



Vue 2017 现状与展望

尤雨溪

@FDCon 2017.04.08

“Just a view layer library”

“Just a view layer library”

The Progressive Framework

渐进式框架

Declarative
Rendering

Component
System

Client-Side
Routing

Large Scale
State
Management

Build
System

Client-Server
Data
Persistence

Ember has taken cue from Vue...

A P P C O M P L E X I T Y



glimmer



```
npm install @ember/router  
npm install @ember/service  
npm install @ember/data
```



ember

Today's Stats

- 接近 50,000 GitHub stars (目前所有项目第 12)
- NPM 每月接近 50 万次下载
- Chrome 开发者插件 14.6 万周活跃用户

国内用户



不完全统计，排名不分先后

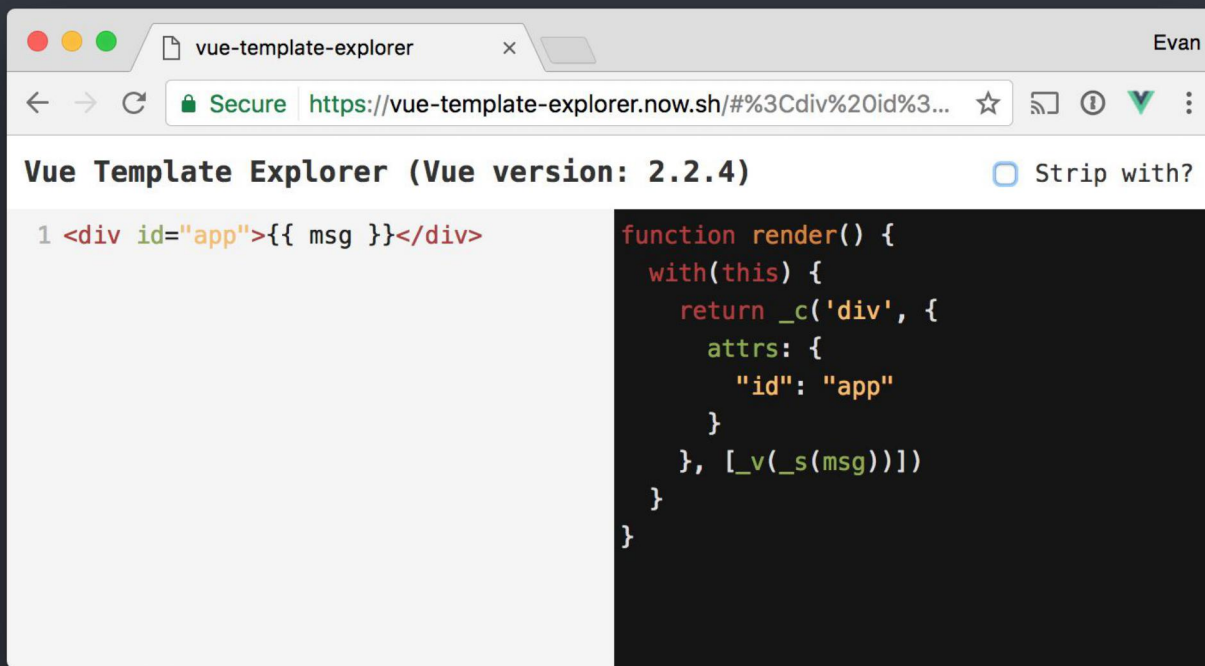
2016.9.30 - Vue 2.0 Ghost in the Shell

2.0

- 从头重写，但保留了绝大部分 API
- 引入 Flow 类型检查，提高代码健壮性
- 基于 Virtual DOM 的渲染机制：更轻，更快；
- 跨平台渲染：
服务端渲染（SSR）+ 移动端原生渲染（Weex）
- 官方 TypeScript 类型声明

模板预编译

Template --[parser]--> AST --[codegen]--> Render Function



The screenshot shows a web browser window titled 'vue-template-explorer' with the URL 'https://vue-template-explorer.now.sh/#%3Cdiv%20id%3...'. The page displays the 'Vue Template Explorer (Vue version: 2.2.4)' interface. On the left, the input template is '

```
function render() {  
  with(this) {  
    return _c('div', {  
      attrs: {  
        "id": "app"  
      }  
    }, [_v(_s(msg))])  
  }  
}
```

vue-template-explorer.now.sh

模板预编译

- 两种编译模式：运行时 or 构建时
- 两段式编译：运行时用基于 with 的简易编译器，构建时额外一次编译去掉 with
- 编译器可剥离：构建时模板编译不需要在运行时包含编译器，可以减少将近 8kb gzip

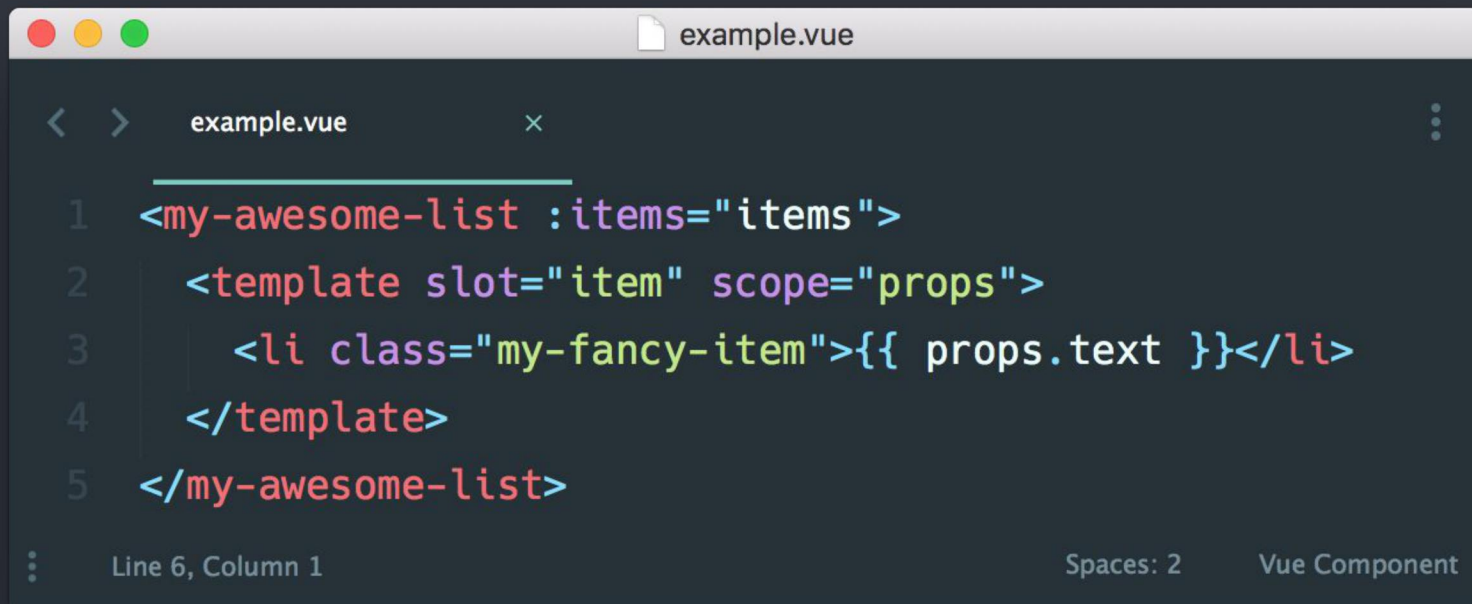
服务端渲染

- 流式渲染：充分利用 Node 和 HTTP 对 streaming 的支持；
- 组件级缓存：大幅度提升抗压能力
- bundleRenderer：解决同构代码在服务端的构建和部署

2016.11.22 - Vue 2.1 Hunter X Hunter

Scoped Slots

- 传递一个可复用的模板片段给子组件



```
example.vue

1 <my-awesome-list :items="items">
2   <template slot="item" scope="props">
3     <li class="my-fancy-item">{{ props.text }}</li>
4   </template>
5 </my-awesome-list>
```

Line 6, Column 1 Spaces: 2 Vue Component

Scoped Slots 有不少有意思的用法...

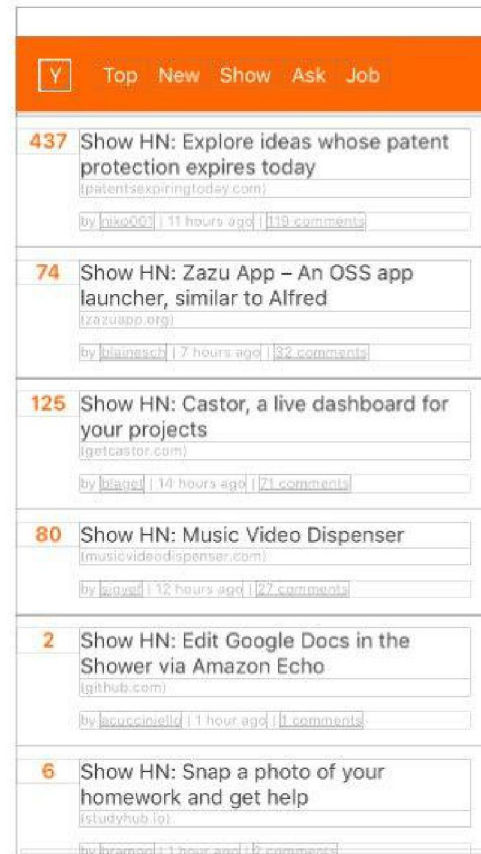
```
example.vue
<
>
example.vue
1 <fetch :url="API_URL">
2   <template scope="{ status, data, error }">
3     <div v-if="status === 'pending'">
4       Loading...
5     </div>
6     <div v-else-if="status === 'success'">
7       {{ data }}
8     </div>
9     <div v-else-if="status === 'error'">
10      Error!
11    </div>
12  </template>
13 </fetch>
14
: Line 7, Column 17; Saved ~/Desktop/example.vue (UTF-8) Spaces: 2 Vue Component
```

2016.12.16 - [Vuex 2.1](#)

Vuex Namespaced Modules

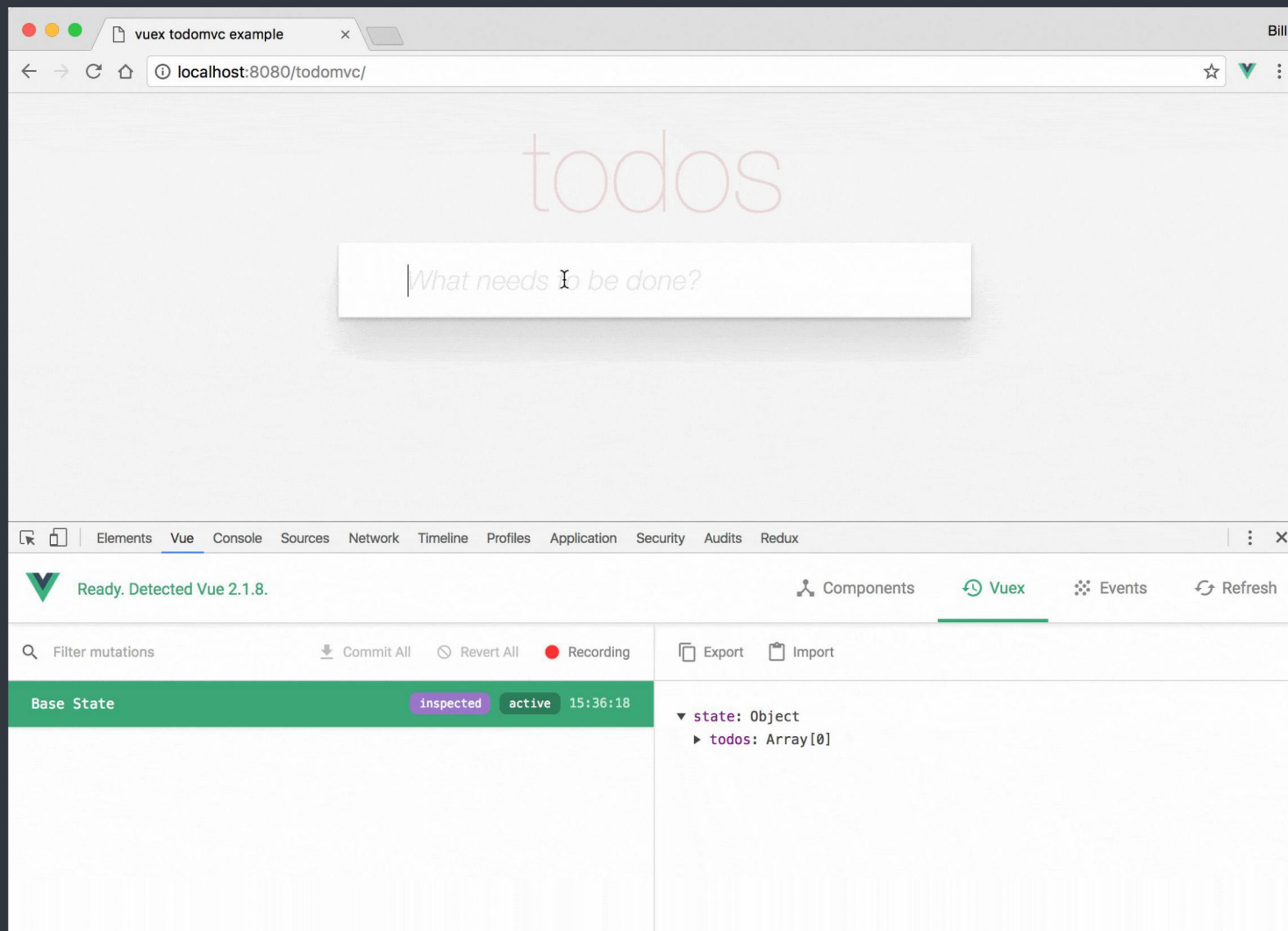
- 带命名空间的模块可以包含自己的 state, actions, mutations 和 getters, 就像一个小 store。
- 通过它们可以轻松地对业务逻辑进行分块封装, 最后只需要在 root store 入口处组合。
- 甚至可以通过动态模块注册来配合代码分割/懒加载

2016.12.22 - Weex 0.9.4
第一个正式包含 Vue 2.0 的 Weex 版本



2017.1.5 - vue-devtools 3.0

大量 UI 细节改进



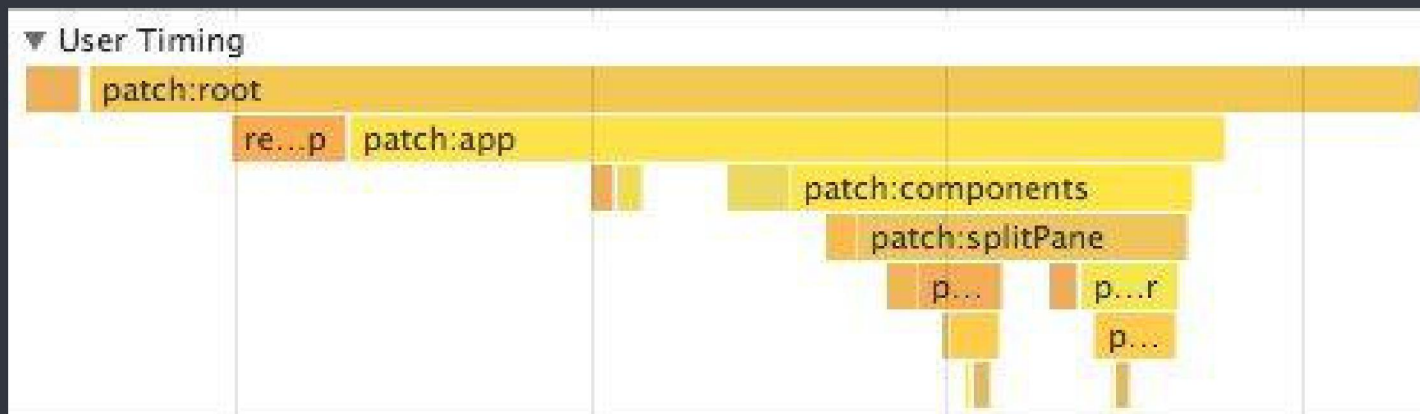
2016.2.26 - Vue 2.2 Initial D

2.2 服务端渲染改进

- 配合 Webpack 完美支持路由级别的代码懒加载（对业务代码无侵入性）
- vue-style-loader 自动抽取 Critical CSS
- bundleRenderer 支持 Source Map

内置 Chrome Timeline 性能追踪支持

- `Vue.config.performance = true`



更完善的异常处理

- 生命周期钩子中抛出的错误不再 crash 整个应用
- `renderError`: 当渲染出错时提供 fallback 组件

Coming soon - Vue 2.3 Jojo's Bizarre Adventure

2.3 服务端渲染改进

- 在存在代码分割的情况下，通过分析 Webpack 服务端和客户端的构建信息，自动推导需要在客户端预加载的文件，生成最优的 `<script>` 和 `<link rel="preload/prefetch">` 链接
- 自动选用不包含 CSS 字符串的异步 JS 文件，避免 Critical CSS 被两次下载

```
<!DOCTYPE html>
<html lang="en">
▼<head>
  <meta charset="utf-8">
  <title>Vue HN 2.0</title>
  <meta name="mobile-web-app-capable" content="yes">
  <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1,
  user-scalable=no, minimal-ui">
  <link rel="shortcut icon" sizes="48x48" href="/public/logo-48.png">
  <meta name="theme-color" content="#f60">
  <link rel="manifest" href="/manifest.json">
  <link rel="preload" href="/dist/manifest.975b5241c7947b0d6429.js" as="script">
  <link rel="preload" href="/dist/vendor.f59d103163ad8310ac6e.js" as="script">
  <link rel="preload" href="/dist/main.6c0e09f20cb9e786e920.js" as="script">
  <link rel="preload" href="/dist/0.106bab27eb10f783d1fd.no-css.js" as="script">
  <link rel="prefetch" href="/dist/1.fe8c3480c791cdb2470a.js" as="script">
  <link rel="prefetch" href="/dist/2.699660539050c557a33d.js" as="script">
  ▶<style data-vue-ssr-id="98ada9f8:0 91ab4ea4:0 95896c92:0 5ecccc62:0">...</style>
</head>
▼<body>
  ▶<div id="app">...</div>
  ▶<script>...</script>
  <script src="/dist/manifest.975b5241c7947b0d6429.js"></script>
... <script src="/dist/0.106bab27eb10f783d1fd.no-css.js"></script> == $0
  <script src="/dist/vendor.f59d103163ad8310ac6e.js"></script>
  <script src="/dist/main.6c0e09f20cb9e786e920.js"></script>
</body>
</html>
```

2.3 异步组件改进

- Loading / Error / Timeout fallback



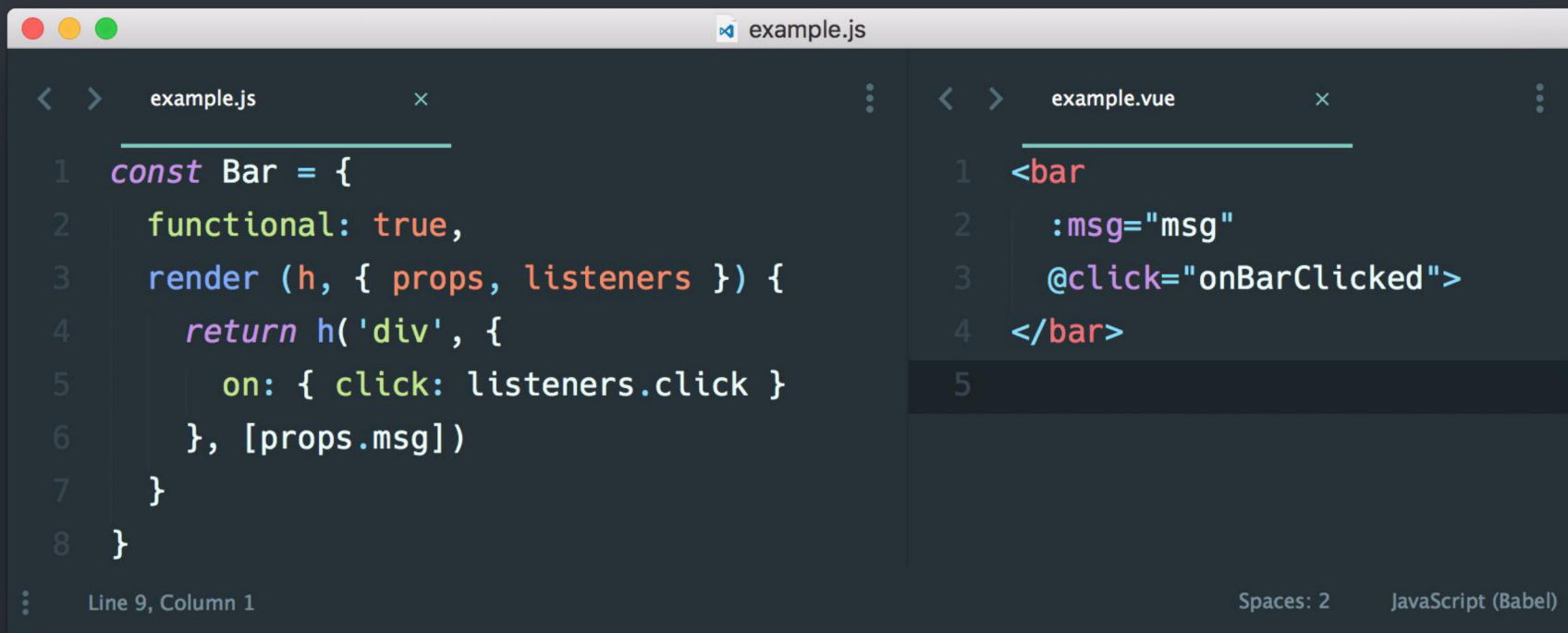
```
example.js

1  const Foo = () => ({
2    component: import('./Foo.vue'),
3    loading: LoadingComponent,
4    error: ErrorComponent,
5    delay: 200,
6    timeout: 3000
7  })

Line 8, Column 1      Spaces: 2      JavaScript (Babel)
```

2.3 函数式组件改进

- 不再需要显式声明 props
- 父组件添加的 v-on 会以 ctx.listeners 的形式提供



The screenshot displays a side-by-side comparison of a functional component's implementation in JavaScript and its equivalent in Vue.js. The left pane, titled 'example.js', shows a `const Bar` function component. It uses `render` to create a `h` (createElement) call with a `div` tag. The props are passed as an object containing `props` and `listeners`. The `on` object maps the `click` event to `listeners.click`. The right pane, titled 'example.vue', shows the corresponding Vue.js template. It uses the `<bar>` tag with a `:msg='msg'` prop and an `@click='onBarClicked'` event listener. The status bar at the bottom indicates 'Line 9, Column 1', 'Spaces: 2', and 'JavaScript (Babel)'.

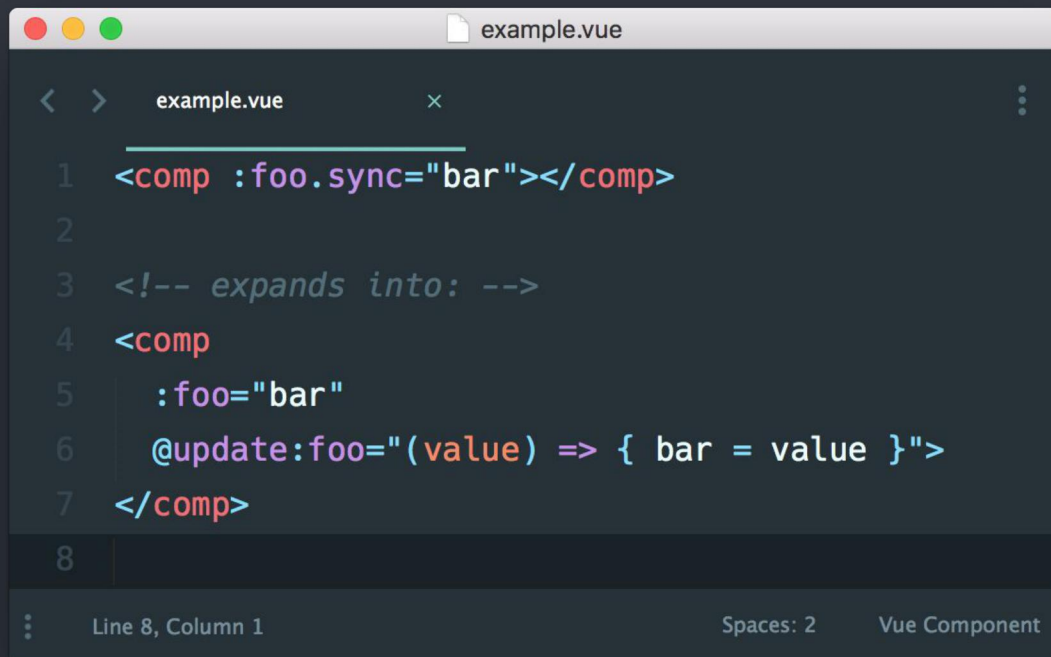
```
example.js
1  const Bar = {
2    functional: true,
3    render (h, { props, listeners }) {
4      return h('div', {
5        on: { click: listeners.click }
6      }, [props.msg])
7    }
8  }

example.vue
1  <bar
2    :msg="msg"
3    @click="onBarClicked">
4  </bar>
```

Line 9, Column 1 Spaces: 2 JavaScript (Babel)

2.3 其他改进

- passive 事件侦听: @touchmove.passive
- .sync 将会以 prop + listener 语法糖的形式回归



```
example.vue
< > example.vue x
1 <comp :foo.sync="bar"></comp>
2
3 <!-- expands into: -->
4 <comp
5   :foo="bar"
6   @update:foo="(value) => { bar = value }">
7 </comp>
8
Line 8, Column 1 Spaces: 2 Vue Component
```

2.3 之后近期计划

- 完善测试最佳实践
- 重构官方 CLI 模板：配置简化 + 依赖化
 - 服务端渲染
 - PWA
 - WEEX
 - TypeScript

社区动向：基于 2.0 的组件库逐渐成熟

- [Element UI](#)
- [iView](#)
- [Muse-UI](#)
- [Vue Material](#)
- [Vuetify](#)
- [Quasar Framework](#)
- [Keen-UI](#)

社区动向：更好的 IDE 支持

- WebStorm
- VSCode(Vetur)

2017 展望

- 更好的 TypeScript 整合：[TypeScript 团队为 Vue 量身打造更好的类型推导](#)
- 单文件组件 CSS 改进：>>> 和 ::slotted 选择器，CSS variable theming
- SSR 性能进一步优化
- Vuex 与 Rx 的整合

2017 展望

- 基于 Proxy 的响应式系统重构
 - 不再需要 `Vue.set`
 - Lazy observation
 - 显式构建响应式对象
- HTML Modules:
类似单文件组件的新标准（由 Google 起草中）