

Thymeleaf - Update Employee



Update Employee - Demo

Employee Directory			
First Name	Last Name	Email	Action
Leslie	Andrews	leslie@luv2code.com	<button>Update</button>
Emma	Baumgarten	emma@luv2code.com	<button>Update</button>
Avani	Gupta	avani@luv2code.com	<button>Update</button>
Yuri	Petrov	yuri@luv2code.com	<button>Update</button>
Juan	Vega	juan@luv2code.com	<button>Update</button>

Update Employee

Step-By-Step

Update Employee

Step-By-Step

Employee Directory

Add Employee

First Name	Last Name	Email	Action
Leslie	Andrews	leslie@luv2code.com	Update
Emma	Baumgarten	emma@luv2code.com	Update
Avani	Gupta	avani@luv2code.com	Update
Yuri	Petrov	yuri@luv2code.com	Update
Juan	Vega	juan@luv2code.com	Update

Update Employee

Step-By-Step

1. "Update" button

Employee Directory			
Add Employee			
First Name	Last Name	Email	Action
Leslie	Andrews	leslie@luv2code.com	Update
Emma	Baumgarten	emma@luv2code.com	Update
Avani	Gupta	avani@luv2code.com	Update
Yuri	Petrov	yuri@luv2code.com	Update
Juan	Vega	juan@luv2code.com	Update

Update Employee

Step-By-Step

1. "Update" button

2. Pre-populate the form

Employee Directory			
Add Employee			
First Name	Last Name	Email	Action
Leslie	Andrews	leslie@luv2code.com	Update
Emma	Baumgarten	emma@luv2code.com	Update
Avani	Gupta	avani@luv2code.com	Update
Yuri	Petrov	yuri@luv2code.com	Update
Juan	Vega	juan@luv2code.com	Update

Update Employee

Step-By-Step

1. "Update" button

2. Pre-populate the form

3. Process form data

Employee Directory			
Add Employee			
First Name	Last Name	Email	Action
Leslie	Andrews	leslie@luv2code.com	<button>Update</button>
Emma	Baumgarten	emma@luv2code.com	<button>Update</button>
Avani	Gupta	avani@luv2code.com	<button>Update</button>
Yuri	Petrov	yuri@luv2code.com	<button>Update</button>
Juan	Vega	juan@luv2code.com	<button>Update</button>

Update Employee

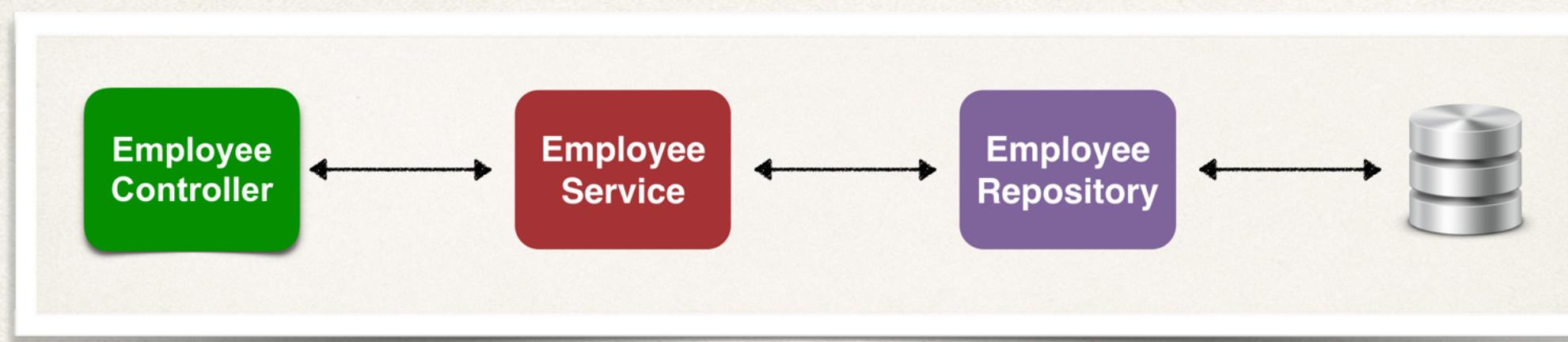
Step-By-Step

1. "Update" button

2. Pre-populate the form

3. Process form data

Employee Directory			
First Name	Last Name	Email	Action
Leslie	Andrews	leslie@luv2code.com	<button>Update</button>
Emma	Baumgarten	emma@luv2code.com	<button>Update</button>
Avani	Gupta	avani@luv2code.com	<button>Update</button>
Yuri	Petrov	yuri@luv2code.com	<button>Update</button>
Juan	Vega	juan@luv2code.com	<button>Update</button>



Step 1: "Update" Button

Step 1: "Update" Button

Employee Directory

Add Employee

First Name	Last Name	Email	Action
Leslie	Andrews	leslie@luv2code.com	<button>Update</button>
Emma	Baumgarten	emma@luv2code.com	<button>Update</button>
Avani	Gupta	avani@luv2code.com	<button>Update</button>
Yuri	Petrov	yuri@luv2code.com	<button>Update</button>
Juan	Vega	juan@luv2code.com	<button>Update</button>

Step 1: "Update" Button

Employee Directory

Add Employee

First Name	Last Name	Email	Action
Leslie	Andrews	leslie@luv2code.com	Update
Emma	Baumgarten	emma@luv2code.com	Update
Avani	Gupta	avani@luv2code.com	Update
Yuri	Petrov	yuri@luv2code.com	Update
Juan	Vega	juan@luv2code.com	Update

Each row has an **Update** link

- current employee id embedded in link

When **clicked**

- will load the employee from database
- prepopulate the form

Step 1: "Update" button

First Name	Last Name	Email	Action
Leslie	Andrews	leslie@luv2code.com	<button>Update</button>
Emma	Raumgarten	emma@luv2code.com	<button>Update</button>

Step 1: "Update" button

- Update button includes employee id

First Name	Last Name	Email	Action
Leslie	Andrews	leslie@luv2code.com	<button>Update</button>
Emma	Raumgarten	emma@luv2code.com	<button>Update</button>

Step 1: "Update" button

- Update button includes employee id

First Name	Last Name	Email	Action
Leslie	Andrews	leslie@luv2code.com	<button>Update</button>
Emma	Raumgarten	emma@luv2code.com	<button>Update</button>

```
<tr th:each="tempEmployee : ${employees}">
```

```
...
```

Step 1: "Update" button

- Update button includes employee id

First Name	Last Name	Email	Action
Leslie	Andrews	leslie@luv2code.com	<button>Update</button>
Emma	Raumgarten	emma@luv2code.com	<button>Update</button>

```
<tr th:each="tempEmployee : ${employees}">  
...  
<td>  
  
    <a th:href="@{/employees/showFormForUpdate(employeeId=${tempEmployee.id})}"  
        class="btn btn-info btn-sm">  
        Update  
    </a>  
  
</td>  
  
</tr>
```

Step 1: "Update" button

- Update button includes employee id

First Name	Last Name	Email	Action
Leslie	Andrews	leslie@luv2code.com	<button>Update</button>
Emma	Raumgarten	emma@luv2code.com	<button>Update</button>

```
<tr th:each="tempEmployee : ${employees}">  
...  
<td>  
  
    <a th:href="@{/employees/showFormForUpdate(employeeId=${tempEmployee.id})}"  
        class="btn btn-info btn-sm">  
        Update  
    </a>  
</td>  
</tr>
```

Appends to URL

?employeeId=xxx

Step 2: Pre-populate Form

Step 2: Pre-populate Form

```
@Controller  
@RequestMapping("/employees")  
public class EmployeeController {  
    ...
```

Step 2: Pre-populate Form

```
@Controller  
@RequestMapping("/employees")  
public class EmployeeController {  
    ...  
  
    @GetMapping("/showFormForUpdate")  
    public String showFormForUpdate(@RequestParam("employeeId") int theId,  
                                    Model theModel) {
```

Step 2: Pre-populate Form

```
@Controller  
@RequestMapping("/employees")  
public class EmployeeController {  
    ...  
  
    @GetMapping("/showFormForUpdate")  
    public String showFormForUpdate(@RequestParam("employeeId") int theId,  
                                    Model theModel) {  
        // Load the employee from the database  
        Employee tempEmployee = employeeService.getEmployee(theId);  
        theModel.addAttribute("employee", tempEmployee);  
  
        // Create the update link  
        Edit Employee Information  
    }  
}
```

Step 2: Pre-populate Form

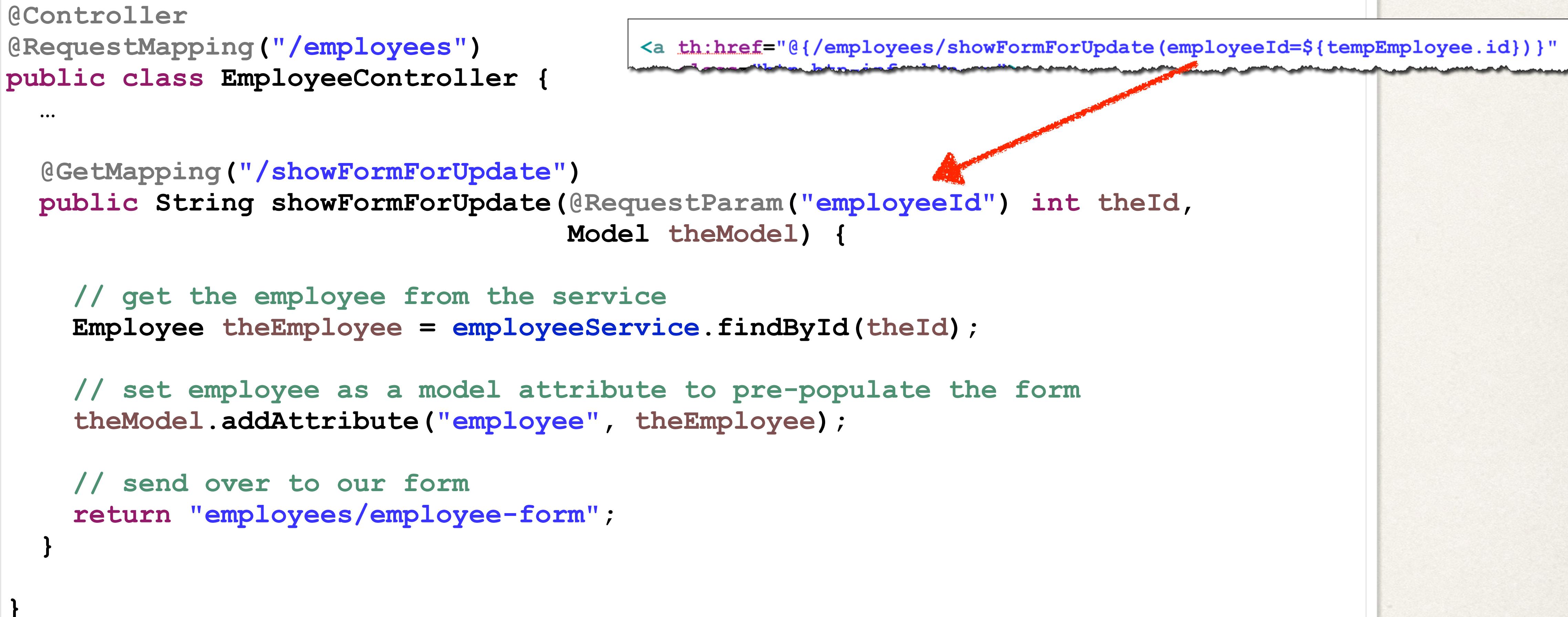
```
@Controller  
@RequestMapping("/employees")  
public class EmployeeController {  
    ...  
  
    @GetMapping("/showFormForUpdate")  
    public String showFormForUpdate(@RequestParam("employeeId") int theId,  
                                    Model theModel) {  
  
        // get the employee from the service  
        Employee theEmployee = employeeService.findById(theId);  
  
        Link to the update form  
    }  
}
```

Step 2: Pre-populate Form

```
@Controller  
@RequestMapping("/employees")  
public class EmployeeController {  
    ...  
  
    @GetMapping("/showFormForUpdate")  
    public String showFormForUpdate(@RequestParam("employeeId") int theId,  
                                    Model theModel) {  
  
        // get the employee from the service  
        Employee theEmployee = employeeService.findById(theId);  
  
        // set employee as a model attribute to pre-populate the form  
        theModel.addAttribute("employee", theEmployee);  
  
        <a th:href="@{/employees/showFormForUpdate(employeeId=${tempEmployee.id})}"  
           th:text="Edit Employee Information" />  
    }  
}
```

Step 2: Pre-populate Form

```
@Controller  
@RequestMapping("/employees")  
public class EmployeeController {  
    ...  
  
    @GetMapping("/showFormForUpdate")  
    public String showFormForUpdate(@RequestParam("employeeId") int theId,  
                                    Model theModel) {  
  
        // get the employee from the service  
        Employee theEmployee = employeeService.findById(theId);  
  
        // set employee as a model attribute to pre-populate the form  
        theModel.addAttribute("employee", theEmployee);  
  
        // send over to our form  
        return "employees/employee-form";  
    }  
}
```



```
<a th:href="@{/employees/showFormForUpdate(employeeId=${tempEmployee.id})}"  
   data-th-link-type="button" data-th-link-method="GET">Edit Employee Information
```

Step 2: Pre-populate Form

Step 2: Pre-populate Form

```
<form action="#" th:action="@{/employees/save}"
      th:object="${employee}" method="POST">

    <!-- Add hidden form field to handle update -->
    <input type="hidden" th:field="*{id}" />

    <input type="text" th:field="*{firstName}"
           class="form-control mb-4 col-4" placeholder="First name">

    <input type="text" th:field="*{lastName}"
           class="form-control mb-4 col-4" placeholder="Last name">

    <input type="text" th:field="*{email}"
           class="form-control mb-4 col-4" placeholder="Email">

    <button type="submit" class="btn btn-info col-2">Save</button>

</form>
```

Step 2: Pre-populate Form

1

When form is **loaded**,
will call:

employee.getFirstName()

...

employee.getLastName

```
<form action="#" th:action="@{/employees/save}"
      th:object="${employee}" method="POST">

    <!-- Add hidden form field to handle update -->
    <input type="hidden" th:field="*{id}" />

    <input type="text" th:field="*{firstName}"
           class="form-control mb-4 col-4" placeholder="First name">

    <input type="text" th:field="*{lastName}"
           class="form-control mb-4 col-4" placeholder="Last name">

    <input type="text" th:field="*{email}"
           class="form-control mb-4 col-4" placeholder="Email">

    <button type="submit" class="btn btn-info col-2">Save</button>

</form>
```

Step 2: Pre-populate Form

1

When form is **loaded**,
will call:

employee.getFirstName()

...

employee.getLastName

```
<form action="#" th:action="@{/employees/save}"
      th:object="${employee}" method="POST">

    <!-- Add hidden form field to handle update -->
    <input type="hidden" th:field="*{id}" />

    <input type="text" th:field="*{firstName}"
           class="form-control mb-4 col-4" placeholder="First name">

    <input type="text" th:field="*{lastName}"
           class="form-control mb-4 col-4" placeholder="Last name">

    <input type="text" th:field="*{email}"
           class="form-control mb-4 col-4" placeholder="Email">

    <button type="submit" class="btn btn-info col-2">Save</button>

</form>
```

This is how form is
pre-populated
Thanks to calls to getters

Step 2: Pre-populate Form

```
<form action="#" th:action="@{/employees/save}"
      th:object="${employee}" method="POST">

    <!-- Add hidden form field to handle update -->
    <input type="hidden" th:field="*{id}" />

    <input type="text" th:field="*{firstName}"
           class="form-control mb-4 col-4" placeholder="First name">

    <input type="text" th:field="*{lastName}"
           class="form-control mb-4 col-4" placeholder="Last name">

    <input type="text" th:field="*{email}"
           class="form-control mb-4 col-4" placeholder="Email">

    <button type="submit" class="btn btn-info col-2">Save</button>

</form>
```

Step 2: Pre-populate Form

```
<form action="#" th:action="@{/employees/save}"
      th:object="${employee}" method="POST">
    <!-- Add hidden form field to handle update -->
    <input type="hidden" th:field="*{id}" />

    <input type="text" th:field="*{firstName}"
           class="form-control mb-4 col-4" placeholder="First name">

    <input type="text" th:field="*{lastName}"
           class="form-control mb-4 col-4" placeholder="Last name">

    <input type="text" th:field="*{email}"
           class="form-control mb-4 col-4" placeholder="Email">

    <button type="submit" class="btn btn-info col-2">Save</button>

</form>
```

Hidden form field
required for updates

Step 2: Pre-populate Form

```
<form action="#" th:action="@{/employees/save}"
      th:object="${employee}" method="POST">
    <!-- Add hidden form field to handle update -->
    <input type="hidden" th:field="*{id}" />

    <input type="text" th:field="*{firstName}"
           class="form-control mb-4 col-4" placeholder="First name">

    <input type="text" th:field="*{lastName}"
           class="form-control mb-4 col-4" placeholder="Last name">

    <input type="text" th:field="*{email}"
           class="form-control mb-4 col-4" placeholder="Email address">

    <button type="submit" class="btn btn-info col-4" value="Save">Save</button>
</form>
```

Hidden form field
required for updates

This binds to the model attribute

Tells your app
which employee to update

Step 3: Process form data to save employee

Step 3: Process form data to save employee

- No need for new code ... we can reuse our existing code

Step 3: Process form data to save employee

- No need for new code ... we can reuse our existing code
- Works the same for add or update :-)

```
@Controller  
@RequestMapping("/employees")  
public class EmployeeController {  
    ...  
  
    @PostMapping("/save")  
    public String saveEmployee(@ModelAttribute("employee") Employee theEmployee) {  
  
        // save the employee  
        employeeService.save(theEmployee);  
  
        // use a redirect to prevent duplicate submissions  
        return "redirect:/employees/list";  
    }  
    ...  
}
```

