**Spring MVC e pagine dinamiche con Thymeleaf**

ResourceHttpRequestHandler gestisce le risorse statiche

Tamplate engine: Un template engine è uno strumento per generare file basati su modelli.

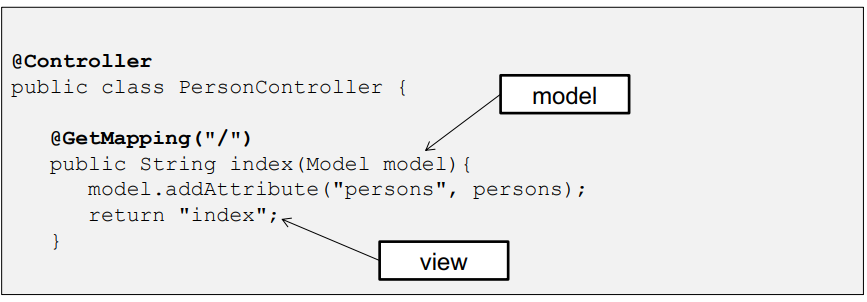
Spring Boot include auto-configurazioni per:

* FreeMarker
* Groovy
* Mustache
* Thymeleaf

I templates sono cercati nella cartella **src/main/resources/templates**

È possibile configurare ogni template engine supportato da Spring Boot

**Spring MVC e pagine web**



Model sarà quello che arriverà al template engine, ci saranno i dati necessari

L’annotazione @RequestParam indica che un parametro di un metodo dev’essere associato a un parametro della richiesta.

Immagine che contiene testo

Descrizione generata automaticamente

Quando si hanno tanti parametri nella richiesta HTTP, piuttosto che aggiungere tanti @RequestParam è più comodo utilizzare un oggetto apposito (in questo esempio di tipo Person)

Immagine che contiene testo

Descrizione generata automaticamente

L’annotazione @ModelAttribute, quando utilizzato come argomento di metodo, indica che il valore dell’argomento deve essere associato ad un attributo del modello, e sarà quindi possibile usarlo nella visualizzazione web.

Immagine che contiene testo

Descrizione generata automaticamente

Se il nome di una view inizia con il prefisso redirect: il View Resolver lo riconoscerà come indicazione che un redirect è necessario

Immagine che contiene testo

Descrizione generata automaticamente

**Thymeleaf**

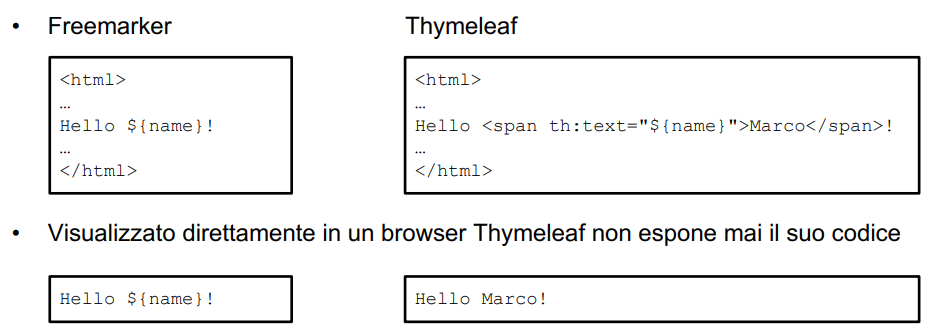


Immagine che contiene testo

Descrizione generata automaticamente

Thymeleaf usa i tag html per assegnare i valori, non visualizzo la sintassi di thymeleaf se si apre il tamplate

Th:text sostituisce il testo con il valore da utilizzare

Immagine che contiene testo

Descrizione generata automaticamente

L’attributo th:text rimpiazza il contenuto di un tag.

In generale se si vuole rendere dinamico un attributo di un elemento HTML è sufficiente aggiungere th: all’attributo.

Immagine che contiene testo

Descrizione generata automaticamente

th:each ripete il tag tante volte quanti sono gli elementi nell’array o la lista ritornata dall’espressione

Immagine che contiene testo

Descrizione generata automaticamente

Tutte le espressioni che iniziano con @{} son dette link expressions e servono

per creare degli URL e sono molto utili perché possono aggiungere il contesto

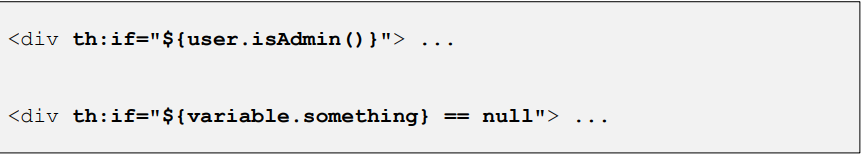
nel quale è l’applicazione.

Immagine che contiene testo

Descrizione generata automaticamente

Si possono creare link dinamici

Utilizzando th:if è possibile porre una condizione per la visualizzazione di un tag



L’attributo th:object permette di semplificare l’accesso ai campi di un oggetto in

un determinato contesto.

Immagine che contiene testo

Descrizione generata automaticamente

Con \*{campo} si accede direttamente al campo

[Evitare th:object]

th:field serve per aggiungere un input in un form.

Immagine che contiene testo

Descrizione generata automaticamente

Nelle select con field seleziona automaticamente l’opzione passata

Immagine che contiene testo

Descrizione generata automaticamente

Esempio:

Immagine che contiene testo

Descrizione generata automaticamente

Immagine che contiene testo

Descrizione generata automaticamente