

# Survey on Template Engines in Java

Jishnu Saurav Mittapalli<sup>1\*</sup> and Menaka Pushpa Arthur<sup>1</sup>

<sup>1</sup>School of Computer Science and Engineering, Vellore Institute of Technology, Chennai Campus, India 600 127

**Abstract** - In today's fast paced world every minute is very important. In the corporate world and also otherwise Documentation is very important for various purposes like version control, proofs, consent, copyrights and expectations and outcomes/reports. So, because of these reasons template engines have become very important and extremely necessary for the world. Template engines are basically software that help us create result documents from data models and templates by combining them. This paper presents a survey on the newest development of research work on template engines for java, along with an in-depth outline of its research. To the best of what developments have been achieved till now, this survey paper has been written. Finally, the differences, advantages and disadvantages of the various template engines for java, have been tabulated as a part of the results.

## 1 INTRODUCTION

A template engine is a software that provides static template files for our application. During runtime, template engines can replace the variables by real values in our file and it helps us to transform the template into a HTML file. This basically supports the developer to make a better HTML webpage. A template language or templating language is the language that the templates are written. a template. Most of the template engines generally have the common features for all high-level programming languages, where features for processing plain text are emphasized [1].

The common features of template engines are as follows:

- Replacement of text.
- Functions and variables
- Loops and conditional statements
- Inclusion of files

In many different contexts, template processing has many interesting uses, also very different and unique ones. One of the main uses is that it depends on the software application that is created for this purpose. Basically, the template engine's flexibility helps us in

- Organization of source code.
- Helps in version control.
- Improves productivity and efficiency.
- Improves teamwork or team collaboration in the projects.

The importance of automatic generation of code is increasing at a tremendous rate. As today in model-driven software applications, flexible code generation is very important. This paper basically discusses the various existing template engines in Java programming language and their pros and cons. Along with this, we recommend an appropriate template engine as per the developers demand and requirement.

many other uncommon ways different from the ones the creators require [2]. The following are the main uses of template engines:

- A software that has modules of all major programming languages and is generally a part of the software development.
- Main functionality of the software - Basically, the source data and the templates are used typically to generate output web pages or parts of the output web pages. Generally, used in web application development.
- Template engines can also be used for generation of source code from very abstract or a little unclear data models for example UML or relational data models. That is, it helps the developer, by simplifying the production process and making his work easier.
- Template engines are also used for documentation purposes, i.e, they help us in generating documents for software products.

The main benefits of using template engines are as follows:

- Saves a lot of time in the software development life cycle.
- Helps in the proper

## 2 APACHE VELOCITY

One of the commonly used templating engines for Java is Apache Velocity. Apache velocity is open source and it is designed to be a web framework and can be used very efficiently in the MVC (Model-View-Controller) architecture. XML files, SQL, PostScript and many other formats can be generated using Apache Velocity. Reading, parsing, content generation and many other things can be made easily by using this apache velocity. It is one of the cleanest ways of incorporating webpage content that is dynamic using references. We can download velocity from their official website. Building web applications is the most major use of apache

\* Corresponding author: [jishnu.2018a@vitstudent.ac.in](mailto:jishnu.2018a@vitstudent.ac.in)

velocity. Servlet-based framework or servlet will be needed by us to develop a web application using apache. Velocity can also be used to generate PDFs from HTML documents. Coming to the price details, apache velocity is a free and open-source templating engine. This can be installed in various IDEs like eclipse, netbeans and also in visual code studio as a plugin and can be used as the template engine. And the free version itself has all the features that we can use in this framework.

The following are the main use-cases of apache velocity:

- **Dynamic Web Pages:** HTML page structures can be rendered Web applications by Web developers. Dynamic information populates the content. VelocityViewServlet or a number of frameworks that support Velocity process the web pages [3][4]. Various applications that require the generation of automatic email reminders for password, reports automatically sent or account signup. The email that is a text file storing the content can be sent and this can be embedded in the java code itself using velocity [5].

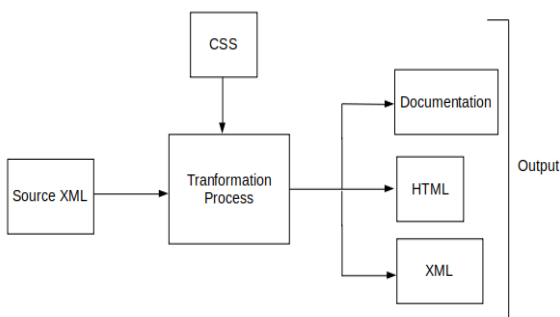
- **AWS API Gateway:** Use the velocity engine for body mapping templates and the output is the JSON file.

- **Conversion of HTML into documentation and vice-versa:** This can be used to convert HTML pages into documents and documents into beautiful web pages.

The following is the technology used behind Apache Velocity:

Context objects are used for pulling references from a set and get methods for setting and receiving objects. The values can be directly inserted into the page. Macros can be created and files should be included. Java methods which are arbitrary can be called and this ability is available to be used multiple times making this a strong and still a simple method with which we can develop text files or other web pages dynamically.

MVC is enforced by velocity and this way of creation where the html code is divided with the Java code is very efficient and reusable. Velocity features are not implemented using other functions unlike PHP., Velocity does not implement features with other functions. The greatest strength of velocity as it allows to create web pages with better designs as it uses MVC [6]. The overall architecture of the Velocity is given in Figure 1.



**Figure 1.** Architecture diagram of Apache Velocity

### 3 APACHE FREEMARKER

Apache Freemarker is also a free Java based template which was mainly designed for dynamic web page generation based on software architecture such as MVC. The main advantage of using this is that it is a general template engine with no particular dependencies and it is not dependent on specific technologies like HTTP or HTML or servlets, therefore allowing us to use it for all purposes of a template engine like document generation and source code generation.

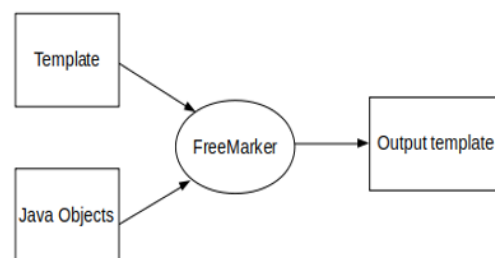
A few important features of FreeMarker are listed below:

- It is really lightweight and multipurpose, has no dependencies at all and any output format is allowed and can be used as just a plugin in text editors like VSC etc. It also has a lot of configuration options.

- It is aware of what kind of program it is working on and when it is locale sensitive with respect to date and timeline. It has a lot of capabilities like XML processing. It can traverse files of various different types and also process them in a more efficient way than many other engines.

- The data model is a very versatile one [3]

Basically, as shown in Figure 2, the template and the given Java objects act as the input to the Apache freemarker and it parses and combines in order to give the output as the template.



**Figure 2.** Architecture diagram of Apache Freemarker

### 4 THYMELEAF

Thymeleaf is basically a template engine for Java that can work on both in non-web as well as servlet-based environments and applications. It is more apt to work with HTML5 for web applications but it can also work well with offline applications. It can basically act as a substitute for JSP, its template files can be opened directly from the browser itself.

The main features are as follows:

- Can be used for full documentation.
- If required it can be used as a template engine framework.

- It can detect the type of document and convert as and when required.

- Very good support

- Reduces the input output to the minimum due to the feature known as parsed template cache which gives a high performance and efficiency.

- It is capable of working in both online and offline environments.

The main advantage of this in comparison to the other engines that we have discussed is that in it templates work as prototypes enabling a natural templating system. The templates open in the browser without any server running i.e., templates are statically displayed. It is basically like a template engine framework.

## 5 APACHE TILES

Apache Tiles basically is a composition framework for templates. Web application development and User interface development in Java was the main aim for the development of apache tiles, but it is no longer restricted to the JavaEE web environment. Here, page fragments are allowed to be defined. And the complete page can be assembled from the page fragments generated at runtime.

A template is basically the layout part of a webpage. Every page created has a structure with a few gaps called attributes, that have to be filled. There are three types of attributes, string, template and definition. A composition that has to be rendered to the end user and this is called a definition. A definition is completely or partially filled attributes, composed of a template. It can be rendered to the end user if all of its attributes are filled and if all of its attributes are not filled it can be called as an abstract definition and can be used for missing attributes or extended deadlines i.e, it can be used a base definition for it and the attributes can be filled at runtime.

Many of the time definitions need to be prepared and only then can they be rendered. And this is the reason we have something called a view Preparer and it can be used and after which the definition is rendered, so that all the things that are required are rendered correctly in the correct form.

## 6 MUSTACHE.JAVA

Basically, mustache is a web-based template system with applications of so many different languages like, PHP, Rust, Xquery, Python, Rupy, C++, R, Scala etc. Another important feature is that mustache is described as a logic less templating engine. The main reason for this is the lack of any explicit conditional / looping statements and the evaluation can be achieved using section tags , lists and lambda functions.

The reason Mustache got its name is because of the huge use of curly braces '{ }', which resembles a side 'mustache'. Mustache is used mainly for mobile and web applications. This is mainly used for mobile and web applications. It is available as plugins in many text editors and syntax highlighting is available in many text editors like Textmat, Vi, Visual studio code, Atom, Coda etc.

The templates of mustache support are there by default in many frameworks including web application frameworks like cakePHP. Client Side, ajax frameworks such as jquery, dojo as well as server-side javascript

programming libraries are also included and are supported by this.

The following are the important features of mustache:

For the mustache engine many implementations are existing, and all of them have a few common specifications that the results have a common syntax. Most important method is the render method and the most common class is the parser class of this technology.

The development of the mustache template engine inspired many JS template libraries that improved the functionality and efficiency of this template/rendering engine. Lack of logic statements and another feature is that it is compatible with all javascript frameworks. On the client side JQuery, YUI, Dojo and on the server side Node.js are supported by this.

Main advantages of this engine are Code is very understandable, Re-use of code, and easy maintenance. Source code, HTML and config files are the major applications of this.

Hash or objects have the values which are used for expanding the tags, and this is how it works.

A Mustache template have A mustache template containing a string and it can have any number of mustache tags. The double mustaches indicate the tags. The sections's tag value can be undefined, false or null and then only an inverted section's block is rendered. Mustache.compile can be used to create javaScript functions by compiling mustache templates for improving the performance of the rendering process.

**Table 1.** Features of Template Engines

Name	Functions	Error handling	Natural Templates	Inheritance
Apache velocity	Yes	Yes	No	No
Freemarker	Yes	Yes	Yes	No
Mustache	Yes	Yes	Yes	No
Thymeleaf	Yes	Yes	Yes	No
Groovy	Yes	Yes	Yes	Yes
Tiles	Yes	No	Yes	No

## 7 GROOVY

A language which can work well with the OOPS concepts especially in Java was developed and called groovy. It is supportive of dynamic as well as static language. It has features similar to ruby, python etc [7]. We can use it as a scripting as well as a programming language for java.

**Table 2.** Comparison of Different Template Engines in Java

Name	Name of developer	Company owned	Year	IDE	Paid/ Free	Online / Offline	Enough documentation	Completed / Ongoing
Apache velocity	Apache Software Foundation	ASF (The Apache Software Foundation)	2001	NetBeans, Eclipse, BBEdit, Dreaweaver, IntelliJ IDEA, JEdit	Free	Both	Yes	Most probably satisfy the requirement of the user
Freemarker	Jonathan Revusky, Attila Szegedi, Dániel Dékány, and others	The Apache Software Foundation	2000	Eclipse, NetBeans	Free	Both	Yes	Most probably satisfy the requirement of the user
Mustache	Chris Wanstrath and other contributors	Available in github	2009	IntelliJ	Free	Offline	MIT Licensed Documentation	Ongoing
Thymeleaf	Daniel Fernández	ASF(The Apache Software Foundation)	2018	IntelliJ, Eclipse	Free	Both	Yes	Completed
Groovy	James, Strachan	AFS (The Apache Software Foundation)	2020	IntelliJ, Netbeans	Free	Both	Yes	ongoing
Tiles	ASF	ASF (The Apache Software Foundation)	2012	IntelliJ, Selenium, Eclipse, NetBeans	Free	Both	Yes	Ongoing

The syntax with a curly bracket is used by groovy. Expressions in strings, multiline strings and closures are supported by groovy.

Most of the valid Java files that are existing are also out of groovy. Groovy code can be very small and clear which is a great advantage.

It is easy for developers to learn and make themselves familiar with the basic java syntax as it does not does not require all the components that normal java requires. The features that groovy offers are not available anywhere else. Official syntax for lists, polymorphic iterations, maps, helper methods are a few to name. From its second version groovy also supports the jar files etc to improve the modularity of the programs. Still research is going on and people are working to improve the efficiency of this platform.

Native support is provided for many markup languages like HTML, XML etc. Not like Java, groovy code can be executed as a script i.e., without compilation itself

## 8 DIFFICULTIES AND DISADVANTAGES OF TEMPLATE ENGINES

The use of template engines itself has quite a few disadvantages. The following are a few of them:

- Firstly, using these is a skill and it has to be acquired, which will take time. As it is not very easy to learn them.
- There definitely will be some amount of performance overhead as it is something additional and extra processing will be required when we use them.

- Time consuming and add a component into the development lifecycle.

## 9 RESULTS AND DISCUSSION

The various features of the template engines in Java such as Apache velocity, Freemarker, Mustache, Thymeleaf, Groovy, Tiles are tabulated in Table 1. and Table 2. We, have presented the review to the best of our knowledge. From Table 2, the following observations we made from our comparison. Other than Mustache, remaining engines are developed by Apache Software Foundation. It shows the quality and feature richness in the tool. As a developer, we can contribute our work in Mustache since the code is in GitHub. Groovy and Tiles are latest release with current trend features. The IDE IntelliJ is almost compatible with all listed template engines. Each template engine has its own pros and cons, we have to choose the template engine based on our requirements.

## 10 RECOMMENDATION

Since Apache Velocity and FreeMarker are free open-source code, students and research scholars can utilize this engine for their application development. Web and non-web applications can be developed by the Apache Velocity template engine. Groovy's template engine is very simple and intended for limited purposes. Like mail merge in the word processor, Groovy can provide such a functionality for Map [8]. Mustache generates an output for the input with simple logic-less tags. Mostly used for dynamic web-page generation and server-side programming. Thymeleaf can be a template engine for source code development and web-based applications.

## 11 CONCLUSION

This paper reviews all template engines that are currently being used in Java. In this paper as we discussed every template engine has its own pros and cons. We have to choose our template engine as and when based on our templating requirements and the type of application we are developing. Though there can be more research and development in this field we still can still make the use of these template engines a little more widespread. These technologies basically are very useful and if used properly as per the requirements we can achieve a greater efficiency in our work and have a more efficient and a faster development lifecycle. So, basically, we in this paper have given the details about the various template engines that are existing in the market to the best of our knowledge and research, for further research and industries to choose the template engine required by them according to their needs.

Many enhancements and different new developments can be made to this field. These technologies can be more widespreadly used and more advancement can be brought in. And these engines can be made more efficient and faster.

## REFERENCES

1. Sturm T., von Voss J., Boger M., Generating Code from UML with Velocity Templates. In: Jézéquel JM., Hussmann H., Cook S. (eds) <<UML>> 2002 — The Unified Modeling Language. UML 2002. Lecture Notes in Computer Science, vol 2460, 2002.
2. Michiaki Tatsubori and Toyotaro Suzumura, HTML templates that fly: a template engine approach to automated offloading from server to client. In Proceedings of the 18th international conference on the World wide web (WWW '09). Association for Computing Machinery, New York, NY, USA, 951–960, 2009.
3. Tatsubori M., Sasaki T., Chiba S., Itano K., A Bytecode Translator for Distributed Execution of “Legacy” Java Software. In: Knudsen J.L. (eds) — Object-Oriented Programming. ECOOP 2001. Lecture Notes in Computer Science, vol 2072. Springer, Berlin, Heidelberg. 2001.
4. Barclay, Kenneth; John Savage, Groovy Programming: An Introduction for Java Developers. ISBN 978-0-12-372507-3. Archived from the original on 2010-01-12. Retrieved 2007-07-19, 2007.
5. Tilevich E., Smaragdakis Y., J-Orchestra: Automatic Java Application Partitioning. In: Magnusson B. (eds) ECOOP 2002 — Object-Oriented Programming. ECOOP 2002. Lecture Notes in Computer Science, vol 2374. Springer, Berlin, Heidelberg, 2002.
6. The Apache Velocity Project, <https://velocity.apache.org/>
7. Kenneth Barclay and John Savage, Chapter 21 - Template engines, Groovy Programming, Morgan Kaufmann, 2007, Pages 297-300
8. Kenneth Barclay and John Savage, Chapter 21 - Template engines, Groovy Programming, Morgan Kaufmann, 2007, Pages 297-302