

# Thymeleaf & Spring Boot Data JPA



Presented By





- Thymeleaf is a modern server-side Java template engine for processing HTML, XML, JavaScript, CSS and even plain text.
- The main goal of Thymeleaf is to provide an elegant and highly-maintainable way of creating templates.
- Standard Dialects such as *attribute processors* are used. This allows browsers to correctly display HTML template files.
- Thymeleaf is an alternative to dynamic web pages., i.e., JSP, ASP, achieved by standard dialects.
- Need to add thymeleaf **xmlns** to **<html>** tag as follows:

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

# Thymeleaf template attributes

**th:text** : Thymeleaf will process the text inside the **th:text** attribute

Ex: `<p th:text="'Thymeleaf welcomes you!'">text</p>`

**th:object** : Represents an object sent to the web page

Ex: `th:object="${product}"`

**th:action** : Represents a hyperlink

Ex: `th:action="@{/save}"`

**th:field** : Represents a field received from an object

Ex: `th:field="*{id}"` -- representing *product.id*



# Thymeleaf template attributes

**th:href:** Represents an anchor tag

Ex: `<a th:href="@{'/edit/'}" -- navigates to the URI`

**th:each :** Represents an iteration attribute as in `for(int score : scores) { }` in Java

Ex: `th:each="product : ${listProducts}"`

To iterate all products using product object from list of products



```
<tr th:each="product : ${listProducts}">
  <td th:text="${product.id}"></td>
  <td th:text="${product.name}"></td>
  <td th:text="${product.brand}"></td>
  <td th:text="${product.madein}"></td>
  <td th:text="${product.price}"></td>
</tr>
```

\*\*Demo



