

Create HTML Tables with Thymeleaf



HTML Tables

Start with plain table
Will add CSS in later videos

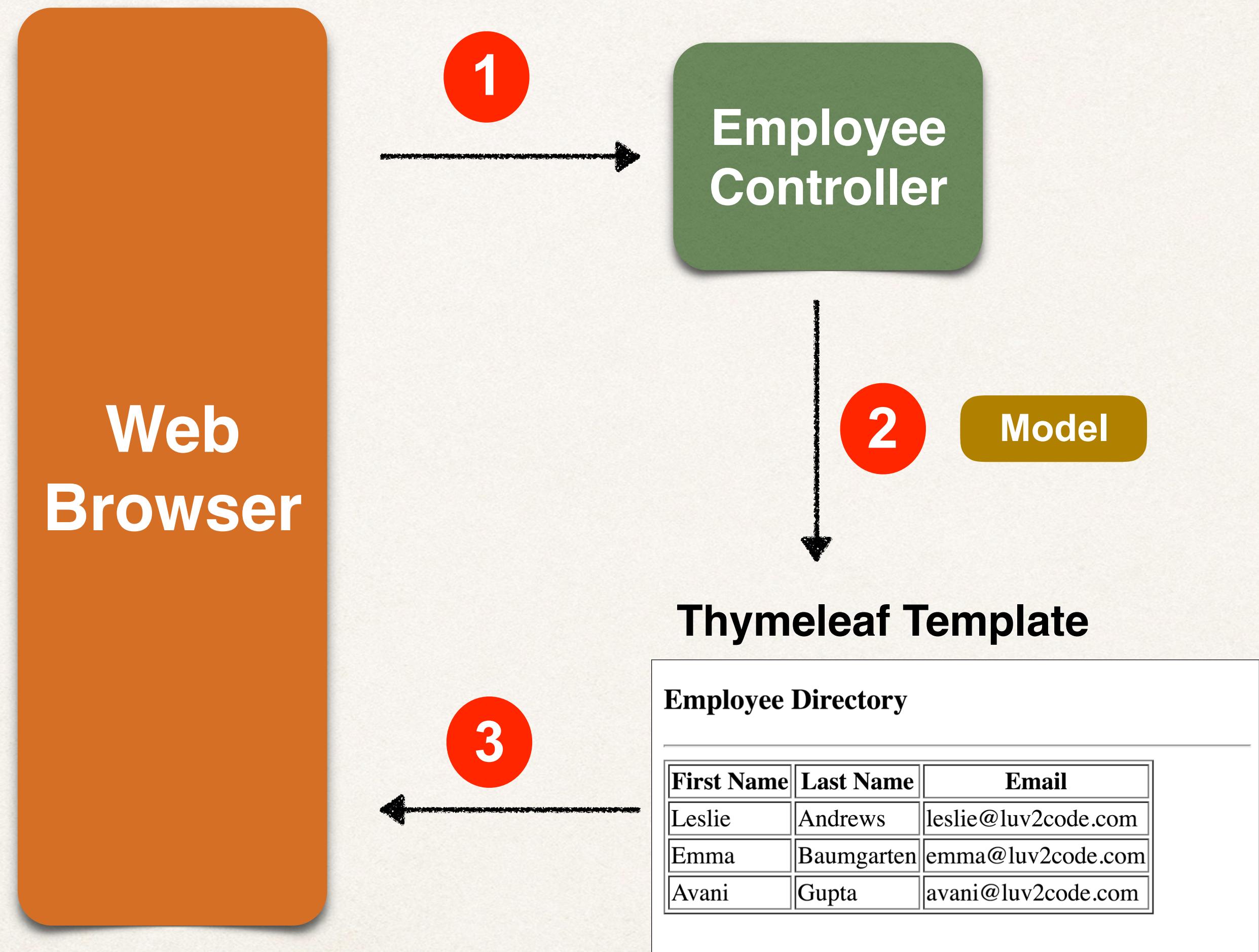
HTML Tables

Start with plain table
Will add CSS in later videos

Employee Directory

First Name	Last Name	Email
Leslie	Andrews	leslie@luv2code.com
Emma	Baumgarten	emma@luv2code.com
Avani	Gupta	avani@luv2code.com

Big Picture



Development Process

Step-By-Step

Development Process

Step-By-Step

1. Create Employee.java

Development Process

Step-By-Step

1. Create Employee.java
2. Create EmployeeController.java

Development Process

Step-By-Step

1. Create Employee.java
2. Create EmployeeController.java
3. Create Thymeleaf template

Step 1: Create Employee.java

```
public class Employee {

    private int id;
    private String firstName;
    private String lastName;
    private String email;

    public Employee() {
    }

    public Employee(int id, String firstName, String lastName, String email) {
        this.id = id;
        this.firstName = firstName;
        this.lastName = lastName;
        this.email = email;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }
}
```

Step 1: Create Employee.java

- Regular Java class: fields, constructors, getters / setters

```
public class Employee {

    private int id;
    private String firstName;
    private String lastName;
    private String email;

    public Employee() {
    }

    public Employee(int id, String firstName, String lastName, String email) {
        this.id = id;
        this.firstName = firstName;
        this.lastName = lastName;
        this.email = email;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }
}
```

Step 2: Create EmployeeController.java

Step 2: Create EmployeeController.java

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

Step 2: Create EmployeeController.java

```
@Controller  
@RequestMapping("/employees")  
public class EmployeeController {  
  
    @GetMapping("/list")  
    public String listEmployees(Model theModel) {
```

Step 2: Create EmployeeController.java

```
@Controller  
@RequestMapping("/employees")  
public class EmployeeController {  
  
    @GetMapping("/list")  
    public String listEmployees(Model theModel) {  
  
        // create employees  
        Employee emp1 = new Employee(1, "Leslie", "Andrews", "leslie@luv2code.com");  
        Employee emp2 = new Employee(2, "Emma", "Baumgarten", "emma@luv2code.com");  
        Employee emp3 = new Employee(3, "Avani", "Gupta", "avani@luv2code.com");  
    }  
}
```

Will add database integration later

Step 2: Create EmployeeController.java

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

    @GetMapping("/list")
    public String listEmployees(Model theModel) {

        // create employees
        Employee emp1 = new Employee(1, "Leslie", "Andrews", "leslie@luv2code.com");
        Employee emp2 = new Employee(2, "Emma", "Baumgarten", "emma@luv2code.com");
        Employee emp3 = new Employee(3, "Avani", "Gupta", "avani@luv2code.com");

        // create the list
        List<Employee> theEmployees = new ArrayList<>();
```

Step 2: Create EmployeeController.java

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

    @GetMapping("/list")
    public String listEmployees(Model theModel) {

        // create employees
        Employee emp1 = new Employee(1, "Leslie", "Andrews", "leslie@luv2code.com");
        Employee emp2 = new Employee(2, "Emma", "Baumgarten", "emma@luv2code.com");
        Employee emp3 = new Employee(3, "Avani", "Gupta", "avani@luv2code.com");

        // create the list
        List<Employee> theEmployees = new ArrayList<>();

        // add to the list
        theEmployees.add(emp1);
        theEmployees.add(emp2);
        theEmployees.add(emp3);
    }
}
```

Step 2: Create EmployeeController.java

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

    @GetMapping("/list")
    public String listEmployees(Model theModel) {

        // create employees
        Employee emp1 = new Employee(1, "Leslie", "Andrews", "leslie@luv2code.com");
        Employee emp2 = new Employee(2, "Emma", "Baumgarten", "emma@luv2code.com");
        Employee emp3 = new Employee(3, "Avani", "Gupta", "avani@luv2code.com");

        // create the list
        List<Employee> theEmployees = new ArrayList<>();

        // add to the list
        theEmployees.add(emp1);
        theEmployees.add(emp2);
        theEmployees.add(emp3);

        // add to the spring model
        theModel.addAttribute("employees", theEmployees);
    }
}
```

Step 2: Create EmployeeController.java

```
@Controller  
@RequestMapping("/employees")  
public class EmployeeController {  
  
    @GetMapping("/list")  
    public String listEmployees(Model theModel) {  
  
        // create employees  
        Employee emp1 = new Employee(1, "Leslie", "Andrews", "leslie@luv2code.com");  
        Employee emp2 = new Employee(2, "Emma", "Baumgarten", "emma@luv2code.com");  
        Employee emp3 = new Employee(3, "Avani", "Gupta", "avani@luv2code.com");  
  
        // create the list  
        List<Employee> theEmployees = new ArrayList<>();  
  
        // add to the list  
        theEmployees.add(emp1);  
        theEmployees.add(emp2);  
        theEmployees.add(emp3);  
  
        // add to the spring model  
        theModel.addAttribute("employees", theEmployees);  
    }  
}
```

Our Thymleaf template will access this data

Step 2: Create EmployeeController.java

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

    @GetMapping("/list")
    public String listEmployees(Model theModel) {

        // create employees
        Employee emp1 = new Employee(1, "Leslie", "Andrews", "leslie@luv2code.com");
        Employee emp2 = new Employee(2, "Emma", "Baumgarten", "emma@luv2code.com");
        Employee emp3 = new Employee(3, "Avani", "Gupta", "avani@luv2code.com");

        // create the list
        List<Employee> theEmployees = new ArrayList<>();

        // add to the list
        theEmployees.add(emp1);
        theEmployees.add(emp2);
        theEmployees.add(emp3);

        // add to the spring model
        theModel.addAttribute("employees", theEmployees);

        return "list-employees";
    }
}
```

Our Thymleaf template will access this data

Step 2: Create EmployeeController.java

```
@Controller  
@RequestMapping("/employees")  
public class EmployeeController {  
  
    @GetMapping("/list")  
    public String listEmployees(Model theModel) {  
  
        // create employees  
        Employee emp1 = new Employee(1, "Leslie", "Andrews", "leslie@luv2code.com");  
        Employee emp2 = new Employee(2, "Emma", "Baumgarten", "emma@luv2code.com");  
        Employee emp3 = new Employee(3, "Avani", "Gupta", "avani@luv2code.com");  
  
        // create the list  
        List<Employee> theEmployees = new ArrayList<>();  
  
        // add to the list  
        theEmployees.add(emp1);  
        theEmployees.add(emp2);  
        theEmployees.add(emp3);  
  
        // add to the spring model  
        theModel.addAttribute("employees", theEmployees);  
  
        return "list-employees";  
    }  
}
```

Our Thymleaf template will access this data

src/main/resources/templates/list-employees.html

Step 3: Create Thymeleaf template

Step 3: Create Thymeleaf template

File: list-employees.html

```
<!DOCTYPE HTML>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
```

To use Thymeleaf
expressions

Step 3: Create Thymeleaf template

File: list-employees.html

```
<!DOCTYPE HTML>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
...
<body>

<h3>Employee Directory</h3>
<hr>
```

To use Thymeleaf
expressions

Employee Directory

First Name	Last Name	Email
Leslie	Andrews	leslie@luv2code.com
Emma	Baumgarten	emma@luv2code.com
Avani	Gupta	avani@luv2code.com

Step 3: Create Thymeleaf template

File: list-employees.html

```
<!DOCTYPE HTML>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
...
<body>

    <h3>Employee Directory</h3>
    <hr>

    <table border="1">
        <!-- Build HTML table based on employees -->
    </table>
```

To use Thymeleaf
expressions

Employee Directory

First Name	Last Name	Email
Leslie	Andrews	leslie@luv2code.com
Emma	Baumgarten	emma@luv2code.com
Avani	Gupta	avani@luv2code.com

Step 3: Create Thymeleaf template

File: list-employees.html

```
<!DOCTYPE HTML>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
...
<body>

    <h3>Employee Directory</h3>
    <hr>

    <table border="1">
        <!-- Build HTML table based on employees -->
    </table>
```

To Do
Add code to loop over employees

To use Thymeleaf
expressions

Employee Directory

First Name	Last Name	Email
Leslie	Andrews	leslie@luv2code.com
Emma	Baumgarten	emma@luv2code.com
Avani	Gupta	avani@luv2code.com

Step 3: Create Thymeleaf template

File: list-employees.html

```
<!DOCTYPE HTML>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
...
<body>

    <h3>Employee Directory</h3>
    <hr>

    <table border="1">
        <!-- Build HTML table based on employees -->
    </table>

</body>
</html>
```

To use Thymeleaf
expressions

Employee Directory

First Name	Last Name	Email
Leslie	Andrews	leslie@luv2code.com
Emma	Baumgarten	emma@luv2code.com
Avani	Gupta	avani@luv2code.com

To Do
Add code to loop over employees

Step 3: Create Thymeleaf template

Step 3: Create Thymeleaf template

File: list-employees.html

```
<table border="1">
```

Employee Directory

First Name	Last Name	Email
Leslie	Andrews	leslie@luv2code.com
Emma	Baumgarten	emma@luv2code.com
Avani	Gupta	avani@luv2code.com

Step 3: Create Thymeleaf template

File: list-employees.html

```
<table border="1">
<thead>
<tr>
<th>First Name</th>
<th>Last Name</th>
<th>Email</th>
</tr>
</thead>
```

Employee Directory

First Name	Last Name	Email
Leslie	Andrews	leslie@luv2code.com
Emma	Baumgarten	emma@luv2code.com
Avani	Gupta	avani@luv2code.com

Step 3: Create Thymeleaf template

File: list-employees.html

```
<table border="1">
  <thead>
    <tr>
      <th>First Name</th>
      <th>Last Name</th>
      <th>Email</th>
    </tr>
  </thead>
```

Employee Directory

First Name	Last Name	Email
Leslie	Andrews	leslie@luv2code.com
Emma	Baumgarten	emma@luv2code.com
Avani	Gupta	avani@luv2code.com

Step 3: Create Thymeleaf template

File: list-employees.html

```
<table border="1">
  <thead>
    <tr>
      <th>First Name</th>
      <th>Last Name</th>
      <th>Email</th>
    </tr>
  </thead>

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

Employee Directory

First Name	Last Name	Email
Leslie	Andrews	leslie@luv2code.com
Emma	Baumgarten	emma@luv2code.com
Avani	Gupta	avani@luv2code.com

Step 3: Create Thymeleaf template

File: list-employees.html

```
<table border="1">
  <thead>
    <tr>
      <th>First Name</th>
      <th>Last Name</th>
      <th>Email</th>
    </tr>
  </thead>

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

Loop over
list of employees

Employee Directory

First Name	Last Name	Email
Leslie	Andrews	leslie@luv2code.com
Emma	Baumgarten	emma@luv2code.com
Avani	Gupta	avani@luv2code.com

Step 3: Create Thymeleaf template

File: list-employees.html

```
<table border="1">
  <thead>
    <tr>
      <th>First Name</th>
      <th>Last Name</th>
      <th>Email</th>
  
```

Loop parameter

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

Loop over
list of employees

Employee Directory

First Name	Last Name	Email
Leslie	Andrews	leslie@luv2code.com
Emma	Baumgarten	emma@luv2code.com
Avani	Gupta	avani@luv2code.com

Step 3: Create Thymeleaf template

File: list-employees.html

```
<table border="1">
  <thead>
    <tr>
      <th>First Name</th>
      <th>Last Name</th>
      <th>Email</th>
```

Loop parameter

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

Loop over
list of employees

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

    @GetMapping("/list")
    public String listEmployees(Model theModel) {

        // create employees
        // create the list
        // add to the list
        ...

        // add to the spring model
        theModel.addAttribute("employees", theEmployees);

        return "list-employees";
    }
}
```

Employee Directory

First Name	Last Name	Email
Leslie	Andrews	leslie@luv2code.com
Emma	Baumgarten	emma@luv2code.com
Avani	Gupta	avani@luv2code.com

Step 3: Create Thymeleaf template

File: list-employees.html

```
<table border="1">
  <thead>
    <tr>
      <th>First Name</th>
      <th>Last Name</th>
      <th>Email</th>
```

Loop parameter

```
<tbody>
  <tr th:each="tempEmployee : ${employees}">
    <td th:text="${tempEmployee.firstName}" />
    <td th:text="${tempEmployee.lastName}" />
    <td th:text="${tempEmployee.email}" />
  </tr>
</tbody>
</table>
```

Loop over
list of employees

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

    @GetMapping("/list")
    public String listEmployees(Model theModel) {

        // create employees
        // create the list
        // add to the list
        ...

        // add to the spring model
        theModel.addAttribute("employees", theEmployees);

        return "list-employees";
    }
}
```

Employee Directory

First Name	Last Name	Email
Leslie	Andrews	leslie@luv2code.com
Emma	Baumgarten	emma@luv2code.com
Avani	Gupta	avani@luv2code.com