

# solutions

陈磊

2018 年 1 月 31 日

## 目录

<b>1</b>	<b>兼容性问题解决</b>	<b>1</b>
1.1	IE: 盒模型 . . . . .	1
1.2	IE 8: map . . . . .	1
1.3	IE 8: fontawesome 图标显示为方块 . . . . .	2
1.4	IE 10+ 浏览器定位 . . . . .	2
1.5	IE 6-8 CSS3 媒体查询 (Media Query) . . . . .	3
<b>2</b>	<b>HTML</b>	<b>3</b>
2.1	参数 (input) 在 form 之外 . . . . .	3
<b>3</b>	<b>模块化与组件化</b>	<b>3</b>
3.1	实现模块化 . . . . .	4
3.2	实现组件化 . . . . .	4
<b>4</b>	<b>Vue.js 权限管理</b>	<b>4</b>
<b>5</b>	<b>系统方案</b>	<b>4</b>
5.1	分角色登录 . . . . .	4

## 1 兼容性问题解决

### 1.1 IE: 盒模型

IE 默认情况下长宽包含 padding 和 border, 和其他浏览器的长宽存在区别, 建议添加 border-sizing 属性. 保持多个浏览器的一致性.

```
1 box-sizing: border-box;
```

## 1.2 IE 8: map

IE8 不支持 JavaScript 原生 map 函数, 可在任意地方加入如下的代码片段<sup>1</sup>.

```
1 (function (fn) {  
2     if (!fn.map) fn.map = function (f) {  
3         var r = [];  
4         for (var i = 0; i < this.length; i++)  
5             if (this[i] !== undefined) r[i] = f(this[i]);  
6         return r  
7     }  
8     if (!fn.filter) fn.filter = function (f) {  
9         var r = [];  
10        for (var i = 0; i < this.length; i++)  
11            if (this[i] !== undefined && f(this[i])) r[i] = this[i];  
12        return r  
13    }  
14 })(Array.prototype);
```

或者用 jQuery 的 map 函数.

```
1 // array.map(function( ) { });  
2 jQuery.map(array, function( ) {  
3 }
```

## 1.3 IE 8: fontawesome 图标显示为方块

对于修饰性不影响功能的图标, 可以做降级处理, 仅在非 IE 或者 IE9+ (条件注释<sup>2</sup>) 情况下引入 fontawesome 图标库. (谷歌搜索了一堆方案都没用, 最后应用这种方式来解决).

```
1 <!--[if (gt IE 8) / !IE]><!-->  
2 <link rel="stylesheet" href="font-awesome.min.css">  
3 <!--<![endif]-->
```

---

<sup>1</sup>Is the javascript .map() function supported in IE8?

<sup>2</sup>About conditional comments

## 1.4 IE 10+ 浏览器定位

IE 10+ 不支持条件注释, 因此需要其他方式定位这些浏览器. 如果只增加 CSS, 可采用以下方式定位<sup>3</sup>.

```
1 /*IE 9+, 以及 Chrome*/
2 @media screen and (min-width:0\0) {
3 }
4
5 /*IE 10*/
6 @media all and (-ms-high-contrast: none), (-ms-high-contrast: active) {
7 }
8
9 /*Edge*/
10 @supports (-ms-accelerator:true) {
11 }
```

另一种方式是 JavaScript 检测浏览器版本, 在 `body` 标签为特定浏览器添加 `class` 属性标识.

## 1.5 IE 6-8 CSS3 媒体查询 (Media Query)

引入 Respond.js.

```
1 <!--[if lt IE 9]>
2 <script src="respond.min.js"></script>
3 <![endif]-->
```

# 2 HTML

## 2.1 参数 (input) 在 form 之外

`input` 在 `form` 之外时, 在 `input` 元素内添加 `form` 属性值为 `form` 的 ID<sup>4</sup>. 这样 `input` 仍然可以看做隶属于此表单, `jQuery $('#formid').serialize();` 能够获取 `form` 之外的输入框值. 或者在提交 (`submit`) 表单时会同样提交 `outside` 这个值.

---

<sup>3</sup>How do I target only Internet Explorer 10 for certain situations like Internet Explorer-specific CSS or Internet Explorer-specific JavaScript code?

<sup>4</sup>PLACING FORM FIELDS OUTSIDE THE FORM TAG

```
1 <form id="formid" method="get">
2
3   <label>Name:</label>
4   <input type="text" id="name" name="name">
5
6   <label>Email:</label>
7   <input type="email" id="email" name="email">
8   <input type="submit" form="contact_form" value="send form" />
9 </form>
10 <input type="text" name="outside" form="formid">
```

注意: IE8 `$('#formid').serialize();` 无法获取 `outside` 值.

## 3 模块化与组件化

内嵌框架图 (embedded)

模块化, 强调内聚, 包含完整的业务逻辑, 可以方便业务的复用. 组件化, 强调复用, 重点在于接口的暴露, 和构件的概念类似.

### 3.1 实现模块化

业务相关的特殊性都应包含在同一个模块内, 具体到前端, 这些特性包括与业务相关的接口、状态、路由等.

### 3.2 实现组件化

关键是如何暴露接口, 方便外部复用.

## 4 Vue.js 权限管理

- 基于 Vue 实现后台系统权限控制
- 用 `addRoutes` 实现动态路由
- 手把手, 带你用 vue 撸后台系列二 (登录权限篇)
- Vue 后台管理控制用户权限的解决方案?
- 自定义指令
- <https://codepen.io/diemah77/pen/GZGxPK>

## 5 系统方案

### 5.1 分角色登录

不同角色/权限登录后所见菜单不一样. 方案如下:

1. 给不同的路由配置其角色/权限属性.
2. 登录进入时, 记录角色/权限.
3. 进入主页, 根据角色/权限构建菜单 (view 中包含全部菜单, 非此角色菜单移除 Dom).
4. 点击菜单进入到对应路由时, 根据判断路由的角色/权限属性是否和登录进入时记录的一样.

此方案包括两部分的权限限制, 其一是将不必要的菜单移除 Dom, 但菜单对应的路由依然可用, 只是在页面上没有对应可操作的视图, 其二是路由和登录的角色/权限匹配. 以 AngularJS 为例, 对应每个步骤的代码如下.

1. 路由属性.

```
1 $stateProvider.state('Registration.Instructors', {
2   url: "/Instructors",
3   templateUrl: '/Scripts/App/Instructors/Templates/instructors.html',
4   controller: 'InstructorController',
5   data: { auth: "Admin" }
6 })
```

2. 登录用户权限/角色信息可记录到 `rootScope` 中, 比如 `rootScope.adminType`  
`↪ = "Admin".`
3. 菜单保留与移除. `ng-if="adminType==='Admin'"`<sup>5</sup>.
4. 路由和登录角色/权限匹配<sup>6</sup>.

```
1 app.run(function($rootScope){
2   $rootScope.$on('$stateChangeStart', function(event, toState, toParams,
3     ↪ fromState, fromParams){
4     if ( toState.data.auth !== $rootScope.adminType ) {
5       event.preventDefault();
6       return false;
7     }
8   });
```

<sup>5</sup> what is the difference between ng-if and ng-show/ng-hide

<sup>6</sup> angularjs: conditional routing in app.config