

View 구현 2

Thymeleaf

- 타임리프에서 자바 클래스 접근하기

- SeedStarter→Feature→name: 중첩 반복문 작성이 까다로움
- 본 예제에서는 특정 SeedStarter에 속한 모든 Feature들을 탐색하며 name을 수집, 하나의 문자열로 concat

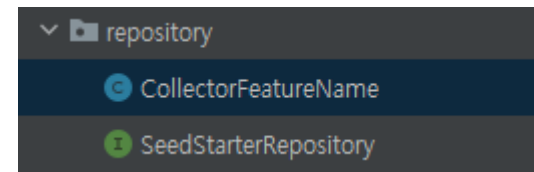
심은날짜	가림유무	소재	특징
2002-04-05	No	PLASTIC	[SUBSTRATE, PH_CORRECTOR]
2001-04-05	No	WOOD	[FERTILIZER]
2003-04-05	Yes	WOOD	[PH_CORRECTOR]

Thymeleaf

- 타임리프에서 자바 클래스 접근하기

- CollectorFeatureName

```
public class CollectorFeatureName {  
  
    public static final List<FeatureType> name(Stream<Feature> stream) {  
        return stream.map(v->v.getName()).collect(Collectors.toList());  
    }  
}
```



- 타임리프에서 메소드에 정의된 클래스에 접근하기 위해(인스턴스 변수로 접근이 아닌) 함수를 static으로 정의

```
<td th:text="${T(com.example.seedstarter.entity.CollectorFeatureName).name(starter.features.stream())}"></td>
```

Thymeleaf

- 테이블 안에 테이블

셀별품종

1	Thymus vularis	10
---	----------------	----

2	Thymus pseudolaginosus	15
---	------------------------	----

3	x citriodorus	20
---	---------------	----

1	Thymus herba-barona	5
---	---------------------	---

2	Thymus serpyllum	5
---	------------------	---

1	New Variety	20
---	-------------	----

Thymeleaf

- 테이블 안에 테이블

```
<td>
  <table>
    <tbody>
      <tr th:each="detail : ${starter.details}">
        <td th:text="${detail.rowNum}">1</td>
        <td th:text="${detail.variety}">Thymus Thymi</td>
        <td th:text="${detail.seedPerCell}">12</td>
      </tr>
    </tbody>
  </table>
</td>
```

Thymeleaf

- 새로운 SeedStarter 추가 form 만들기

```
<div class="container alert alert-success">
  <form action="#" th:action="@{/seedstartermng}" th:object="{seedStarterAddForm}" method="post">
    <fieldset>
      <legend>새로운 Seed Starter추가</legend>
      <div class="form-group row">
        <label for="plantedDateTextBox" class="col-sm-2 col-form-label"
          th:text="#{seedstarter.datePlanted}"></label>
        <div class="col-sm-10">
          <input type="text" id="plantedDateTextBox" th:field="*{datePlanted}"/>
        </div>
      </div>

      <button type="submit" class="btn btn-primary">추가</button>
    </fieldset>
  </form>
</div>
```

Thymeleaf

- Selection expressions

- `*{...}` : th:object로 정의된 변수(object)에 포함된 값을 의미

```
<form action="#" th:action="@{/seedstartermng}" th:object="${seedStarterAddForm}" method="post">
```

```
<input type="text" id="plantedDateTextBox" th:field="*{datePlanted}"/>
```

```
@Getter
@Setter
public class SeedStarterAddForm {
    private String datePlanted;
}
```

Thymeleaf

- Selection expressions

- Method level에 @ModelAttribute를 사용하여 모델에 공통 속성 등록

```
public class SeedStarterMngController {  
  
    @ModelAttribute  
    public void addAttributes(Model model) {  
        model.addAttribute("seedStarterAddForm", new SeedStarterAddForm());  
    }  
  
    @PostMapping("/{seedstartermng}")  
    public String addNewSeedStarter(@ModelAttribute SeedStarterAddForm seedStarterAddForm){  
        System.out.println("seedStarter = " + seedStarterAddForm.getDatePlanted());  
        return "seedstartermng1";  
    }  
}
```


Thymeleaf

- Selection expressions

- SeedStarter리스트 표시에 필요한 부분도 모델에 공통 속성 등록

```
public class SeedStarterMngController {  
    private final SeedStarterService seedStarterService;  
    private final ObjectMapper mapper;  
    @ModelAttribute  
    public void addAttributes(Model model) {  
        List<SeedStarter> seedStarterWithFeature = seedStarterService.findWithFeature();  
        List<SeedStarter> seedStarterWithDetail = seedStarterService.findWithDetail();  
        model.addAttribute("seedStarterWithFeature", seedStarterWithFeature);  
        model.addAttribute("seedStarterWithDetail", seedStarterWithDetail);  
        model.addAttribute("seedStarterAddForm", new SeedStarterAddForm());  
    }  
}
```

Thymeleaf

- 새로운 SeedStarter 추가 form 만들기

```
<div class="form-group row">
  <div class="col-sm-2" th:text="#{seedstarter.covered}"></div>
  <div class="col-sm-10">
    <div class="form-check">
      <input class="form-check-input" type="checkbox" id="coveredChkBox" th:field="*{covered}">
      <label class="form-check-label" for="coveredChkBox" ></label>
    </div>
  </div>
</div>
```

```
public class SeedStarterAddForm {
    private String datePlanted;
    private boolean covered;
}
```

Thymeleaf

- 새로운 SeedStarter 추가 form 만들기

```
<div class="form-row row">
  <div class="col-sm-2" th:text="#{seedstarter.type}"></div>
  <div class="col-sm-2">
    <select id="seedStarterType" class="form-control">
      <option selected>PLASTIC</option>
      <option>WOOD</option>
    </select>
  </div>
</div>
```

```
@Getter
@Setter
public class SeedStarterAddForm {
    private String datePlanted;
    private boolean covered;
    private String type;
}
```

Thymeleaf

- 새로운 SeedStarter 추가 form 만들기

```
<div class="form-row row">
  <div class="col-sm-2" th:text="#{seedstarter.features}"></div>
  <div class="col-sm-2">
    <div class="form-check">
      <input class="form-check-input" type="checkbox" name="features" value="SUBSTRATE" th:field="*{substrate}">
      <label class="form-check-label">특정 기질</label>
    </div>
    <div class="form-check">
      <input class="form-check-input" type="checkbox" name="features" value="FERTILIZER" th:field="*{fertilizer}">
      <label class="form-check-label">비료 사용</label>
    </div>
    <div class="form-check">
      <input class="form-check-input" type="checkbox" name="features" value="PH_CORRECTOR" th:field="*{phCorrector}">
      <label class="form-check-label">수소 농도 교정</label>
    </div>
  </div>
</div>
```

Thymeleaf

- 새로운 SeedStarter 추가 form 만들기

```
@Getter
@Setter
public class SeedStarterAddForm {
    private String datePlanted;
    private boolean covered;
    private String type;
    private String substrate;
    private String fertilizer;
    private String phCorrector;
}
```