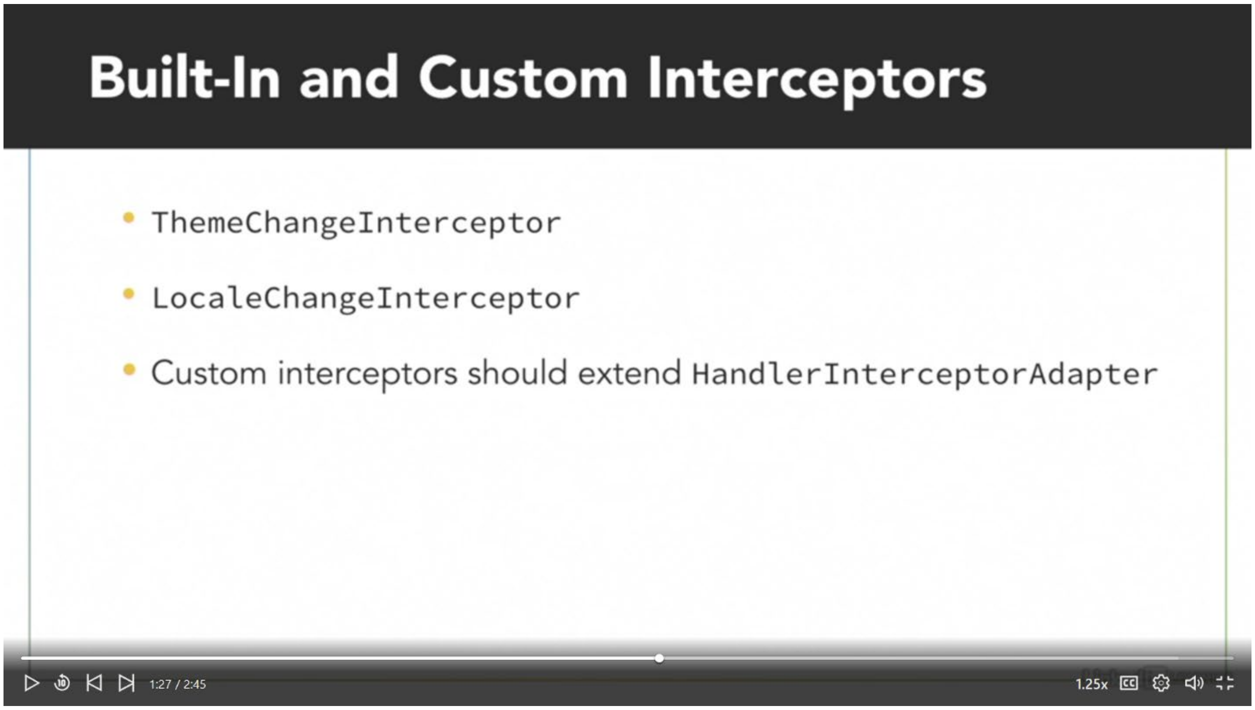
Interceptors in Spring MVC

- Similar to filters in Servlet API
- Used for pre- and post-processing of the request



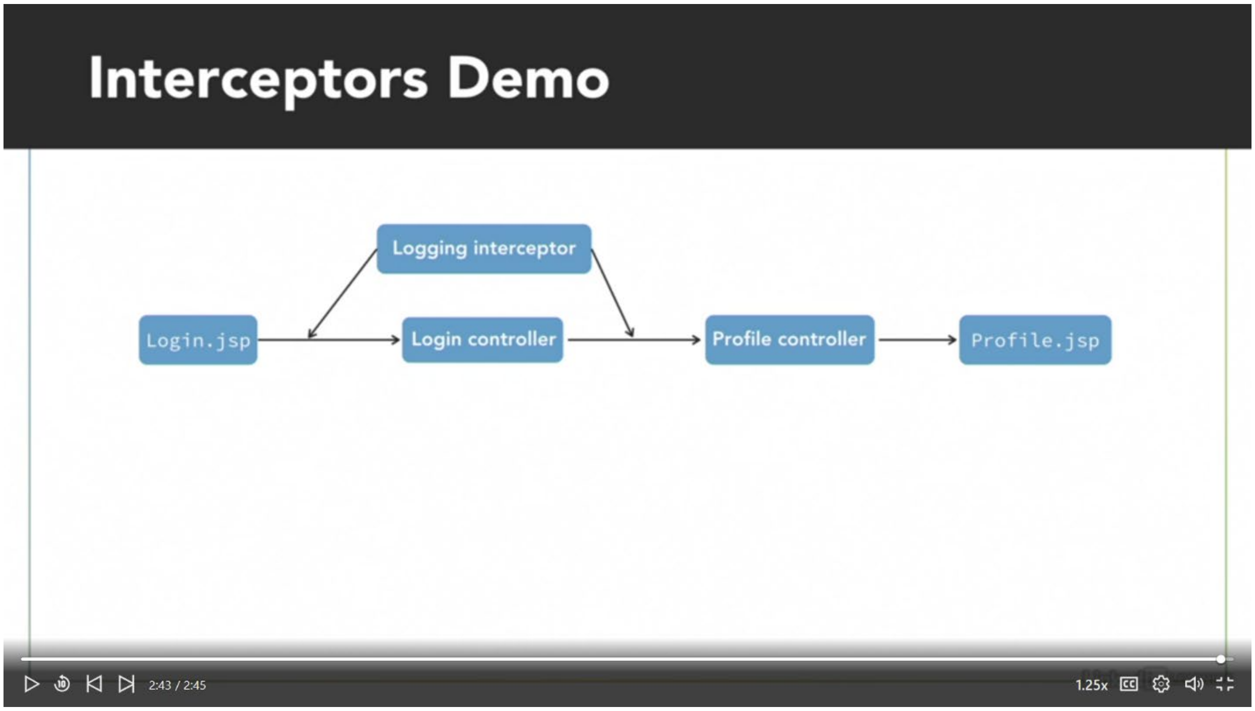
Interceptors in Spring MVC ctd...

- Implementations of `HandlerInterceptorAdapter`
- `preHandle` – before request is processed
- `postHandle` – after request is processed
- `afterCompletion` – after response is committed

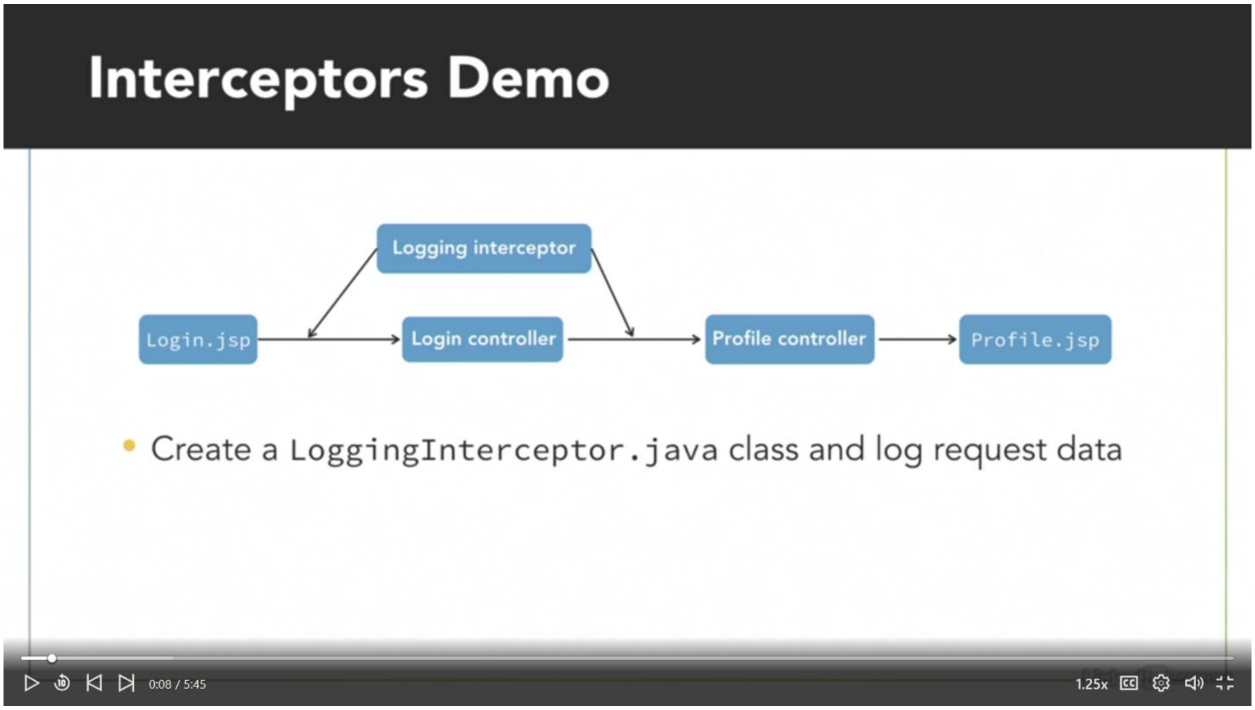


Built-In and Custom Interceptors

- ThemeChangeInterceptor
- LocaleChangeInterceptor
- Custom interceptors should extend HandlerInterceptorAdapter



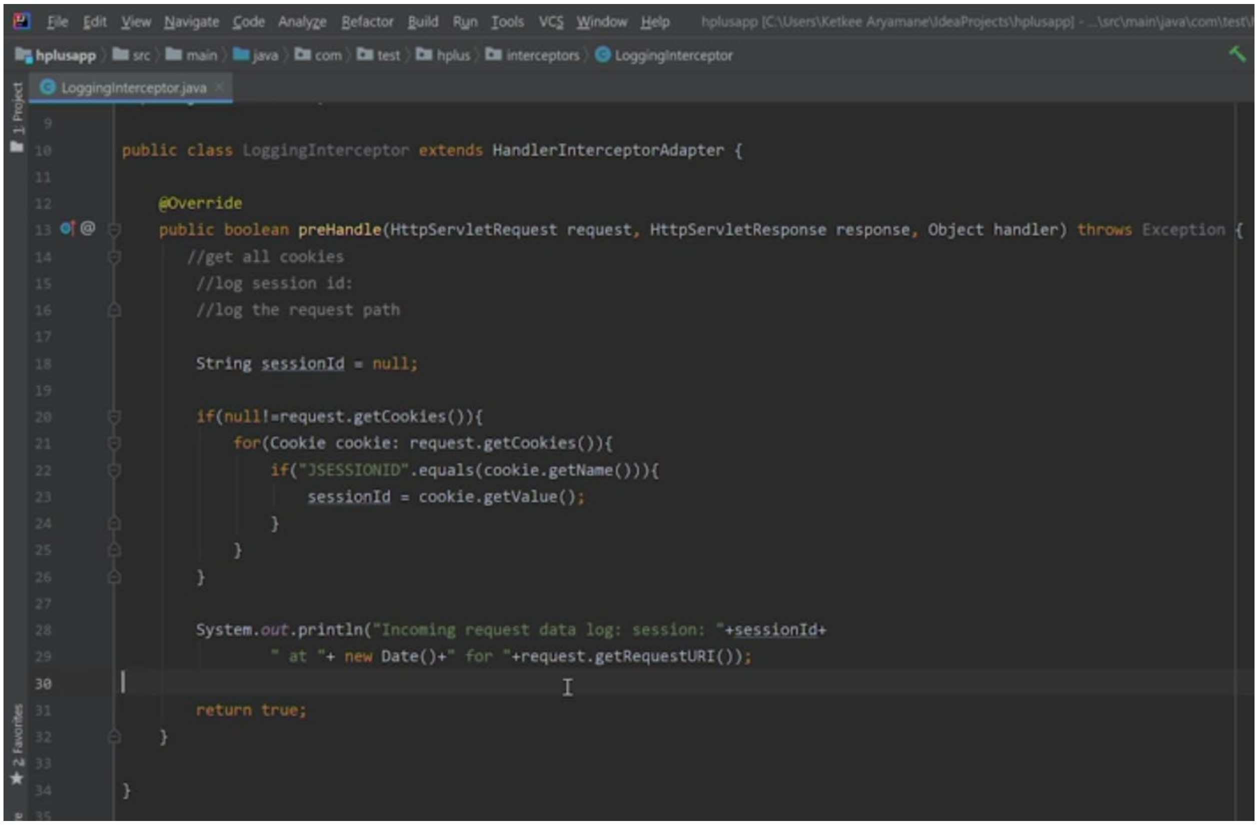
Interceptors Demo



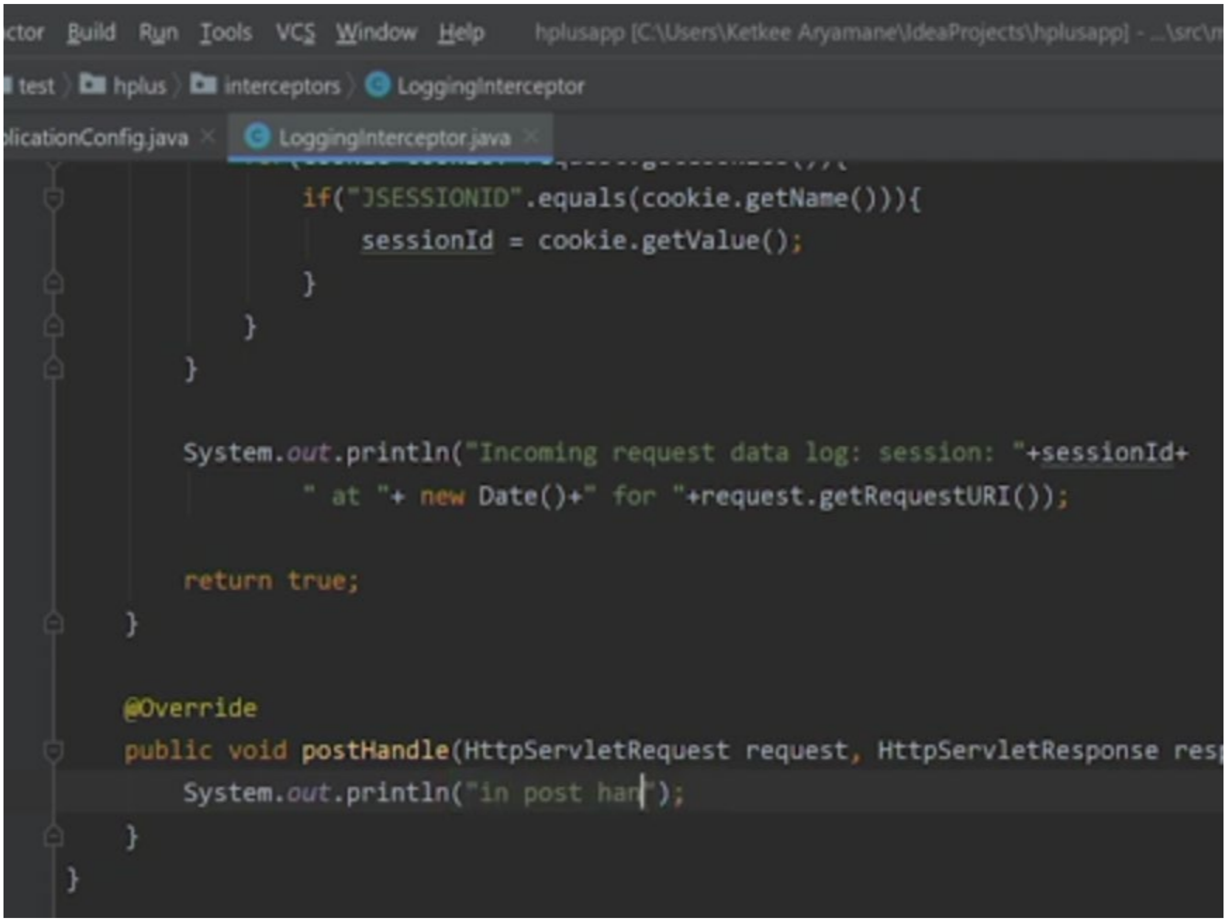
Interceptors Demo

- Create a `LoggingInterceptor.java` class and log request data

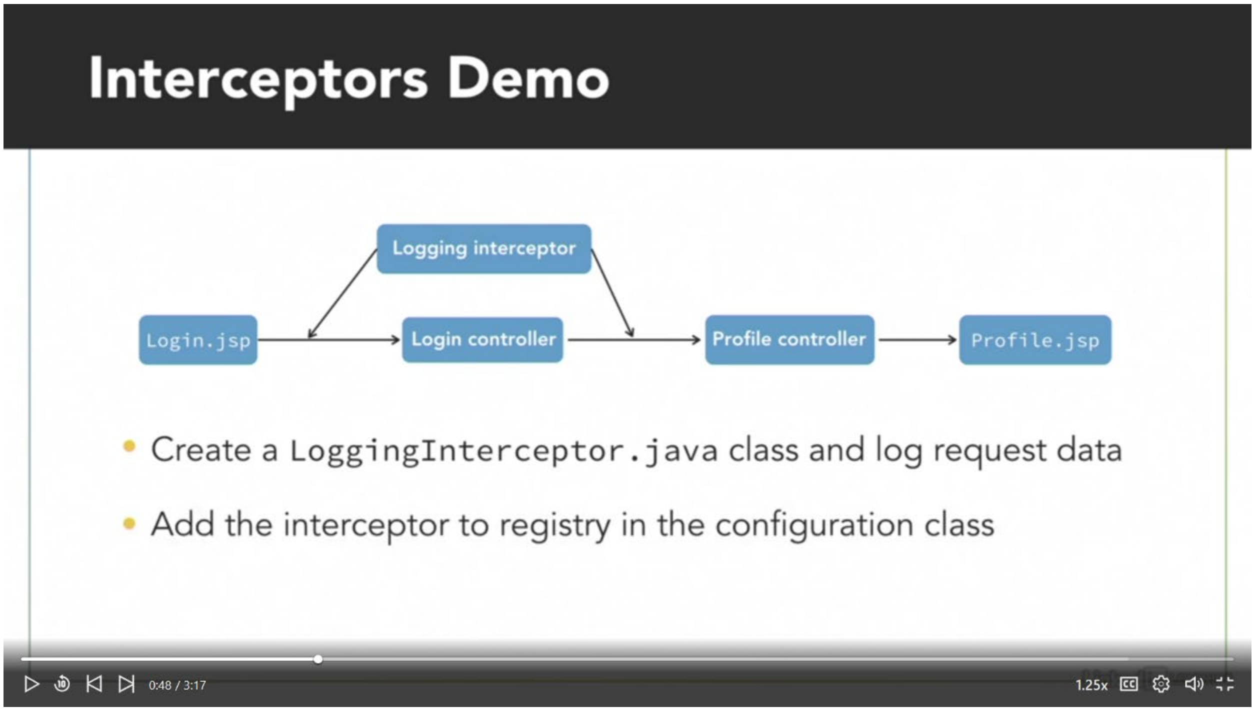
Create a LoggingInterceptor class that extends HandlerInterceptorAdapter and override its functions.


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
hplusapp [C:\Users\Ketkee Aryamane\IdeaProjects\hplusapp] - ...\src\main\java\com\test\
hplusapp
src
main
java
com
test
hplus
interceptors
LoggingInterceptor
LoggingInterceptor.java
1: Project
2: Favorites

```java
public class LoggingInterceptor extends HandlerInterceptorAdapter {

    @Override
    public boolean preHandle(HttpServletRequest request, HttpServletResponse response, Object handler) throws Exception {
        //get all cookies
        //log session id:
        //log the request path

        String sessionId = null;

        if(null!=request.getCookies()){
            for(Cookie cookie: request.getCookies()){
                if("JSESSIONID".equals(cookie.getName())){
                    sessionId = cookie.getValue();
                }
            }
        }

        System.out.println("Incoming request data log: session: "+sessionId+
                " at "+ new Date()+" for "+request.getRequestURI());

        return true;
    }

}
```


ctor Build Run Tools VCS Window Help
hplusapp [C:\Users\Ketkee Aryamane\IdeaProjects\hplusapp] - ...\src\m
test 〉 hplus 〉 interceptors 〉 LoggingInterceptor
plicationConfig.java
LoggingInterceptor.java

```
                if("JSESSIONID".equals(cookie.getName())){
                    sessionId = cookie.getValue();
                }
            }
        }

        System.out.println("Incoming request data log: session: "+sessionId+
                " at "+ new Date()+" for "+request.getRequestURI());

        return true;
    }

    @Override
    public void postHandle(HttpServletRequest request, HttpServletResponse res
        System.out.println("in post han");
    }
}
```



Interceptors Demo

Logging interceptor

Login.jsp

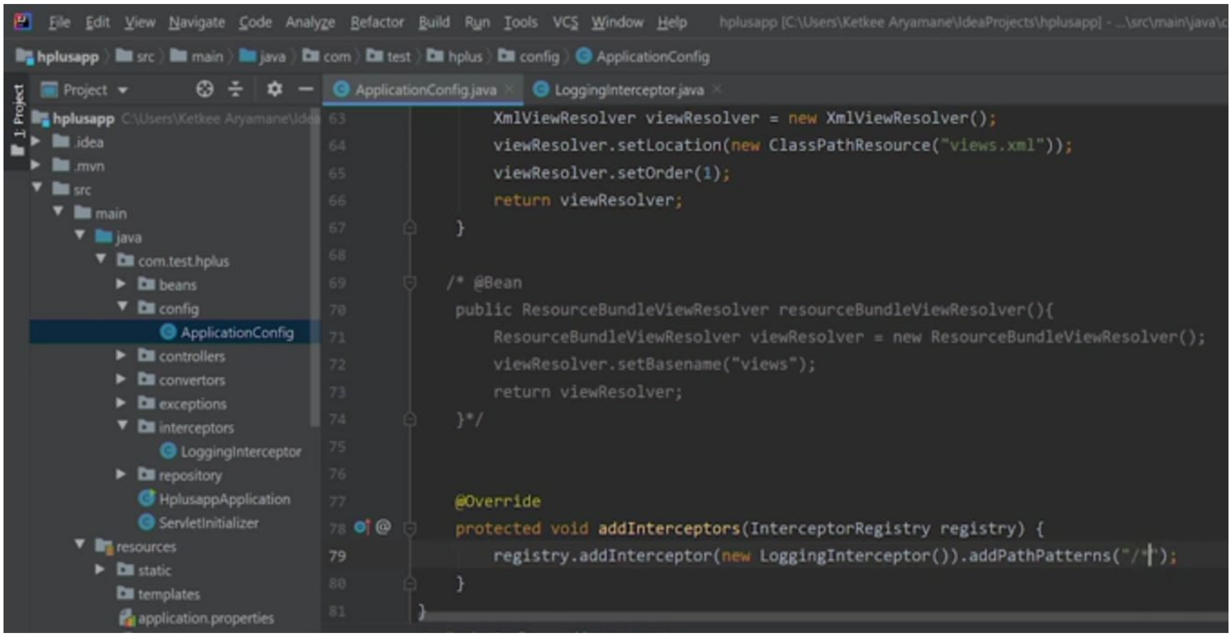
Login controller

Profile controller

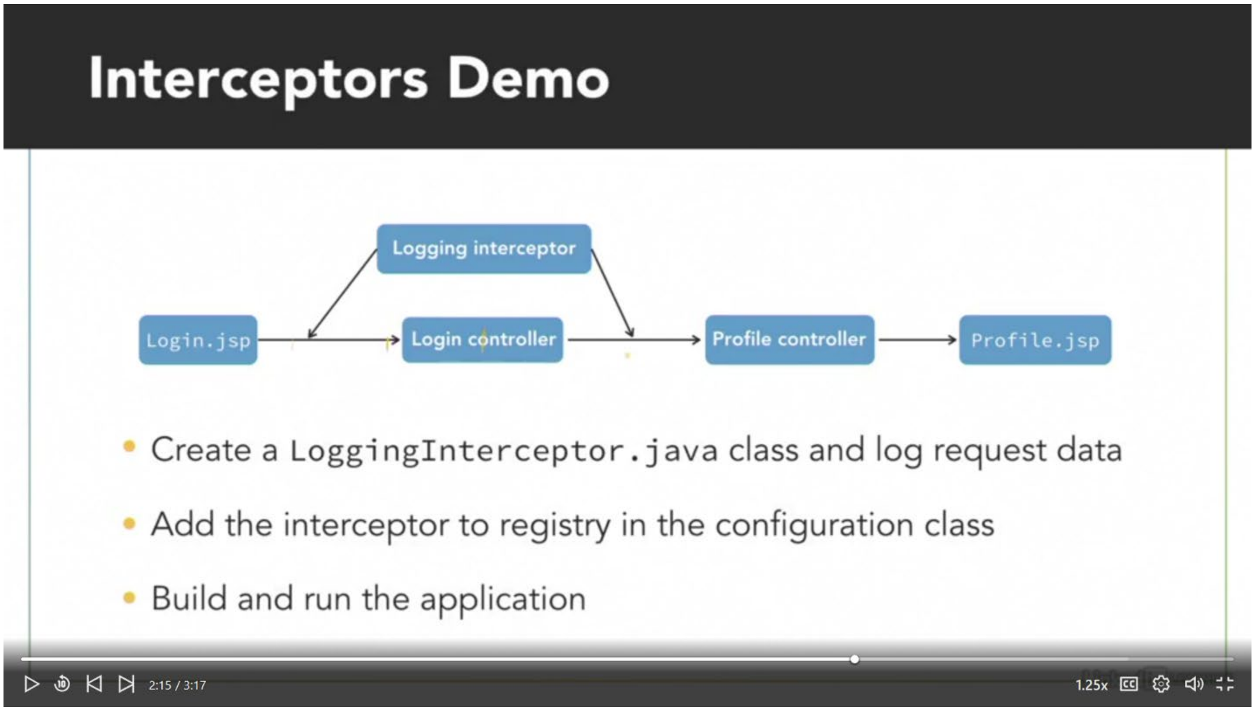
Profile.jsp

- Create a `LoggingInterceptor.java` class and log request data
- Add the interceptor to registry in the configuration class

Override the addInterceptor() method in ApplicationConfig.java class to register the newely created LoggingInterceptor with pathPattern to all urls /*


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
hplusapp [C:\Users\Ketkee Aryamane\IdeaProjects\hplusapp] - ...\src\main\java\c
hplusapp › src › main › java › com › test › hplus › config › ApplicationConfig
Project
ApplicationConfig.java
LoggingInterceptor.java
hplusapp C:\Users\Ketkee Aryamane\Idea
.idea
.mvn
src
main
java
com.test.hplus
beans
config
ApplicationConfig
controllers
convertors
exceptions
interceptors
LoggingInterceptor
repository
HplusappApplication
ServletInitializer
resources
static
templates
application.properties

```
        XmlViewResolver viewResolver = new XmlViewResolver();
        viewResolver.setLocation(new ClassPathResource("views.xml"));
        viewResolver.setOrder(1);
        return viewResolver;
    }

   /* @Bean
    public ResourceBundleViewResolver resourceBundleViewResolver(){
        ResourceBundleViewResolver viewResolver = new ResourceBundleViewResolver();
        viewResolver.setBasename("views");
        return viewResolver;
    }*/


    @Override
    protected void addInterceptors(InterceptorRegistry registry) {
        registry.addInterceptor(new LoggingInterceptor()).addPathPatterns("/*");
    }
}
```

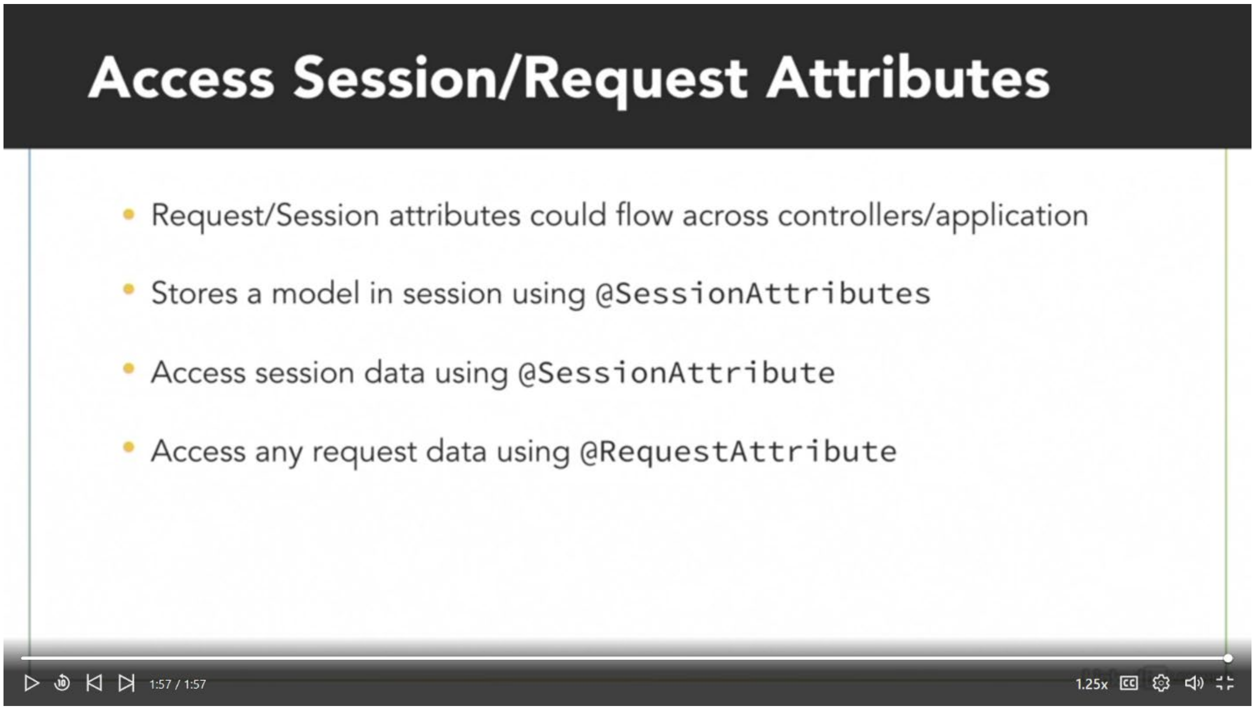


Interceptors Demo

Logging interceptor

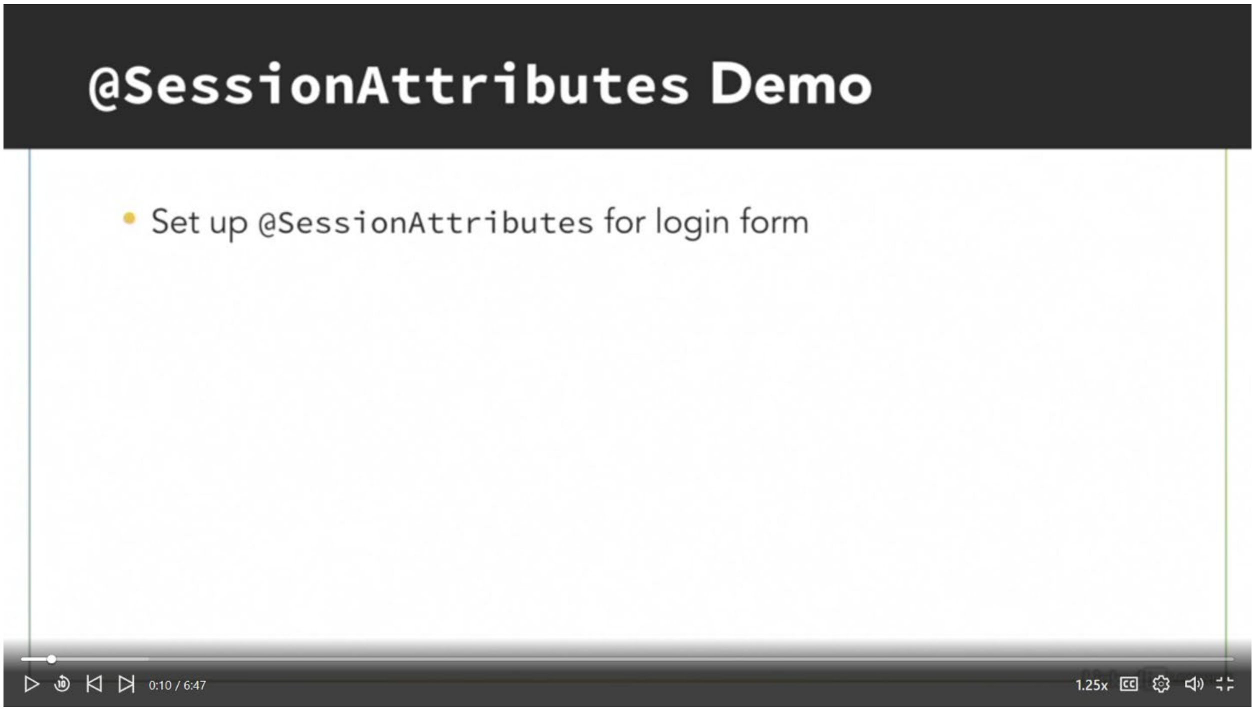
Login.jsp → Login controller → Profile controller → Profile.jsp

- Create a LoggingInterceptor.java class and log request data
- Add the interceptor to registry in the configuration class
- Build and run the application



Access Session/Request Attributes

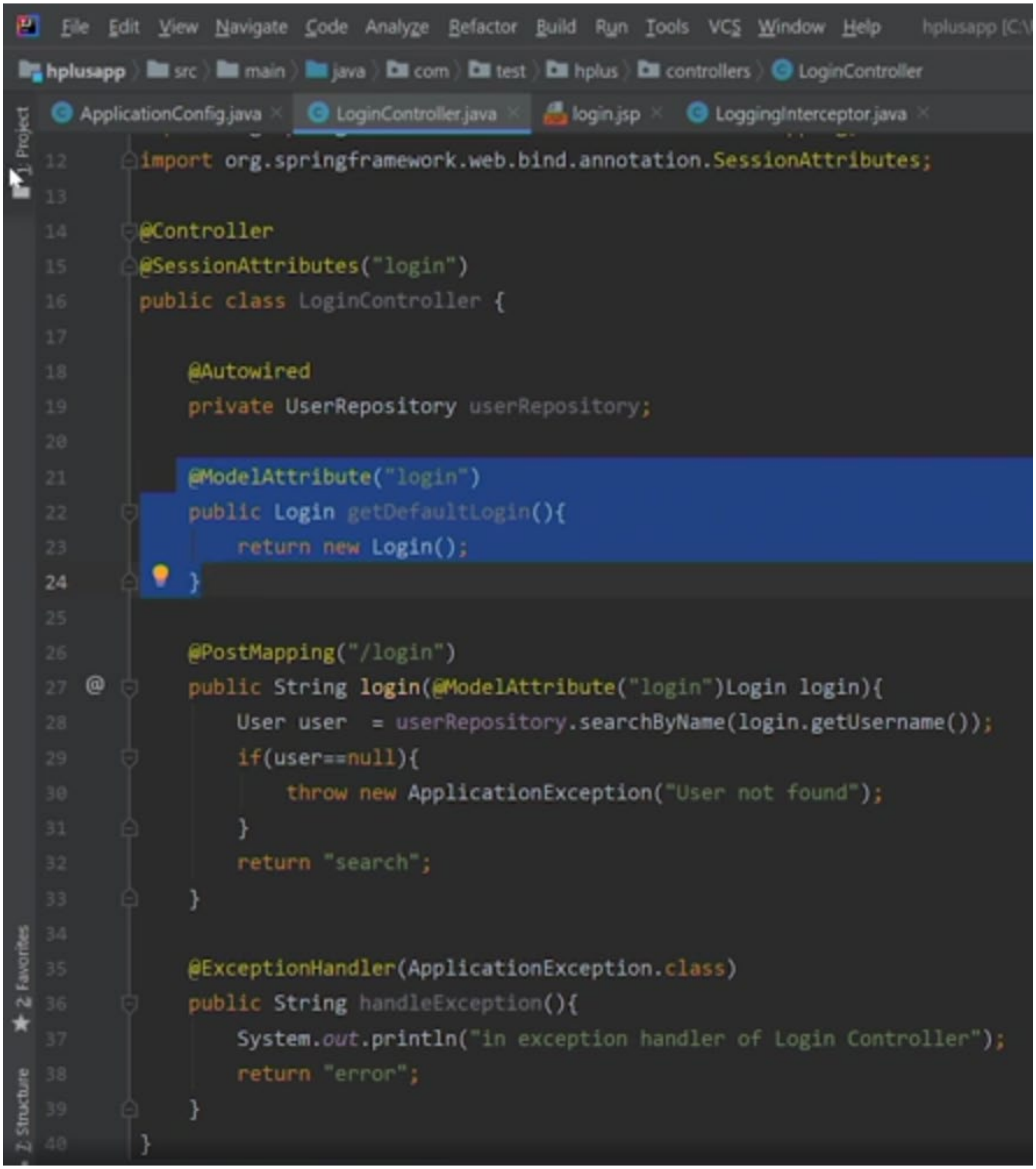
- Request/Session attributes could flow across controllers/application
- Stores a model in session using `@SessionAttributes`
- Access session data using `@SessionAttribute`
- Access any request data using `@RequestAttribute`



`@SessionAttributes` Demo

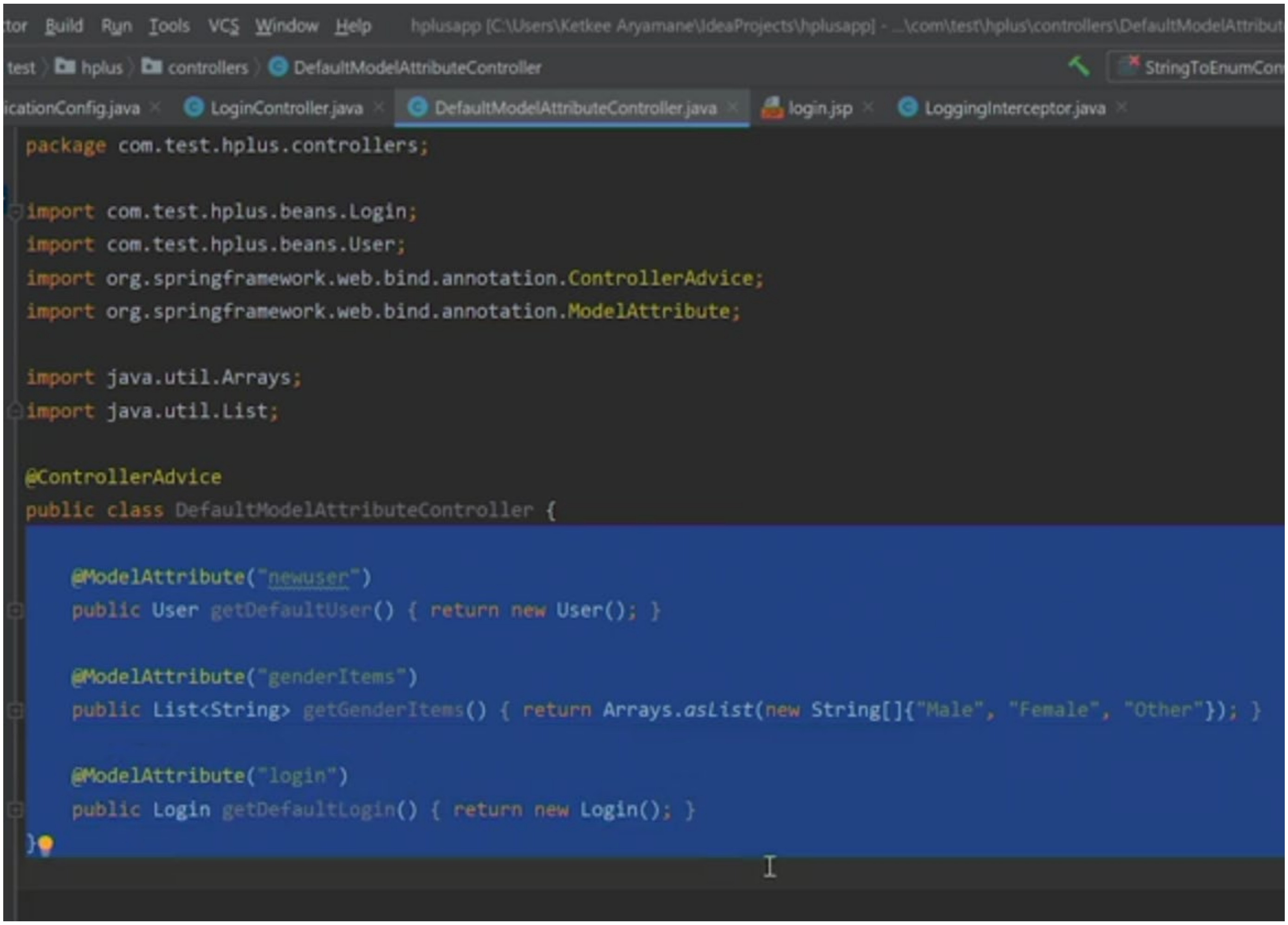
- Set up `@SessionAttributes` for login form

So for SessionAttribute to set values in a session Model, the Model should exists before it sets the values in a Model in SessionAttribute. So the controller should have a getDefaultLogin method to return that model.


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
hplusapp [C:\
hplusapp › src › main › java › com › test › hplus › controllers › LoginController
ApplicationConfig.java
LoginController.java
login.jsp
LoggingInterceptor.java
1: Project
2: Favorites
7: Structure

```java
import org.springframework.web.bind.annotation.SessionAttributes;

@Controller
@SessionAttributes("login")
public class LoginController {

    @Autowired
    private UserRepository userRepository;

    @ModelAttribute("login")
    public Login getDefaultLogin(){
        return new Login();
    }

    @PostMapping("/login")
    public String login(@ModelAttribute("login")Login login){
        User user  = userRepository.searchByName(login.getUsername());
        if(user==null){
            throw new ApplicationException("User not found");
        }
        return "search";
    }

    @ExceptionHandler(ApplicationException.class)
    public String handleException(){
        System.out.println("in exception handler of Login Controller");
        return "error";
    }
}
```

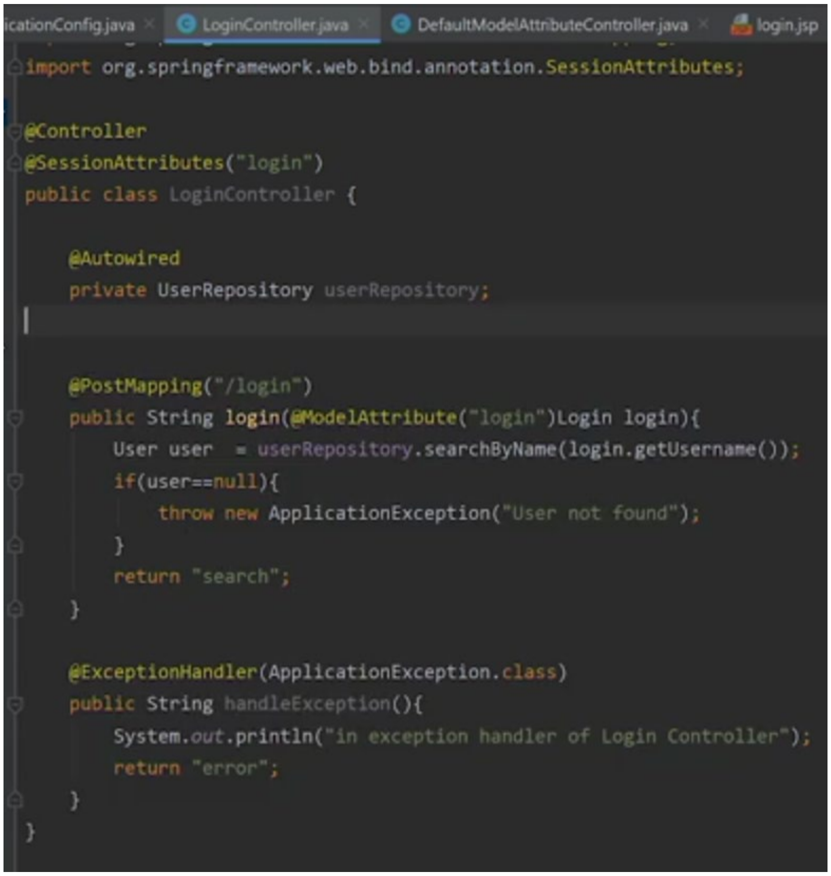
However we have already have these defaults model in one of our ControllerAdvice that is DefaultModelAttribute so our Loginontroller do not need to reimplement the login ModelAttribute.



tor Build Run Tools VCS Window Help hplusapp [C:\Users\Ketkee Aryamane\IdeaProjects\hplusapp] - ...\com\test\hplus\controllers\DefaultModelAttribut
test hplus controllers DefaultModelAttributeController
StringToEnumCon
icationConfig.java LoginController.java DefaultModelAttributeController.java login.jsp LoggingInterceptor.java

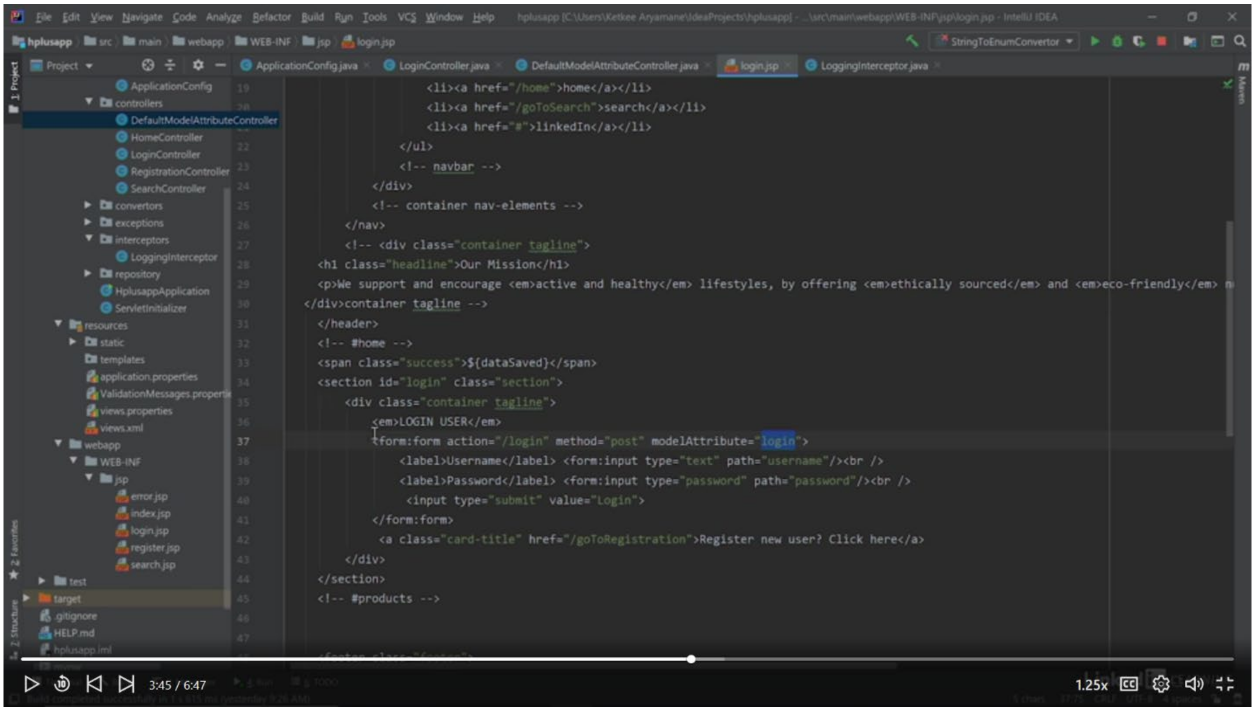
```java
package com.test.hplus.controllers;

import com.test.hplus.beans.Login;
import com.test.hplus.beans.User;
import org.springframework.web.bind.annotation.ControllerAdvice;
import org.springframework.web.bind.annotation.ModelAttribute;

import java.util.Arrays;
import java.util.List;

@ControllerAdvice
public class DefaultModelAttributeController {

    @ModelAttribute("newuser")
    public User getDefaultUser() { return new User(); }

    @ModelAttribute("genderItems")
    public List<String> getGenderItems() { return Arrays.asList(new String[]{"Male", "Female", "Other"}); }

    @ModelAttribute("login")
    public Login getDefaultLogin() { return new Login(); }
}
```

So no need to reimplement LoginAttribute in LoginController

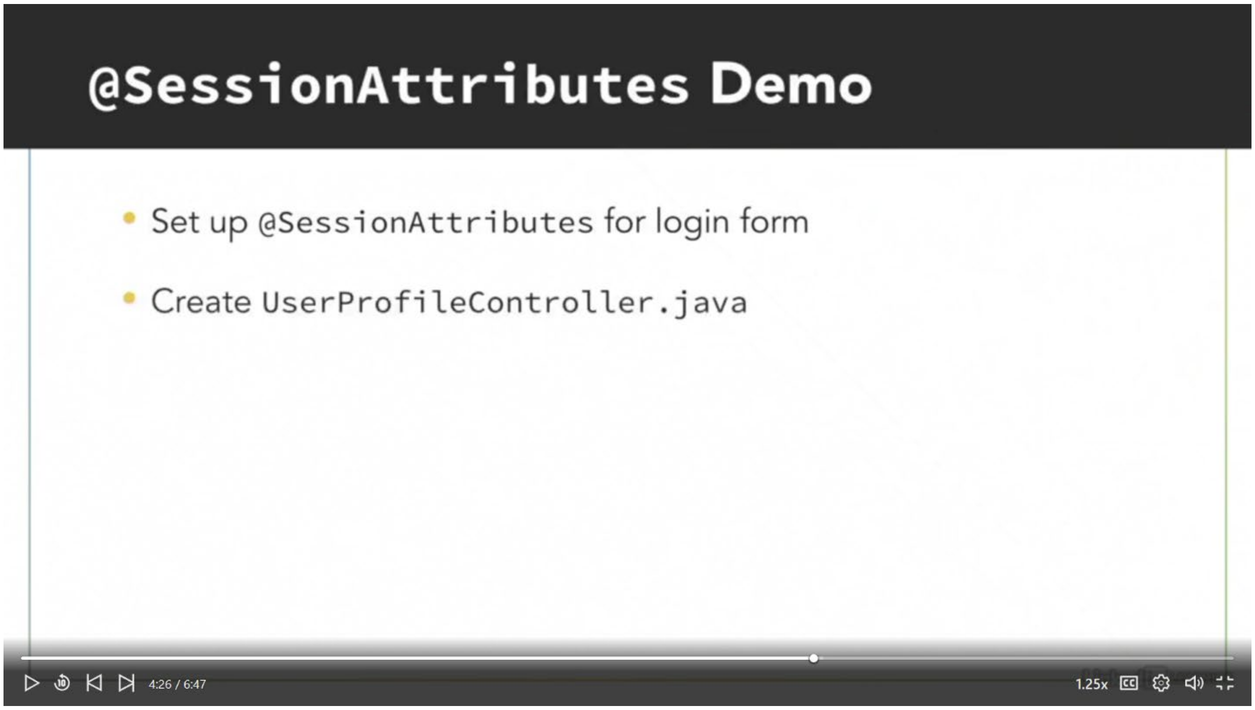


```java
import org.springframework.web.bind.annotation.SessionAttributes;

@Controller
@SessionAttributes("login")
public class LoginController {

    @Autowired
    private UserRepository userRepository;


    @PostMapping("/login")
    public String login(@ModelAttribute("login")Login login){
        User user  = userRepository.searchByName(login.getUsername());
        if(user==null){
            throw new ApplicationException("User not found");
        }
        return "search";
    }

    @ExceptionHandler(ApplicationException.class)
    public String handleException(){
        System.out.println("in exception handler of Login Controller");
        return "error";
    }
}
```

Get the login ModelAttribute from the login.jsp page

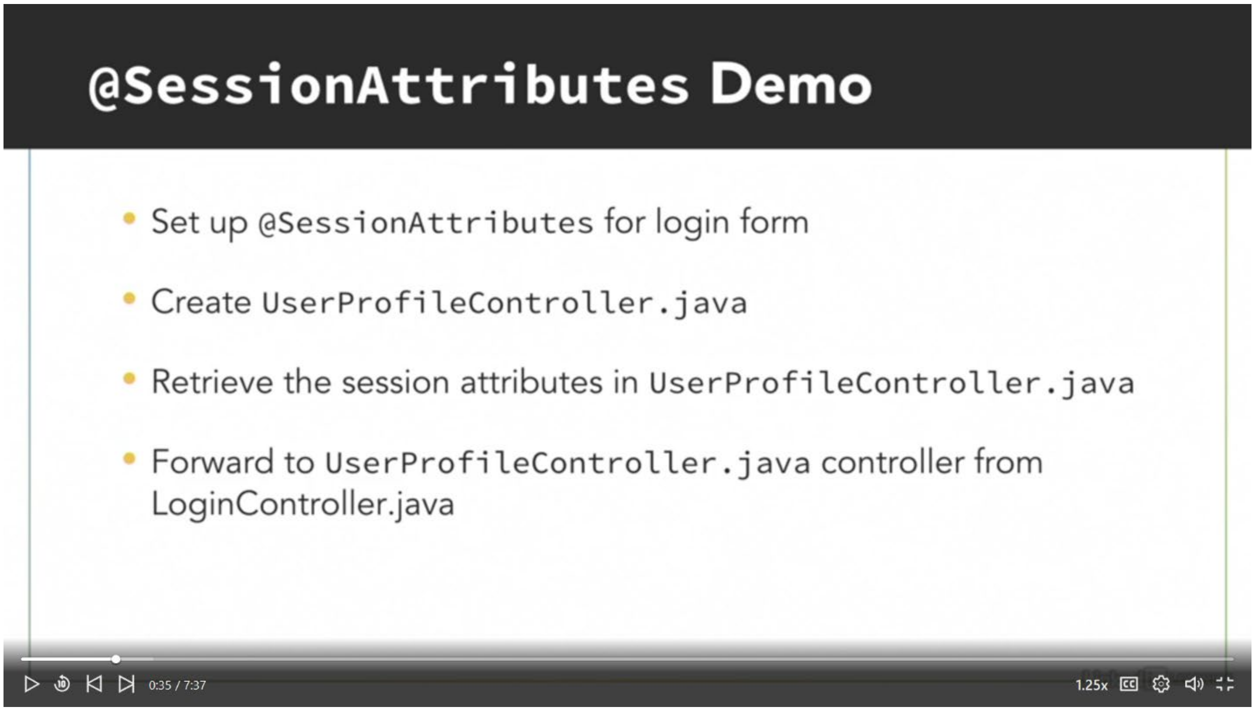


```html
                    <li><a href="/home">home</a></li>
                    <li><a href="/goToSearch">search</a></li>
                    <li><a href="#">linkedIn</a></li>
                </ul>
                <!-- navbar -->
            </div>
            <!-- container nav-elements -->
        </nav>
        <!-- <div class="container tagline">
    <h1 class="headline">Our Mission</h1>
    <p>We support and encourage <em>active and healthy</em> lifestyles, by offering <em>ethically sourced</em> and <em>eco-friendly</em> n
  </div>container tagline -->
    </header>
    <!-- #home -->
    <span class="success">${dataSaved}</span>
    <section id="login" class="section">
        <div class="container tagline">
            <em>LOGIN USER</em>
            <form:form action="/login" method="post" modelAttribute="login">
                <label>Username</label> <form:input type="text" path="username"/><br />
                <label>Password</label> <form:input type="password" path="password"/><br />
                 <input type="submit" value="Login">
            </form:form>
             <a class="card-title" href="/goToRegistration">Register new user? Click here</a>
        </div>
    </section>
    <!-- #products -->
```



@SessionAttributes Demo

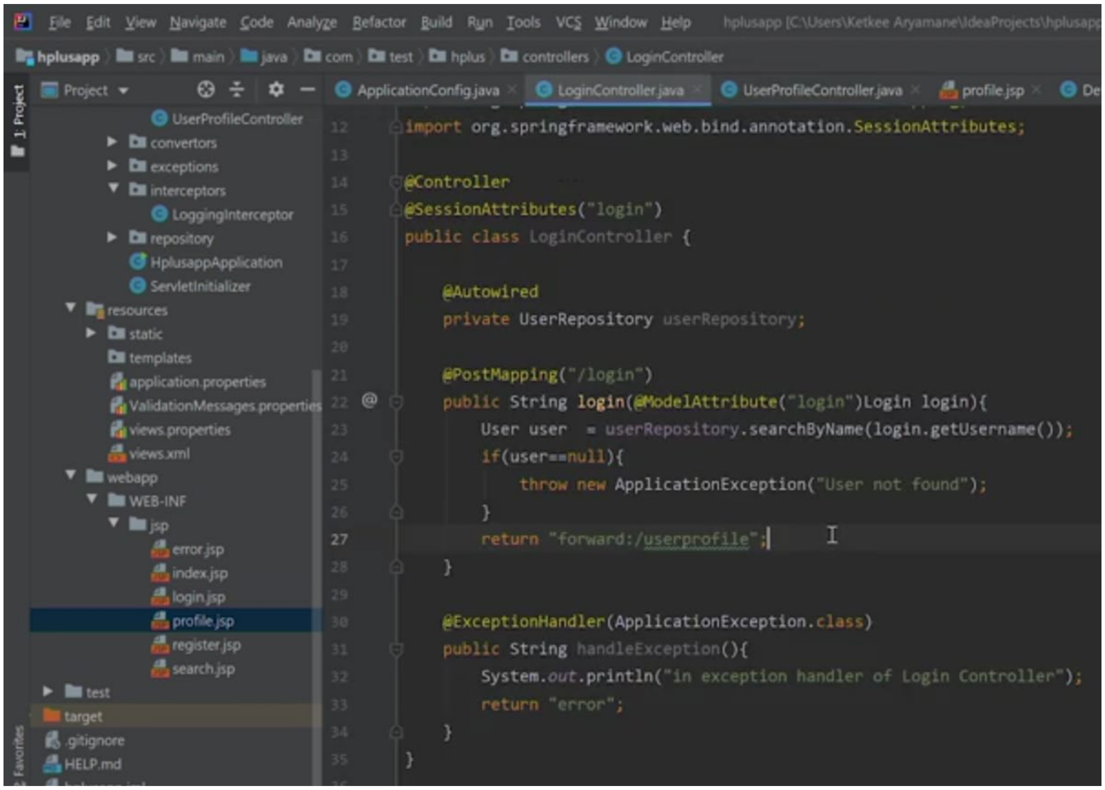
- Set up @SessionAttributes for login form
- Create UserProfileController.java



@SessionAttributes Demo

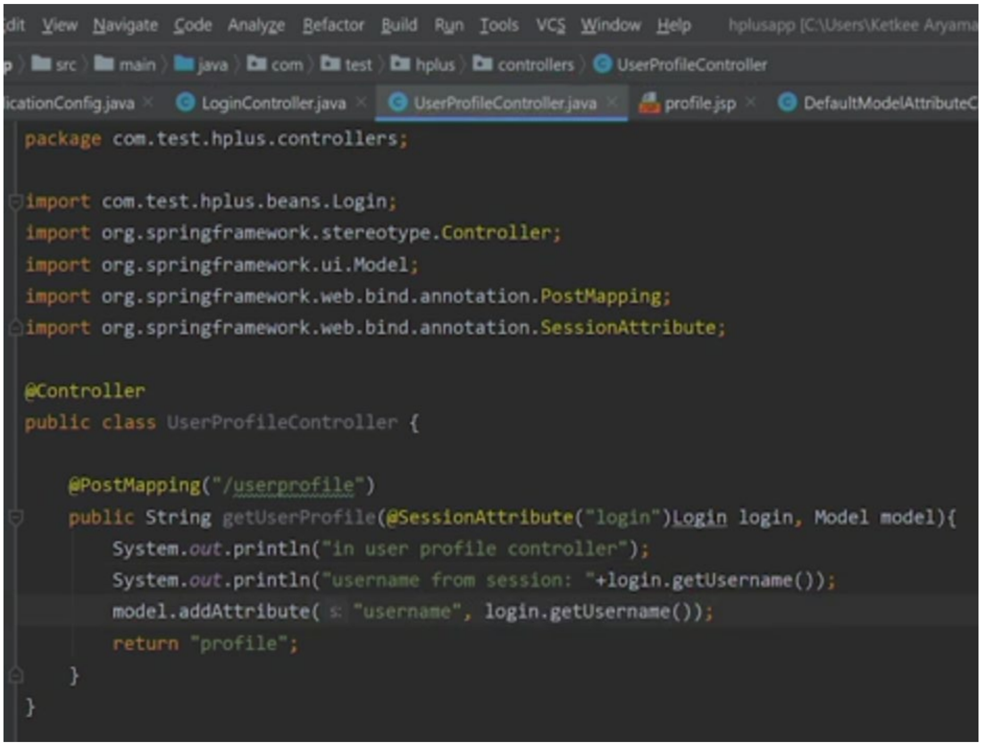
- Set up @SessionAttributes for login form
- Create UserProfileController.java
- Retrieve the session attributes in UserProfileController.java
- Forward to UserProfileController.java controller from LoginController.java

Set the login ModelAttribute on SessionAttribute and forward the controller request to user profile controller from LoginController



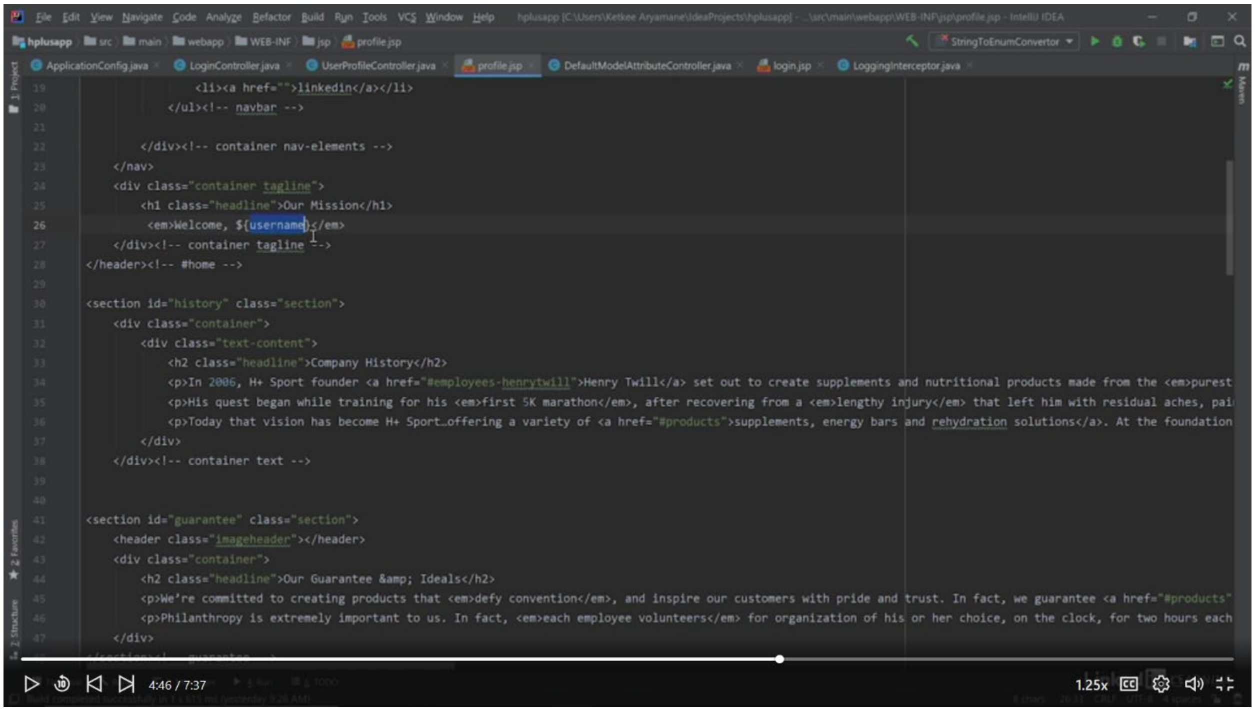
```java
import org.springframework.web.bind.annotation.SessionAttributes;

@Controller
@SessionAttributes("login")
public class LoginController {

    @Autowired
    private UserRepository userRepository;

    @PostMapping("/login")
    public String login(@ModelAttribute("login")Login login){
        User user  = userRepository.searchByName(login.getUsername());
        if(user==null){
            throw new ApplicationException("User not found");
        }
        return "forward:/userprofile";
    }

    @ExceptionHandler(ApplicationException.class)
    public String handleException(){
        System.out.println("in exception handler of Login Controller");
        return "error";
    }
}
```

Create a UserProfileController and access username from the SessionAttribute

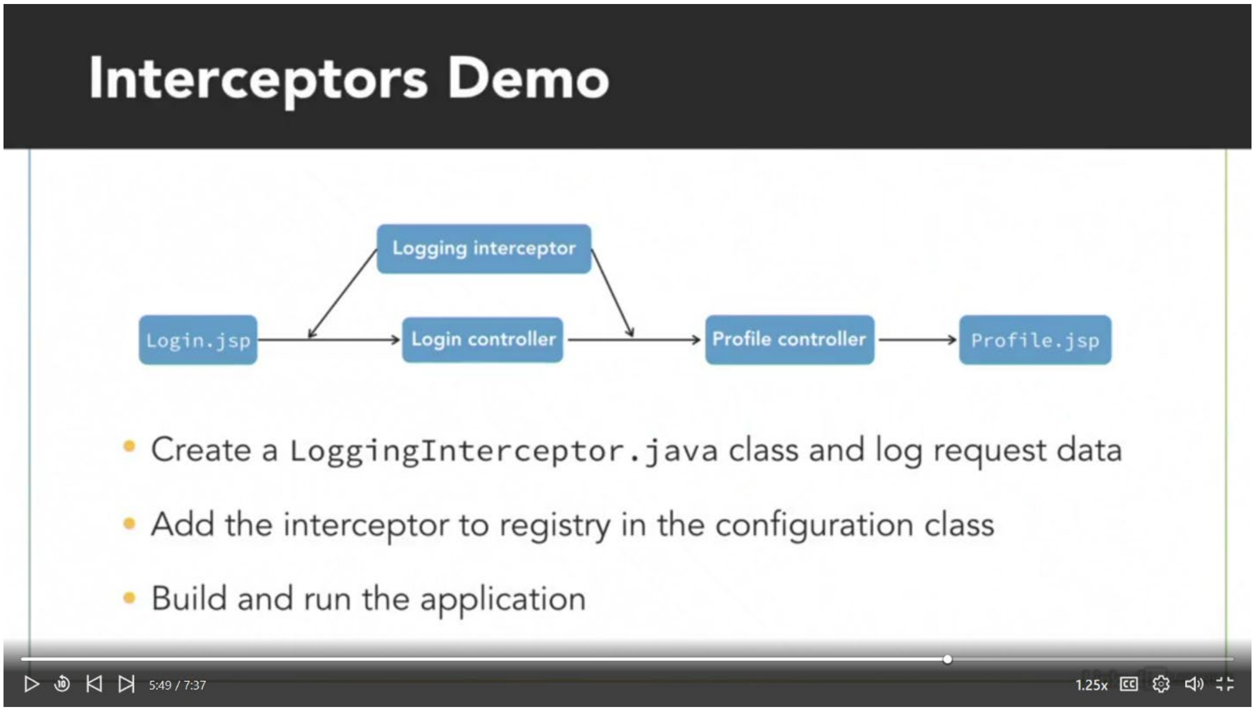

dit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
hplusapp [C:\Users\Ketkee Aryama
p 〉 src 〉 main 〉 java 〉 com 〉 test 〉 hplus 〉 controllers 〉 UserProfileController
icationConfig.java
LoginController.java
UserProfileController.java
profile.jsp
DefaultModelAttributeC

```java
package com.test.hplus.controllers;

import com.test.hplus.beans.Login;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.SessionAttribute;

@Controller
public class UserProfileController {

    @PostMapping("/userprofile")
    public String getUserProfile(@SessionAttribute("login")Login login, Model model){
        System.out.println("in user profile controller");
        System.out.println("username from session: "+login.getUsername());
        model.addAttribute( s: "username", login.getUsername());
        return "profile";
    }
}
```

Create a userProfile.jsp and access the username from the stored in ModelAttribute that is set via the SessionAttribute in UserProfileController.

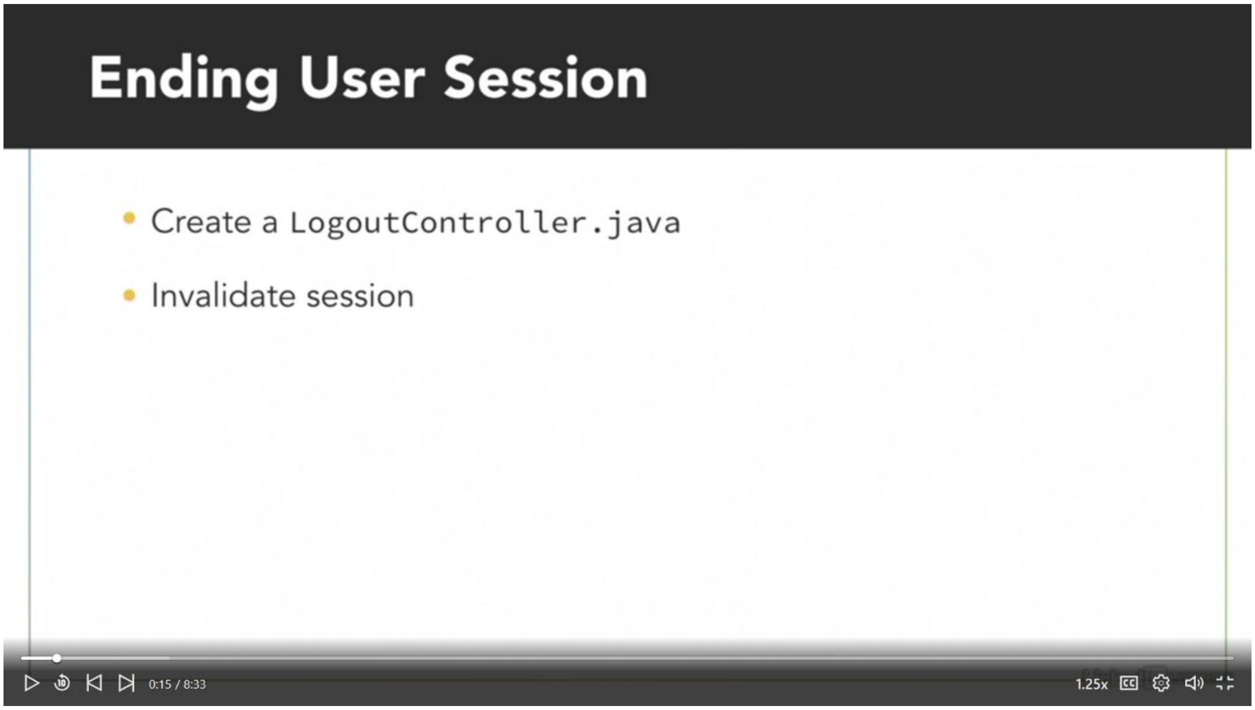

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
hplusapp [C:\Users\Ketkee Aryamane\IdeaProjects\hplusapp] - ...\src\main\webapp\WEB-INF\jsp\profile.jsp - IntelliJ IDEA
hplusapp › src › main › webapp › WEB-INF › jsp › profile.jsp
StringToEnumConvertor
ApplicationConfig.java
LoginController.java
UserProfileController.java
profile.jsp
DefaultModelAttributeController.java
login.jsp
LoggingInterceptor.java
4:46 / 7:37
1.25x

```
                <li><a href="">linkedin</a></li>
            </ul><!-- navbar -->

        </div><!-- container nav-elements -->
    </nav>
    <div class="container tagline">
        <h1 class="headline">Our Mission</h1>
         <em>Welcome, ${username}</em>
    </div><!-- container tagline -->
</header><!-- #home -->

<section id="history" class="section">
    <div class="container">
        <div class="text-content">
            <h2 class="headline">Company History</h2>
            <p>In 2006, H+ Sport founder <a href="#employees-henrytwill">Henry Twill</a> set out to create supplements and nutritional products made from the <em>purest
            <p>His quest began while training for his <em>first 5K marathon</em>, after recovering from a <em>lengthy injury</em> that left him with residual aches, pai
            <p>Today that vision has become H+ Sport…offering a variety of <a href="#products">supplements, energy bars and rehydration solutions</a>. At the foundation
        </div>
    </div><!-- container text -->


<section id="guarantee" class="section">
    <header class="imageheader"></header>
    <div class="container">
        <h2 class="headline">Our Guarantee &amp; Ideals</h2>
        <p>We're committed to creating products that <em>defy convention</em>, and inspire our customers with pride and trust. In fact, we guarantee <a href="#products"
        <p>Philanthropy is extremely important to us. In fact, <em>each employee volunteers</em> for organization of his or her choice, on the clock, for two hours each
    </div>
```



Interceptors Demo

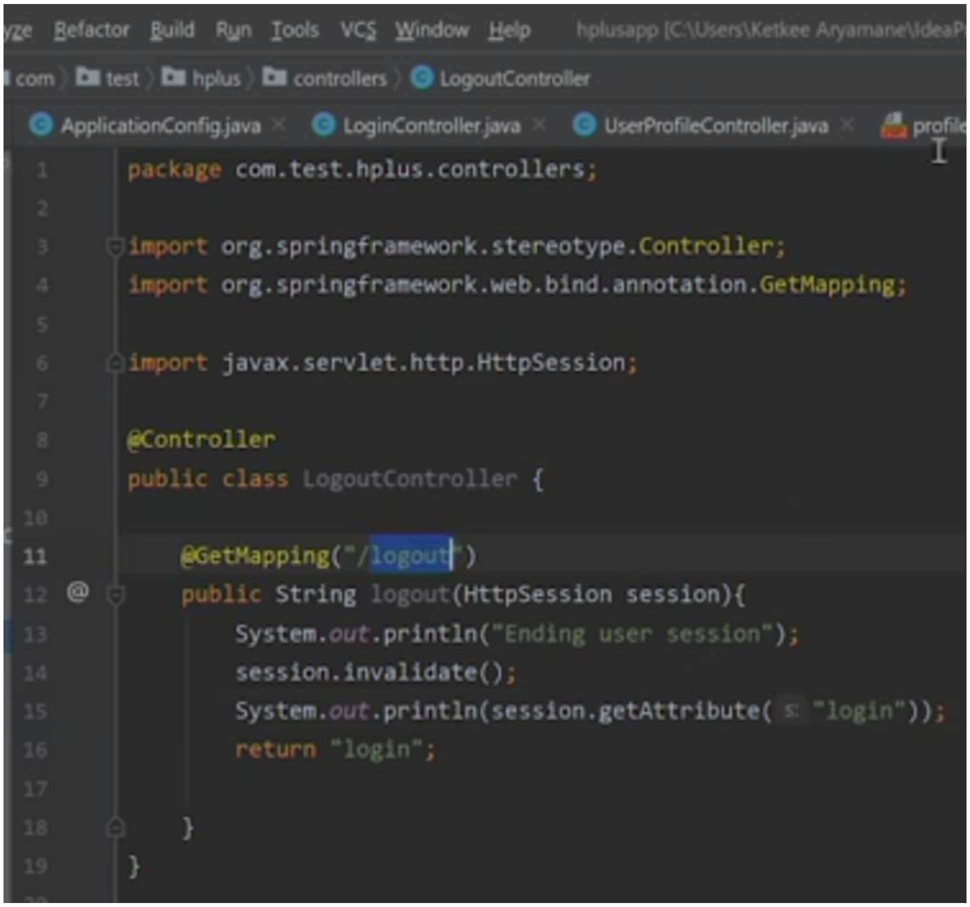
- Create a LoggingInterceptor.java class and log request data
- Add the interceptor to registry in the configuration class
- Build and run the application



Ending User Session

- Create a `LogoutController.java`
- Invalidate session

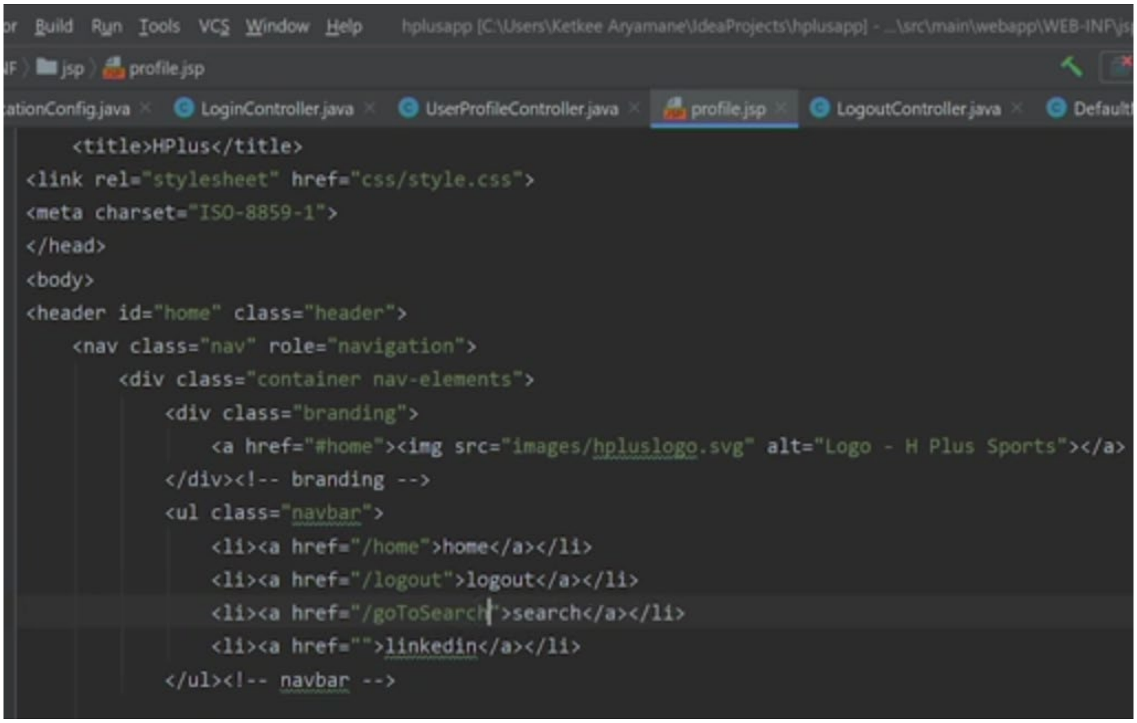
Create a LogOutController and create a @GetMapping(“logout”) and in the logout() method invalidate the session as session.invalidate();



yze Refactor Build Run Tools VCS Window Help hplusapp [C:\Users\Ketkee Aryamane\IdeaP
com 〉 test 〉 hplus 〉 controllers 〉 LogoutController
ApplicationConfig.java × LoginController.java × UserProfileController.java × profile

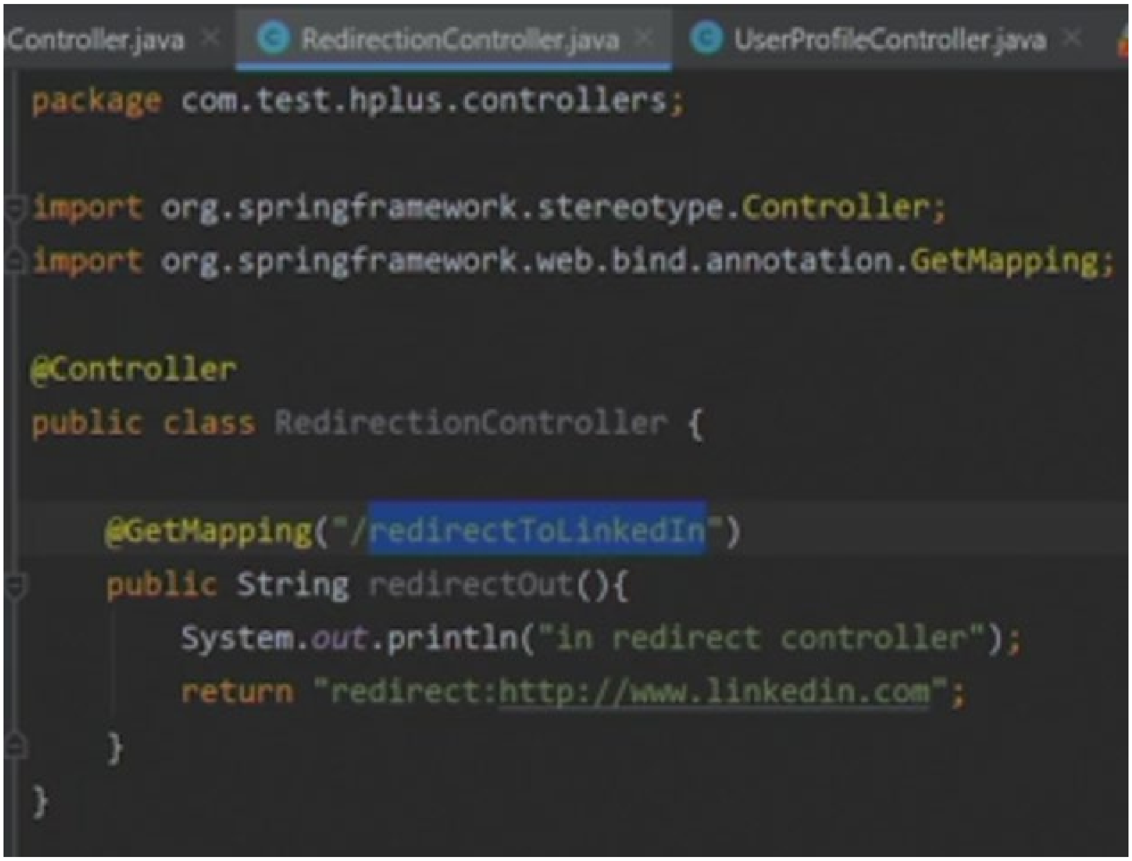
```java
package com.test.hplus.controllers;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;

import javax.servlet.http.HttpSession;

@Controller
public class LogoutController {

    @GetMapping("/logout")
    public String logout(HttpSession session){
        System.out.println("Ending user session");
        session.invalidate();
        System.out.println(session.getAttribute( s: "login"));
        return "login";

    }
}
```

Inside userprofile.jsp just have the link of logout to /logout


or Build Run Tools VCS Window Help
hplusapp [C:\Users\Ketkee Aryamane\IdeaProjects\hplusapp] - ...\src\main\webapp\WEB-INF\js
IF jsp profile.jsp
ationConfig.java
LoginController.java
UserProfileController.java
profile.jsp
LogoutController.java
Default

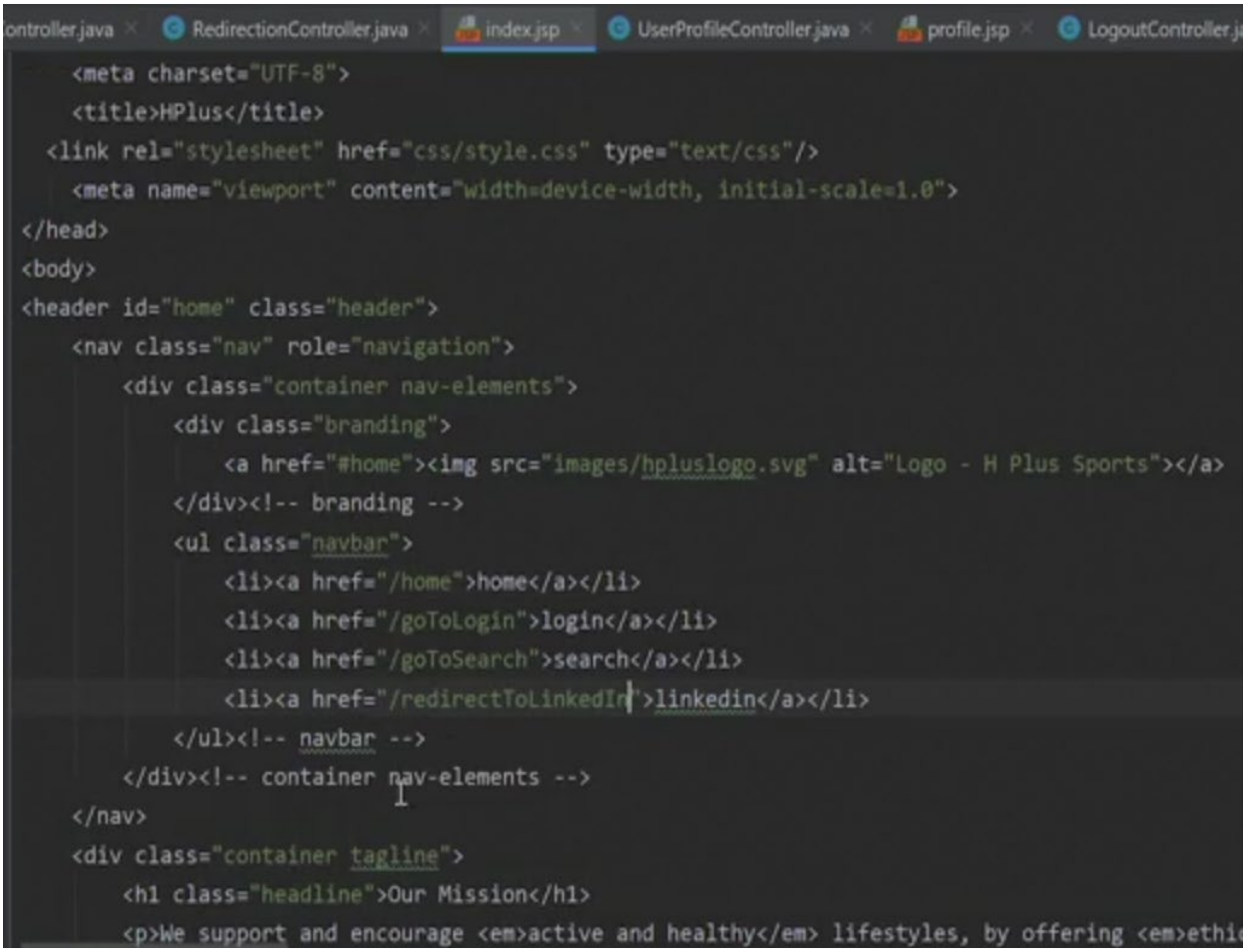
```
    <title>HPlus</title>
<link rel="stylesheet" href="css/style.css">
<meta charset="ISO-8859-1">
</head>
<body>
<header id="home" class="header">
    <nav class="nav" role="navigation">
        <div class="container nav-elements">
            <div class="branding">
                <a href="#home"><img src="images/hpluslogo.svg" alt="Logo - H Plus Sports"></a>
            </div><!-- branding -->
            <ul class="navbar">
                <li><a href="/home">home</a></li>
                <li><a href="/logout">logout</a></li>
                <li><a href="/goToSearch">search</a></li>
                <li><a href="">linkedin</a></li>
            </ul><!-- navbar -->
```

Redirecting outside of an application : Create a RedirectionController for it



```
package com.test.hplus.controllers;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;

@Controller
public class RedirectionController {

    @GetMapping("/redirectToLinkedIn")
    public String redirectOut(){
        System.out.println("in redirect controller");
        return "redirect:http://www.linkedin.com";
    }
}
```

Now for all the linekedIn urls map the url in jsps



```
    <meta charset="UTF-8">
    <title>HPlus</title>
  <link rel="stylesheet" href="css/style.css" type="text/css"/>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>
<header id="home" class="header">
    <nav class="nav" role="navigation">
        <div class="container nav-elements">
            <div class="branding">
                <a href="#home"><img src="images/hpluslogo.svg" alt="Logo - H Plus Sports"></a>
            </div><!-- branding -->
            <ul class="navbar">
                <li><a href="/home">home</a></li>
                <li><a href="/goToLogin">login</a></li>
                <li><a href="/goToSearch">search</a></li>
                <li><a href="/redirectToLinkedIn">linkedin</a></li>
            </ul><!-- navbar -->
        </div><!-- container nav-elements -->
    </nav>
    <div class="container tagline">
        <h1 class="headline">Our Mission</h1>
        <p>We support and encourage <em>active and healthy</em> lifestyles, by offering <em>ethi
```

Now run your application .