

准备工作

2015年7月20日 19:02

使用GitHub进行版本控制:

<https://github.com/yjflike/hyoito.git>

<http://git.oschina.net/hyoito/hyoito>

C:\Users\Administrator\.ssh\id_rsa

sublime text3

下载sublime text3

<http://www.sublimetext.com/3>

安装后打开, ctr+` 号 打开控制台并输入下面代码 安装 package controler

```
import urllib.request,os; pf = 'Package Control.sublime-package'; ipp = sublime.installed_packages_path();
urllib.request.install_opener( urllib.request.build_opener( urllib.request.ProxyHandler()) ); open(os.path.join(ipp, pf),
'wb').write(urllib.request.urlopen( 'http://sublime.wbond.net/' + pf.replace(' ','%20')).read())
```

等待安装后 重启st

ctrl+shift+P 找到package install

等待后可以搜索插件

angular , emmet , view in browser,HTML-CSS-JS Prettify(全选后ctrl+shift+H快速格式化)

Sublime CodeIntel(自动补全代码)

view in browser配置(用默认的浏览器打开):

preference - 按键绑定-用户(key binding-user):

```
{ "keys": ["ctrl+shift+enter"], "command": "open_in_browser" }
```

From <<http://www.zhugexiaoju.com/note/view-336.html>>

下载git和TortoiseGit

<http://git-scm.com/download/>

<http://download.tortoisegit.org/>

安装

安装nodejs

<https://nodejs.org/>

使用cnpm替代npm

```
npm install -g cnpm --registry=https://registry.npm.taobao.org
```

来自 <<http://npm.taobao.org/>>

cmd:

```
npm install grunt
```

angular 包替换官方的

<http://apps.bdimg.com/libs/angular.js/1.2.16/angular.min.js>

note

使用

<http://apps.bdimg.com/libs/angular.js/1.2.16/angular.min.js>

"<http://www.hubwiz.com/scripts/angular.min.js>

来自 <<http://www.hubwiz.com/class/547c3e3b88dba0087c55b4e5>>

OR 百度开源静态链接

(<http://cdn.code.baidu.com/>)

<http://apps.bdimg.com/libs/angular.js/1.2.16/angular.min.js>

代替原有的angular链接

不要复用 controller,一个控制器一般只负责一小块视图

不要在controller中操DOM,这不是控制器的职责(速度低,导致浏览器的重新布局)

不要在controller 中做数据过滤

一般来说,controller不会互相调用,要通过事件进行交互

angularjs的mvc都是借助于\$scope(作用域)来实现的!

\$scope是一个pojo(plain old javascript object)

\$scope提供一些工具方法\$watch()\$apply

\$scope是一个树形结构,与dom标签平行

\$scope可以传播事件,类似DOM事件可以向上也可以向下

每个angular应该有且仅有一个根\$scope(一般位于ng-app上)

可以用 angular.element(\$0).scope()进行调试

CSS

Thursday, July 23, 2015 17:16



CSS

@keyframes 动画式样

```
1 }
2 @keyframes slideOutLeft {
3     to {
4         transform: translateX(-100%);
5     }
6 }
7 @keyframes rotateOutNewspaper {
8     to {
9         transform: translateZ(-3000px) rotateZ(360deg);
10        opacity: 0;
11    }
12 }
13 @keyframes scaleUp {
14     from {
15         opacity: 0.3;
16         -webkit-transform: scale(0.8);
17     }
18 }
19 @keyframes slideInRight {
20     from {
21         transform: translateX(100%);
22     }
23 }
```

angular

Friday, July 24, 2015

12:52

四大核心特性

MVC

模块化和依赖注入

双向绑定

指定

控制器+model

2015年7月21日 23:13

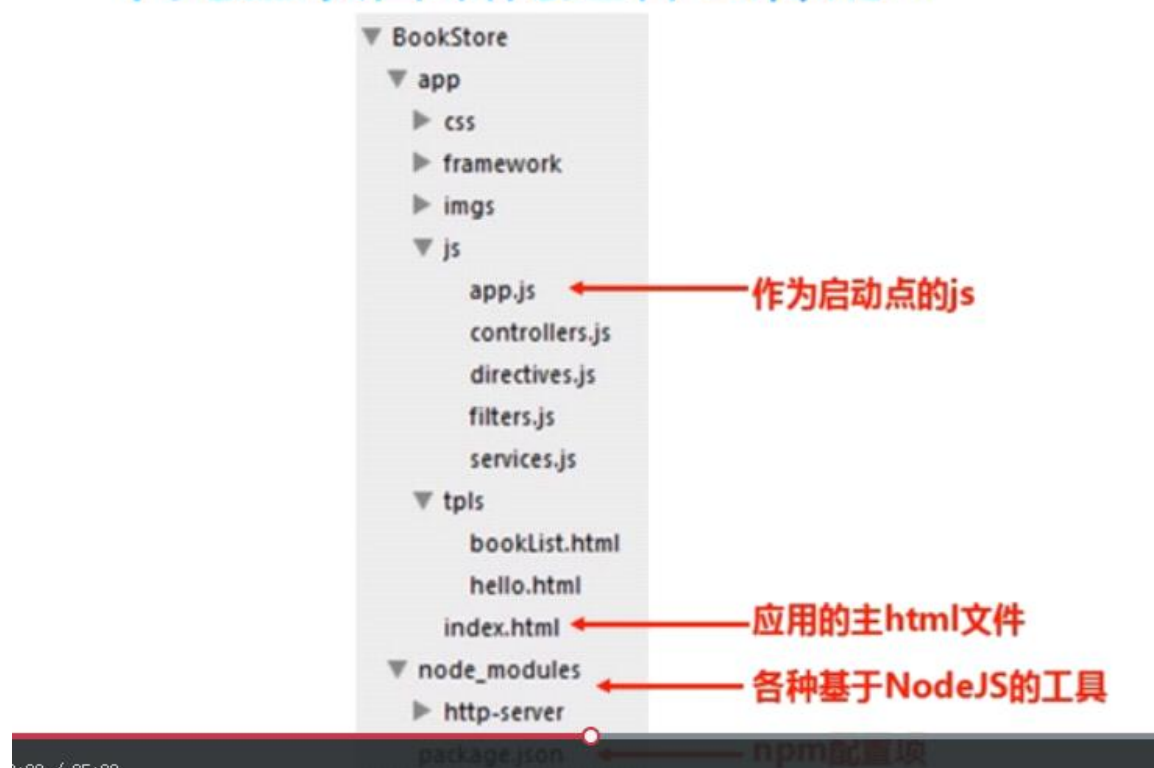
一个项目有且仅有一个ng-app作为主入口，入口指向一个js文件。

```
<html ng-app="HelloAngular">
```

入口文件中的控制器可以直接调用

项目结构

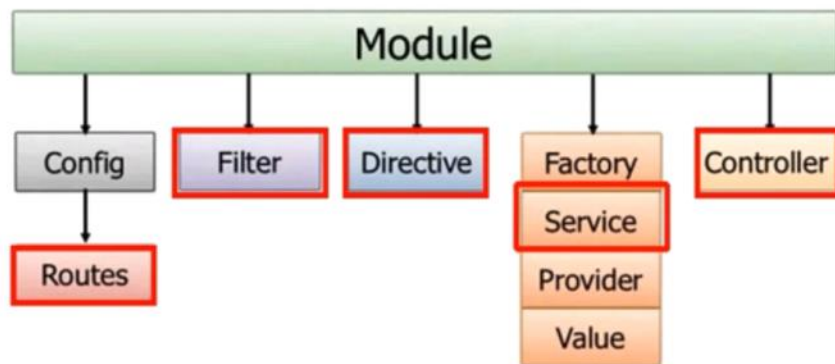
一个完整项目结构是什么样的？



一切都是从模块开始的

Modules are Containers

```
<html ng-app="moduleName">
```



慕课网

模块之间的依赖 是靠依赖注入实现的

双向绑定

Wednesday, July 22, 2015 12:55

index使用 ng-bind替代`{{}}`的方式进行数据绑定可以避免网络慢的时候显示代码



hyoito

bootstrap

Wednesday, July 22, 2015 13:39

```
<link href="http://libs.baidu.com/bootstrap/3.0.3/css/bootstrap.min.css" rel="stylesheet">
<script src="http://libs.baidu.com/bootstrap/3.0.3/js/bootstrap.min.js"></script>
```

bootstrap 横向布局共分为12格

```
<form class="form-horizontal" role="form">
  <div class="form-group">
    <label class="col-md-2 control-label">    指定 占用布局2格
      Email:
    </label>
    <div class="col-md-10">    placeholder用于提示
      <input type="email" class="form-control" placeholder="input your email address">
    </div>
  </div>
</form>
```

```
<div class="col-md-offset-2 col-md-10">    布局偏移
```

sublime text 的bootstrap snippet插件

bst-*使用

bs3-*使用

★ div中嵌套的内层div布局col-md-*是把外层div再次分12格后的布局

路由

Friday, July 24, 2015 11:36

Ajax请求不会留下历史记录(用户不能通过加书签,分享链接重新访问页面)

Ajax对SEO(搜索引擎优化)是个灾难:没办法通过SEO进行索引→对搜索引擎不友好

前端路由的基本原理

哈希 #

html5中新的history API

路由的核心是给应用定义"状态"

考虑兼容性问题与"优雅降级"

指令

Friday, July 24, 2015

12:53

指令

- 解析最简单的指令hello : 匹配模式restrict
- 解析最简单的指令hello : template、templateUrl、\$templateCache
- 解析最简单的指令hello : replace与transclude
- compile与link (操作元素、添加CSS样式、绑定事件)
- 指令与控制器之间的交互
- 指令间的交互
- scope的类型与独立scope
- scope的绑定策略

- AngularJS内置的指令
 - 实例解析Expander
 - 实例解析Accordion
 - 指令的运行原理 : compile和link
 - 总结 : ERP类型的系统必备的UI组件
 - 总结 : 互联网/电商型系统必备的UI组件
 - 第三方指令库angular-ui
 - Directive思想的起源和原理概述

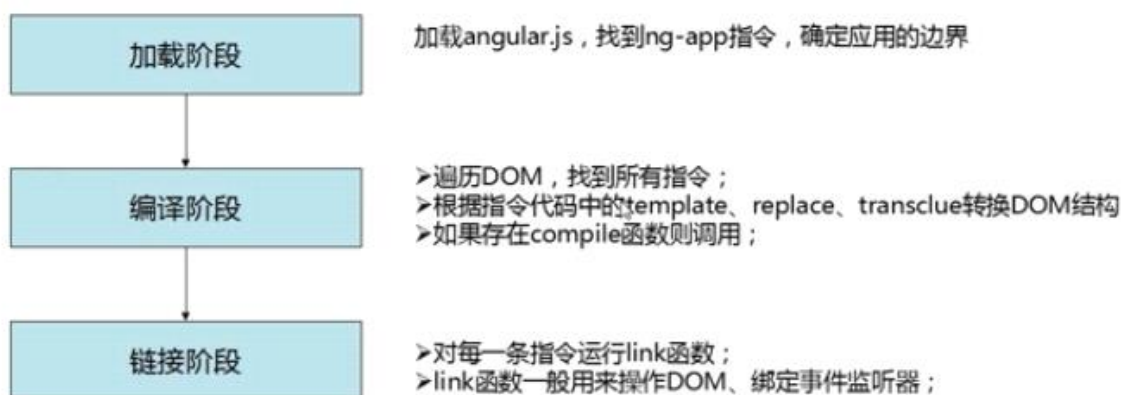
指令1

Friday, July 24, 2015 12:57

四中方式表示指令,最后一种使用注释的方式表示指令时,两边要有空格!

```
<hello></hello>  
<div hello></div>  
<div class="hello"></div>  
<!-- directive:hello -->  
<div></div>
```

compile与link



- compile 函数用来对模板自身进行转换, 而link 函数负责在模型和视图之间进行动态关联;
- 作用域在链接阶段才会被绑定到编译之后的link 函数上;
- compile 函数仅仅在编译阶段运行一次, 而对于指令的每个实例, link 函数都会执行一次;
- compile可以返回preLink和postLink函数, 而link函数只会返回postLink函数;
- 如果需要修改DOM 结构, 应该在postLink 中来做这件事情, 而如果在preLink 中做这件事情会导致错误;
- 大多数时候我们只要编写link函数即可;

angular在指令(directive)中的link函数中操作DOM

指令和控制器/指令之间的交互

Friday, July 24, 2015 15:13

指令和控制器之间的交互

```
link: function($scope, iElm, iAttrs, controller) {
    iElm.bind("mouseenter", function(){ // 绑定鼠标滑入事件时执行的方法
        //$scope.loadData(); // 除了直接用scope调用之外, 还可以用apply()
        //$scope.$apply("loadData()")
        // iAttrs.howtoload() // 在页面调用的是大写, 但是在js中要用小写

        // 通过iAttrs.howtoload实现了指令的复用
        $scope.$apply(iAttrs.howtoload); // 这里不要写函数调用的形式, 而是用属性调用的形式, howtoload是属性!
    });
}
```

controller 中放的方法是需要给外部调用的方法;

link中放的是内部的方法处理指定内部的事件: 绑定.....

```
restrict: 'AE',
controller: function($scope) { // 与angular中的控制器不同, 是给指令暴露出一组public方法给外部调用
    $scope.abilities = [];
    this.addStrength = function() {
        $scope.abilities.push("strength");
    };
    this.addSpeed = function() {
        $scope.abilities.push("speed");
    };
    this.addLight = function() {
        $scope.abilities.push("light");
    };
},

link: function($scope, iElm, iAttrs, controller) {
    iElm.addClass('btn btn-primary');
    iElm.bind("mouseenter", function() { // 绑定一个mouseenter事件
        console.log($scope.abilities);
    });
}
```

```
require: '^superman', // 表示该指令依赖于superman指令。
```

独立scope以及scope的绑定策略

Friday, July 24, 2015 17:54

```
angular.module('MyModule', []);
myModule.directive('hello', function(){
  // Runs during compile
  return {
    restrict: 'AE',
    template: '<div><input type="text" ng-model="userName" />{{userName}}</div>',
    replace: true
  }
});
```

```
<hello></hello>
<hello></hello>
<hello></hello>
<hello></hello>
```

33333	33333
33333	33333
33333	33333
33333	33333

当没有创建独立的scope对象时,指令是共同的,会互相影响

添加完scope:{}配置项之后,即可创建独立的scope空间,消除相互之间的影响

```
// Runs during compile
return {
  scope:{},
  restrict: 'AE',
  template: '<div><input type="text" ng-model="userName" />{{userName}}</div>',
  replace: true
};
```

123	123
321	321
11	11
22	22

scope的绑定策略

@	把当前属性作为字符串传递。你还可以绑定来自外层scope 的值，在属性值中插入({})即可
=	与父scope中的属性进行双向绑定
&	传递一个来自父scope 的函数，稍后调用

sublime text3

Tuesday, July 21, 2015 17:53

快捷键

Tuesday, July 21, 2015 17:53

以下是个人总结不完全的快捷键总汇，祝愿各位顺利解放自己的鼠标。

选择类

Ctrl+D 选中光标所占的文本，继续操作则会选中下一个相同的文本。

Alt+F3 选中文本按下快捷键，即可一次性选择全部的相同文本进行同时编辑。举个栗子：快速选中并更改所有相同的变量名、函数名等。

Ctrl+L 选中整行，继续操作则继续选择下一行，效果和 Shift+↓ 效果一样。

Ctrl+Shift+L 先选中多行，再按下快捷键，会在每行行尾插入光标，即可同时编辑这些行。

Ctrl+Shift+M 选择括号内的内容（继续选择父括号）。举个栗子：快速选中删除函数中的代码，重写函数体代码或重写括号内里的内容。

Ctrl+M 光标移动至括号内结束或开始的位置。

Ctrl+Enter 在下一行插入新行。举个栗子：即使光标不在行尾，也能快速向下插入一行。

Ctrl+Shift+Enter 在上一行插入新行。举个栗子：即使光标不在行首，也能快速向上插入一行。

Ctrl+Shift+[选中代码，按下快捷键，折叠代码。

Ctrl+Shift+] 选中代码，按下快捷键，展开代码。

Ctrl+K+O 展开所有折叠代码。

Ctrl+← 向左单位性地移动光标，快速移动光标。

Ctrl+→ 向右单位性地移动光标，快速移动光标。

shift+↑ 向上选中多行。

shift+↓ 向下选中多行。

Shift+← 向左选中文本。

Shift+→ 向右选中文本。

Ctrl+Shift+← 向左单位性地选中文本。

Ctrl+Shift+→ 向右单位性地选中文本。

Ctrl+Shift+↑ 将光标所在行和上一行代码互换（将光标所在行插入到上一行之前）。

Ctrl+Shift+↓ 将光标所在行和下一行代码互换（将光标所在行插入到下一行之后）。

Ctrl+Alt+↑ 向上添加多行光标，可同时编辑多行。

Ctrl+Alt+↓ 向下添加多行光标，可同时编辑多行。

编辑类

Ctrl+J 合并选中的多行代码为一行。举个栗子：将多行格式的CSS属性合并为一行。

Ctrl+Shift+D 复制光标所在整行，插入到下一行。

Tab 向右缩进。

Shift+Tab 向左缩进。

Ctrl+K+K 从光标处开始删除代码至行尾。

Ctrl+Shift+K 删除整行。

Ctrl+/ 注释单行。

Ctrl+Shift+/ 注释多行。

Ctrl+K+U 转换大写。

Ctrl+K+L 转换小写。

Ctrl+Z 撤销。

Ctrl+Y 恢复撤销。

Ctrl+U 软撤销，感觉和 Ctrl+Z 一样。

Ctrl+F2 设置书签

Ctrl+T 左右字母互换。

F6 单词检测拼写

搜索类

Ctrl+F 打开底部搜索框，查找关键字。

Ctrl+shift+F 在文件夹内查找，与普通编辑器不同的地方是sublime允许添加多个文件夹进行查找，略高端，未研究。

Ctrl+P 打开搜索框。举个例子：1、输入当前项目中的文件名，快速搜索文件，2、输入@和关键字，查找文件中函数名，3、输入：和数字，跳转到文件中该行代码，4、输入#和关键字，查找变量名。

Ctrl+G 打开搜索框，自动带：，输入数字跳转到该行代码。举个例子：在页面代码比较长的文件中快速定位。

Ctrl+R 打开搜索框，自动带@，输入关键字，查找文件中的函数名。举个例子：在函数较多的页面快速查找某个函数。

Ctrl+： 打开搜索框，自动带#，输入关键字，查找文件中的变量名、属性名等。

Ctrl+Shift+P 打开命令框。场景栗子：打开命名框，输入关键字，调用sublime text或插件的功能，例如使用package安装插件。

Esc 退出光标多行选择，退出搜索框，命令框等。

显示类

Ctrl+Tab 按文件浏览过的顺序，切换当前窗口的标签页。

Ctrl+PageDown 向左切换当前窗口的标签页。

Ctrl+PageUp 向右切换当前窗口的标签页。

Alt+Shift+1 窗口分屏，恢复默认1屏（非小键盘的数字）

Alt+Shift+2 左右分屏-2列

Alt+Shift+3 左右分屏-3列

Alt+Shift+4 左右分屏-4列

Alt+Shift+5 等分4屏

Alt+Shift+8 垂直分屏-2屏

Alt+Shift+9 垂直分屏-3屏

Ctrl+K+B 开启/关闭侧边栏。

F11 全屏模式

Shift+F11 免打扰模式

From <<http://www.douban.com/note/362268947/>>