# Spring Boot WEB开发- Thymeleaf 模极引擎

Spring Boot 提供了大量模板引擎，包含括FreeMarker 、Groovy 、Thymeleaf、Velocity 和Mustache, Spring Boot 中推荐使用Thymeleaf 作为模板引擎，因为Thymeleaf 提供了完美的Spring MVC 的支持。

Thymeleaf 是一个Java 类库，它是一个xml/xhtml/html5 的模板引擎，可以作为MVC 的Web 应用的View 层。Thymeleaf 还提供了额外的模块与Spring MVC 集成，所以我们可以使用Thymeleaf 完全替代JSP。

## 引入Thymeleaf

|  |
| --- |
| <**html xmlns:th="http://www.thymeleaf.org"**><!-- 1 --> <**head**>  <**meta content="text/html;charset=UTF-8"**/>  <**meta http-equiv="X-UA-Compat**<**div class="panel-body"**>  <**ul class="list-group"**>  <**li class="list-group-item" th:each="person:${people}"**>  <**span th:text="${person.name}"**></**span**>  <**span th:text="${person.age}"**></**span**>  <**button class="btn" th:onclick="'getName(\'' + ${person.name} + '\');'"**>获得名字</**button**>  </**li**>  </**ul**> </**div**>  **ible" content="IE=edge"**/>  <**meta name="viewport" content="width=device-width, initial-scale=1"**/>  <**link th:href="@{bootstrap/css/bootstrap.min.css}" rel="stylesheet"**/>*<!-- 2 -->*  <**link th:href="@{bootstrap/css/bootstrap-theme.min.css}" rel="stylesheet"**/>*<!-- 2 -->* </**head**> <**body**> <**script th:src="@{jquery-1.10.2.min.js}" type="text/javascript"**></**script**>*<!-- 2 -->* <**script th:src="@{bootstrap/js/bootstrap.min.js}"**></**script**>*<!-- 2 -->* </**body**> </**html**> |

说明：通过xmlns:th=http://www.thymeleaf.org 命名空间，将镜头页面转换为动态的视图。需要进行动态处理的元素将使用“th:”为前级；过“@{}”引用Web 静态资源。

## 访问model 中的数据

通过“${}”访问model 中的属性。使用<span th:text="${singlePerson.name}"></span>访问model 中的singlePerson 的name 属性。注意：需要处理的动态内容需要加上“th :”前缀。

|  |
| --- |
| <**div class="panel panel-primary"**>  <**div class="panel-heading"**>  <**h3 class="panel-title"**>访问model</**h3**>  </**div**>  <**div class="panel-body"**>  <**span th:text="${singlePerson.name}"**></**span**>  </**div**> </**div**> |

## model 中的数据迭代

使用th:each 来做循环迭代（ th:each=person: ${people} ), person 作为迭代元素来使用，然后像上面一样访问迭代元素中的属性。

|  |
| --- |
| <**div class="panel-body"**>  <**ul class="list-group"**>  <**li class="list-group-item" th:each="person:${people}"**>  <**span th:text="${person.name}"**></**span**>  <**span th:text="${person.age}"**></**span**>  </**li**>  </**ul**> </**div**> |

## 数据判断

通过$ {not #lists.isEmpty(people）｝表达式判断people 是否为空。Thymeleaf 支持>、<、>=、<=、!=作为比较条件，同时也支持将SpringEL 表达式语言用于条件中。

|  |
| --- |
| <**div th:if="${not #lists.isEmpty(people)}"**> </**div>** |

## 在JavaScript 中访问model

在JavaScript 访问model 中的似，在Thymeleaf实现代码如下：

|  |
| --- |
| <**script th:inline="javascript"**>  **var *single*** = [[${singlePerson}]];  **console**.log(***single***.**name** + **"/"** + ***single***.age) </**script**> |

通过th:inline＝”javascript”添加到script 标签，这样JavaScript代码即可访问model中的属性；通过“[[${}]]”格式获得实际的值。

在html代码中获取model属性。

|  |
| --- |
| <**button class="btn" th:onclick="'getName(\'' + ${person.name} + '\');'"**>获得名字</**button**> |