

有哪些好用的前端模板引擎?

关注问题

写回答

前端开发 JavaScript

关注者

被浏览

1,775

543,445

有哪些好用的前端模板引擎?

- 欢迎大家列举:
- * 自己用过的 (jQuery也算, Backbone, Anguar, React当然算)
 - * 自己喜爱的 (除了纯前端工程美学角度的喜爱, 还有生产实践角度的喜爱)
 - * 自己看到并觉得不错的 (包括在github上面众多的前端模板引擎)
 - * 自己公司用的 (包括BAT网易自己的较好实现)
 - * 自己写的 (Oh Yeah我也写过一个来贴上来! :))

- 也不限于前端模板引擎,
- * 前端模板引擎
 - * 后端模板引擎
 - * 前后端通用模板引擎
 - * 前端库/框架
 - * 新式的前端开发框架

等等都算。

前端领域轮子真多!!!! 大家造轮子的激情真的好厉害!!!!

关注问题

写回答

邀请回答

好问题 14

5 条评论

分享

收起

44 个回答

默认排序

KEEP CALM AND MAKE EPIC SHIT

尤雨溪 前端开发话题下的优秀答主

+ 关注

484 人赞同了该回答

我对题主的问题有些意见。

首先, [jQuery.templ](#) 是一个插件, jQuery 本身跟模板毫无关系。Backbone 所谓的 template 功能是 underscore 里面的一个函数, 是可替换的, 你手动拼接 html 字符串也可以, 说是 underscore template 更准确一点 (其实原版还是 John Resig 写的)。

其次, 我很反对把 Angular React 这样的东西和模板引擎混为一谈。“[前端框架](#)”和“模板引擎”是两个概念。我看到有人把所有的框架都笼统的叫做模板引擎, 就像看到有人把 HTML5 叫成 H5 一样不舒服。我们可以说 Angular 有针对模板的处理策略, 但它无论是理念、实现和使用方式和通常所说的[模板引擎](#)都是完全不一样的, 所包含的功能也远远超过模板引擎的范畴。React 就更不一样了, 因为丫根本就没有模板的概念。

严格的模板引擎的定义, 输入模板字符串 + 数据, 得到渲染过的字符串。实现上, 从[正则替换](#)到拼 function 字符串到正经的 AST 解析各种各样, 但从定义上来说都是差不多的。字符串渲染的性能其实也就在后端比较有意义, 毕竟每一次渲染都是在消耗服务器资源, 但在前端, 用户只有一个, 几十毫秒的[渲染时间](#)跟请求延迟比起来根本不算瓶颈。倒是前端的后续更新是[字符串模板引擎](#)的软肋, 因为用渲染出来的字符串整个替换 [inne](#)rHTML 是一个效率很低的更新方式。所以这样的模板引擎如今在纯前端情境下已经不再是好的选择, 意义更多是在于方便前后端[共用模板](#)。

相比之下 Angular 是 DOM-based templating, 直接解析 live DOM 来提取绑定, 如果是字符串模板则是先转化成 live DOM 再解析。数据更新的时候直接通过绑定做局部更新。其他 MVVM 如 Knockout, Vue, Avalon 同理。缺点是没有现成的服务端渲染, 要做服务端渲染基本等于重写一个



相关问题

有哪些好用的前端模板引擎值得分享? 0 个回答

有哪些好用的前端模板引擎值得推荐的? 1 个回答

哪些常见的前端模板引擎可以使用? 0 个回答

前端模板引擎的作用有哪些? 1 个回答

前端模板引擎哪些值得推荐? 0 个回答

有哪些好用的前端模板引擎？

框架里面也有在前端解析字符串^Q模板到静态 AST 再生成 live DOM 做局部更新的，比如 Ractive 和

@郑海波

的 Regular。这一类的实现因为解析到 AST 的这一步已经在框架内部完成了，所以做服务端渲染几乎是现成的，另外也可以在构建时进行预编译^Q。

然后说说 React，JSX 根本就不是模板，它就是带语法糖的手写 AST，并且把语法糖^Q的处理放到了构建阶段。因为运行时不需要解析，所以 virtual DOM 可以每次渲染都重新生成整个 AST，在客户端^Q用 diff + patch，在服务端则直接 serialize 成字符串。所有其他 virtual DOM 类方案同理。像 virtual-dom，mithril^Q之类的连语法糖都不带。

最后说下我的看法：如果是静态内容为主，那就直接服务端渲染好了，首屏^Q加载速度快。如果是动态的应用界面，那就不应该用拼模板的思路去做，而是用做应用的架构（MV*，组件树^Q）思路去做。

编辑于 2015-07-21 04:28

▲ 赞同 484 ▼

● 26 条评论

🔗 分享

★ 收藏

♥ 喜欢

...

收起 ^



张立理 前端开发话题下的优秀答主

+ 关注

168 人赞同了该回答

模板引擎^Q有很多，比较知名的在各回答里也都提及了，而我要说的是一个更根本的问题，怎么样的模板引擎是适合前端的

首先说明一点，underscore.template这种东西不叫模板引擎，他还不够格来配上“引擎”这两个字，充其量就是一个模板函数^Q，因此不会在这个回答的探讨范围里

由于这个推理的过程相当冗长，基本是用好几年的实际经验和踩过无数坑后堆出来的，别的不说光我们团队自己用的模板引擎都前后有3个版本，现在我正在思考第4个版本应该有的特性（某实现者已吐血），可见我们对此的思考从来没有停止。所以这边只说一些结论性的东西，一个前端的模板引擎需要有怎么样的特性

以下特性没有特定的排序，不分先后，纯粹想到一条写一条出来，也作为工作到现阶段在模板引擎这一话题上的总结吧

前端模板引擎需要有开发时的透明性

我认为前端任何框架和工具都要有对开发的透明性，模板引擎也不例外。所谓透明性即指我在搭建好开发环境^Q后，随手写代码随手刷新浏览器就能看到最新的效果，而不需要额外地执行任何命令或有任何的等待过程

所以一切依赖编译过程的模板引擎并不适合前端使用，编译只能是模板引擎的一个特性，而不能是使用的前提

更严格地说，使用FileWatch等手段进行文件变更检测并自动编译也不在我的考虑范围之内，因为这会造成额外的等待，像我这种手速极快的人可能编译速度跟不上

由此可以推出，前端的模板引擎应该是具备可在纯前端环境中解析使用的能力的

前端模板引擎要有良好的运行时调试能力



帮助中心

知乎隐私保护指引 申请开通机构号 联系我们

举报中心

涉未成年举报 网络谣言举报 涉企侵权举报 更多

关于知乎

下载知乎 知乎招聘 知乎指南 知乎协议 更多

京 ICP 证 110745 号 · 京 ICP 备 13052560 号 - 1 ·
京公网安备 11010802020088 号 · 京网文
[2022]2674-081 号 · 药品医疗器械网络信息服务备
案（京）网药械信息备字（2022）第00334号 · 广
播电视节目制作经营许可证：（京）字第06591号 ·
服务热线：400-919-0001 · Investor Relations ·
© 2024 知乎 北京智者天下科技有限公司版权所有 ·
违法和不良信息举报：010-82716601 · 举报邮箱：
jubao@zhihu.com



有哪些好用的前端模板引擎?

也导致前端必然存在需要直接在线上排查问题的情况

而当问题出现在模板引擎这一层时,就需要模板引擎提供良好的调试能力

一般来说,编译后生成的函数的调试能力是弱于原先手动编写的模板片断的,因为自动生成的函数基本不具备可读性和可断点跟踪性

因此在这一点上,一个供前端使用的模板引擎应该具备在特定情况下从“执行编译后函数获取HTML”换回“解析原模板再执行函数获取HTML”的模式,即应该支持在两种模式间切换

或者更好地,一个强大的前端模板引擎编译生成的函数,可以使用Source Map或其它自定义的手段直接映射回原模板片段,不过现在并没有什么模板引擎实现了这一功能

前端模板引擎要对文件合并友好

在HTTP/2普及之前,文件合并依旧是前端性能优化中的一个重要手段,模板作为文件的一部分,依旧是需要合并的

在提供编译功能的模板引擎中,我们可以使用编译的手段将模板变为JavaScript源码,再在JavaScript的基础上做文件合并

但是如果我们出于上文所说的调试能力等原因希望保留原模板片段^Q,那就需要模板引擎本身支持模板片段合并为一个文件了

大部分仅支持将一段输入的字符串作为模板解析的引擎并不具备这一能力,他们天生并不能将一整个字符串切分为多个模板片段,因而无法支持模板片段层面上的文件合并

需要实现对文件合并的支持,最好的办法就是让模板的语法是基于“片段”的,如下面的模板

```
[[title]]
<h2>{{title}}</h2>

[[content]]
<section class="article-content">
  {{content}}
</section>
```

就由title和content两个片段组成,多个片段可以出现在一个文件中,这就在语法层面上支持了我们的文件合并

前端模板引擎要担负XSS的防范

从安全性上来说,前端对XSS的控制是有严格要求的

我在
[单页面\(SPA\)开发会不会比多页面有更多的安全问题? - 张立理的回答](#)

中有提到过,前端对XSS的防范比较合适的方法是使用“默认转义”的白名单^Q式策略

基于此,一个合理的模板引擎是**必须**支持默认转义的,即所有数据的输出都默认经过escape的逻辑处理,将**关键符号**^Q转为对应的HTML实体符号,以从根源上杜绝XSS的入侵路径

当然并不是所有的内容都必须经过转义的,在系统中免不了有对用户输入富文本的需求,因此需要支持特定的语法来产生**无转义**^Q的输出,但时刻注意无转义输出才是特例,默认情况下必须是转义输出的

有哪些好用的前端模板引擎?

这可不是前端模板引擎的需求，事实上任何模板引擎都应该支持片段的复用，后端如Velocity、Smarty等无不拥有此功能

所谓[片段复用](#)^Q，应该有以下几个层次的应用：

1. 一个片段可以被引入到另一处，相当于一个变量到处用的效果
2. 一个片段被引入时，可以向其传递不同的数据，相当于一个函数到处用的效果
3. 一个片段可以被外部替换，但外部不提供此片段的话保持一个默认的内容，类似设计模式中的[策略模式](#)^Q

满足第1和第2点的模板引擎并不少，而满足第3点的前端模板引擎却不多见，而后端的Razor、Smarty等都具备这一功能

话说我当时设计我们自己的模板引擎的第3个版本时，就想出了block这一个概念来实现第3点，在做完交付将近半年之后，有人告诉我说Smarty上就有这概念，顿时有种不知应该高兴还是悲伤的不知所措感。还好他并没有怀疑我直接抄了别人的功能，不然真是冤枉

前端模板引擎要支持数据输出时的处理

所谓数据输出时处理，指一个数据要在输出时做额外的转换，最常见的如字符串的trim操作，比较技术性的如markdown的转换等

诚然数据的转换完全可以在将数据交给模板引擎前就通过JavaScript的逻辑处理完，但这会导致不少有些丑陋又有些冗余的代码，对逻辑本身的复用性也会造成负面的影响

通常模板引擎对数据做额外处理会使用filter的形式实现，类似[bash](#)^Q中的管道的逻辑。filter的实现和注册也会有不同的设计，如[mustache](#)^Q其实注册的是filter工厂，而另一些模板引擎则会直接注册filter本身，不同设计有不同的考量点，我们很难说谁好谁坏

但是，模板引擎支持数据的输出处理后，会另我们在编码过程中产生一个新的纠结，即哪些数据处理应该交由模板引擎的filter实现，哪些应该在交给模板引擎前由自己的逻辑逻辑实现。这个话题展开来又是一篇长长的论述，于当前的话题无关就略过吧

前端模板引擎要支持动态数据

在开发过程中，其实有不少数据并不是静态的，如EmberJS就提供了Computed Property这样的概念，Angular也有类似的东西，Backbone则可以通过重写Model的get方法来变相实现

虽然ES5在语言层面上直接提供了getter的支持，但我们在[前端开发](#)^Q的大部分场景下依旧不会使用这一语言特性，而会选择将动态的数据封装为某种对象的get等方法

而模板引擎在将数据转为HTML片段的过程中，同样应该关注这一点，对这些动态计算的数据有良好的支持

说得更明白一些，模板引擎不应该仅仅接受纯对象（Plain Object）作为输入，而应该更开放地接受类似带有get方法的动态的数据

一个比较合理的逻辑是，如果一个对象有一个get方法（模板引擎决定这个接口），则数据通过该方法获取，其它情况下视输入的对象为纯对象（Plain Object），使用标准的属性获取逻辑

前端模板引擎要与[异步流程](#)^Q紧密结合

前端有一个很大的特点，就是到处充斥着异步的流程。由于JavaScript在浏览器提供的引擎中单线程执行的特性、大部分与IO相关的API都暴露为异步的事实，以及多数模块定义规范中模板的动态获取是异步的这一现象，注定我们无法将这个世界当作完全同步来看

有哪些好用的前端模板引擎?

模板引擎，但这是一种业务与视图相对耦合的玩法，出于强迫症我并不觉得这是一个漂亮的设计，所以我们希望直接在模板中这么写：

```
<footer>{{ $globals.ICP_SERIAL }}</footer>
```

这是我假想的一个语法，通过\$globals可以使用AMD Loader获取globals这一模块，随后获取其中的ICP_SERIAL属性输出

模板引擎支持异步是一个比较具有挑战性的话题，我的计划是在我们自己的模板引擎的下一个版本中尝试实现。这其中涉及很多的技术点，比如：

- 模板的输出本身成了异步的方法，而不再像现在一样直接返回字符串
- 分析模板对异步操作的依赖，整个字符串的拼接逻辑被打断成多个异步
- 异步是需要等待的，且等待是未知的，从性能上考虑，是否需要考虑Stream式的输出，以便完成一段提供一段
- 是提供内置的固定几种异步逻辑，还是基于Promise支持任何自定义的异步逻辑，在复杂度和实用性上作出平衡

至今我还没有完全明确模板与异步结合的方式和接口，这个话题也没办法继续深入探讨了

前端模板引擎要支持不同的开发模式

前端发展至今，有很多不同的开发模式，比如：

- 最普通的HTML页面，使用DOMContentLoaded等事件添加逻辑，特定交互下局部刷新页面
- 采用传统的MVC模型进行单页式开发
- 使用MVVM方式以数据为核心，数据与视图方向绑定进行开发
- 基于Immutable Data进行数据比对Diff转DOM更新的开发（其中可能有Virtual DOM的引入）

一个模板引擎要能支持这么多种不同的模式是一个非常大的挑战，特别是对双向绑定的支持尤为突出。至今为止几乎所有的支持双向绑定的开发框架都自带了专用的模板引擎，这是因为双向绑定对模板有两大要求：

- 能够从模板中提取“这一模板对哪些数据有依赖”的元信息
- 能够知道一个数据变化引擎的是模板的哪一块，而不至于整个刷新

而通用模板引擎很少提供这两个特性，所以没办法对不同的前端开发模式进行全面到位的支持

从模板引擎本身的实现上来说，一种方法是直接将模板解析后的类似AST的结构暴露出去，供其他框架合理地处理，同时提供对模板局部的刷新功能（也可与前面所说的模板片段一起考虑），但是大部分模板引擎为了性能等考虑，是不会解析出类似AST的语法结构来的

前端模板引擎要有实例间的隔离

在大型的前端项目，特别是单页式的项目中，会有完全未知个数的模板片段同时存在，如果这些片段是带有名称（出于复用的考虑）的，就很容易造成名称上的冲突

对于同一层级的逻辑（如大家都是业务层代码，或者大家都是控件层代码），名称冲突是可以通过一些开发时的约定来解决的。但不同层之间，由于封装性的要求，外部不应该知道一些仅内部使用的片段的名称，此时如果不幸有名称与其它层有冲突，会让情况变得比较麻烦，这类问题甚至都不容易跟踪，往往会导致大量的精力和时间的浪费

因此，一个好的模板引擎应该是多实例的，且不同实例间应该相互具备隔离性，不会出现这种不可预期的冲突

有哪些好用的前端模板引擎?

实例的关系是一种组合依赖但又具备基本的封装和隔离的状态

说了这么多，其实现在市面上大致是没有任何一个模板引擎能满足全部这些条件的，至少“[异步流程集成^Q](#)”和“多开发模式支持”这两点就不是一个普通的模板引擎会去考虑的场景，只有我们这种大型的系统才会遇到并在这方向上产生一些思考

因此在选择或设计开发模板引擎时，这些准则可以给你作一个参考，却不需要完全去遵从

推荐自己做的东西就好像把初中二年级写的小说给别人看一样充满着耻感，所以我不推销我们自己的产品了.....

为啥我不把这些写成博客去骗下季度的部门稿费--

编辑于 2015-07-20 02:06

▲ 赞同 168 ▼ ● 28 条评论 ↗ 分享 ★ 收藏 ♥ 喜欢 ... 收起 ^



Jim Liu

前端开发话题下的优秀答主

98 人赞同了该回答

首先要区分出（传统的）string-based模板引擎和（新潮的）dom-based模板引擎。

string based模板引擎的目标是输出字符串，几乎是所有时候，它的输入也是字符串，也就是所谓的模板。

string based[模板引擎^Q](#)当中应用最广泛的，我猜是PHP的Smarty模板引擎，如果你对此有了解，那么所谓string based模板引擎的理解就没什么问题了。

string based模板引擎的输出并不一定是要HTML，其实你想输出什么都可以，比如我经常用模板引擎来输出ini文件神马的，还有公司使用的各种结构化格式的[配置文件^Q](#)，还用来生成过CSS文件（iconfont平台的自动构建神马的），甚至用来输出过shell和SQL脚本（所谓板砖一般的代码）。

但是类似Jade这类的引擎，它基本上就只能输出[标记语言^Q](#)，比如HTML/XML了，它就是为此而设计的。

至于题目“[有哪些好用的前端模板引擎^Q](#)”，JS这面我觉得佳作来自国内的

[ecomfe/etpl · GitHub](#)

和

[aui/artTemplate · GitHub](#)

，首先它们的速度都非常快，这对于大流量的服务端来说是很重要的，其次它们的需求出发点都是国内的知名互联网公司，比较符合国情（笑）。根据我自己的体验（用的不多），比国外的[轮子^Q](#)要好。当然我自己也造了一个轮子

[LiuJi-Jim/raze-tpl · GitHub](#)

，吸收了这两个的一些优点，也有自己的风格和优点，不过属于玩票性质，BUG没功夫修，不建议任何人拿去用在自己的项目里（我也只是用在我自己的项目里，没在公司项目用，逃）.....

而[dom-based模板引擎^Q](#)，通常输出就直接是dom了（当然，很多dom-based模板引擎也可以很方便的挂载string输出端，从而在服务端也能输出），输入也不一定，可能就是已经存在的DOM树（比如AngularJS直接写在页面上的时候），可能是字符串（如AngularJS使用字符串模板的时候）

有哪些好用的前端模板引擎?

语法”，比如变量binding^Q啊、比如循环、条件啊什么的。

dom-based模板引擎基本上不考虑输出HTML/XML以外的东西。

相比之下dom-based模板引擎可以实现像Matt-Esch/virtual-dom • GitHub这类的东西，在数据更新的时候实现最小操作，并不像string-based模板引擎那样，可能改了一个值就要重新生成巨大的一部分字符串。

string-based模板引擎在数据更新这个问题对于交互比较多的页面这是不友好的，意味着即使你用模板引擎生成了HTML代码，再把它innerHTML成一堆DOM，你还是得用原本的方式（比如jQuery）去操作这些DOM实现交互，这个过程中，模板引擎仅仅起到了帮你生成那堆DOM这一个问题，后面怎么去玩他们的问题就没下文了。

个人认为，在浏览器端，string-based模板引擎的应用场景越来越有限，string-based模板引擎对现在提倡的组件化^Q开发其实是挺不友好的。与此同时，因为MV*——尤其是MVVM已经越来越被认可代表了先进生产力^Q的发展方向，而MVVM对dom-based模板的依赖几乎是与连体婴的关系。

随着node.js^Q服务端越来越被人接受，string-based模板引擎在做一些以输出页面为主的服务当中还是很好用的。但在dom-based模板引擎大炒“同构渲染^Q”的趋势之下，string-based模板引擎的生存空间还是略微受到一些挤压，但因为在需求有限的场景中，简单粗暴出活快不用加班才是目的，string-based模板引擎依然顽强的生存着.....

编辑于 2015-07-22 01:41

▲ 赞同 98 ▼ ● 6 条评论 ↗ 分享 ★ 收藏 ♥ 喜欢 ... 收起 ^

智谱清言

广告 ▼

来智谱清言“AI搜索”一下吧！

智谱清言GLMs智能体的高级联网功能，不仅可以精准识别用户意图，还能快速准确回答问题，无需自主分辨信息，智谱清言直接给出答案 ~ [查看详情](#)



顾轶灵

前端开发话题下的优秀答主

18 人赞同了该回答

问题不要问太大.....后面那两个一加^Q这个问题就凌乱了。

我还是回答你模板引擎的问题吧：

- [Handlebars.js: Minimal Templating on Steroids](#)
- [ETpl - Enterprise Template](#)

有一个针对国外的模板引擎^Q的选择工具：

[So you need a template engine..](#)

--

另外，

@贺师俊

的

有哪些好用的前端模板引擎?

也马上就要发正式版本了, 可以期待一下。

编辑于 2015-07-20 01:17

▲ 赞同 18 ▼ ● 4 条评论 ↗ 分享 ★ 收藏 ♥ 喜欢 ... 收起 ^



颜海镜

《现代JavaScript库开发: 原理、技术与实战》作者

13 人赞同了该回答

[yanhaijing/template.js](#) · GitHub 一款javascript模板引擎^Q, 简单, 好用

唯一官方支持webpack和fis的插件

兼容IE6, 安卓2.3, iphone4

JavaScript语法, 零学习成本

(○v○)嗯, **我写的**, 就酱紫

编辑于 2017-09-08 03:55

真诚赞赏, 手留余香

赞赏

还没有人赞赏, 快来当第一个赞赏的人吧!

▲ 赞同 13 ▼ ● 4 条评论 ↗ 分享 ★ 收藏 ♥ 喜欢 ... 收起 ^