

北京泰合佳通信息技术有限公司

前端框架开发指南

WebFramework2.0

版本	时间	修改人	内容
0.1.0304	2016-3-4	林俊杰	初稿

目录

- 1 框架介绍.....3
- 2 框架组成.....3
 - 2.1 文件结构.....4
 - 2.2 主要目录定义.....4
 - 2.3 框架提供.....5
 - 2.4 辅助功能.....6
- 3 实例.....7
 - 3.1 引入组件.....7
 - 3.2 调用组件.....9

1 框架介绍

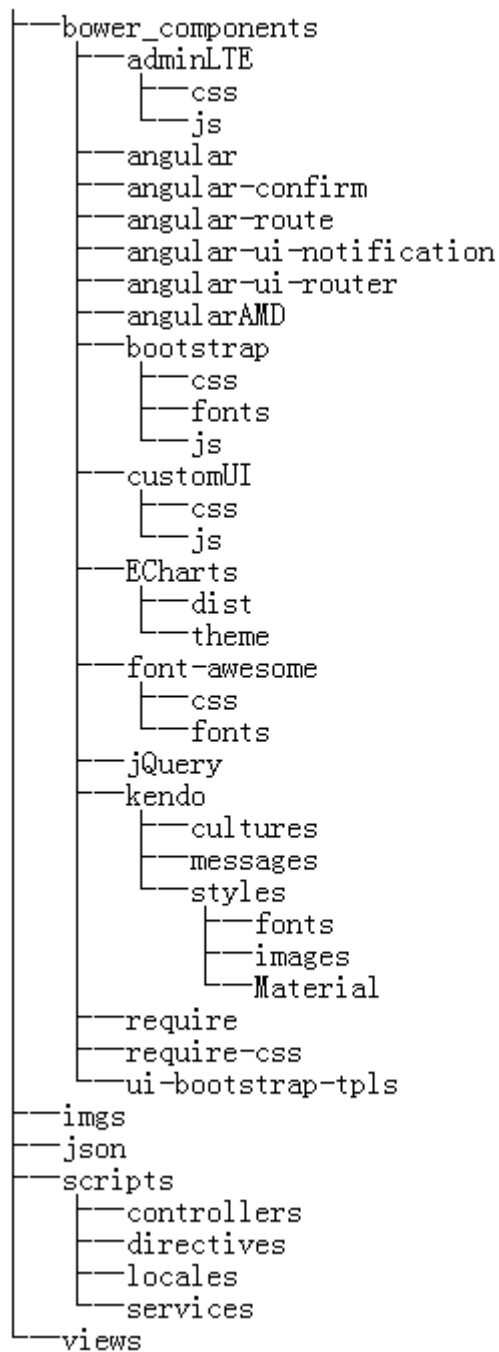
WebFramework2.0 是一个以 AngularJs 为基础进行二次扩展和封装的 MVVM 框架。框架中不仅保留了 AngularJs 中的所有特性，还在原来的基础上扩展了模块化加载、按需加载、多层路由扩展和丰富的第三方控件库，简单易用的特性和模块化的开发模式使得前端开发人员在大大降低学习和开发的成本的同时多人协作开发的时候也不互相干扰，可以更关注于功能业务的实现。

WebFramework2.0 在 1.0 版本的基础上实现了脚本模块和样式表的按需加载功能，重构后的框架更加精简和便于使用。新版前端框架有着诸多特性，最为核心的是：MVVM、模块化、按需加载、自动化双向数据绑定、语义化标签、依赖注入等等。

2 框架组成

框架的文件系统结构和主要组件的介绍

2.1 文件结构



2.2 主要目录定义

/bower_components:存放第三方模块和扩展插件

/imgs:存放图片目录

/json:存放序列化的 json 字符串文件,一般用于配置或者测试

/scripts/controllers:存放业务功能所需脚本控制器

/scripts/directive:存放自定义指令

/scripts/locales:存放汉化配置

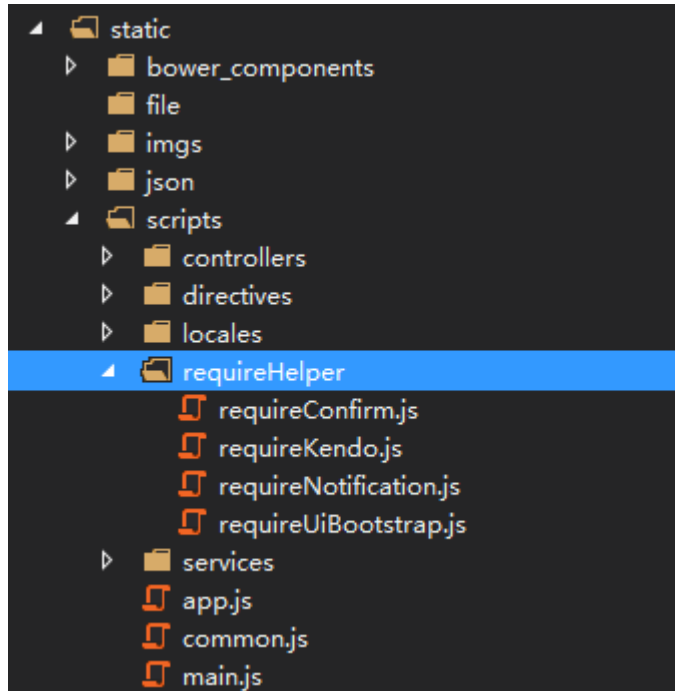
/scripts/services:存放通讯相关服务

/scripts/views:存放业务界面模板

2.3 框架提供

- RequireJs: 小巧的 JavaScript 模块载入框架, 是 AMD 规范最好的实现者之一。实现前端脚本的按需加载, 防止污染全局变量。
- Require-css: 实现样式表文件的按需加载
- Angular: 实现前端 MVVM、类库、数据双向绑定、提供丰富的内置指令
- AngularAMD: 结合 Requirejs 和 Angular, 实现包括控制器的按需加载和模块的动态注入
- Angular-route: 通过路由和 Html 模板的渲染实现 Web 无刷新视图切换的单页面应用
- Angular-ui-router: 扩展 Angular-route, 弥补 Angular-route 的无法实现多层路由嵌套功能不足
- jQuery: 优秀的 Javascript 库, 为系统第三方控件和样式提供支撑
- Bootstrap: 为前端响应式布局提供更便捷的实现方案
- Ui-bootstrap: 对 Bootstrap 组件进行封装, 以便在框架中使用指令的模式调用 Bootstrap 中提供的组件
- Angular-confirm: 在 Ui-bootstrap 的 Modal 组件上进行封装的弹出确认框
- Angular-ui-notification: 提供即时的消息通知提示框
- Kendo: 功能丰富强大的第三方控件库, 提供了适用于 Angular 框架中调用的指令, 包括常用的 Web 控件和图表控件
- ECharts: 开源的第三方图表控件库, 提供带交互式的图表控件

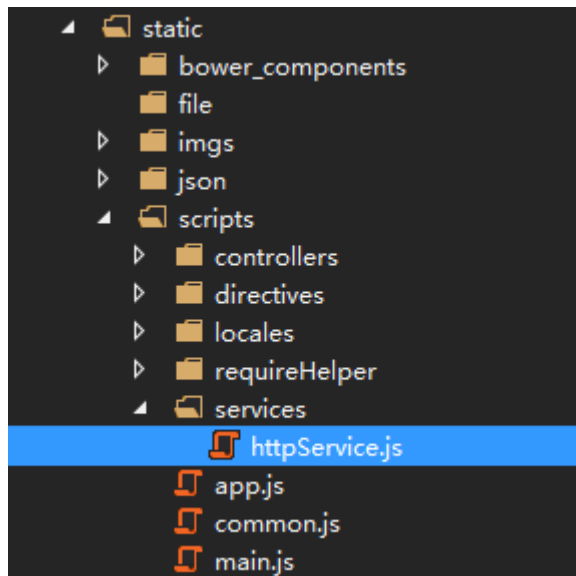
2.4 辅助功能



1.

requireHelper: 包括 requireConfirm 、 requireKendo 、 requireNotification 、 requireUiBootstrap, 分别抽象于 Angular-confirm、Kendo、Angular-ui-notification、Ui-bootstrap 所需要加载的文件。引用的时候只需调用对应的辅助脚本即可加载对应组件的样式表和脚本, 如需要用到 Kendo:

```
define([
    'scripts/requireHelper/requireKendo'
],
function () {
    return ['$scope',
    function ($scope) {
        //此处放置具体业务代码
    }]
});
```



2.

httpService: 封装了调用后台服务的方法，可以使用 Post 或者 Get 的形式调用后台服务，如 Post 形式：

```
$scope.httpServiceDemo = function () {  
    httpService.post('CompusNet', 'AreaService', 'GetAreas',  
        {regionType: 2}  
    ).then(function (data) {  
        //数据具体处理方法  
    },  
    function (errorMessage) {  
        Notification.error({  
            message: errorMessage,  
            delay: 5000  
        });  
    });  
};
```

“CompusNet”代表后台服务项目名称，根据具体项目而定

“AreaService”代表后台服务名称，根据具体项目而定

“GetArea”代表后台服务方法名称，根据具体项目而定

{regionType: 2}代表具体的参数对象，属性名称必须和后台服务参数名相对应

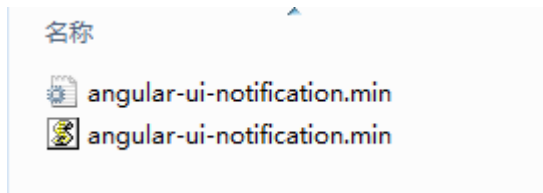
3 实例

实例中讲解如何在框架中引入一个适用于 Angular 的第三方组件，然后新建一个新的页面，并在页面中调用引入的组件功能

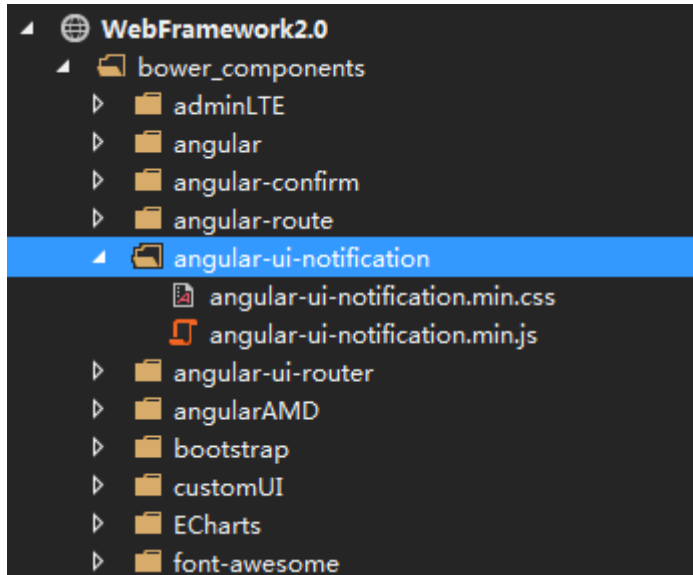
3.1 引入组件

以 Angular-ui-notification 为例，整个引入的过程分以下步骤：

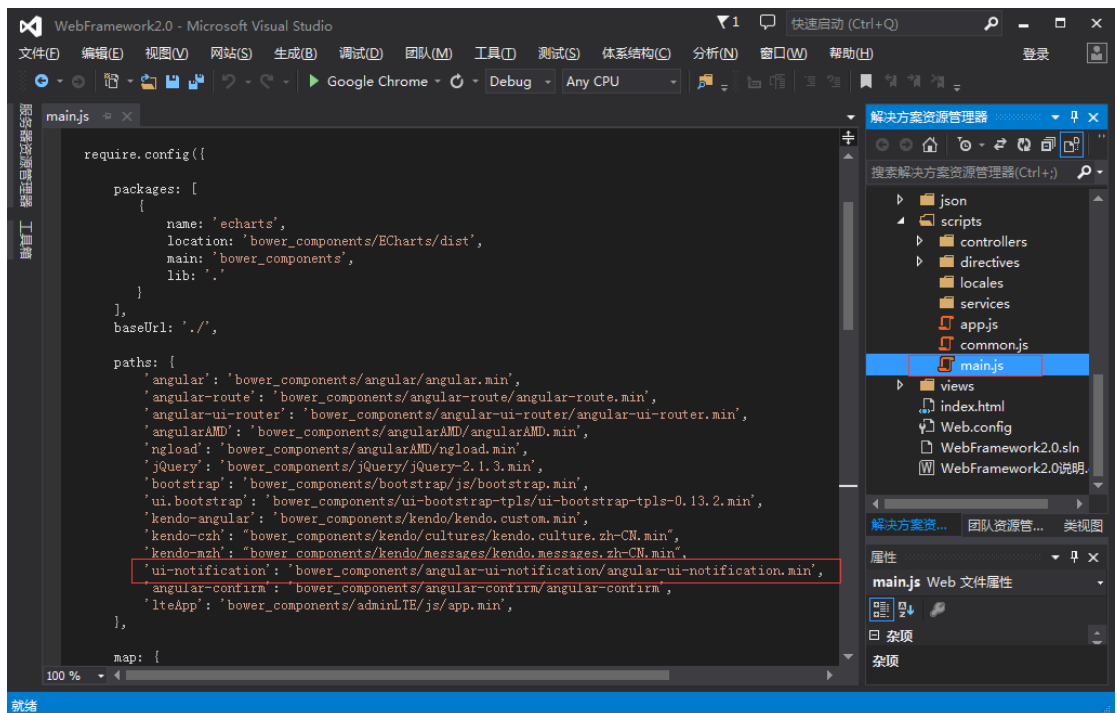
1. 下载组件文件所需文件

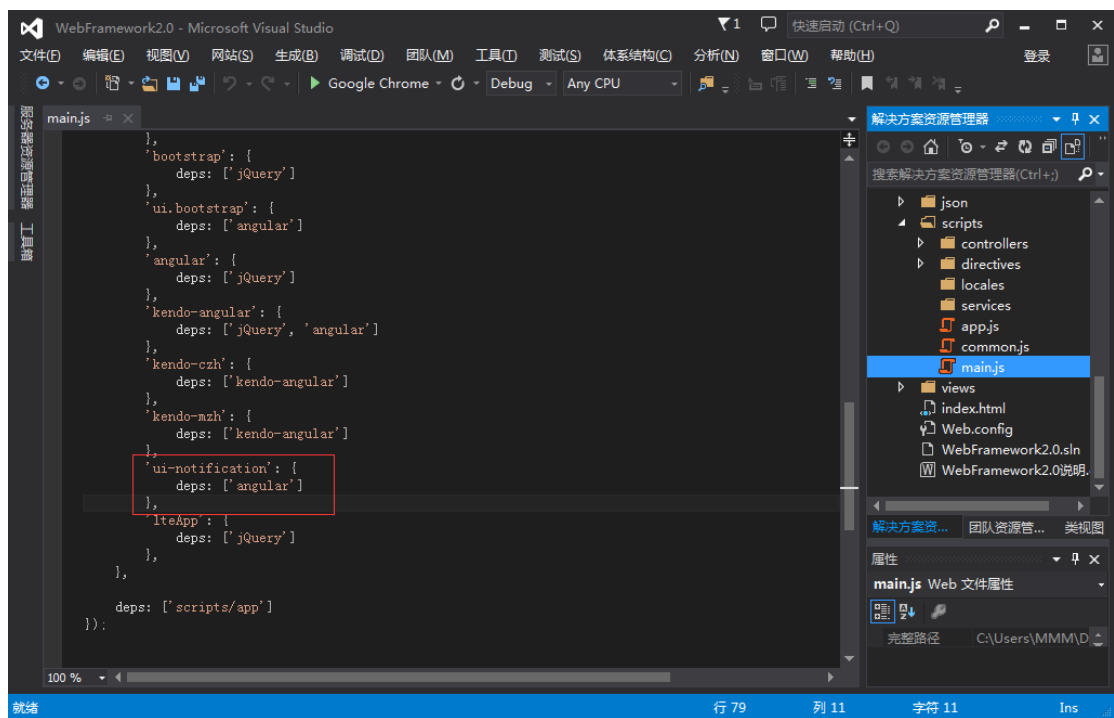


2. 将文件引入到项目第三方引用对应文件夹下



3. 配置 Requirejs 配置文件和依赖关系

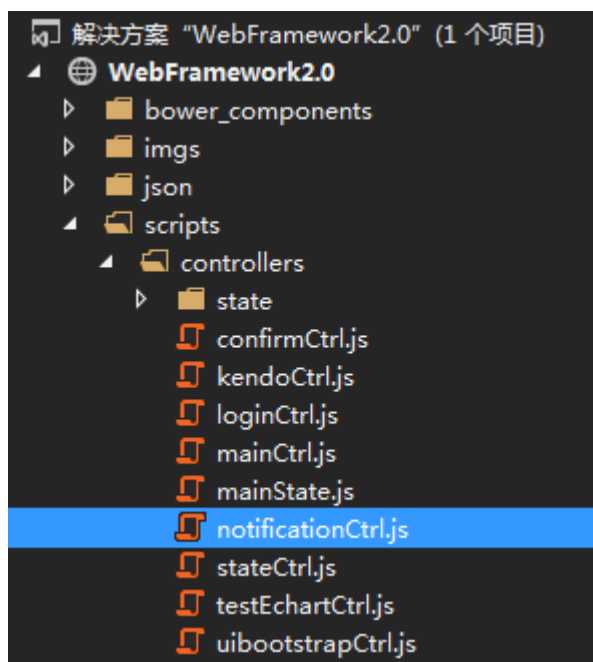


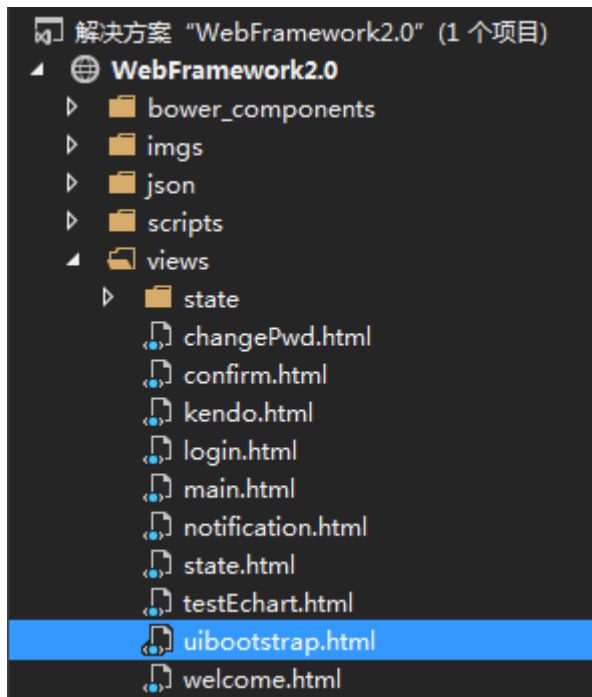


3.2 调用组件

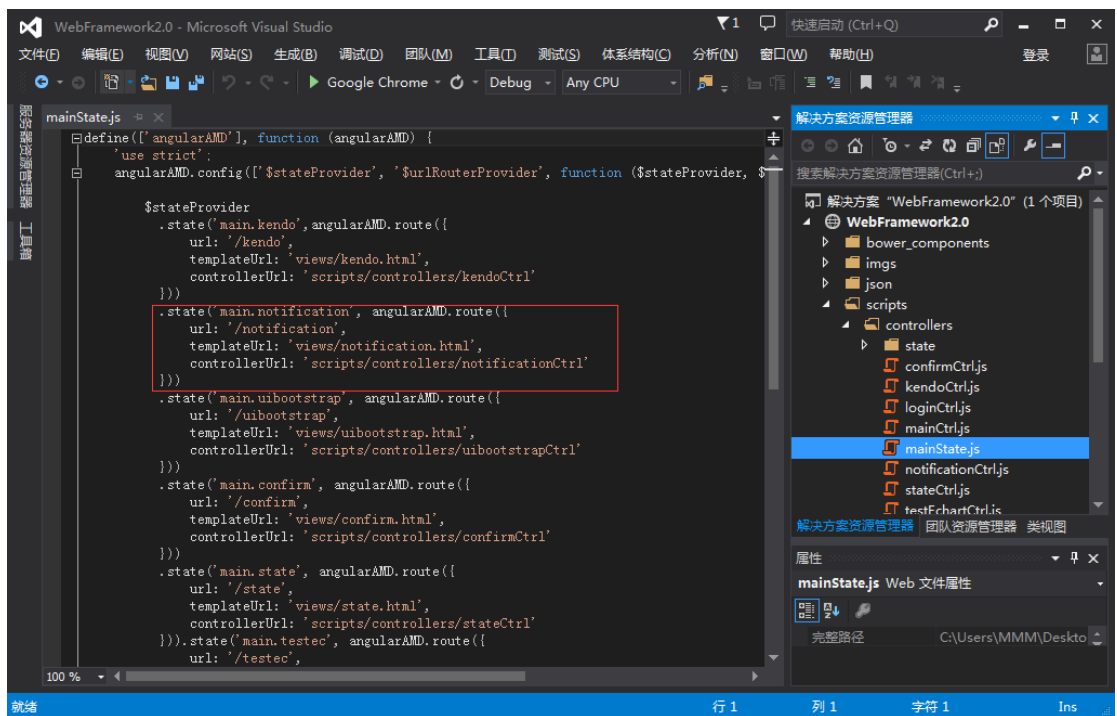
新建一个页面并调用所引入的 Angular-ui-notification 组件分以下过程步骤:

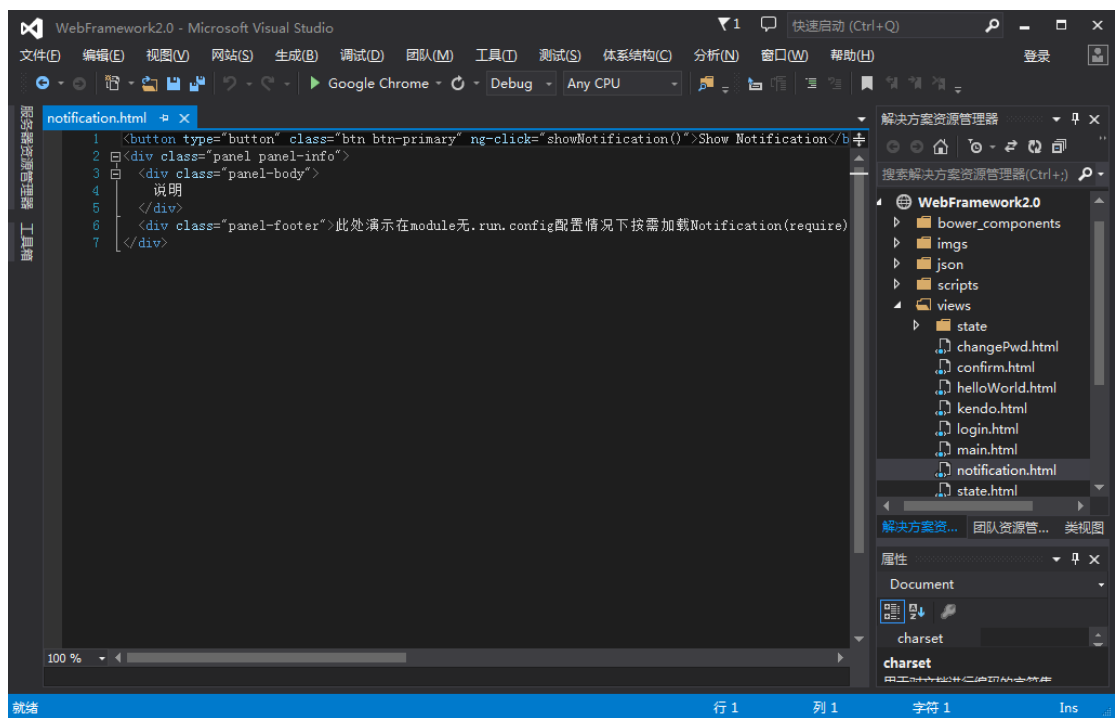
1. 新建控制器和视图页面





2. 配置页面路由和菜单





4. 页面呈现样式

