Introduzione a FreeMarker

Cosa imparerai

- Panoramica su FreeMarker e le sue funzionalità

paolopreite

Introduzione a FreeMarker

Apache FreeMarker™ è un *template engine*.

Cos'è un *template engine*?

È uno strumento in grado di generare output (ad esempio codice HTML, ma non solo…) attraverso un algoritmo interno (da cui il nome *engine*) sulla base di modelli (chiamati *template*).

Conosciamo FreeMarker…

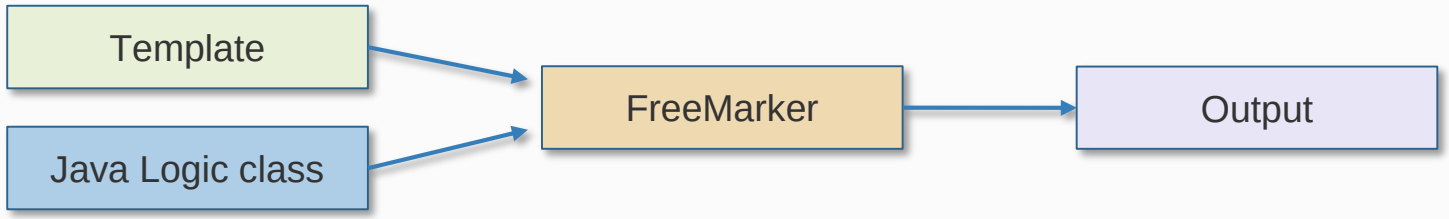
FreeMarker è un *template engine* open source, rilasciato con licenza *Apache Version 2.0*.

Maggiori dettagli su licenza ed utilizzo li trovate qui: https://freemarker.apache.org/

Introduzione a FreeMarker

Come funziona FreeMarker

FreeMarker è scritto in Java (e distribuito sotto forma di jar) ed è in grado di generare output di testo (pagine HTML, e-mail, ecc…) sulla base di *template* e dati.



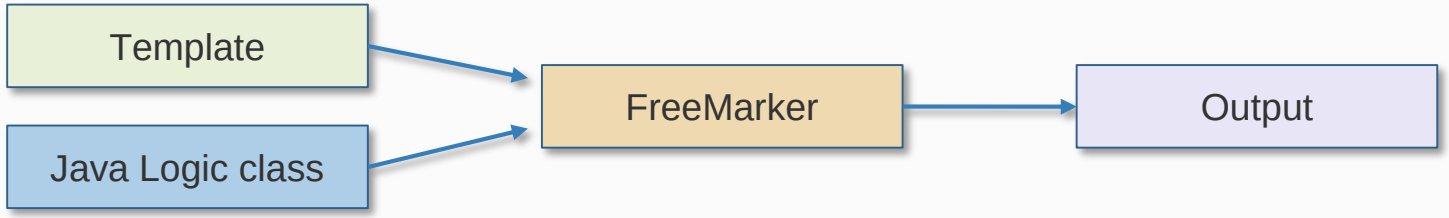
I *template* sono scritti in **FreeMarker Template Language (FTL)**, un linguaggio semplice e specializzato.

I dati vengono prodotti dal software (ad es. un controller Spring che invoca un DAO…).

FreeMarker visualizza i dati utilizzando i template.

Introduzione a FreeMarker

Come funziona FreeMarker



Con FreeMarker siamo in grado di disaccoppiare la logica di preparazione dei dati dalla logica di presentazione.

- **Nel template** ci occupiamo della presentazione dei dati.
- **Fuori dal template** (ad es. nel controller Spring) ci occupiamo della preparazione dei dati da presentare.

È il pattern MVC (Model View Controller).

NOTA: FreeMarker è nato come strumento per generare pagine HTML in applicazioni Web MVC. Tuttavia non è vincolato al web e può essere utilizzato anche in applicazioni standalone.

Introduzione a FreeMarker

Alcune caratteristiche di FreeMarker

- **È un linguaggio potente**: consente di creare blocchi condizionali, iterazioni, assegnazioni, operazioni su stringhe, operazioni aritmetiche, formattazione, ecc…
- **È leggero**: non ha dipendenze, per utilzzarlo è sufficiente importare il jar di FreeMarker nel proprio progetto
- **È flessibile**: è in grado di generare qualsiasi formato di output, non solo HTML ed è altamente configurabile

Introduzione a FreeMarker

Facciamo un esempio…

Supponiamo di avere una pagina HTML di benvenuto da far vedere all'utente autenticato.

```
<html>
  <head>Home</head>
  <body>
    <h1>Ciao, Paolo!</h1>
    <p>Hai 3 messaggi da leggere.</p>
  </body>
</html>
```

Il nome ed il numero di messaggi variano in base all'utente autenticato e pertanto non possiamo scriverlo nel codice HTML altrimenti tutti vedrebbero "**Ciao, Paolo!**".

Abbiamo bisogno di qualcosa di dinamico…

Introduzione a FreeMarker

Facciamo un esempio…

Prima si usavano gli scriptlet nelle JSP (*NOTA: si trovano ancora web app scritte in questo modo da manutenere per cui è importante sapere come funzionano le JSP…*).

Con FreeMarker possiamo usare un template che utilizza delle variabili opportunamente impostate nella logica Java della web application.

Avremo pertando un template di questo tipo:

```
<html>
  <head>Home</head>
  <body>
    <h1>Ciao, ${userFirstname}!</h1>
    <p>Hai ${unreadMessages} messaggi da leggere.</p>
  </body>
</html>
```

```
<html>
  <head>Home</head>
  <body>
    <h1>Ciao, Paolo!</h1>
    <p>Hai 3 messaggi da leggere.</p>
  </body>
</html>
```

```
<html>
  <head>Home</head>
  <body>
    <h1>Ciao, Maria!</h1>
    <p>Hai 25 messaggi da leggere.</p>
  </body>
</html>
```

```
<html>
  <head>Home</head>
  <body>
    <h1>Ciao, Alessandra!</h1>
    <p>Hai 0 messaggi da leggere.</p>
  </body>
</html>
```

```
<html>
  <head>Home</head>
  <body>
    <h1>Ciao, Antonio!</h1>
    <p>Hai 12 messaggi da leggere.</p>
  </body>
</html>
```

Introduzione a FreeMarker

IMPORTANTE!

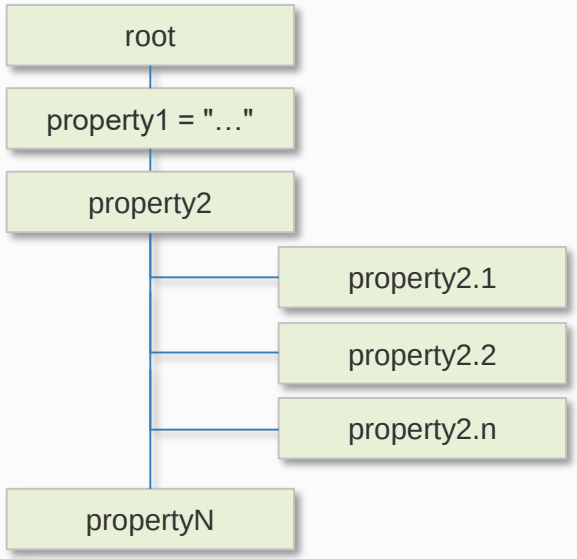
- Il template contiene solo logica di presentazione dei dati ed in generale dell'output.
- Il template NON accede a Database, NON esegue query ecc…queste istruzioni vengono eseguite lato controller.
- A chi realizza i template, NON importa come sono generati i dati; a lui interessa solo sapere come si chiamano le variabili e cosa contengono.
- Teoricamente chi realizza il template potrebbe non essere uno sviluppatore Java ma potrebbe avere forti competenze su web design. In linea teorica dovremmo avere: il Java Developer che implementa la logica di recupero ed elaborazione dei dati, il Web Designer che implementa la logica di presentazione dei dati ***(*)***.

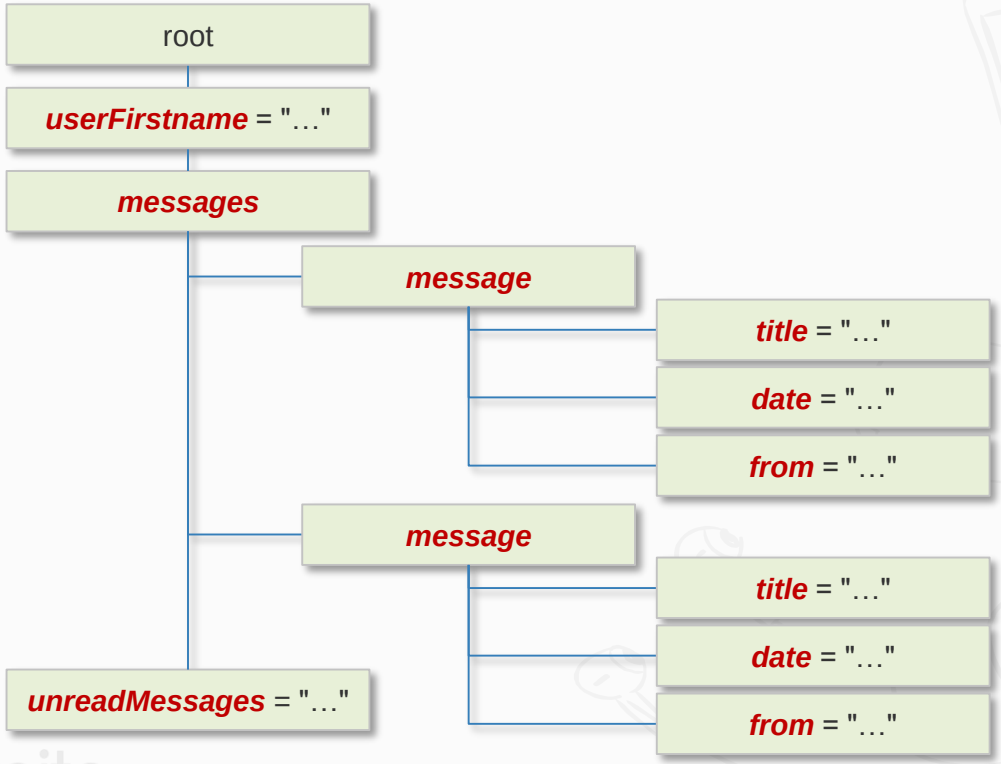
(*) *NOTA: spesso troviamo la figura del Full Stack Developer che fa tutt'e due le cose…ahimè.*

Introduzione a FreeMarker

I dati elaborate dall'engine di FreeMarker per i template sono chiamati **data-model**.

Il *data-model* è un oggetto (nel disegno è *root*) che contiene tanti oggetti Java organizzati con una struttura ad albero (come se fossero tante directory annidate…).





Introduzione a FreeMarker

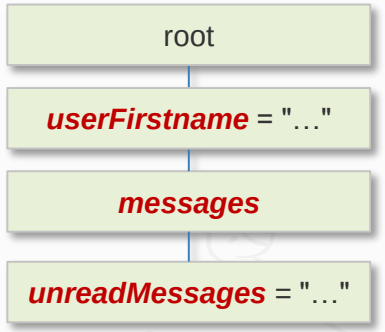
Se fossimo in ambito Java per accedera ai vari oggetti useremmo la notazione getObject()…

Ad esempio: getUserFirstName(), getMessages()…

Con FreeMarker possiamo accedere ai singoli oggetti usando la seguente notazione:

${object1} oppure ${object2.object21} …

Ad esempio: ${userFirstName} … ${unreadMessages}



Di cosa abbiamo parlato in questa lezione

- Panoramica su FreeMarker e le sue funzionalità