

# Thymeleaf and Validations - Exercise

## 1. Functionality Overview

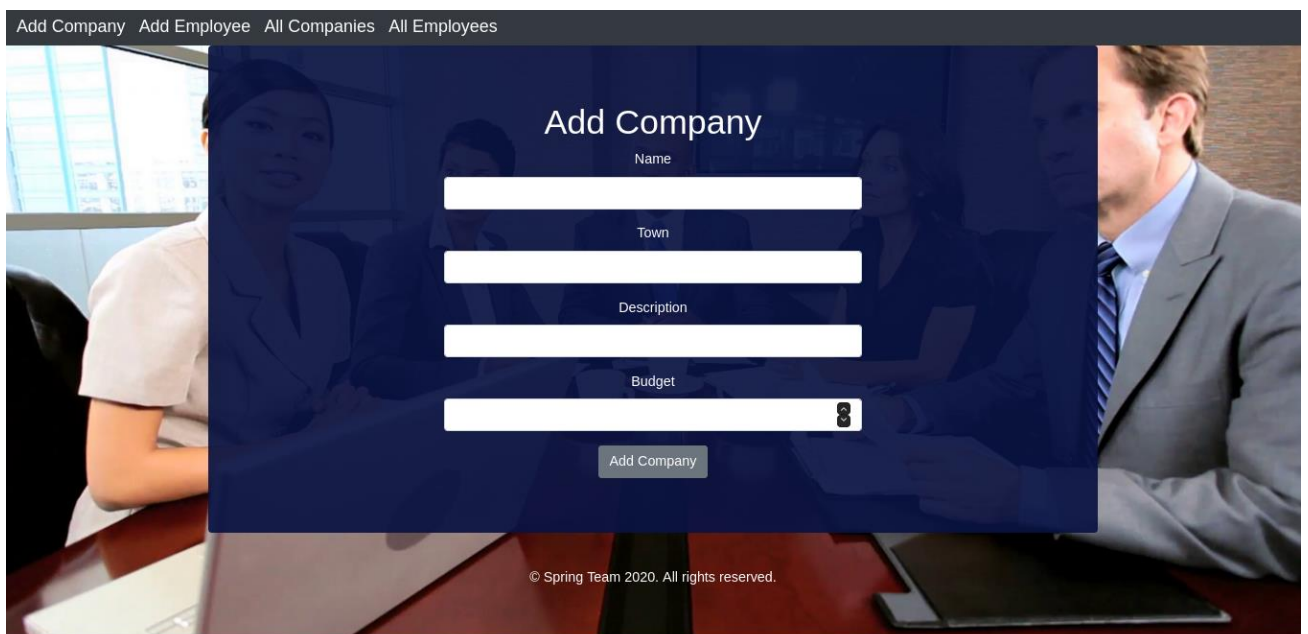
The application should be able to easily **accept** hard-formatted. The application is called – **LinkedOut**.

Look at the pictures below to see what must happen:

- `route("/")`



- `route("/companies/add")`



# Add Company

Name

A

Name length must be between 2 and 10

Town

A

Town length must be between 2 and 10

Description

Invalid

Description length must be more than 10

Budget

-1

- `route("/employees/add")`

Add Company Add Employee All Companies All Employees

## Add Employee

First Name

Last Name

Choose Education Level

☐ Master

☐ Bachelor

☐ Secondary

All available companies

Job Title

Birth Date

mm / dd / yyyy

Salary

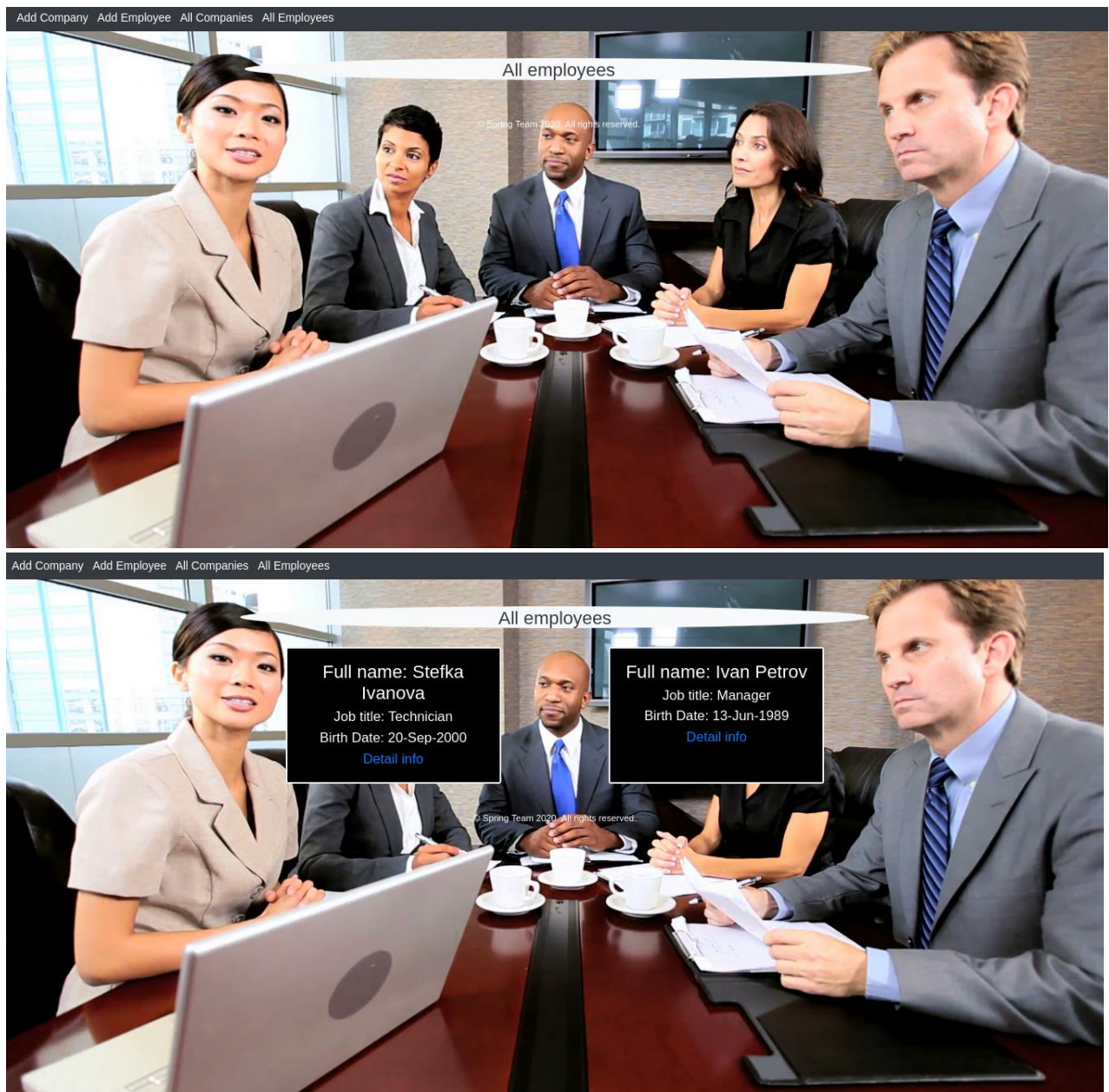
Add Employee

- `route("/companies/all")`





- `route("/employees/all")`



## 2. Model Definition

There are 2 main models that the **LinkedOut database** application should contain in its functionality.

Design them in the **most appropriate** way, considering the following **data constraints**:

### Company

- **id** – a **char sequence**
- **budget** – a **number**. (must be a positive number). Cannot be **null**.
- **description** – a **very long char sequence**. Cannot be **null**. Must be **at least 10 characters**.
- **name** – a **char sequence**. Must be **unique** and cannot be **null**. Must be **between 2 and 10 characters**.
- **town** – a **char sequence**. Cannot be **null**. Must be **between 2 and 10 characters**.

## Employee

- **id** – a **char sequence**
- **birthdate** – a **date**. Cannot be **null**.
- **educationLevel** – a **char sequence**. Cannot be **null**.
- **firstName** – a **char sequence**. Cannot be **null**. Must be at least **2 characters**. Cannot be **null**.
- **jobTitle** – a **char sequence**. Cannot be **null**.
- **lastName** – a **char sequence**. Cannot be **null**. Must be at least **2 characters**. Cannot be **null**.
- **salary** – a **number** (must be a positive number). Cannot be **null**.

**NOTE:** Name the entities and their class members, **exactly** in the **format stated** above. Do not name them in snake case with the dashes, of course.

## Relationships

Your partners gave you a little hint about the more complex relationships in the database, so that you can implement it correctly.

One **Employee** may be in only one **Company**, and one **Company** may have many **Employees**.

