

artTemplate-3.0

新一代 javascript 模板引擎

目录

- 特性
- 快速上手
- 模板语法
- 方法
- NodeJS
- 使用预编译
- 更新日志
- 授权协议

特性

1. 性能卓越，执行速度通常是 Mustache 与 tpl 的 20 多倍 ([性能测试](#))
2. 支持运行时调试，可精确定位异常模板所在语句 ([演示](#))
3. 对 NodeJS Express 友好支持
4. 安全，默认对输出进行转义、在沙箱中运行编译后的代码 (Node版本可以安全执行用户上传的模板)
5. 支持include语句
6. 可在浏览器端实现按路径加载模板 ([详情](#))
7. 支持预编译，可将模板转换为非常精简的 js 文件
8. 模板语句简洁，无需前缀引用数据，有简洁版本与原生语法版本可选
9. 支持所有流行的浏览器

快速上手

编写模板

使用一个type="text/html"的script标签存放模板：

```
1 <script id="test" type="text/html">
2 <h1>{{title}}</h1>
3 <ul>
4   {{each list as value i}}
5   <li>索引 {{i + 1}} : {{value}}</li>
6   {{/each}}
7 </ul>
8 </script>
```

渲染模板

```
1 var data = {
2   title: '标签',
3   list: ['文艺', '博客', '摄影', '电影', '民谣', '旅行', '吉他']
4 };
5 var html = template('test', data);
6 document.getElementById('content').innerHTML = html;
```

[演示](#)

模板语法

有两个版本的模板语法可以选择。

简洁语法

推荐使用，语法简单实用，利于读写。

```
1 {{if admin}}
2 {{include 'admin_content'}}
3
4 {{each list}}
5 <div>{{index}}. {{value.user}}</div>
6 {{/each}}
7 {{/if}}
```

[查看语法与演示](#)

原生语法

```
1 <%if (admin){%>
2 <%include('admin_content')%>
3
4 <%for (var i=0;i<list.length;i++) {%>
5 <div><%=i%>. <%=list[i].user%></div>
6 <%}%>
7 <%}%>
```

[查看语法与演示](#)

下载

- [template.js](#) (简洁语法版, 2.7kb)
- [template-native.js](#) (原生语法版, 2.3kb)

方法

template(id, data)

根据 id 渲染模板。内部会根据document.getElementById(id) 查找模板。

如果没有 data 参数，那么将返回一渲染函数。

template.compile(source, options)

将返回一个渲染函数。[演示](#)

template.render(source, options)

将返回渲染结果。

template.helper(name, callback)

添加辅助方法。

例如时间格式器：[演示](#)

template.config(name, value)

更改引擎的默认配置。

openTag	String	'{'	逻辑语法开始标签
closeTag	String	'}'	逻辑语法结束标签
escape	Boolean	true	是否编码输出 HTML 字符
cache	Boolean	true	是否开启缓存（依赖 options 的 filename 字段）
compress	Boolean	false	是否压缩 HTML 多余空白字符

使用预编译

可突破浏览器限制，让前端模板拥有后端模板一样的同步“文件”加载能力：

一、按文件与目录组织模板

```
1 template('tpl/home/main', data)
```

二、模板支持引入子模板

1	<code>{{include '../public/header'}}</code>
---	---

基于预编译:

- 可将模板转换为非常精简的 js 文件（不依赖引擎）
- 使用同步模板加载接口
- 支持多种 js 模块输出: AMD、CMD、CommonJS
- 支持作为 GruntJS 插件构建
- 前端模板可共享给 NodeJS 执行
- 自动压缩打包模板

预编译工具: [TmodJS](#)

NodeJS

安装

1	<code>npm install art-template</code>
---	---------------------------------------

使用

1	<code>var template = require('art-template');</code>
2	<code>var data = {list: ["au", "test"]};</code>
3	
4	<code>var html = template(__dirname + '/index/main', data);</code>

配置

NodeJS 版本新增了如下默认配置:

base	String	"	指定模板目录
extname	String	'html'	指定模板后缀名
encoding	String	'utf-8'	指定模板编码

配置base指定模板目录可以缩短模板的路径, 并且能够避免include语句越级访问任意路径引发安全隐患, 例如:

```
template.config('base', __dirname);
```

```
var html = template('index/main', data)
```

NodeJS + Express

1	<code>var template = require('art-template');</code>
2	<code>template.config('base', '');</code>
3	<code>template.config('extname', '.html');</code>
4	<code>app.engine('.html', template._express);</code>
5	<code>app.set('view engine', 'html');</code>
6	<code>//app.set('views', __dirname + '/views');</code>

运行 demo:

1	<code>node demo/node-template-express.js</code>
---	---

若使用 js 原生语法作为模板语法, 请改用 `require('art-template/node/template-native.js')`

/*

模板引擎:

template
\${}

js -> JQ -> template (相对JQ提升了性能问题) [artTemplate]

chrome
342
5

```

        13      5
    iE
        2689

        122

        196      17

    */
/*
    <%=title%>    //输出
    <%for () {%>    //标签对

    <%}%>

```

1.引入template.js

2.创建script标签并且type="text/html" 给id

模板代码就放到这个script标签中

3.页面中创建一个box，给id

4.在自己的script中，调用template

第一个参数:第2步的ID

第二个参数:放入数据

5.将template的返回值放到第三步的容器中

```

    */

```