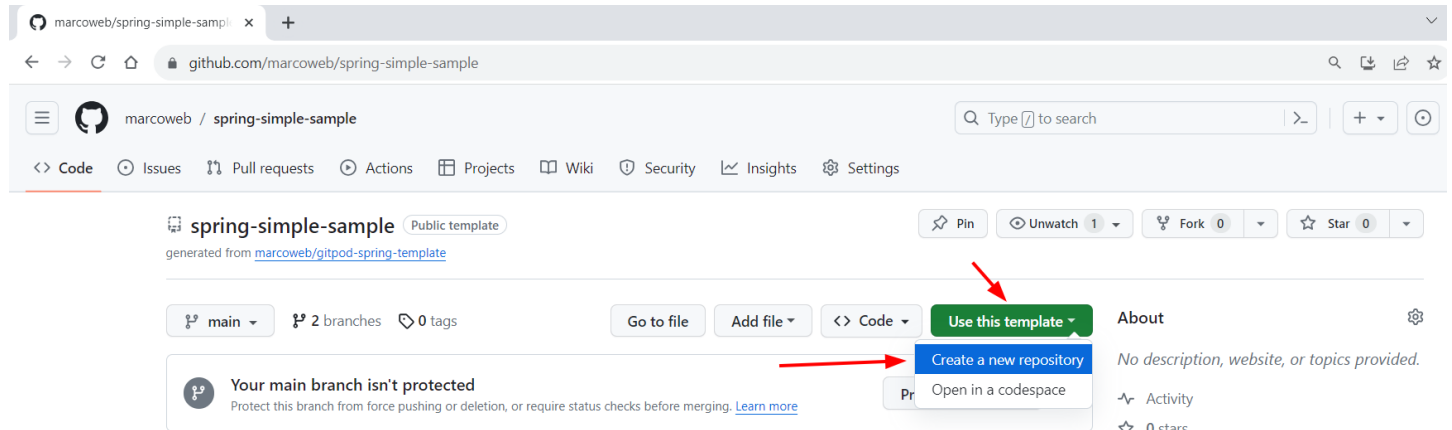


# Autenticação Simples utilizando o Spring Security

Crie uma nova aplicação com base no modelo <https://github.com/marcoweb/spring-simple-sample>:




Nomeie o repositório à ser criado:

## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

*Required fields are marked with an asterisk (\*).*

### Repository template


 **marcoweb/spring-simple-sample** ▾

Start your repository with a template repository's contents.

☐ **Include all branches**

Copy all branches from marcoweb/spring-simple-sample and not just the default branch.

Owner \*

 **marcoweb** ▾

Repository name \*

/ **security-tutorial**

✓ security-tutorial is available.

Great repository names are short and memorable. Need inspiration? How about [curly-octo-adventure](#) ?

Description (optional)

☒ **Public**

Anyone on the internet can see this repository. You choose who can commit.

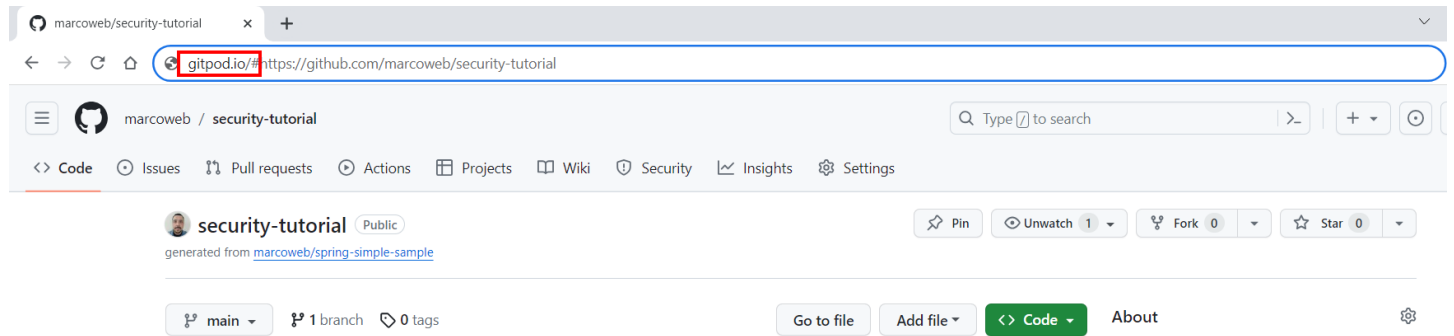
☐ **Private**

You choose who can see and commit to this repository.

 You are creating a public repository in your personal account.

**Create repository**

Abra o projeto criado no gitpod:



Teste a aplicação de modelo:

```
gitpod /workspace/security-tutorial (main) $ gradle bootRun
```

Observe as páginas ("/" e "/admin"):

8080-marcoweb-securitytutori-i6qwhpudkyv.ws-us105.gitpod.io

# Olá Spring!!!

8080-marcoweb-securitytutori-i6qwhpudkyv.ws-us105.gitpod.io/admin

## Admin

### Adicionando o Spring Security

Adicione a biblioteca Spring Security no arquivo "build.gradle":

build.gradle M X

```
app > build.gradle
1  plugins {
2      id 'java'
3      id 'org.springframework.boot' version '3.0.6'
4      id 'io.spring.dependency-management' version '1.1.0'
5  }
6
7  repositories {
8      mavenCentral()
9  }
10
11 dependencies {
12     implementation 'org.springframework.boot:spring-boot-starter-web'
13     implementation 'org.apache.tomcat.embed:tomcat-embed-jasper'
14     implementation 'org.springframework.boot:spring-boot-starter-data-jpa:3.1.5'
15     implementation 'mysql:mysql-connector-java:8.0.33'
16     implementation 'org.springframework.boot:spring-boot-starter-security:3.1.5'
17 }
```

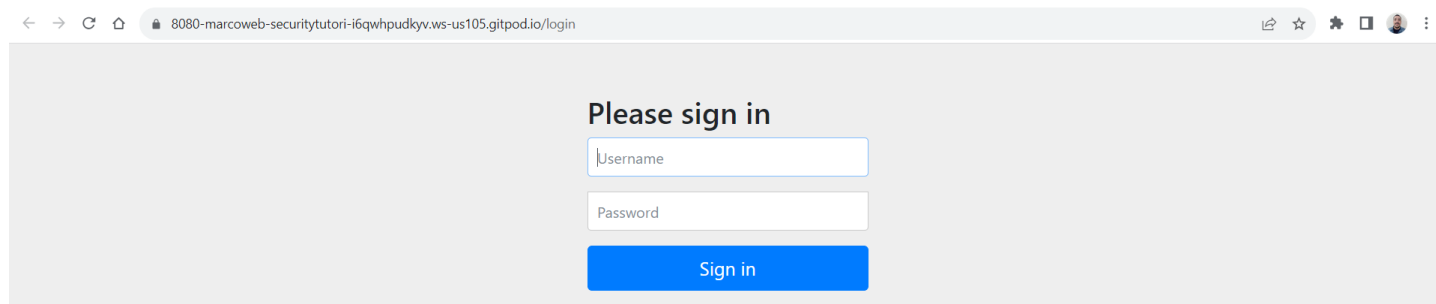
Construa a aplicação:

```
gitpod /workspace/security-tutorial (main) $ gradle build  
Picked up JAVA_TOOL_OPTIONS: -Xmx3489m  
  
BUILD SUCCESSFUL in 6s  
5 actionable tasks: 2 executed, 3 up-to-date
```

Execute a aplicação:

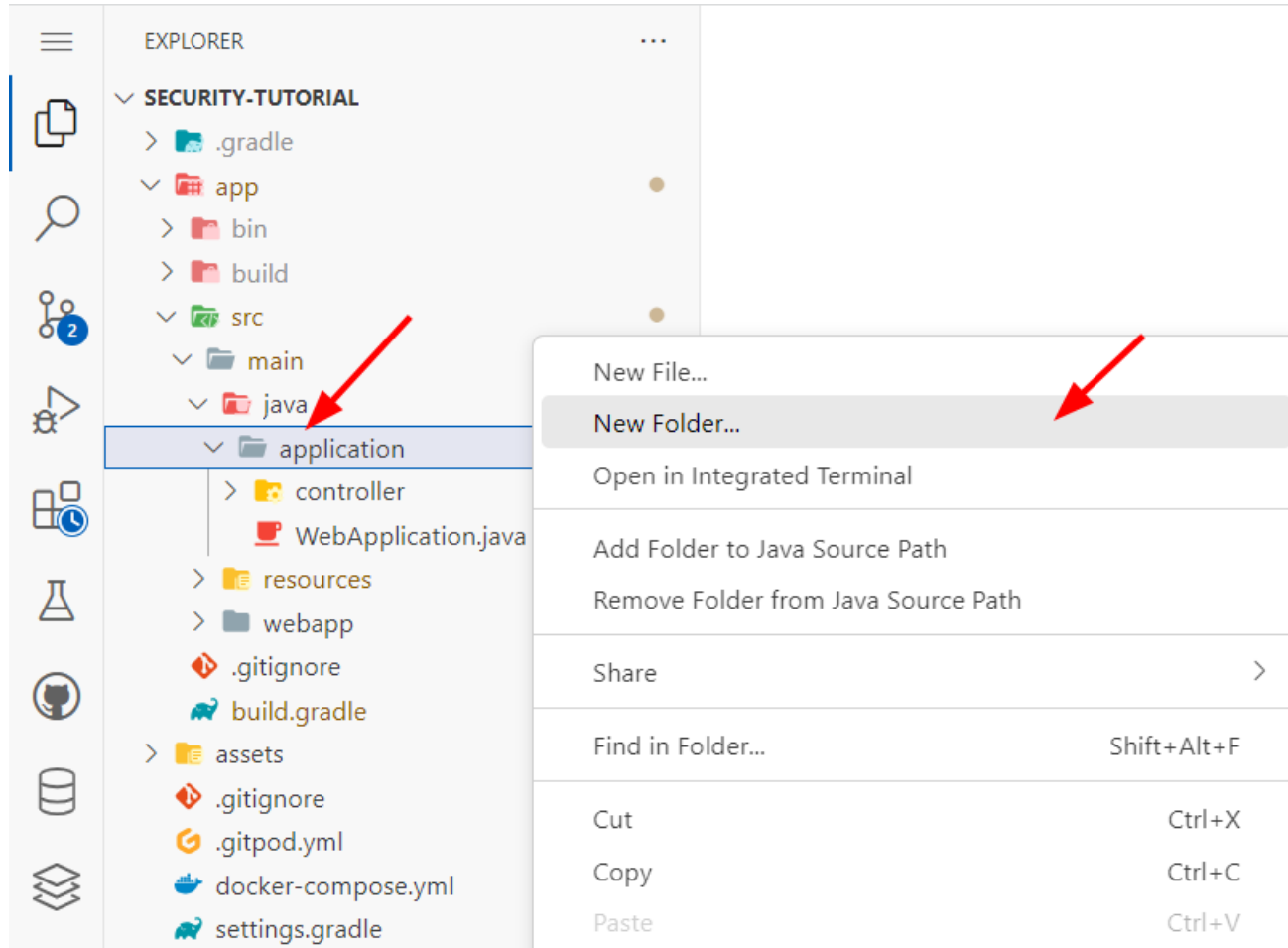
```
gitpod /workspace/security-tutorial (main) $ gradle bootRun
```

Observe que a aplicação já possui segurança:

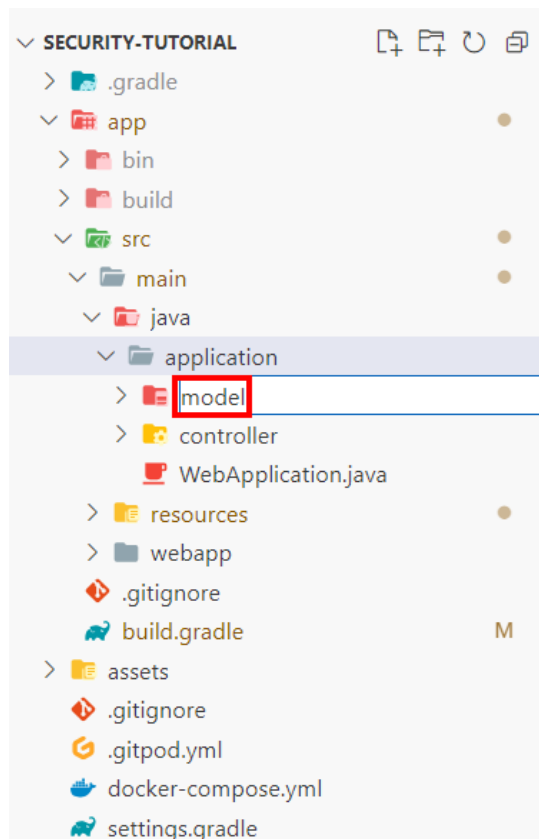


## Criando o Modelo de Dados de Usuários

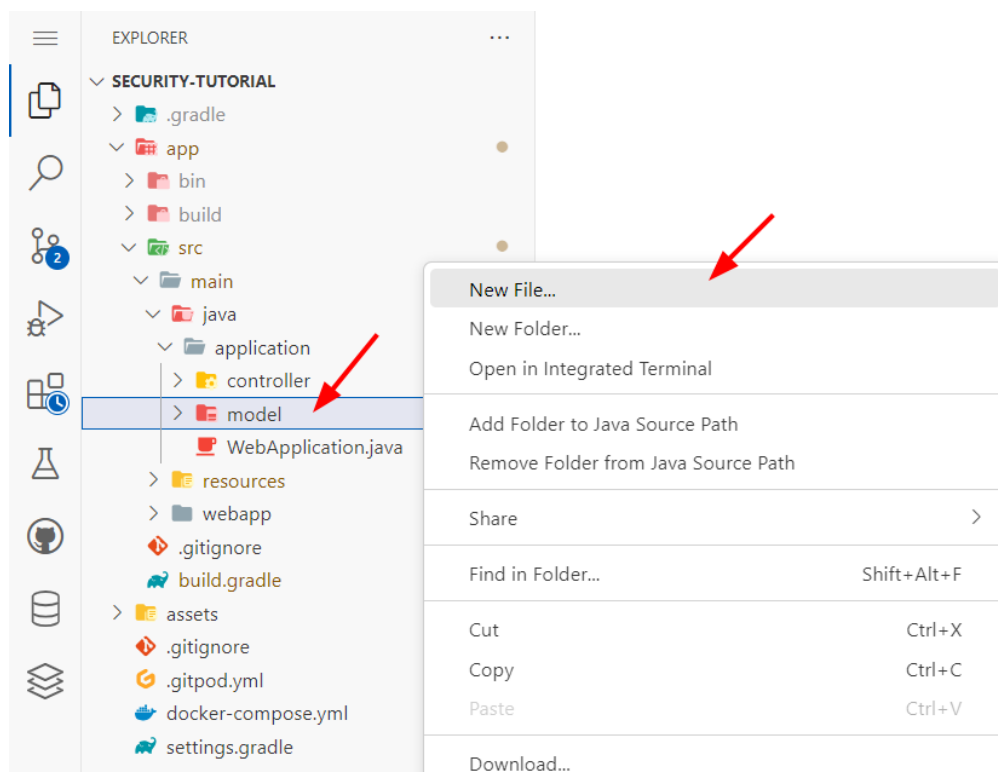
Crie uma nova pasta:



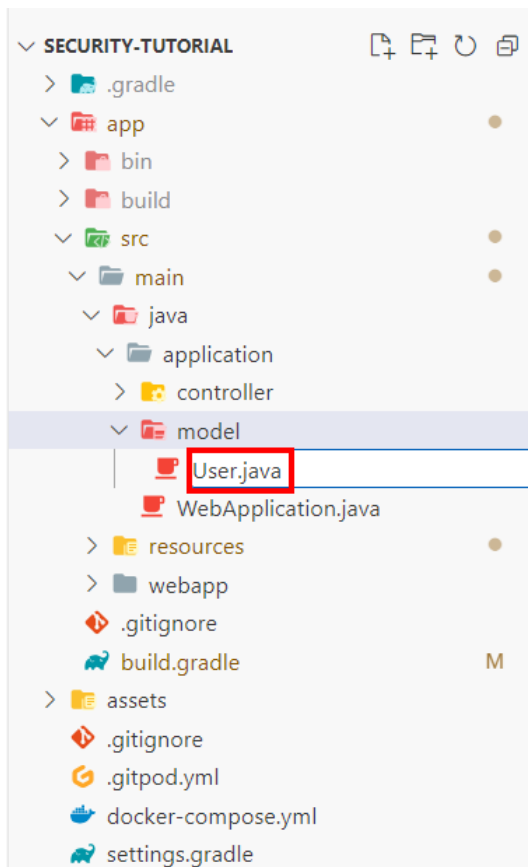
Nomeie a pasta criada como “model”:



Crie um novo arquivo:



Nomeie o arquivo criado como “User”:

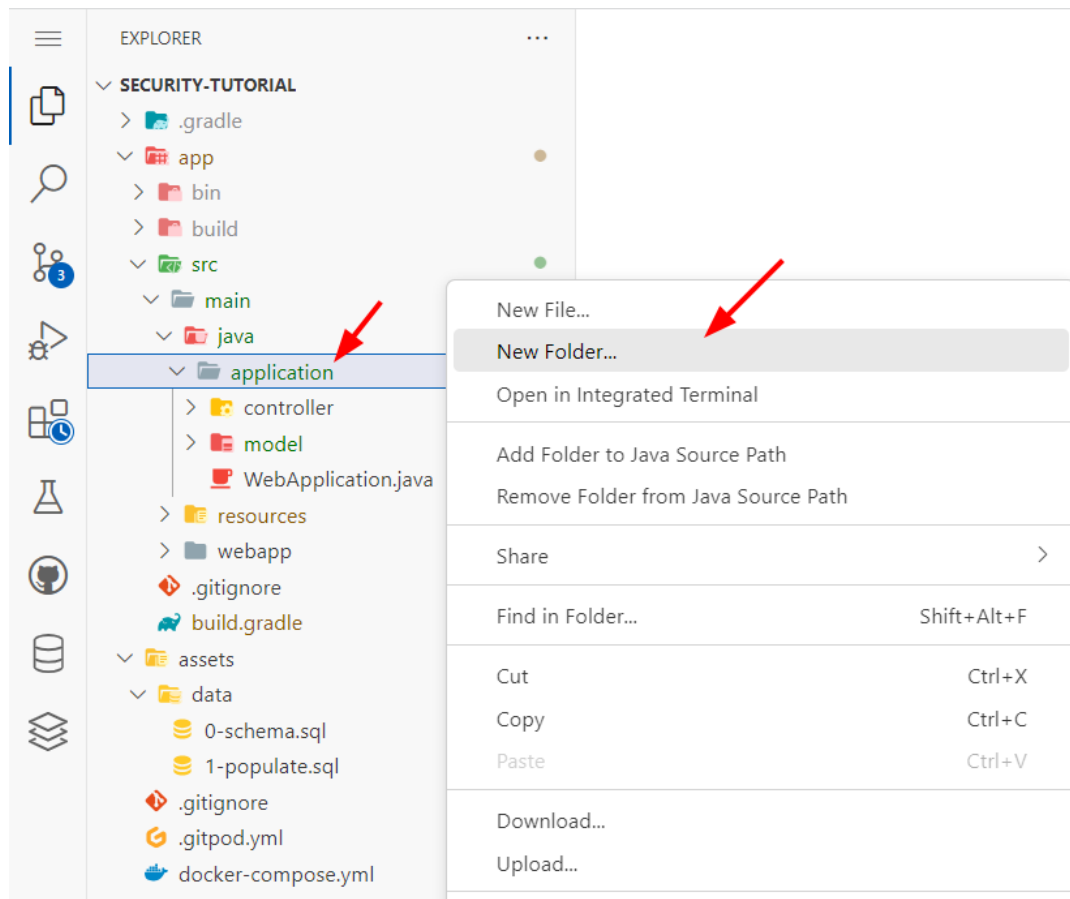


Insira o conteúdo da listagem a seguir ao arquivo criado:

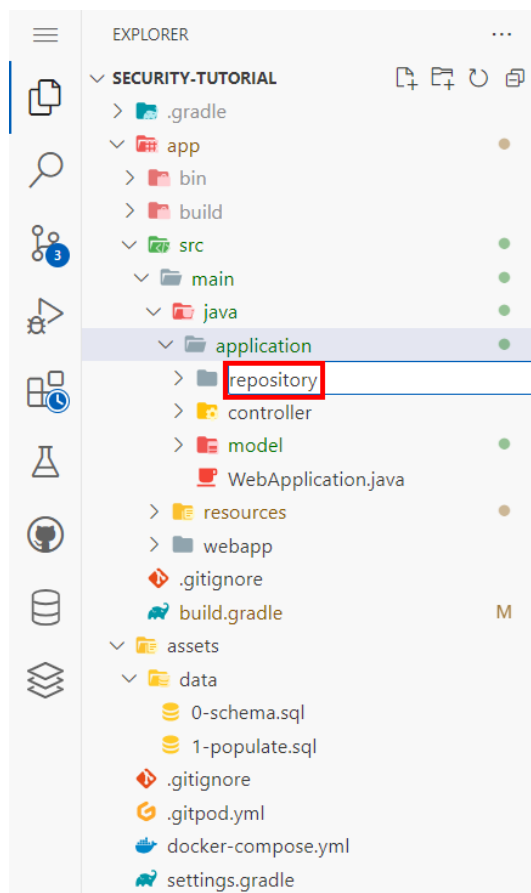
app > src > main > java > application > model >  User.java > ...

```
1  package application.model;
2
3  import jakarta.persistence.Entity;
4  import jakarta.persistence.GeneratedValue;
5  import jakarta.persistence.GenerationType;
6  import jakarta.persistence.Id;
7  import jakarta.persistence.Table;
8
9  @Entity
10 @Table(name = "users")
11 public class User {
12     @Id
13     @GeneratedValue(strategy = GenerationType.IDENTITY)
14     private long id;
15     private String username;
16     private String password;
17
18     public long getId() {
19         return id;
20     }
21     public void setId(long id) {
22         this.id = id;
23     }
24     public String getUsername() {
25         return username;
26     }
27     public void setUsername(String username) {
28         this.username = username;
29     }
30     public String getPassword() {
31         return password;
32     }
33     public void setPassword(String password) {
34         this.password = password;
35     }
36 }
```

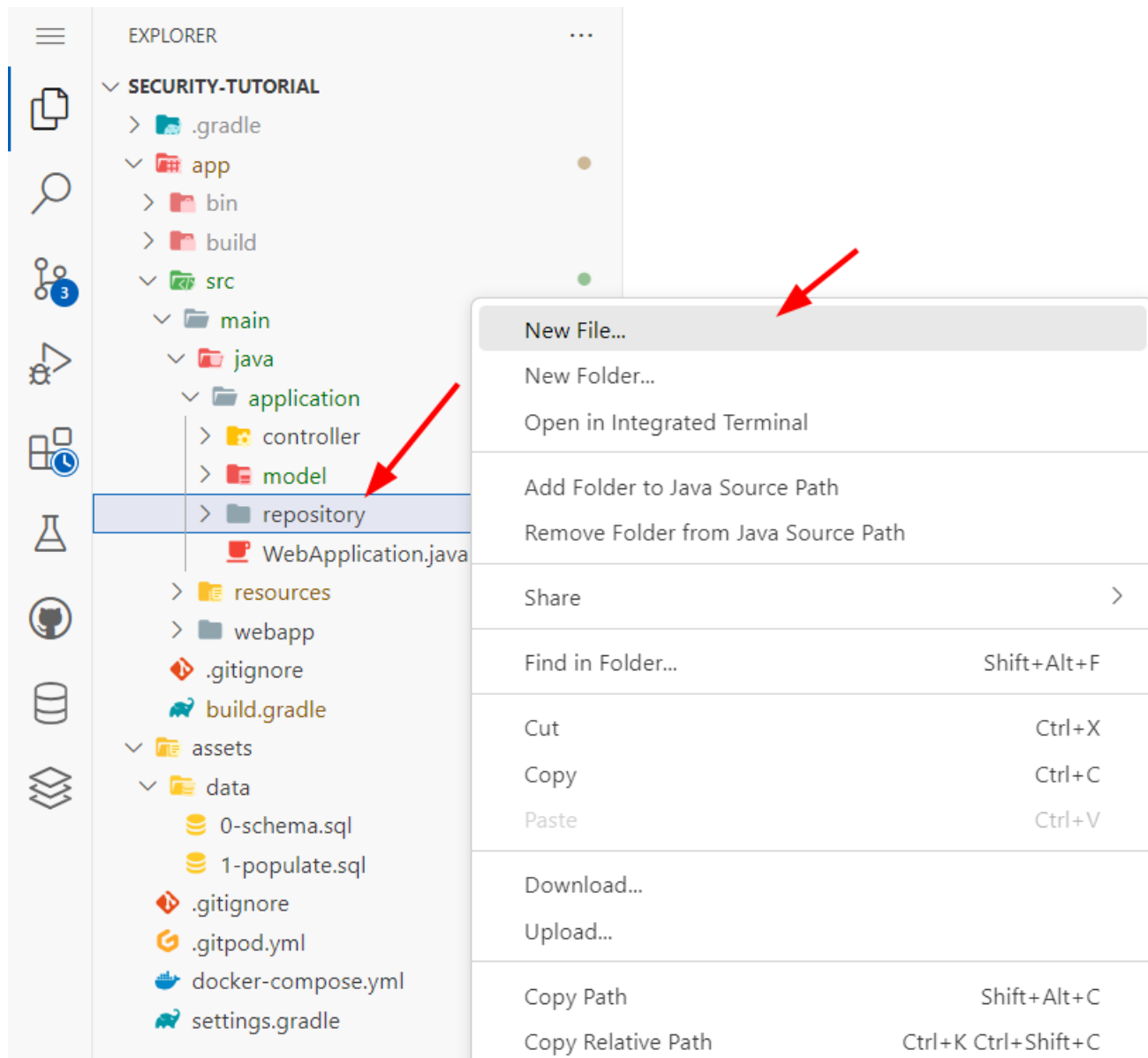
Crie uma nova pasta:



Nomeie a pasta criada como “repository”:

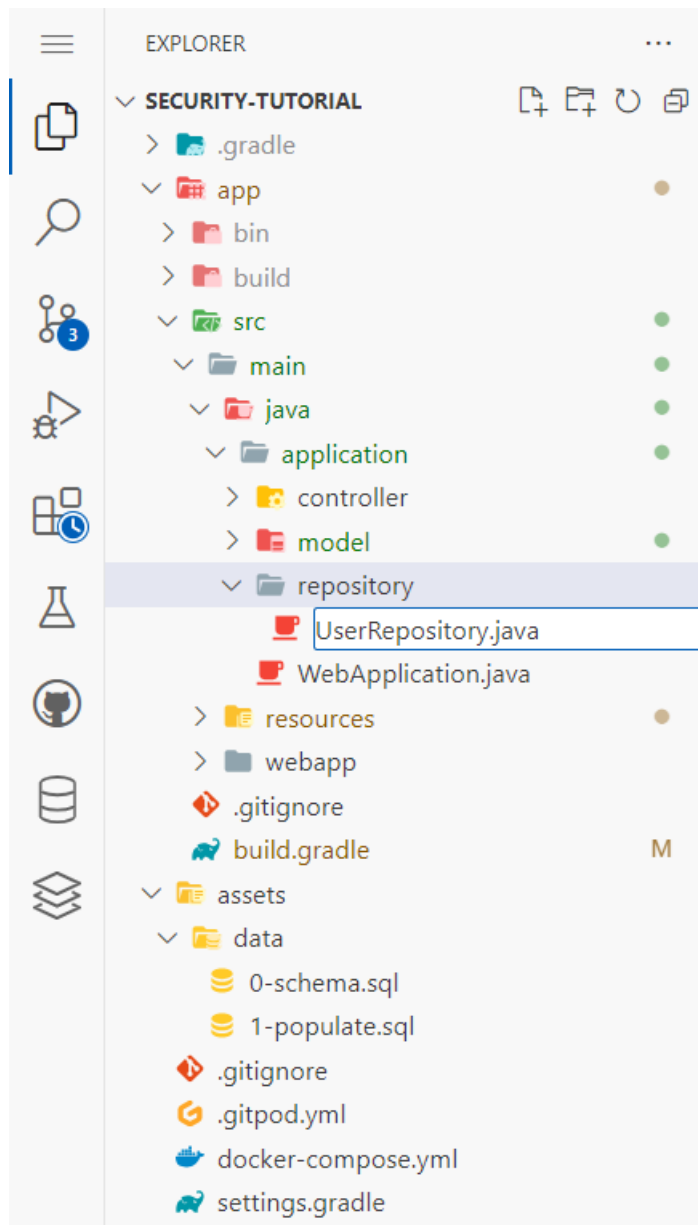


Crie um novo arquivo em “repository”:






Nomeie o arquivo como “UserRepository.java”:

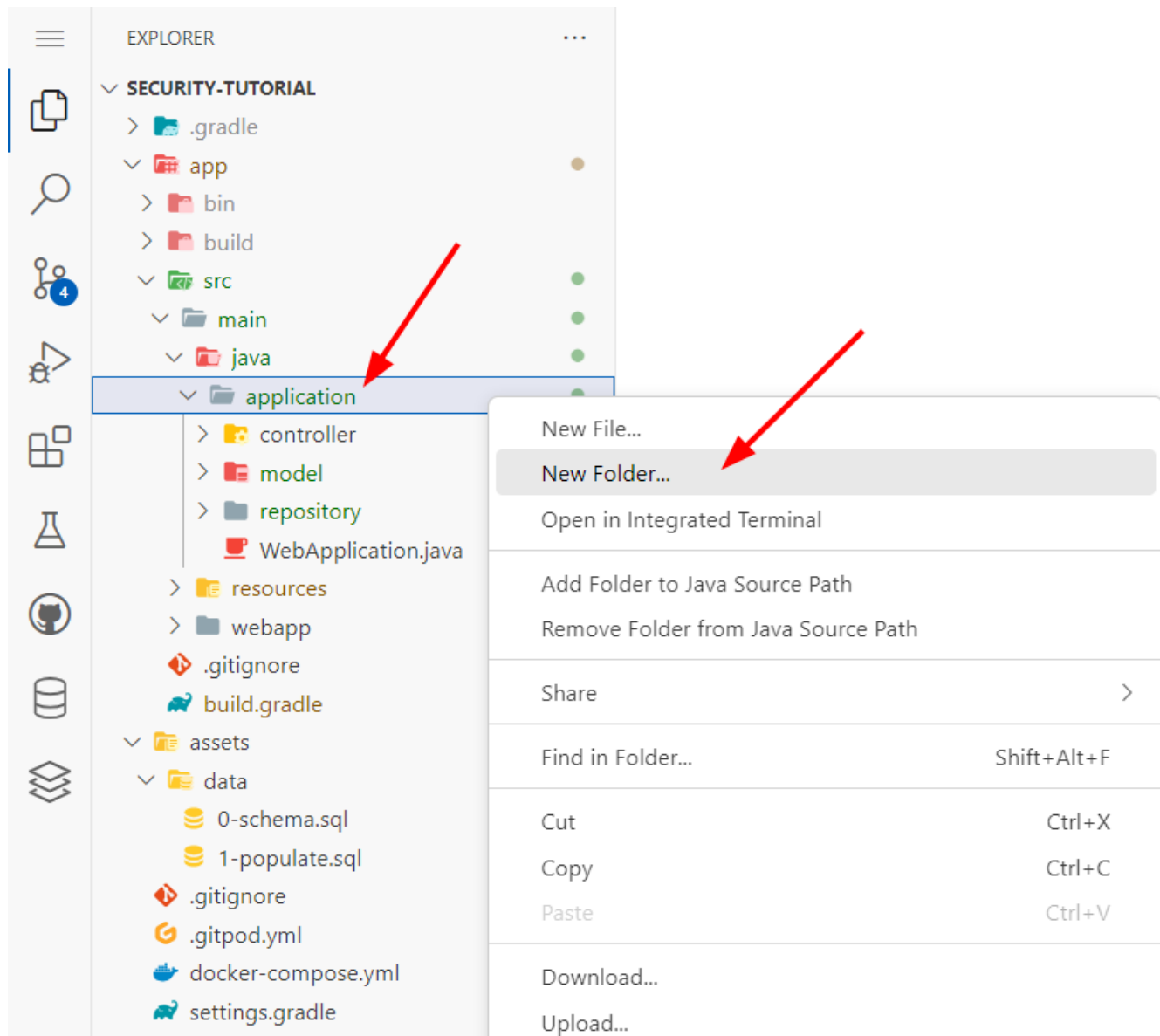


Insira o conteúdo da listagem a seguir ao arquivo criado:

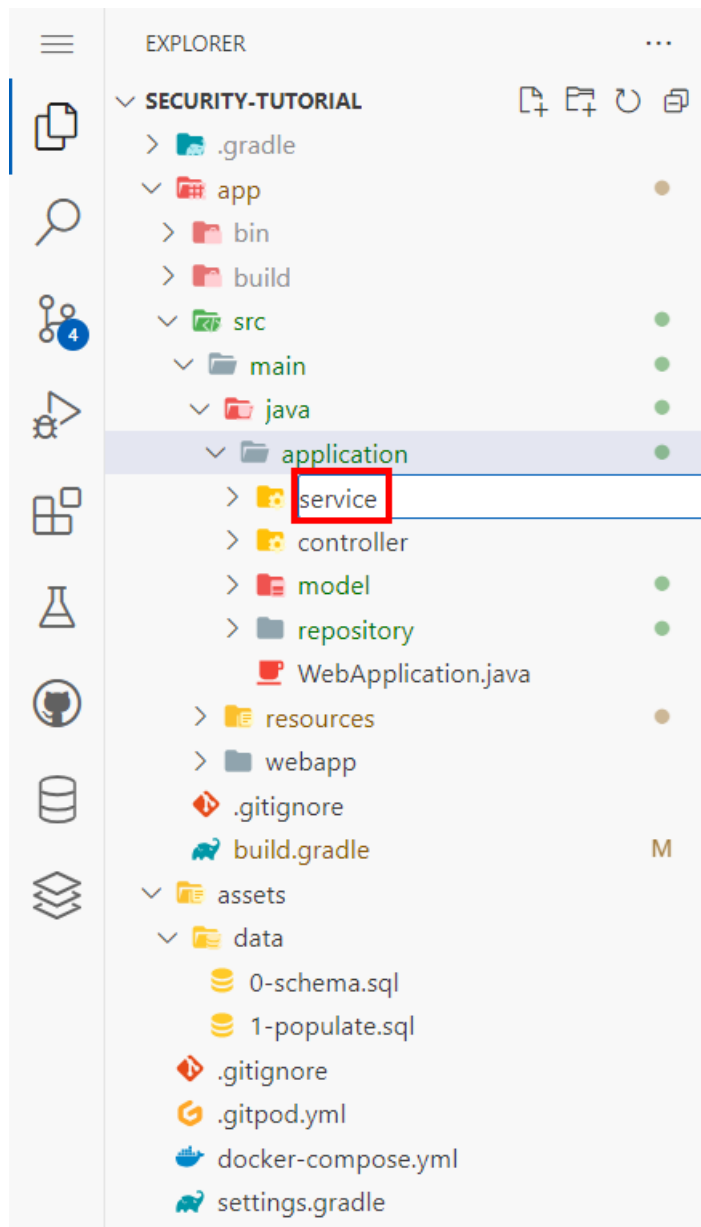
```
app > src > main > java > application > repository >  UserRepository.java > ...  
1 package application.repository;  
2  
3 import org.springframework.data.repository.CrudRepository;  
4  
5 import application.model.User;  
6  
7 public interface UserRepository extends CrudRepository<User, Long> {  
8     public User findByUsername(String username);  
9 }
```

## Criando o Serviço

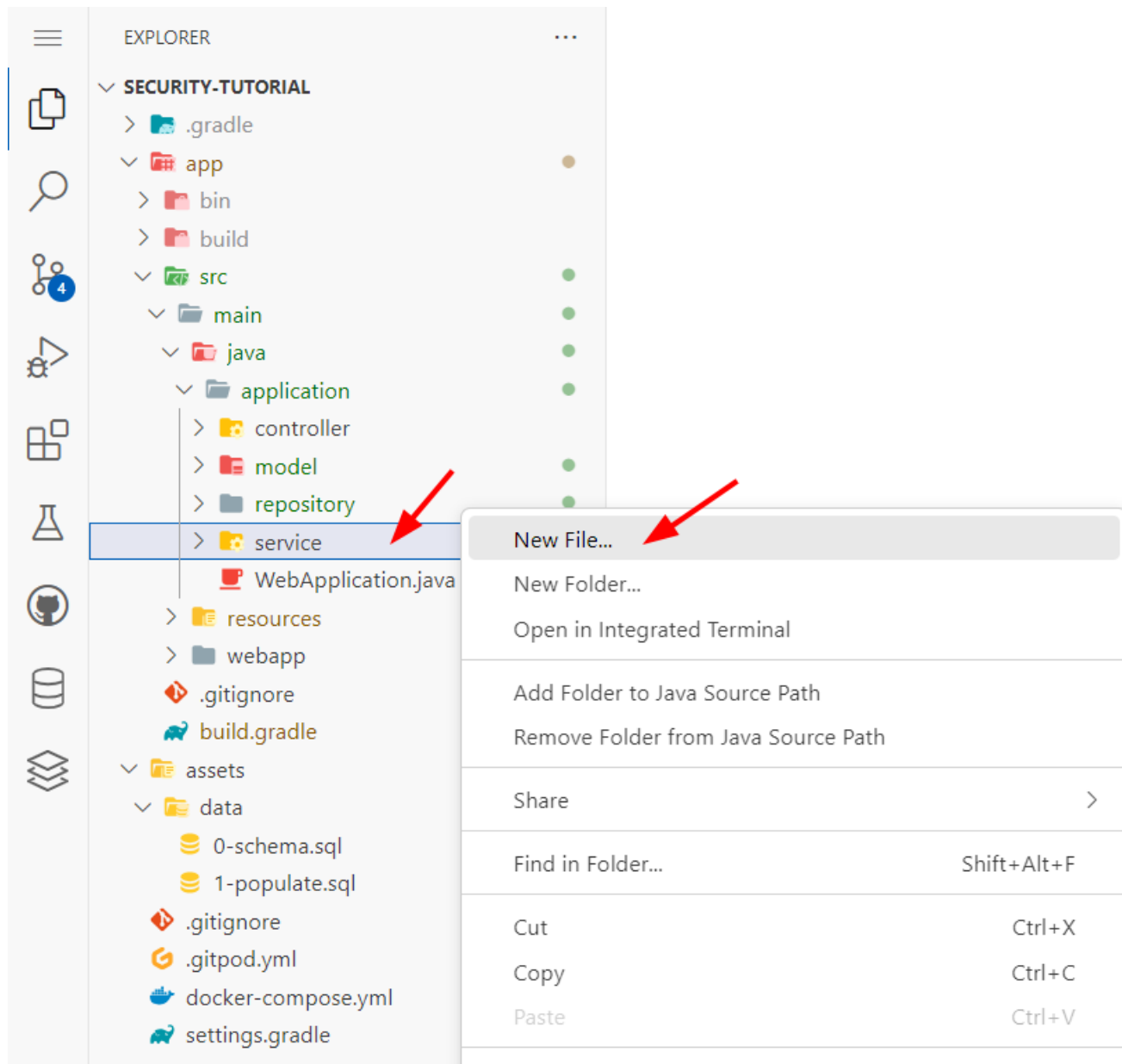
Crie uma pasta em “application”:



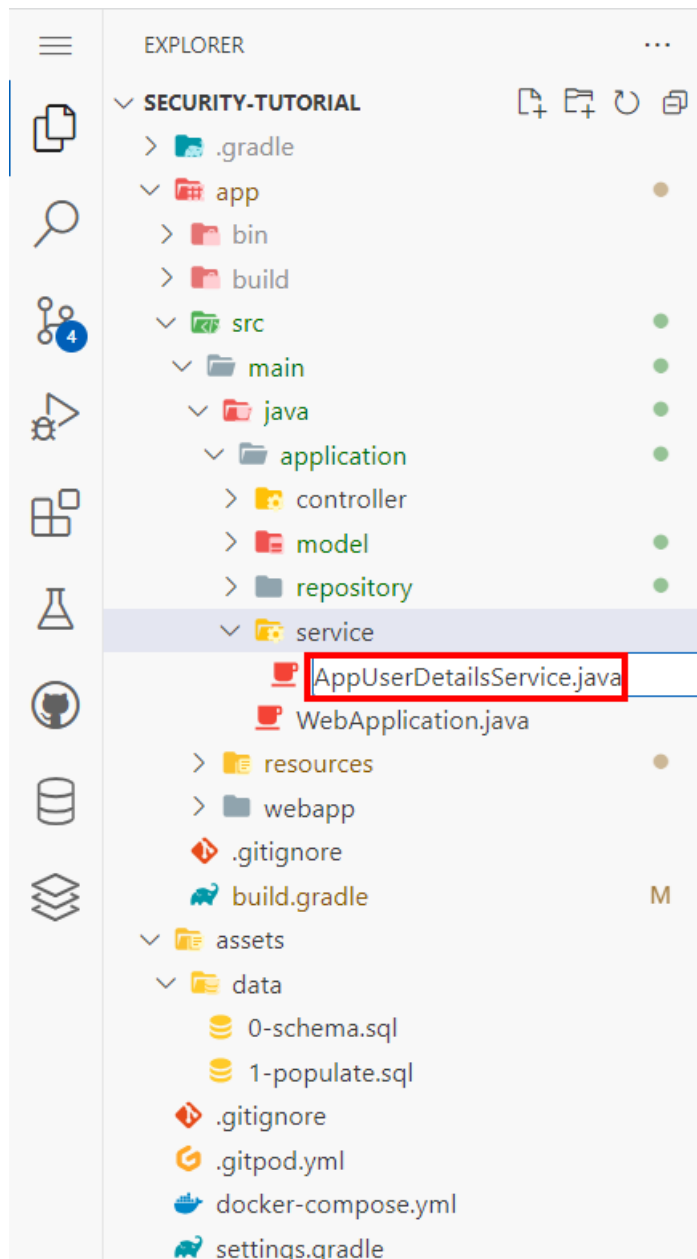
Nomeie como service:




Crie um arquivo em service:



Nomeie o arquivo criado:



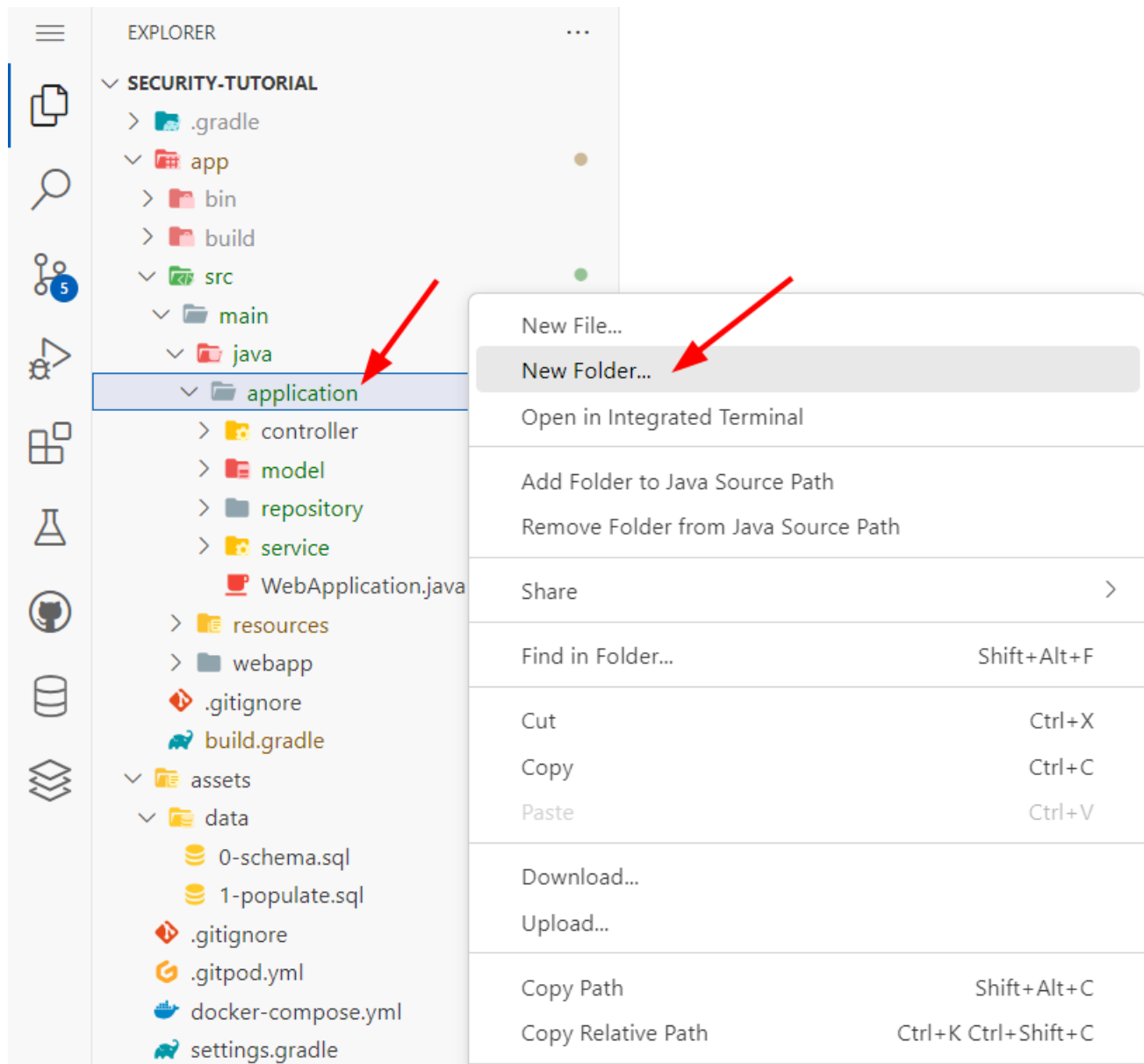
Insira a listagem à seguir ao arquivo criado:

app > src > main > java > application > service >  AppUserDetailsService.java > ...

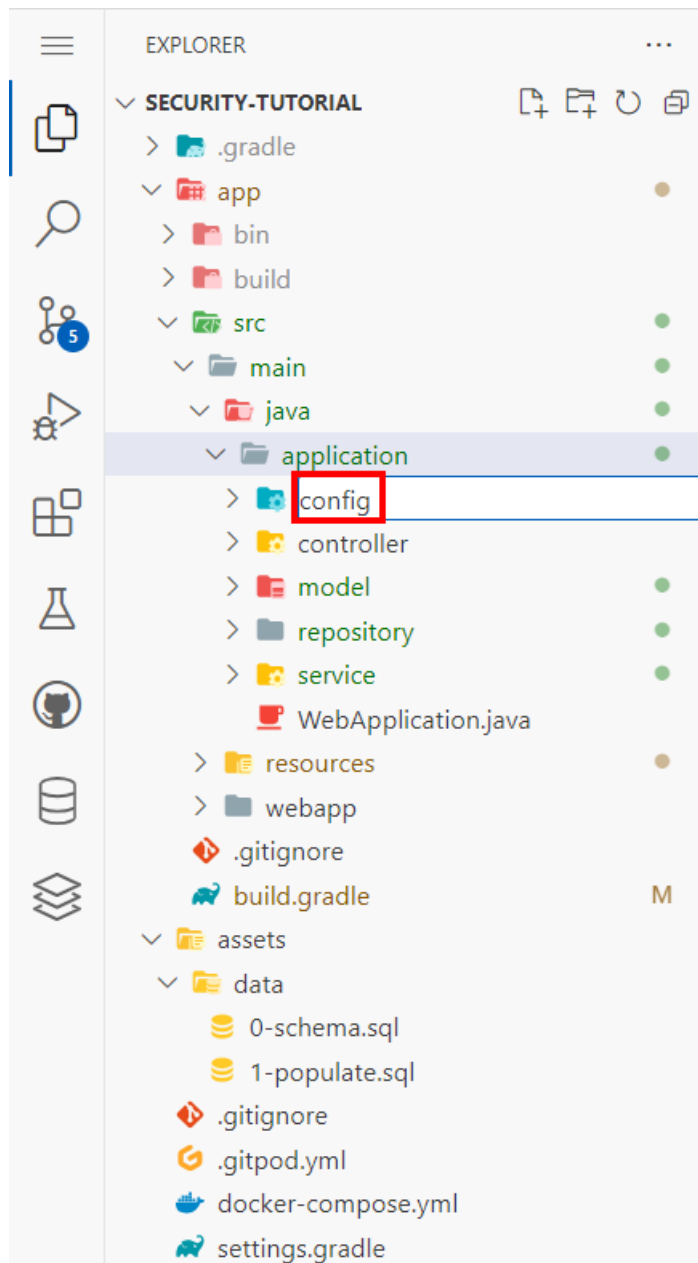
```
1  package application.service;
2
3  import org.springframework.beans.factory.annotation.Autowired;
4  import org.springframework.security.core.userdetails.UserDetails;
5  import org.springframework.security.core.userdetails.UserDetailsService;
6  import org.springframework.security.core.userdetails.UsernameNotFoundException;
7  import org.springframework.stereotype.Service;
8
9  import application.model.User;
10 import application.repository.UserRepository;
11
12 @Service
13 public class AppUserDetailsService implements UserDetailsService {
14     @Autowired
15     private UserRepository userRepo;
16
17     @Override
18     public UserDetails loadUserByUsername(String username) throws UsernameNotFoundException {
19         User user = userRepo.findByUsername(username);
20         if(user == null){
21             throw new UsernameNotFoundException("Usuário Não Encontrado");
22         }
23         UserDetails userDetails =
24             org.springframework.security.core.userdetails.User.builder()
25                 .username(user.getUsername())
26                 .password(user.getPassword())
27                 .roles("USER")
28                 .build();
29
30         return userDetails;
31     }
32 }
```

## Crie a classe de configurações

Crie uma nova pasta:

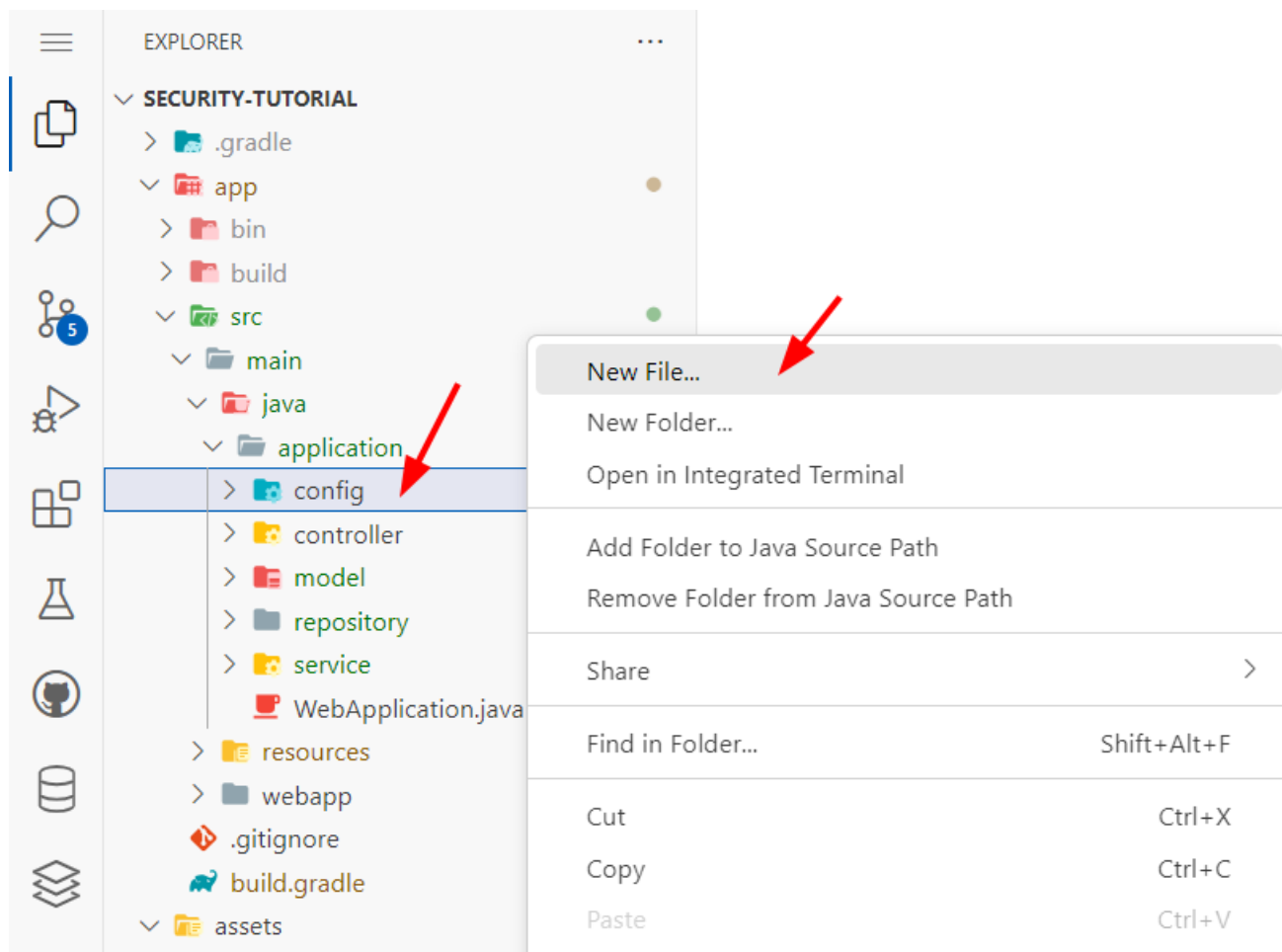


Nomeie como “config”:

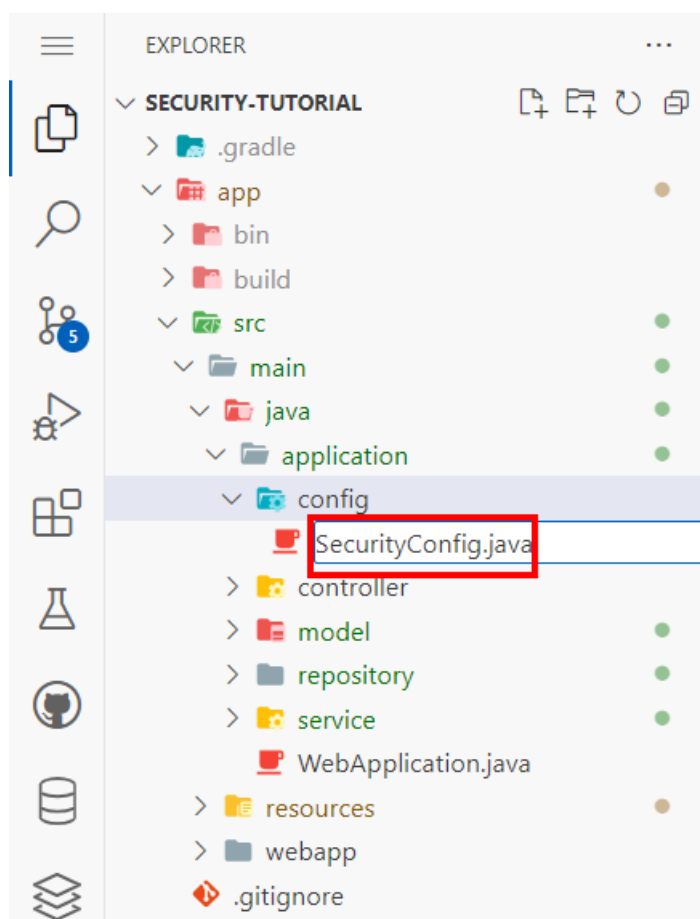





Crie um arquivo:



Nomeie o arquivo criado:



Insira o conteúdo da listagem a seguir ao arquivo criado:

app > src > main > java > application > config >  SecurityConfig.java > ...

```
1  package application.config;
2
3  import org.springframework.context.annotation.Bean;
4  import org.springframework.context.annotation.Configuration;
5  import org.springframework.security.config.annotation.web.builders.HttpSecurity;
6  import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;
7  import org.springframework.security.crypto.password.NoOpPasswordEncoder;
8  import org.springframework.security.web.SecurityFilterChain;
9
10 @Configuration
11 @EnableWebSecurity
12 public class SecurityConfig {
13     @Bean
14     public SecurityFilterChain securityFilterChain(HttpSecurity http) throws Exception {
15         http
16             .csrf().disable()
17             .authorizeHttpRequests()
18             .anyRequest().authenticated();
19         http
20             .formLogin();
21         return http.build();
22     }
23
24     @SuppressWarnings("deprecation")
25     @Bean
26     public NoOpPasswordEncoder passwordEncoder() {
27         return (NoOpPasswordEncoder) NoOpPasswordEncoder.getInstance();
28     }
29 }
30 }
```


Execute a aplicação:

```
gitpod /workspace/security-tutorial (main) $ gradle bootRun
```

Teste a aplicação com as credenciais a seguir:

8080-marcoweb-securitytutori-i6qwhpudkyv.ws-us105.gitpod.io/login

Please sign in



Sign in

Observe o acesso:

← → ↻ 🏠 🔒 8080-marcoweb-securitytutori-i6qwhpudkyv.ws-us105.gitpod.io/?continue

# Olá Spring!!!

Acesse a rota “/logout” para sair:

← → ↻ 🏠 🌐 https://8080-marcoweb-securitytutori-i6qwhpudkyv.ws-us105.gitpod.io/logout

# Olá Spring!!!

← → ↻ 🏠 🔒 8080-marcoweb-securitytutori-i6qwhpudkyv.ws-us105.gitpod.io/login?logout

Please sign in

You have been signed out

Username

Password

Sign in