artTemplate

**插件描述：高性能JavaScript模板引擎**

artTemplate-3.0

新一代 javascript 模板引擎

特性

1. 性能卓越，执行速度通常是 Mustache 与 tmpl 的 20 多倍（[性能测试](http://aui.github.com/artTemplate/test/test-speed.html)）
2. 支持运行时调试，可精确定位异常模板所在语句（[演示](http://aui.github.io/artTemplate/demo/debug.html)）
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](https://github.com/aui/artTemplate#%E4%BD%BF%E7%94%A8%E9%A2%84%E7%BC%96%E8%AF%91)）
7. 支持预编译，可将模板转换成为非常精简的 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：

<script id="test" type="text/html">

<h1>{{title}}</h1>

<ul>

    {{each list as value i}}

        <li>索引 {{i + 1}} ：{{value}}</li>

    {{/each}}

</ul>

</script>

渲染模板

var data = {

    title: '标签',

    list: ['文艺', '博客', '摄影', '电影', '民谣', '旅行', '吉他']

};

var html = template('test', data);

document.getElementById('content').innerHTML = html;

[演示](http://aui.github.com/artTemplate/demo/basic.html)

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。

{{if admin}}

    {{include 'admin\_content'}}

    {{each list}}

        <div>{{$index}}. {{$value.user}}</div>

    {{/each}}

{{/if}}

[查看语法与演示](https://github.com/aui/artTemplate/wiki/syntax:simple)

原生语法

<%if (admin){%>

    <%include('admin\_content')%>

    <%for (var i=0;i<list.length;i++) {%>

        <div><%=i%>. <%=list[i].user%></div>

    <%}%>

<%}%>

[查看语法与演示](https://github.com/aui/artTemplate/wiki/syntax:native)

下载

* [template.js](https://raw.github.com/aui/artTemplate/master/dist/template.js) (简洁语法版, 2.7kb)
* [template-native.js](https://raw.github.com/aui/artTemplate/master/dist/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](http://aui.github.com/artTemplate/demo/compile.html)

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](http://aui.github.com/artTemplate/demo/helper.html)

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**

template('tpl/home/main', data)

二、**模板支持引入子模板**

{{include '../public/header'}}

基于预编译：

* 可将模板转换成为非常精简的 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](http://github.com/aui/tmodjs/)

NodeJS

安装

npm install art-template

使用

var template = require('art-template');

var data = {list: ["aui", "test"]};

var html = template(\_\_dirname + '/index/main', data);

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

var template = require('art-template');

template.config('base', '');

template.config('extname', '.html');

app.engine('.html', template.\_\_express);

app.set('view engine', 'html');

//app.set('views', \_\_dirname + '/views');

运行 demo:

node demo/node-template-express.js

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