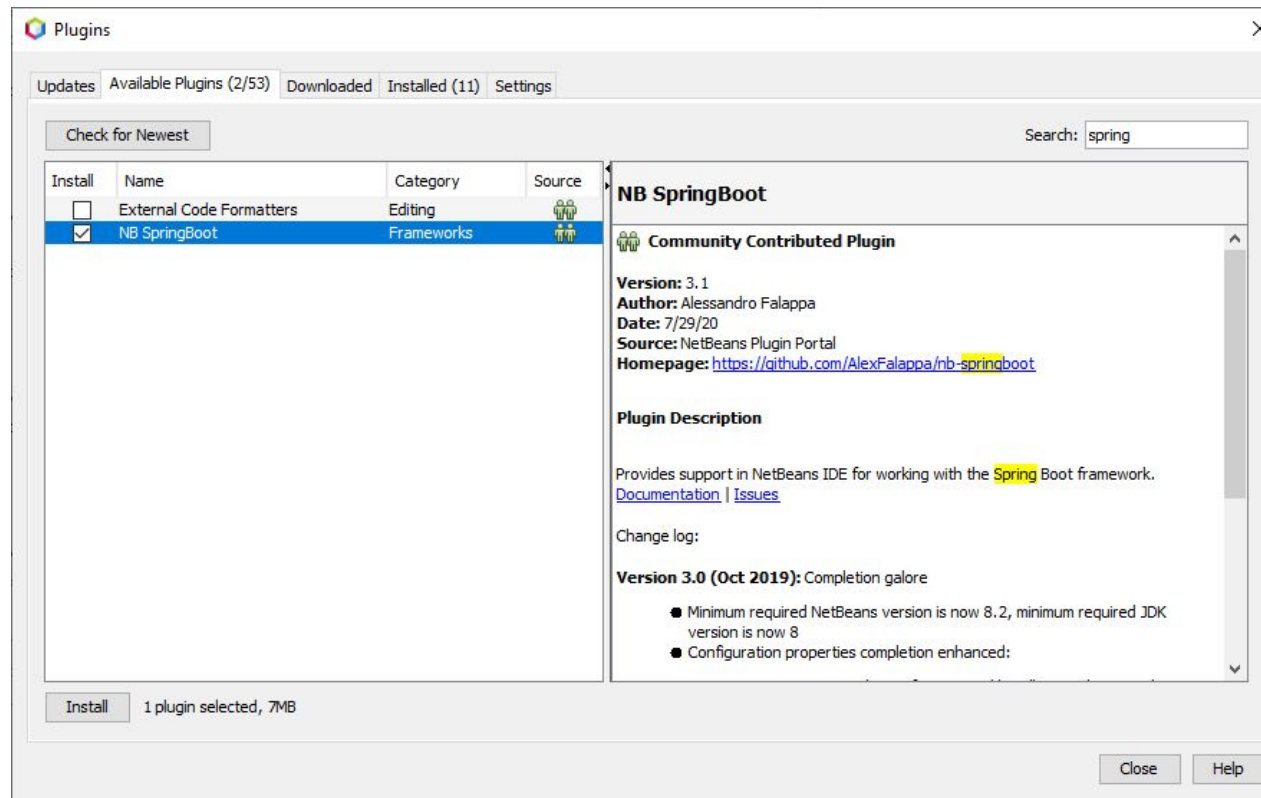


TH- Web app với Thymeleaf

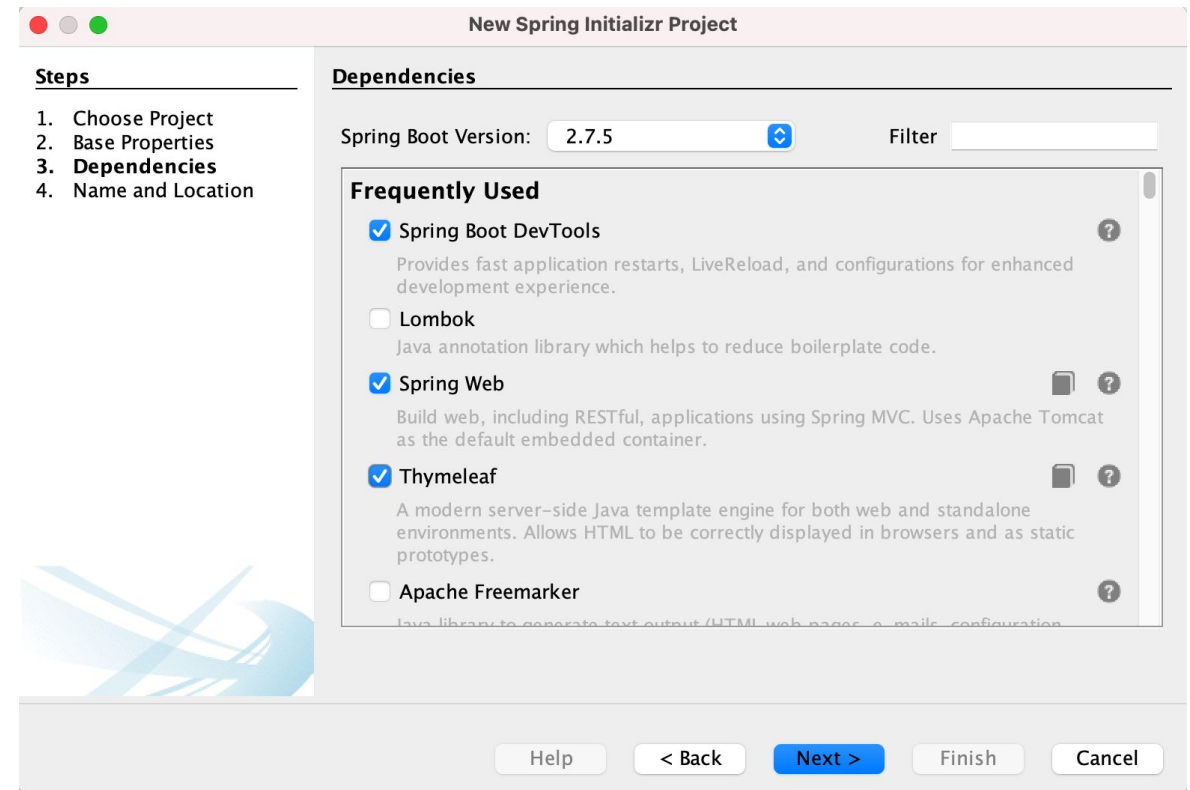
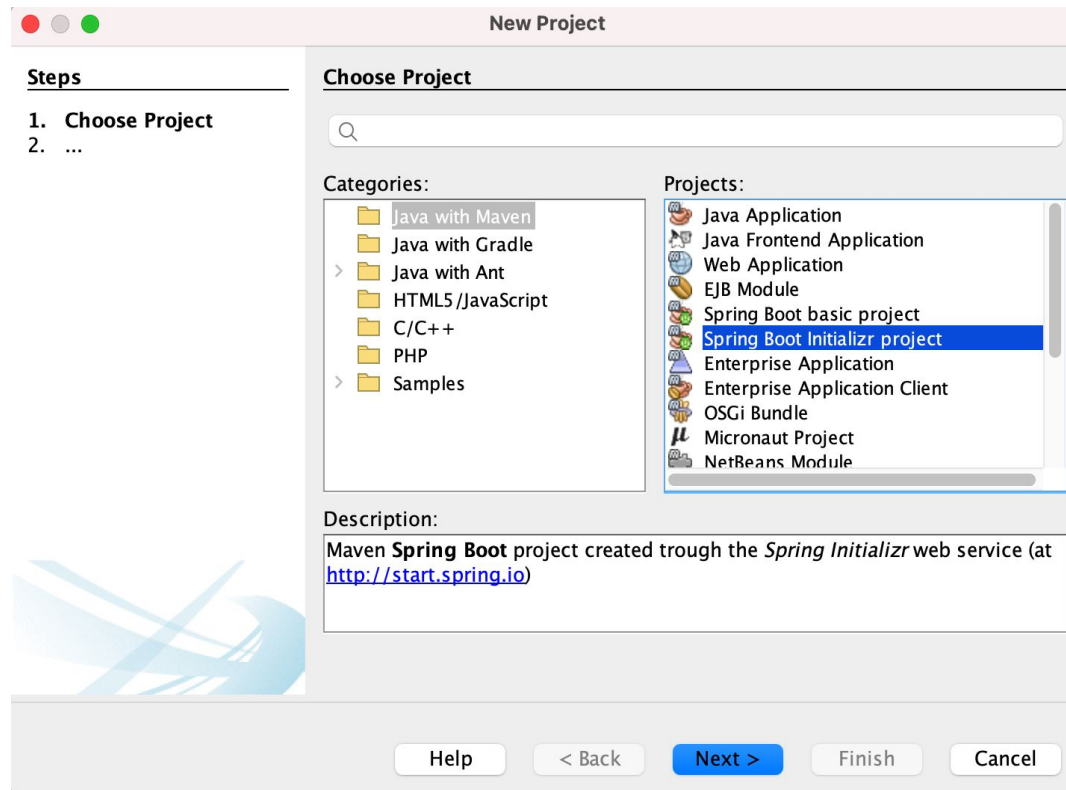
1. Cài đặt Spring Boot

- Cài đặt Plugin: NB SpringBoot



2. Tạo dự án web

- Tạo mới dự án Spring Boot Initializr project
- Thêm Dependencies: Web, Thymeleaf, Lombok

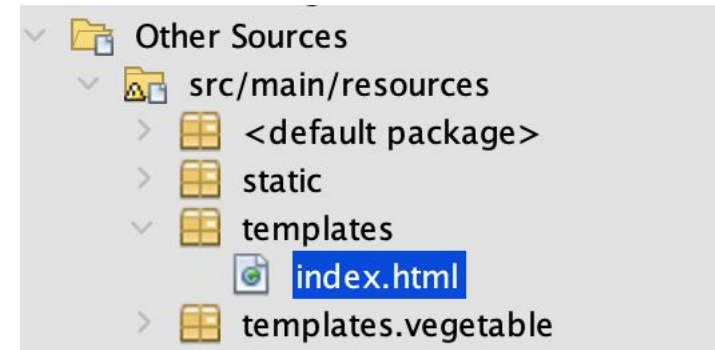


3. Tạo controller

- Tạo **HomeController**
 - Có 1 phương thức homepage □ trang index
 - @GetMapping ("/home"): địa chỉ url

```
@Controller
public class HomeController {
    @GetMapping("/home")
    public String homepage()
    {
        return "index";
    }
}
```

- Tạo **trang index**
 - Trong thư mục resources/templates, tạo trang index.html



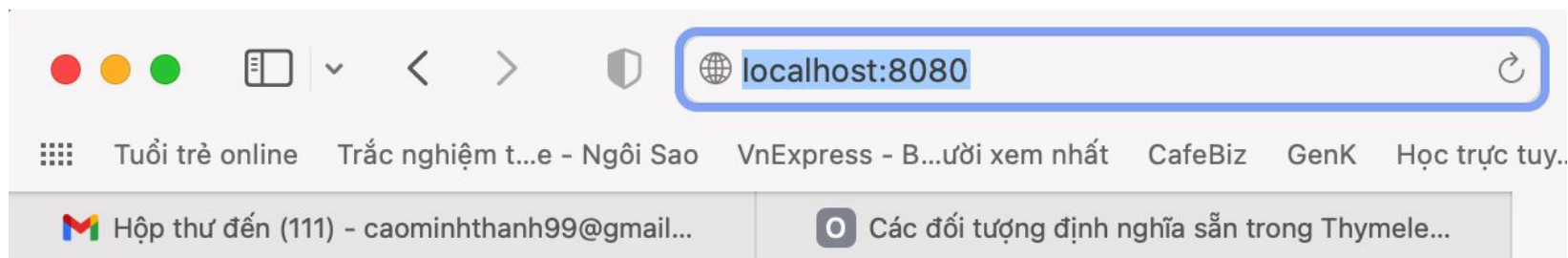
4. Chạy web

- Run lớp DemoSpringBootApplication (lớp do framework tự sinh ra)

```
1  package SpringBoots.demo;  
2  
3  import org.springframework.boot.SpringApplication;  
4  import org.springframework.boot.autoconfigure.EnableAutoConfiguration;  
5  import org.springframework.boot.autoconfigure.SpringBootApplication;  
6  
7  
8  @SpringBootApplication  
9  public class DemoSpringBootApplication {  
10  
11      public static void main(String[] args) {  
12          SpringApplication.run(DemoSpringBootApplication.class, args);  
13      }  
14  
15  }  
16
```

5. Chạy demo

`https://localhost:8080`



TODO write content

6. Tạo Model

```
@Data
public class Vegetable {

    private int VegetableID;

    private String VegetableName;

    private String Unit;

    private int Amount;

    private double Price;

    public Vegetable(int VegetableID, String VegetableName,
                    String Unit, int Amount, double Price) {
        this.VegetableID = VegetableID;
        this.VegetableName = VegetableName;
        this.Unit = Unit;
        this.Amount = Amount;
        this.Price = Price;
    }

    public Vegetable() {
    }
}
```

7. Tạo Controller

@Controller

```
public class VegetableController {  
    private static ArrayList<Vegetable> list = new ArrayList();  
    static {  
        list.add(new Vegetable(1, "Tomato", "Kg", 30, 20000));  
        list.add(new Vegetable(2, "Watermelon", "Kg", 50, 30000));  
    }  
    @GetMapping("/vegetable")  
    public String homepage(Model model) {  
        model.addAttribute("list", list);  
        return "vegetable/index";  
    }  
}
```

View: index.html

```
<div>List vegetable</div>  
<table border="1">  
    <tr>  
        <th>ID</th>  
        <th>Name</th>  
        <th>Unit</th>  
        <th>Amount</th>  
        <th>Price</th>  
        <th>Edit</th>  
    </tr>  
    <tr th:each="row, state: ${list}">  
        <td th:utext="${state.count}">...</td>  
        <td th:utext="${row.VegetableName}">...</td>  
        <td th:utext="${row.Unit}">...</td>  
        <td th:utext="${row.Amount}">...</td>  
        <td th:utext="${row.Price}">...</td>  
        <td><a th:href="'/vegetable/edit/' + ${state.count}">Edit</a></td>  
    </tr>  
</table>  
<p><a href="/vegetable/add">Add vegetable</p>
```

List vegetable

ID	Name	Unit	Amount	Price	Edit
1	Tomato	Kg	30	20000.0	Edit
2	Watermelon	Kg	50	30000.0	Edit

[Add vegetable](#)

7. Tạo Controller

View: add.html

```
@GetMapping(value = {"vegetable/add"})  
  
public String showAddForm(Model model) {  
    Vegetable vegetable = new Vegetable();  
    model.addAttribute("vegetable", vegetable);  
    return "vegetable/add";  
}
```

Add vegetable form

New ID:	<input type="text" value="3"/>
Name:	<input type="text" value="Mang cut"/>
Unit:	<input type="text" value="kg"/>
Amount:	<input type="text" value="300"/>
Price:	<input type="text" value="200000"/>
<input type="button" value="Create"/>	

```
<div>  
    <h2>Add vegetable form</h2>  
    <form th:action="@{/vegetable/save}" th:object="${vegetable}" method="POST">  
        <label>New ID:</label>  
        <input type="number" th:field="*{VegetableID}" />  
        <br/>  
        <label>Name:</label>  
        <input type="text" th:field="*{VegetableName}" />  
        <br/>  
        <label>Unit:</label>  
        <input type="text" th:field="*{Unit}" />  
        <br/>  
        <label>Amount:</label>  
        <input type="text" th:field="*{Amount}" />  
        <br/>  
        <label>Price:</label>  
        <input type="text" th:field="*{Price}" />  
        <br/>  
        <input type="submit" value="Create" />  
    </form>  
</div>
```

7. Tạo Controller

```
@RequestMapping(value = {"/vegetable/save"}, method = RequestMethod.POST)
public String save(Model model, @ModelAttribute("vegetable") Vegetable vegetable) {
    int id = vegetable.getVegetableID();
    String vegetableName = vegetable.getVegetableName();
    String unit = vegetable.getUnit();
    int amount = vegetable.getAmount();
    double price = vegetable.getPrice();
    Vegetable v = new Vegetable(id, vegetableName, unit, amount, price);
    list.add(v);
    model.addAttribute("list", list);
    return "redirect:/vegetable";
}
```

7. Tạo Controller

```
@GetMapping(value = {"vegetable/edit/{id}"})  
public String showEditForm(@PathVariable("id") int id, Model model) {  
    Vegetable vegetable = list.get(id - 1);  
    model.addAttribute("vegetable", vegetable);  
    return "vegetable/edit";  
}
```

Edit vegetable form

ID:

Name:

Unit:

Amount:

Price:

```
<h2>Edit vegetable form</h2>  
<form th:action="@{/vegetable/update/}" th:object="${vegetable}" method="POST">  
    ID:  
    <input type="number" th:field="*{VegetableID}" />  
    <br/>  
    Name:  
    <input type="text" th:field="*{VegetableName}" />  
    <br/>  
    Unit:  
    <input type="text" th:field="*{Unit}" />  
    <br/>  
    Amount:  
    <input type="text" th:field="*{Amount}" />  
    <br/>  
    Price:  
    <input type="text" th:field="*{Price}" />  
    <br/>  
    <input type="submit" value="Update" />  
</form>
```

7. Tạo Controller

```
@RequestMapping(value = {"/vegetable/update"}, method = RequestMethod.POST)
public String update(Model model, @ModelAttribute("vegetable") Vegetable vegetable) {
    int id = vegetable.getVegetableID();
    int index = this.findIndexOfVegetable(id);
    if (index >= 0) {
        Vegetable v = list.get(index);
        v.setVegetableName(vegetable.getVegetableName());
        v.setUnit(vegetable.getUnit());
        v.setAmount(vegetable.getAmount());
        v.setPrice(vegetable.getPrice());
        list.set(index, v);
    }
    return "redirect:/vegetable";
}

private int findIndexOfVegetable(int vegetableID) {
    for (int i = 0; i < list.size(); i++) {
        if (list.get(i).getVegetableID() == vegetableID) {
            return i;
        }
    }
    return -1;
}
```