Java WebService code genaration using FreeMarker

# Soap Web-service template

Sometimes we need to generate, template source code upon which developer do the modifications.

It will save a lot of time for developer. Without Automated generation of source code developers need to create source files and require methods repetitively, tools will generate files for them only developer has to put their logic on generate files.

Say, for all JPA entities in a project basic CRUD operations will expose as web service. Either Developer create Web service manually or we can create the same by tool which will generate the same.

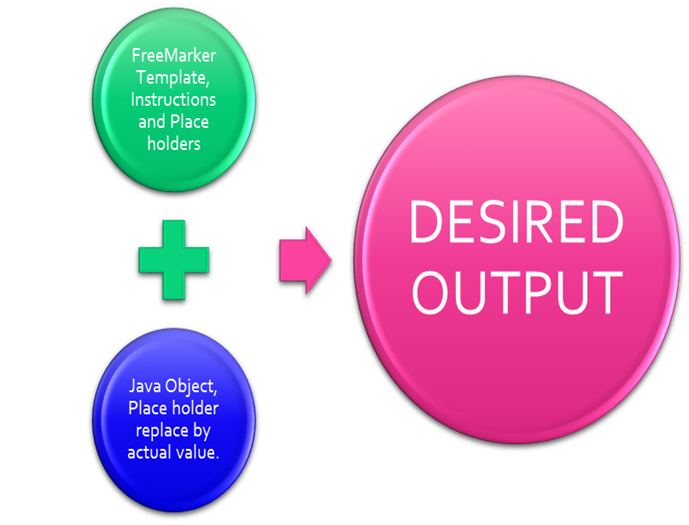
In this article we will generate a simple **java web service** using **Free Marker**. Later you can modify the same to fit it in your project.

Free marker is a **java template engine** which will dynamically create an output. To generate Output

Free Marker needs a FTL file (Free marker template language), Where instructions and place holder are written.

And a Java file where you provide the actual data which will replace the place holder.

Free maker engine combines them and generates the desired Output.



**FreeMarker output generation Process**

**Benefits of Free marker (taken from free marker website):**

1. Powerful template language: Conditional blocks, iterations, assignments, string and arithmetic operations and formatting, macros and functions, including other templates, escaping by default (optional), and many more.
2. Multipurpose and lightweight: Zero dependencies, any output format, can load templates from any place (pluggable), many configuration options
3. Internationalization/localization-aware: Locale sensitive number and date/time formatting, localized template variations.
4. XML processing capabilities: Drop XML DOM-s into the data-model and traverse them, or even process them declaratively
5. Versatile data-model: Java objects are exposed to the template as a tree of variables through pluggable adapters, which decides how the template sees them.

To generate a Java Webservice file usinf FreeMarker following steps are needed.

1, Create a FTL file, file extension will be . ftl. Here we define commands for output define palace holder, which will be replaced by actual value from java objects.

${package}

import javax.jws.\*;

@WebService()

public class ${name}

{

@WebMethod()

public ${return} ${methodname}(${params})

{

${body}

return ${val}

}

}

1. Next create a java file which will provide data for place holder and says the location where Output will be saved.

package com.example;

import java.io.File;

import java.io.FileWriter;

import java.io.IOException;

import java.io.OutputStreamWriter;

import java.io.Writer;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

import freemarker.template.Configuration;

import freemarker.template.Template;

import freemarker.template.TemplateException;

public class WebServiceGenerator {

private static WebServiceGenerator engine = new WebServiceGenerator();

private Template template=null;

Map<String, Object> dataMap = new HashMap<String, Object>();

private WebServiceGenerator()

{

init();

}

private void init()

{

Configuration cfg = new Configuration();

try {

template = cfg.getTemplate("src/javaGen.ftl");

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

public static WebServiceGenerator get()

{

return engine;

}

public WebServiceGenerator buildData()

{

dataMap.put("package", this.getClass().getPackage()+";");

dataMap.put("name", "HelloWorldservice");

dataMap.put("return", "String");

dataMap.put("methodname", "hello");

dataMap.put("params", "String name");

dataMap.put("body", "String res= \"Hi\" + name;\n System.out.println(res);");

dataMap.put("val", "res;");

System.out.println("Preparing Data");

return engine;

}

public void writeFile()

{

Writer file=null;

try {

file = new FileWriter (new File("C:\\installer\\HelloWorldservice.java"));

template.process(dataMap, file);

file.flush();

System.out.println("Success");

}catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

finally

{

try {

file.close();

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

public static void main(String[] args) {

WebServiceGenerator.get().buildData().writeFile();

}

}

Output :

**package** com.example;

**import** javax.jws.\*;

@WebService()

**public** **class** HelloWorldservice

{

@WebMethod()

**public** String hello(String name)

{

String res= "Hi" + name;

System.***out***.println(res);

**return** res;

}

}