Spring Boot CRUD Tutorial

Step 2

MainController.

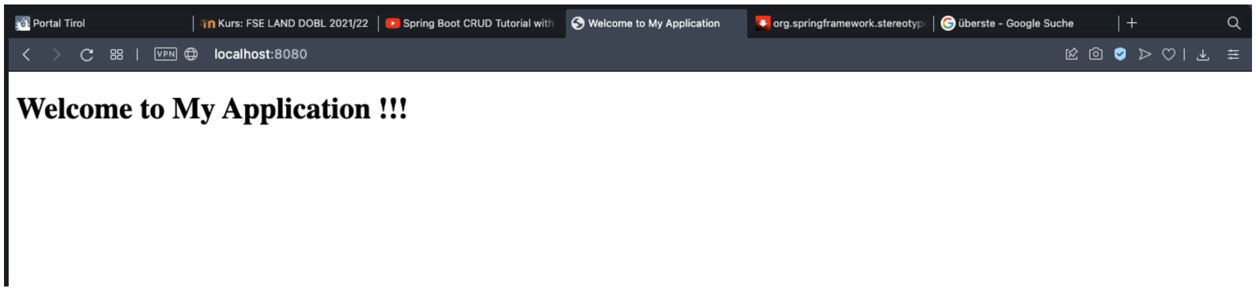
```
package com.mycompany;

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


@Controller
public class MainController {

    @GetMapping("")
    public String showHomePage(){
        return "index";
    }
}
```

index.html → templates

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Welcome to My Application</title>
</head>
<body>
<h1>Welcome to My Application !!!</h1>
<a></a>

</body>
</html>
```

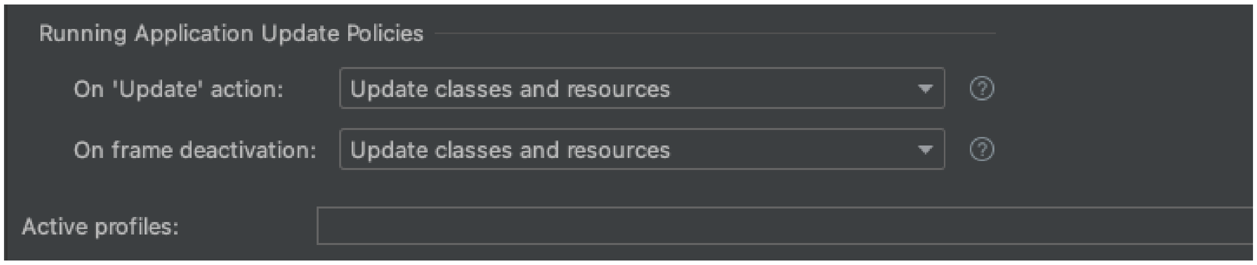
Ausgabe Website

localhost:8080


Portal Tirol
Kurs: FSE LAND DOBL 2021/22
Spring Boot CRUD Tutorial with
Welcome to My Application
org.springframework.stereotyp
überste - Google Suche
localhost:8080

Welcome to My Application !!!

```
<dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
</dependency>
```



Running Application Update Policies

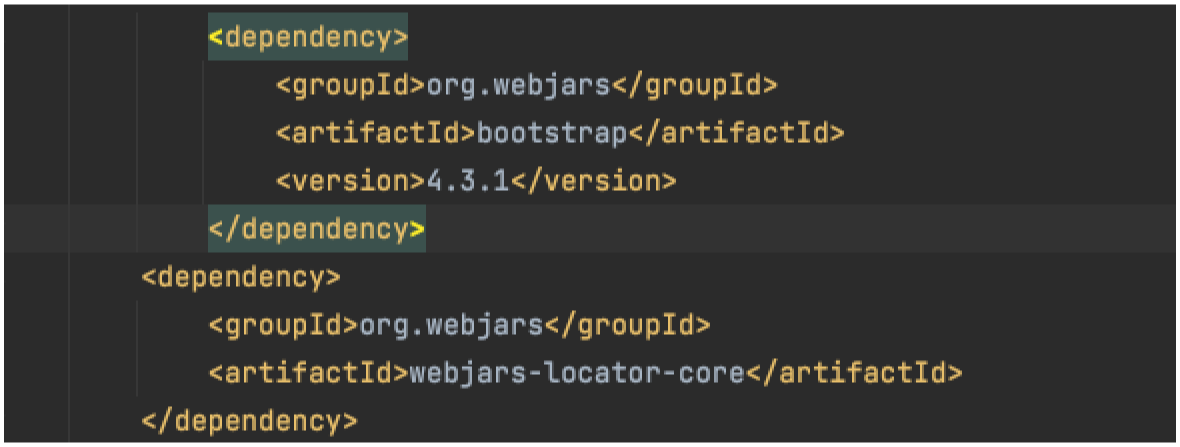
On 'Update' action: Update classes and resources

On frame deactivation: Update classes and resources

Active profiles:

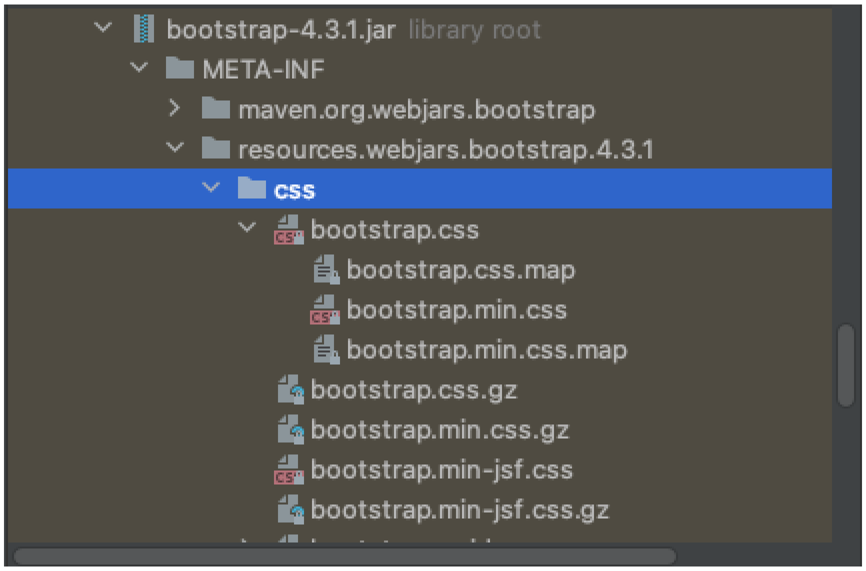
einzige Änderung notwendig

Bootstrap Verwendung



```xml
        <dependency>
            <groupId>org.webjars</groupId>
            <artifactId>bootstrap</artifactId>
            <version>4.3.1</version>
        </dependency>
    <dependency>
        <groupId>org.webjars</groupId>
        <artifactId>webjars-locator-core</artifactId>
    </dependency>
```

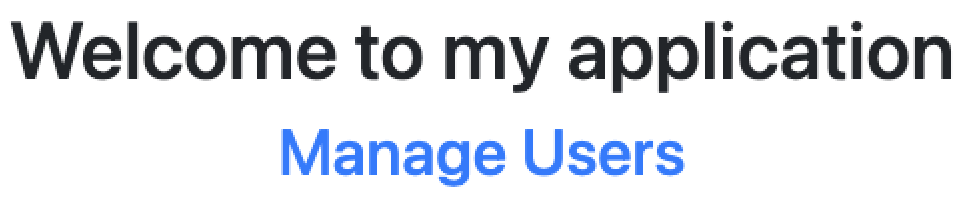
Hier wird Bootstarp hinzugefügt über die Dependencys



```
bootstrap-4.3.1.jar library root
  META-INF
    maven.org.webjars.bootstrap
    resources.webjars.bootstrap.4.3.1
      css
        bootstrap.css
          bootstrap.css.map
          bootstrap.min.css
          bootstrap.min.css.map
        bootstrap.css.gz
        bootstrap.min.css.gz
        bootstrap.min-jsf.css
        bootstrap.min-jsf.css.gz
```

Nachkontrolle in den External Libraries

!!(Wichtig Maven refreshen da Bootstrap nicht direkt eingefügt wird)!!



Test für Bootstrap mit einem Heading 2

```
<h2><a class="h2" href="@{/user}">Manage Users</a></h2>
```

Code Data Access Layer

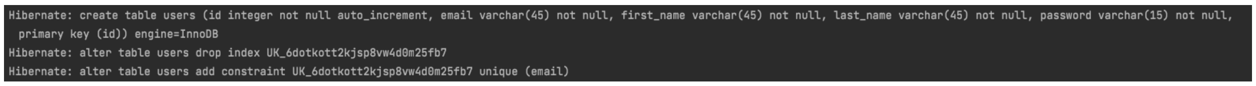
(Repository Layer)

Zuerst wird eine Klasse erstellt für Datenbank(User)

```
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name = "users")
public class User {

}
```

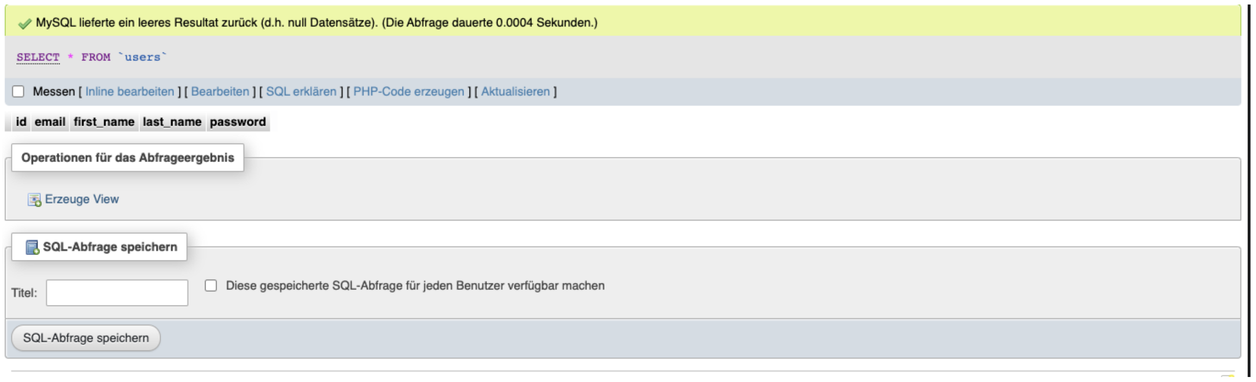
```
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Integer id;

    @Column(nullable = false, unique = true, length = 45)
    private String email;

    @Column(length = 15, nullable = false)
    private String password;

    @Column(length = 45, nullable = false, name = "first_name")
    private String firstName;

    @Column(length = 45, nullable = false, name = "last_name")
    private String lastName;
```

```
import org.springframework.data.repository.CrudRepository;

public interface UserRepository extends CrudRepository<User, Integer> {

}
```



```
Hibernate: create table users (id integer not null auto_increment, email varchar(45) not null, first_name varchar(45) not null, last_name varchar(45) not null, password varchar(15) not null,
  primary key (id)) engine=InnoDB
Hibernate: alter table users drop index UK_6dotkott2kjsp8vw4d0m25fb7
Hibernate: alter table users add constraint UK_6dotkott2kjsp8vw4d0m25fb7 unique (email)
```

Die Compilerasugabe für die Kontrolle der erzeugten Entity.


MySQL lieferte ein leeres Resultat zurück (d.h. null Datensätze). (Die Abfrage dauerte 0.0004 Sekunden.)

```
SELECT * FROM `users`
```

Messen [Inline bearbeiten] [Bearbeiten] [SQL erklären] [PHP-Code erzeugen] [Aktualisieren]

id	email	first_name	last_name	password

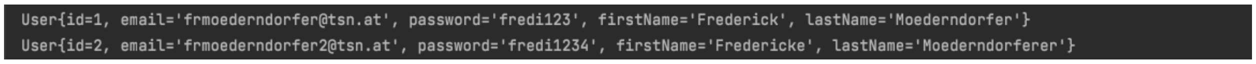
Operationen für das Abfrageergebnis
Erzeuge View
SQL-Abfrage speichern
Titel:
Diese gespeicherte SQL-Abfrage für jeden Benutzer verfügbar machen
SQL-Abfrage speichern

Zweite Überprüfung.

Code Unit Tests

```
@DataJpaTest
@AutoConfigureTestDatabase(replace = AutoConfigureTestDatabase.Replace.NONE)
@Rollback(false)
public class JavaRepositoryTests {

    @Autowired private UserRepository repo;

    @Test
    public void testAddNew(){
        User user = new User();
        user.setEmail("frmoederndorfer2@tsn.at");
        user.setPassword("fredi1234");
        user.setFirstName("Fredericke");
        user.setLastName("Moederndorferer");

        User savedUser = repo.save(user);

        Assertions.assertThat(savedUser).isNotNull();
        Assertions.assertThat(savedUser.getId()).isGreaterThan(0);

    }
}
```

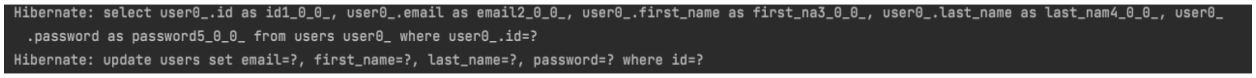
```
@Test
    public void testListAll(){
        Iterable<User> users = repo.findAll();

        Assertions.assertThat(users).hasSizeGreaterThan(0);

        for (User user: users) {
            System.out.println(user);
        }
    }
```



```
User{id=1, email='frmoederndorfer@tsn.at', password='fredi123', firstName='Frederick', lastName='Moederndorfer'}
User{id=2, email='frmoederndorfer2@tsn.at', password='fredi1234', firstName='Fredericke', lastName='Moederndorferer'}
```

Ausgabe der testListAll Methode

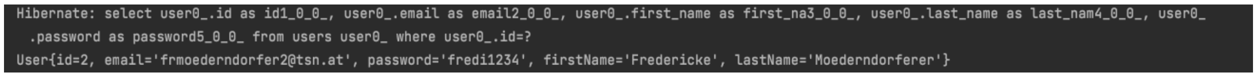
```
@Test
    public void testUpdate(){
        int userId = 1;
        Optional<User> userOptional = repo.findById(userId);
        User user = userOptional.get();
        user.setPassword("hello2000");
        repo.save(user);

        User updatedUser = repo.findById(userId).get();
        Assertions.assertThat(updatedUser.getPassword()).isEqualTo("hello2000");
    }
```



```
Hibernate: select user0_.id as id1_0_0_, user0_.email as email2_0_0_, user0_.first_name as first_na3_0_0_, user0_.last_name as last_nam4_0_0_, user0_
  .password as password5_0_0_ from users user0_ where user0_.id=?
Hibernate: update users set email=?, first_name=?, last_name=?, password=? where id=?
```

Ausgabe der testUpdate Methode

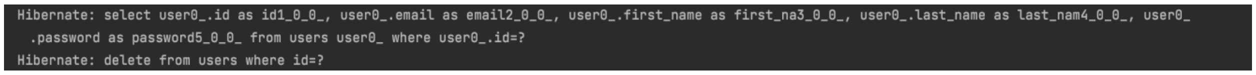
```
@Test
    public void testGet(){
        int userId = 2;
        Optional<User> userOptional = repo.findById(userId);
        Assertions.assertThat(userOptional).isPresent();
        System.out.println(userOptional.get());
    }
```



```
Hibernate: select user0_.id as id1_0_0_, user0_.email as email2_0_0_, user0_.first_name as first_na3_0_0_, user0_.last_name as last_nam4_0_0_, user0_
  .password as password5_0_0_ from users user0_ where user0_.id=?
User{id=2, email='frmoederndorfer2@tsn.at', password='fredi1234', firstName='Fredericke', lastName='Moederndorferer'}
```

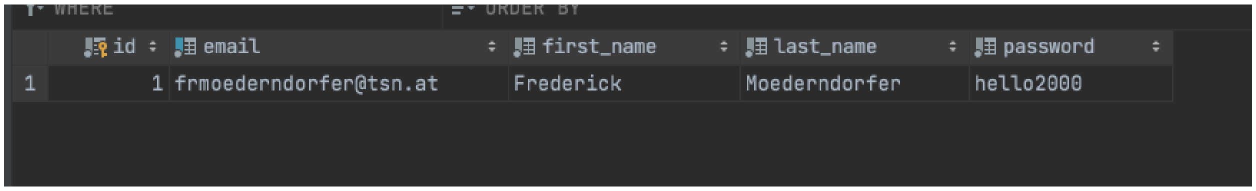
Ausgabe der testGet Methode

```
@Test
    public void testDelete(){
        int userId = 2;
        repo.deleteById(userId);
        Optional<User> userOptional = repo.findById(userId);
        Assertions.assertThat(userOptional).isNotPresent();
    }
```



```
Hibernate: select user0_.id as id1_0_0_, user0_.email as email2_0_0_, user0_.first_name as first_na3_0_0_, user0_.last_name as last_nam4_0_0_, user0_
  .password as password5_0_0_ from users user0_ where user0_.id=?
Hibernate: delete from users where id=?
```

Ausgabe der testDelete Methode



	id	email	first_name	last_name	password
1	1	frmoederndorfer@tsn.at	Frederick	Moederndorfer	hello2000

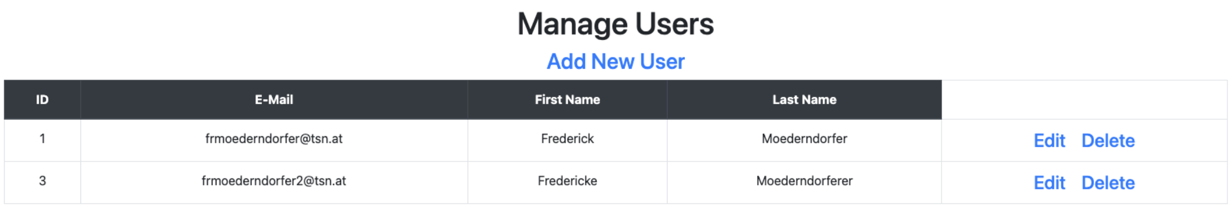
DatenbankBild

Code Users Listing Page

```
@Controller
public class UserController {

    @Autowired private UserService service;
    @GetMapping("/users")
    public String showUserList(Model model){
        List<User> listUsers = service.listAll();
        model.addAttribute("listUsers", listUsers);
        return "users";
    }
}
```

```
@Service
public class UserService {

    @Autowired private UserRepository repo;


    public List<User> listAll(){
        return (List<User>) repo.findAll();
    }
}
```

```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org" >
<head>
    <meta charset="UTF-8">
    <title>Manage Users</title>
    <link rel="stylesheet" type="text/css" th:href="@{/webjars/bootstrap/css/bootstrap.mi
n.css}" />
</head>
<body>
<div class="container-fluid text-center">
    <div><h1>Manage Users</h1></div>
    <div class="m-2">
        <a class="h3" th:href="@{/users/new}">Add New User</a>
    </div>

    <div>
        <table class="table table-bordered">
            <thead class="thead-dark">
            <tr>
                <th>ID</th>
                <th>E-Mail</th>
                <th>First Name</th>
                <th>Last Name</th>
            </tr>
            </thead>
            <tbody>
            <th:block th:each="user : ${listUsers}">
            <tr>
                <td>[[${user.id}]]</td>
                <td>[[${user.email}]]</td>
                <td>[[${user.firstName}]]</td>
                <td>[[${user.lastName}]]</td>
                <td>
                    <a class="h4 mr-3" th:href="@{'/users/edit/' + ${user.id}}">Edit</a>
                    <a class="h4" th:href="@{'/users/delete/' + ${user.id}}">Delete</a>
                </td>
            </tr>
            </th:block>
            </tbody>
        </table>

    </div>



</div>
</body>
</html>
```



Manage Users

Add New User

ID	E-Mail	First Name	Last Name	
1	frmoederndorfer@tsn.at	Frederick	Moederndorfer	Edit Delete
3	frmoederndorfer2@tsn.at	Fredericke	Moederndorferer	Edit Delete

Code Add User Function

Updaten in der User Klasse da wir einen Enabled Spalte hinzugefügt haben.

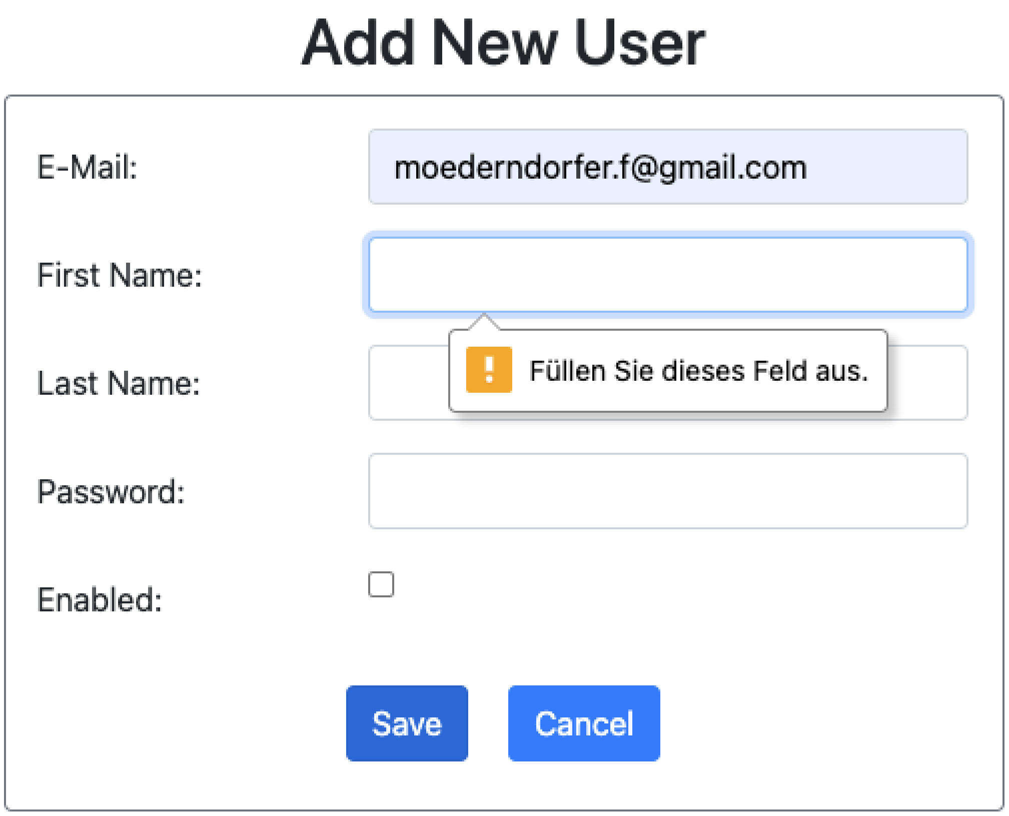
Mit Getter und Setter Methoden.

```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org" >
<head>
    <meta charset="UTF-8">
    <title>Add New User</title>
    <link rel="stylesheet" type="text/css" th:href="@{/webjars/bootstrap/css/bootstrap.mi
n.css}" />
</head>
<body>
<div class="container-fluid">
   <div class="text-center"> <h2>Add New User</h2></div>

    <form th:action="@{/user/save}" method="post" th:object="${user}"
        style="max-width: 500px; margin: 0 auto;">
        <div class="border border-secondary rounded p-3">
            <div class="form-group row">
                <label class="col-sm-4 col-form-label">E-Mail:</label>
                <div class="col-sm-8">
                    <input type="email" th:field="*{email}" class="form-control" />
                </div>
            </div>

            <div class="form-group row">
                <label class="col-sm-4 col-form-label">First Name:</label>
                <div class="col-sm-8">
                    <input type="text" th:field="*{firstName}" class="form-control" />
                </div>
            </div>

            <div class="form-group row">
                <label class="col-sm-4 col-form-label">Last Name:</label>
                <div class="col-sm-8">
                    <input type="text" th:field="*{lastName}" class="form-control" />
                </div>
            </div>

            <div class="form-group row">
                <label class="col-sm-4 col-form-label">Password:</label>
                <div class="col-sm-8">
                    <input type="password" th:field="*{password}" class="form-control" />
                </div>
            </div>

            <div class="form-group row">
                <label class="col-sm-4 col-form-label">Enabled:</label>
                <div class="col-sm-8">
                    <input type="checkbox" th:field="*{enabled}"/>
                </div>
            </div>
```

```
              <div class="text-center">
                  <button type="submit" class="btn btn-primary m-2">Save</button>
                  <button type="button" class="btn btn-primary m-2">Cancel</button>
              </div>
          </div>
      </form>
  </div>
  </body>
  </html>
```

Code Form Validation

Client seitige Validieren mittel required, max und minlength;

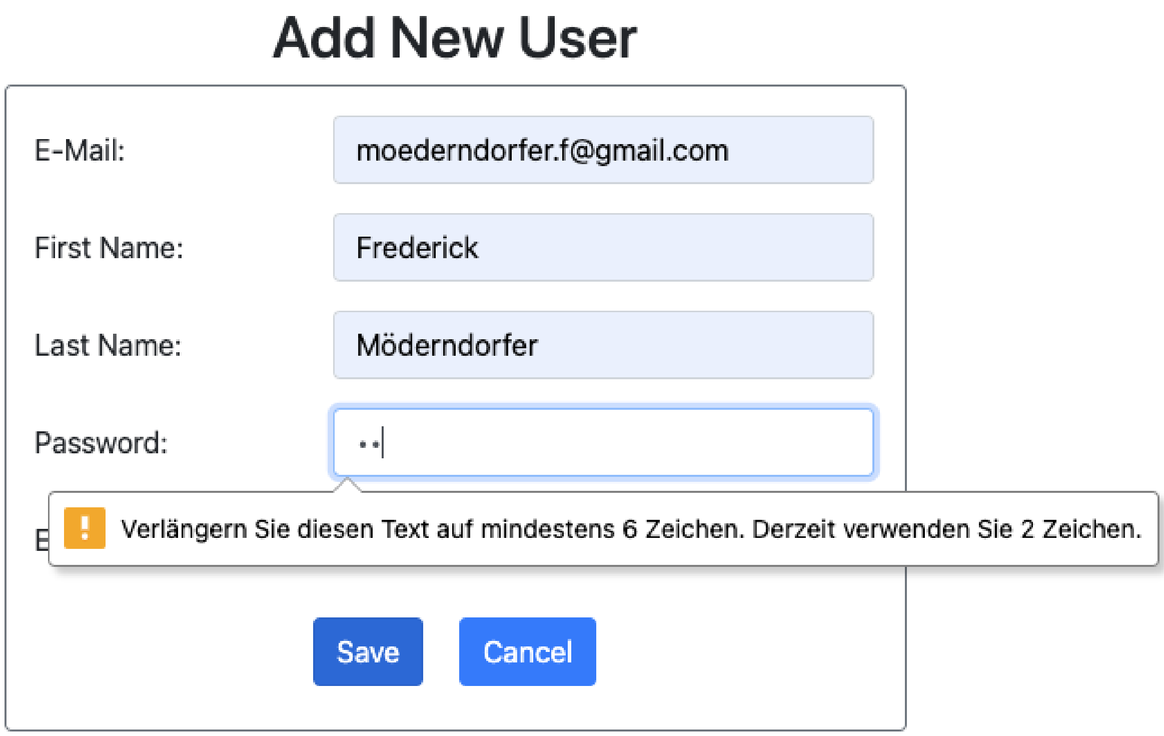


Add New User

E-Mail:	moederndorfer.f@gmail.com
First Name:	
Last Name:	
Password:	
Enabled:	☐

Füllen Sie dieses Feld aus.

Meldung



Verlängern Sie diesen Text auf mindestens 6 Zeichen. Derzeit verwenden Sie 2 Zeichen.

Meldung

```
<div class="text-center">
                <button type="submit" class="btn btn-primary m-2">Save</button>
                <button type="button" class="btn btn-primary m-2" onclick="cancelForm()">C
ancel</button>
            </div>
        </div>
    </form>
</div>
<script type="text/javascript">
    function cancelForm(){
window.location = "[[@{/users}]]";
    }
</script>
```

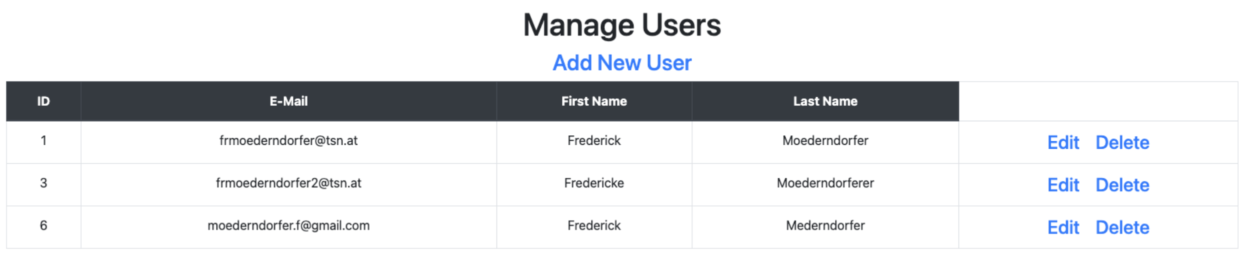
Durch diese zwei neuen Methoden können wir jetzt Daten hinzufügen ohne meiner Test Klasse.

Wichtig!! Beim ersten versuch entstanden bei mir Zwei neue Spalten mit firstName und lastName.

Das sollte eigentlich nicht sein aber es war kein Fehler im Code.

```
@PostMapping("/users/save")
    public String saveUser(User user){
        service.save(user);
        return "redirect:/users";
    }
}
```

```
public void save(User user) {
    repo.save(user);
}
```

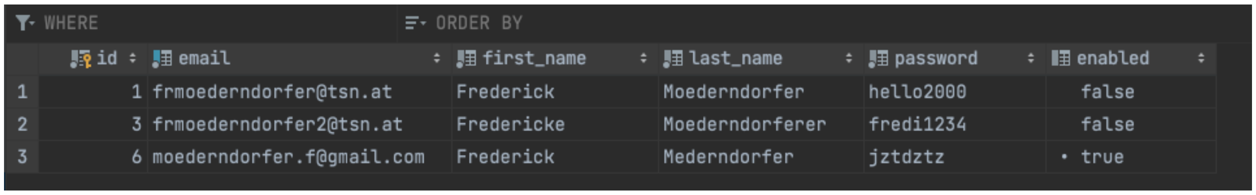


Manage Users

Add New User

ID	E-Mail	First Name	Last Name	
1	frmoederndorfer@tsn.at	Frederick	Moederndorfer	Edit Delete
3	frmoederndorfer2@tsn.at	Fredericke	Moederndorferer	Edit Delete
6	moederndorfer.f@gmail.com	Frederick	Mederndorfer	Edit Delete

Ausgabe nach dem einfügen.



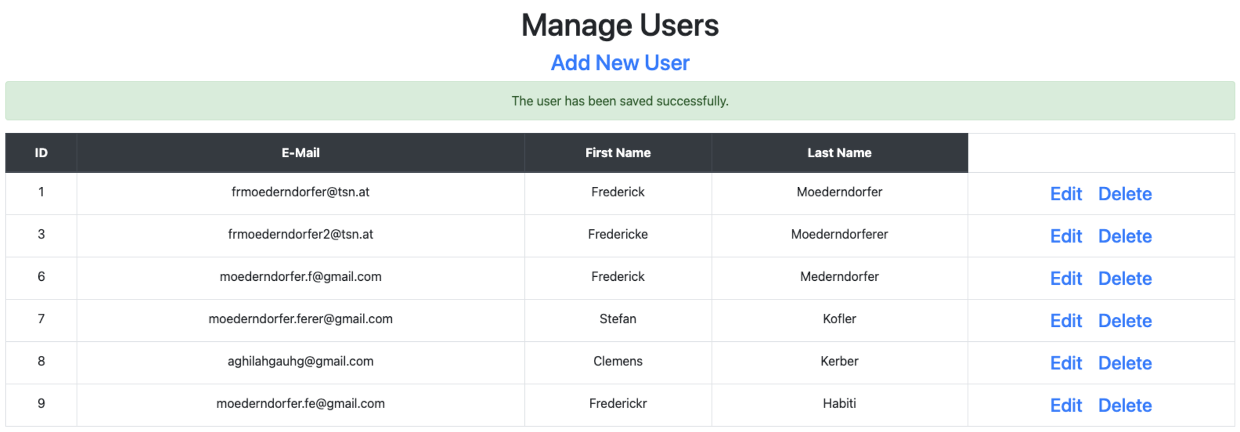
	id	email	first_name	last_name	password	enabled
1	1	frmoederndorfer@tsn.at	Frederick	Moederndorfer	hello2000	false
2	3	frmoederndorfer2@tsn.at	Fredericke	Moederndorferer	fredi1234	false
3	6	moederndorfer.f@gmail.com	Frederick	Mederndorfer	jztdztz	true

Datenbank anzeige über Intellij

Code Add User Function

```
<div class="container-fluid text-center">
    <div><h1>Manage Users</h1></div>
    <div class="m-2">
        <a class="h3" th:href="@{/users/new}">Add New User</a>
    </div>

    <div th:if="${message}" class="alert alert-success text-center">[[${message}]]</div>

    <div>
        <table class="table table-bordered">
            <thead class="thead-dark">
            <tr>
```

```
@PostMapping("/users/save")
public String saveUser(User user, RedirectAttributes ra){
    service.save(user);
    ra.addFlashAttribute("message", "The user has been saved successfully.");
    return "redirect:/users";
}
```


Manage Users

Add New User

The user has been saved successfully.

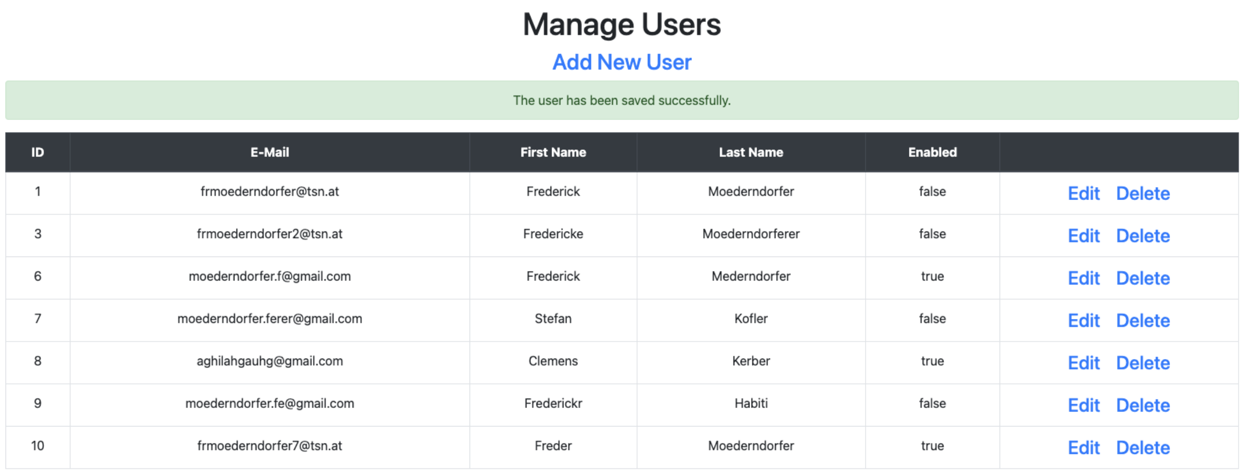
ID	E-Mail	First Name	Last Name	
1	frmoederndorfer@tsn.at	Frederick	Moederndorfer	Edit Delete
3	frmoederndorfer2@tsn.at	Fredericke	Moederndorferer	Edit Delete
6	moederndorfer.f@gmail.com	Frederick	Mederndorfer	Edit Delete
7	moederndorfer.ferer@gmail.com	Stefan	Kofler	Edit Delete
8	aghilahgauhg@gmail.com	Clemens	Kerber	Edit Delete
9	moederndorfer.fe@gmail.com	Frederickr	Habiti	Edit Delete

Code Edit/ Update User Function

```
@PostMapping("/users/save")
public String saveUser(User user, RedirectAttributes ra){
    service.save(user);
    ra.addFlashAttribute("message", "The user has been saved successfully.");
    return "redirect:/users";
}
```

```
@GetMapping("/users/edit/{id}")
public String showEditForm(@PathVariable("id") Integer id, Model model, RedirectAttributes
ra){
    try {
        User user = service.get(id);
        model.addAttribute("user", user);
        model.addAttribute("pageTitle", "Edit User (ID: " + id + ")");

        return "user_form";
    } catch (UserNotFoundExeption e) {
        ra.addFlashAttribute("message", "The user has been saved successfully.");
        return "redirect:/users";
    }
```

```
<div class="container-fluid">
   <div class="text-center"> <h2>[[${pageTitel}]]</h2></div>

    <form th:action="@{/users/save}" method="post" th:object="${user}"
        style="max-width: 500px; margin: 0 auto;">
        <input type="hidden" th:field="*{id}">
        <div class="border border-secondary rounded p-3">
            <div class="form-group row">
                <label class="col-sm-4 col-form-label">E-Mail:</label>
                <div class="col-sm-8">
                    <input type="email" th:field="*{email}" class="form-control" required
 minlength="8" maxlength="45"/>
                </div>
            </div>
```


Manage Users

Add New User

The user has been saved successfully.

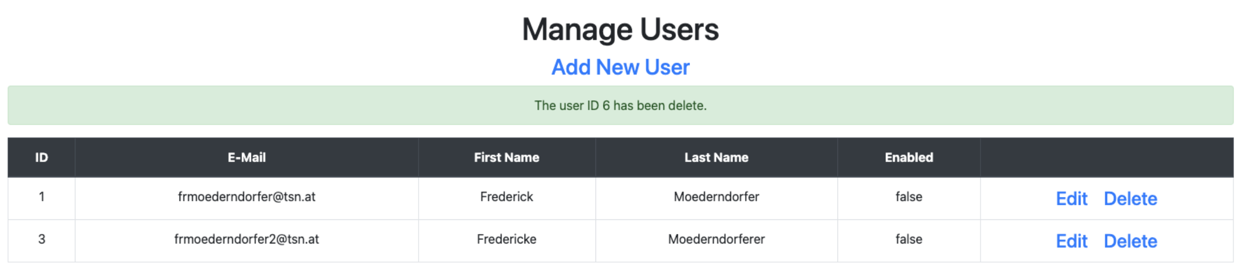
ID	E-Mail	First Name	Last Name	Enabled	
1	frmoederndorfer@tsn.at	Frederick	Moederndorfer	false	Edit Delete
3	frmoederndorfer2@tsn.at	Fredericke	Moederndorferer	false	Edit Delete
6	moederndorfer.f@gmail.com	Frederick	Mederndorfer	true	Edit Delete
7	moederndorfer.ferer@gmail.com	Stefan	Kofler	false	Edit Delete
8	aghilahgauhg@gmail.com	Clemens	Kerber	true	Edit Delete
9	moederndorfer.fe@gmail.com	Frederickr	Habiti	false	Edit Delete
10	frmoederndorfer7@tsn.at	Freder	Moederndorfer	true	Edit Delete

Code Delete User Function

```
@GetMapping("/users/edit/{id}")
public String showEditForm(@PathVariable("id") Integer id, Model model, RedirectAttributes
ra){
    try {
        User user = service.get(id);
        model.addAttribute("user", user);
        model.addAttribute("pageTitle", "Edit User (ID: " + id + ")");

        return "user_form";
    } catch (UserNotFoundExeption e) {
        ra.addFlashAttribute("message", e.getMessage());
        return "redirect:/users";
    }
}
```

```
@GetMapping("/users/delete/{id}")
    public String deleteUser(@PathVariable("id") Integer id, RedirectAttributes ra){
        try {
            service.delete(id);
        } catch (UserNotFoundExeption e) {
            ra.addFlashAttribute("message", e.getMessage());
        }
        return "redirect:/users";
    }
```

```
public void delete(Integer id) throws UserNotFoundExeption {
    Long count = repo.countById(id);
    if(count == null || count == 0){
        throw new UserNotFoundExeption("Could not find any users with ID " + id);
    }
    repo.deleteById(id);
}
```



Manage Users

Add New User

The user ID 6 has been delete.

ID	E-Mail	First Name	Last Name	Enabled	
1	frmoederndorfer@tsn.at	Frederick	Moederndorfer	false	Edit Delete
3	frmoederndorfer2@tsn.at	Fredericke	Moederndorferer	false	Edit Delete