jade这种js模板真的好吗?

模板 模板引擎 javascript jade node.js

Tychio 2013年03月25日提问

2

jade完全改变了html的语法，像官方这个示例

```
doctype 5
html(lang="en")
  head
    title= pageTitle
    script(type='text/javascript')
      if (foo) {
         bar()
      }
  body
    h1 Jade - node template engine
    #container
      if youAreUsingJade
        p You are amazing
      else
        p Get on it!
```

确实，它变得快捷了很多，但是这样是否降低了可移植性？比如页面设计师制作的html就必须被重写成jade的语法，而当我们要更换模板时，成本更高。
而且论效率，我们有zencoding，为什么要在模板上动手脚?

我的问题是，是否有什么特别的优势让jade也能在众多模板中占有一席之地？比如一种转换为普通html的工具或者是兼容html语法之类的。

2013年03月25日提问 1 评论 邀请回答 编辑 ···

7个回答

默认排序 时间排序

6

遇到不少人问我这个问题，看到此问题，特地写了一篇博客。 http://willerce.com/post/the-views-of...

已采纳

作为 Express 的默认模版，Jade 进入了我的视野，如同习惯了 Javascript 的语法，再见到 Coffescript 时的惊艳。优雅、简单是对 Jade 的第一印象，在 Github 上拥有4K的Star足见其受欢迎的程度。

我在刚开发 Noderce 时就用上了 Jade，经过一番学习使用起来倒没有什么问题，但使用 Jade 这个过程并不像想象中的愉快，我遇到了一些的问题，并且耗费在写HTML上的时间有所增加。在之后的 Node.js 项目中我改用 EJS，非常的舒适习惯、效率很高。

我整理了自己使用 Jade 后的一些看法，希望对大家在选择模版引擎时有所帮助。

不能减少你的输入

这是 Jade 官网给的示例

```
doctype 5
  html(lang="en")
    head
    title= pageTitle
    script(type='text/javascript')
    if (foo) {
        bar()
    }
  body
    h1 Jade - node template engine
    #container
    if youAreUsingJade
      p You are amazing
    else
      p Get on it!
```

太棒了，相对手写HTML减少了大量的输入，并且漂亮、缩进清晰、层级明了。但是，似乎用 zencoding 的话，输入量应该是在一个层级，仅看 HTML 部分的话，使用 zencoding 输入量更少。我只需要输入以下内容，以及少许的光标移动

```
html:5
script
h1
#container
p
```

在逻辑方面， Jade 做得不错，不需要 `<%%>` 这样的额外的输入。但总体来说，Jade 并不能降低输入量。

可移植性低

当你的项目使用了一些别的地方拷贝过来的HTML代码时，你需要使用 html2jade 这样的工具来进行转换后才能使用。

调试困难

如果页面中某部分出了问题，首先前端会通过 firebug 定位问题出在哪里，如果是普通模板，直接就

位置，如果页面复杂，这种转换需要的时间也就更长。[via]
(http://segmentfault.com/q/10100000001...)

性能低

[WEB模板jade、ejs、handlebars 万行代码解释效率比较，jade完败]
(http://cnodejs.org/topic/50e70edfa7e6...)

这篇文章提供的效率比较结果（平均消耗时间，约数）
jade 287ms > ejs 43ms > handlebars 28ms

性能不应该成为不使用 Jade 的原因，但快一点总是好的，对吧?

2013年04月01日更新　5 评论　¥ 赞赏　编辑

4

公司有一技术非常热衷这样的模板语言，虽然有点 python 的感觉（我是 Python 死粉），但我还是强烈反对的，理由如下。

1. **前端人员难以接受：**以于公司的前端和设计，它们有时需要为我们调整页面，不要指望他们会喜欢这种语法，如果她们同步静态原型和动态页面的修改，将会非常麻烦（项目中开始的原型是最终定稿的情况非常少）
2. **语言支持太少：**这已经不是 HTML 页面了，页面各IDE对它的语法高亮和缩进，补全等的支持有限（虽然我们 vim ，但我不强制别人也用 vim）
3. **太过小众：**即便同样是开发，也不见得大家都会习惯这种语法
4. **调试困难：**如果页面中某部分出了问题，首先前端会通过 firebug 定位问题出在哪里，如果是普通模板，直接就找到源码中的目标位置了，如果使用了此类模板，好吧，先用脑子翻译成这种模板，然后再去找目标位置，如果页面复杂，这种转换需要的时间也就更长。

总之做团队的项目，大家语言模板上一定要统一，这种自己喜欢玩 Geek 的人，会给后面别人维护时带来无穷无尽的麻烦。**简化开发是好事，但是过犹不及。**

当然，如果是自己的个人项目，哪怕用 CHTML 和 grass-mud-horse 语言都没有关系的。

2013年03月25日回答　8 评论　¥ 赞赏　编辑

1　确实是这样的，但它却存在了，所以我才想知道，那些支持它的人有什么正面的理由吗？它的优势在哪里？难道只是为了简化开发？那zencoding应该足够了。

— **Tychio** · 2013年03月25日

展开评论

2

哈哈，我不知道 sass 出来的时候有没有这样的争论，有没有很多前端报这种心态。

面工具支持。
jade 有 foreach , 有 mixin, 有 include,
如果你用一些框架开发，或者html组件化比较标准的话，你会发现 mixin 不是 zencoding 或者 emmet 能做的。如下面代码放在另一个文件

```
mixin list_a(title, tail)
  li.list-item
    a.list-content
      .title=title
      if tail
        .list-after=tail=='哈哈' ? '啦啦': tail
```

然后在其他用的地方引用：

```
+list_a('标题', '标题尾巴')
+list_a('你好', '哈哈')
```

各位知道会生成什么嘛？会减少你的输入吗？
当你使用 Framework7 之类框架的时候，就知道会减少多少手写代码量了，其他框架请自行脑补。

由此得到的，你所面对的代码会变得很少且清晰。
至于调试，你完全可以继续让服务器使用生成的 html ，或者混合其他模板引擎代码。

jafe 提供了2种方式原样输出，一个是行首竖线 如

```
|<h1>xxx...
```

另一个是行尾点号，如

```
.abc.
  <h1>...
```

至于用作后端语言的模板引擎，不知道预编译和缓存原生模板吗？

至于说看代码时要先在脑袋里翻译成 html 才能...
完全是想当然了，你在处理 html 的时候需要在脑袋里想浏览器里的效果吗？你在拼音输入的时候需要先想拼音吗？

手机打的，有疑问，有兴趣再继续哈

1

首先书写方式简单了很多，写起来很爽，可以提高开发效率，熟悉后阅读起来更直观。模板引擎需要嵌入简单的程序代码，这种情况下用html标签非常的繁琐和不直观，用zencoding也没办法很好的解决。

另外模板引擎通常学习成本不高，并且能够很好的集成到各种web框架中,可以替换一些web框架内置的模板的，这样大部分时间都可以只使用一种模板引擎了。

虽然有将html转换为这种模板引擎的工具，但个人觉得这种html模板引擎不适合用于对浏览器兼容性要求严格项目当中（比如要兼容ie6的），也不适合对这种模板引擎接受度不高的团队。

2013年04月01日回答 1 评论 ¥ 赞赏 编辑

1

jade在用,我来说下我的理解吧.楼上都是从前端考虑的,但是其实某些公司里面有分成重构和前台的.重构专门写html和css.

> 我的角度看,他的定位是增强html的复用能力,适合用于大量的html文件的组织管理
>
> [这里的html并不是针对程序页面模板的,仅仅是静态的原型的重构页面而已]
>
> jade mixin include 都在这里提供了很好的复用html的方式

jade其实是可复用的文件的简单代码逻辑的 **zencoding++**

2013年10月10日更新 评论 ¥ 赞赏 编辑

1

不要使用jade,虽然它被express默认支持,但终究会被淘汰出局!
原因有:
1.前台美工强烈反对
2.无法调试,出错后的调试异常低效
3.运维效率低下,这是大型并发网站决不能忍受的.服务器要承担的工作太多
4.学习成本高
5.不要总以程序员的眼光看问题,要从美工及运维方面综合考虑.开发者不要单干

2015年08月23日更新 评论 ¥ 赞赏 编辑

---

1 个回答被忽略

撰写答案

撰写答案...