## 简介

Web模板引擎是为了使用用户界面与业务数据（内容）分离而产生的，它可以生成特定格式的文档，通常是标准的HTML文档。当然不同的开发语言有不同模板引擎，如JavaScript下的Hogan、ASO下的aspTemplate、以及PHP下的Smarty，这里主要介绍基于JavaScript语言的模板引擎，目前流行有Mustache、Hogan、doT.js、JsRender、Kendo UI Templates等，<http://jsperf.com>上可以看到他们的性能对比。

Mustache是一个轻逻辑的模板语言（Logic-less templates），mustache是一个js模板，用于展示和js分离，它的优势在于可以应用在JavaScript、PHP、Python、Perl等多种编程语言中。原本是基于JavaScript实现的模板引擎，类似于freemark和valicity，但是比freemark和valicity更加轻量级更加容易使用，经过拓展目前支持JavaScript、Java、.NET，PHP，C++等多种平台下开发！

项目首页：<http://mustache.github.com/>

项目文档：<http://mustache.github.com/mustache.5.html>

Demo: [http://mustache.github.com/#demo](http://mustache.github.com/#23demo)

## 模板

1. js模板可以是一个html文件

<h1>Hello {{name}}, it is {{timeNow}}.</h1>

1. js模板可以是一个值是html代码的js变量

var template = "<p>Hello {{name}}, it is {{timeNow}}.</p>";

1. js模板可以是一个script片段

<script id="tpl-greeting" type="text/html">

<dl>

<dt>Name</dt>

<dd>{{name}}</dd>

<dt>Time</dt>

<dd>{{timeNow}}</dd>

</dl>

</script>

## mustache模板语言

1. 变量：变量标记将当前上下文的变量通过模板渲染，如果当前上细纹不存在该变量，则渲染为空串。

默认变量会被经过HTML编码处理，如需显示原始值，用三个大括号或者在模板标记的初始加入&符号。

{{变量名}}, {{{变量名}}}

如果当前键为基本或对象，则渲染一次，如果为数组，则渲染数组长度次数。节点以#号开始，以/结束。

模板{{#数组}}{{数组内的key}}{{/数组}}

json: {"stooges":[{"name":"王升"},{"name":"梁文彦"},{"name":"石洋"}]};

模板：{{#stooges}}<b>{{name}}</b><br>{{/stoogles}}

1. 填充数组节点以#开始，以/结束，则渲染数组长度次数。

模板{{#数组}}{{数组内的key}}{{/数组}}

var template = "{{#stooges}}<b>{{name}}</b><br>{{/stooges}}";

var view = {"stooges":[{"name":"王升"},{"name":"梁文"},{"name":"石洋"}]}

1. 函数作为模板的变量，该函数会在当前列表的每一个元素的上下文迭代执行。

var template = "{{#beatles}}\* {{name}}<br/>{{/beatles}}";

var view = {

"beatles": [

{ "firstName": "John", "lastName": "Lennon" },

{ "firstName": "Paul", "lastName": "McCartney" },

{ "firstName": "George", "lastName": "Harrison" },

{ "firstName": "Ringo", "lastName": "Starr" }],

"name": function () {

return this.firstName + " " + this.lastName;}

};

如果节点键的值为函数，注意该函数在执行时的两个参数，第一个为该节点变量的直接值，第二个为函数，其执行的上下文对应视图对象。

var template = "{{#bold}}Hi {{name}} {{lastName}}.{{/bold}}";

var view = {"name": "John","lastName": "Lennon","bold": function () {

　　　　return function (text, render) {return "<b>" + render(text) + "</b>";}

}}

1. mustache模板使用

模板变量在上下文找模板变量需要的数据进行填充。json数据的key根模板的变量对应就会填充。模板变量和json的key对应不上就不填充。生成展示代码没填充的变量不显示用Mustache.render(template, view)方法填充数据生成展示代码。

view——为json数据，作为模板上下文

template——为模板对象

//模板

var template = "<p>Hello {{name}}, it is {{{ignore name}}}.it is {{&name}}</p>";

//数据

var date = new Date();

var view = {name: "<b>Jonny</b>",timeNow: date.getHours() + ':' + date.getMinutes() };

使用模板文件要用该方法填充数据生成展示代码：

$.get('模板文件', function(templates) {

var template = $(templates).html();

$Mustache.render(template, view);

});

## Mustache语法

Mustache的模板语法很简单，就那么几个，用于HTML，配置文件，源代码等。它的工作方式是通过以哈希值或者对象的方式扩展模板标签。

{{keyName}}

{{#keyName}} {{/keyName}}

{{^keyName}} {{/keyName}}

{{.}}

{{<partials}}

{{{keyName}}}

{{!comments}}

...

<script type="text/javascript" src="mustache.js"></script>

<script type="text/javascript">

var data = {

"company": "Apple",

"address": {

"street": "1 Infinite Loop Cupertino</br>",

"city": "California ",

"state": "CA ",

"zip": "95014 "

},

"product": ["Macbook ","iPhone ","iPod ","iPad "]

}

var tpl = '<h1>Hello {{company}}</h1>';

var html = Mustache.render(tpl, data);

console.log( html )

</script>

...

//运行后 Console 输出：

<h1>Hello Apple</h1>

1. {{keyName}}

{{}}jiushi Mustache的标示符，花括号里的keyName表示键名，这句的作用是直接输出与键名匹配的键值，例如：

1. {{#keyName}}{{/keyName}}

以#开始，以/结束表示区块，它会根据当前上下文中的键值来对区块进行一次或多次渲染，例如改写下Demo中的tpl：

注意：如果{{#keyName}}{{/keyName}}中的keyName值为null，undefined、false；则不渲染输出任何内容。

1. {{^keyName}}{{/keyName}}

该语法与{{#keyName}}{{/keyName}}类似，不同在于它是当keyName值为null，undefined，false时才渲染输出该区块内容。

1. {{.}}

{{.}}表示枚举，可循环输出整个数组，例如：

1. {{>partials}}

以>开始表示字模块，如{{>address}};当结构比较复杂时，我们可以使用该语法将复杂的结构分成几个小的字模块，例如：

1. {{{keyName}}}

{{keyName}}输出会将等特殊字符转译，如果想保持内容原样输出可以使用{{{}}}，例如：

1. {{!comments}}

!表示注释，注释后不会渲染输出任何内容。

## 4.Mustache.java和Mustache.js开发

java开发参见<https://github.com/spullara/mustache.java>

js开发参见<https://github.com/janl/mustache.js>