## 系统架构

* 1. **MEAN架构概述**  
     MEAN指的是以JavaScript为平台的现代WEB开发的技术栈，它是MongoDB + Express +AngularJS + NodeJS 四个框架的第一个字母组合。 在JavaScript界流行着一句话 “LAMP is so yesterday, MEAN is so tomorrow”，意为：以JavaScript为核心的全栈框架正在成为WEB开发界的主流。  
     
  2. **AngularJS在MSE-FOOD中的应用**

最近几年前端技术飞速发展，从前端只是切图做页面，JavaScript做动效、做表单校验的过程到了现在前端成为一个完整的软件工程生态圈，JavaScript可以担任起各种复杂的业务逻辑的处理。JavaScript随着项目的复杂度日益增加，类似AngularJS这样的MVC框架应运而生，它借鉴了很多传统服务端面向对象的语言的一些特性，使得JavaScript开发也变得更为严肃，更为工程化。  
  
**在**MSE-FOOD项目中完整使用了AngularJS的大部分的特性：

* + - 1. Module (模块化)  
         项目分为四大模块：用户模块、菜品模块、订单模块、系统模块。AngularJS的模块化特性使各个模块之间解除耦合、更好的组织代码，在大型项目中模块化的特性可方便团队之间独立的高效开发。
      2. Router (路由)  
         在路由中配置了各个url所对应的template和controller。 项目中使用了angular-ui 开发的ui-router组件，可以使得view能够多级嵌套，能够实现复杂的页面布局。
      3. Service (服务层)  
         在服务层使用了$resource 内置provider，只需短短几行代码即可方便实现与服务端的RESTful 服务的交互。
      4. Contoller (控制器层)  
         在控制器层主要对RESTful获得的数据进行处理和对提交表单的数据进行处理。
      5. Directive (指令)  
         所有所需要操作DOM的代码均放入了指令中。一些可以重复的使用的组件也抽象出来放入了指令中。
      6. Template (模版)  
         AngularJS自带的模版引擎也非常强大，可以在模版中对数据进行遍历操作、逻辑操作等。
      7. Two-Way-Bind (双向绑定)  
         无需任何事件绑定即可方便对表单中的数据进行处理、验证等操作，在对页面的数据处理过程中几乎不需要任何DOM操作，大大提高开发效率。
      8. Inject (注入依赖)  
         借鉴自Java的设计模式。可以方便的使得Service注入到Controller层中使用Service中的方法，也可使得不同的模块之间进行交互，并且可以重用Service。

AngularJS为我们造好平时需要花费大量精力构建的模式、框架，使我们开发者只需要关注业务实现。在我平时工作的使用中也对团队协作的效率提升很大，并且团队中大家写的代码风格也更为规范、统一。

* 1. **Node.JS + Express在MSE-FOOD中的应用**Express是Node.JS的一个web应用开发框架，也是一个非常典型的MVC框架，它已经为我们封装好了http的request、response的操作。  
       
     在MSE-FOOD项目中使用到的Express特性：
     + 1. Router (路由)  
          Express的路由定义应用的端点（URIs）以及如何响应客户端的请求，将请求地址在回调中进行处理或分发到对应的中间件进行处理。
       2. Middleware (中间件)  
          Express的本质就是在应用中调用各种中间件。它可以访问请求对象（request）, 响应对象（response）, 和 web 应用中处于请求-响应循环流程中的中间件，一般被命名为 next 的变量。
       3. Template (模版引擎)  
          Express的模版引擎和AngularJS的模版引擎类似，只不过模版在服务端直接渲染后显示到客户端。MSE-FOOD项目使用的是单页面应用的设计方案，因此仅首页是通过Express在服务端渲染的。

* 1. **MongoDB在MSE-FOOD中的应用**MongoDB是目前比较流行的NoSQL，以键值对的形式存储为BSON格式(Binary JSON)，可以极为方便的通过Node.JS获取到MongoDB的数据，甚至在数据通过RESTful传到客户端的过程中无需对数据进行任何处理。并且可以通过MapReduce对数据进行分布式聚合，满足高并发项目的需求。  
     在MSE-FOOD项目中使用到的MongoDB特性：
     + 1. 面向集合  
          数据被分组存储在数据集中，被称为一个集合（Collenction)。每个集合在数据库中都有一个唯一的标识名，并且可以包含无限数目的文档。集合的概念类似关系型数据库（RDBMS）里的表（table），不同的是它不需要定义任何模式（schema)。
       2. 模式自由  
          存储在mongodb数据库中的文件，在插入新数据时不需要知道它的任何结构定义。并且可以把不同结构的文件存储在同一个集合里。
  2. **Mongoose在MSE-FOOD中的应用**Mongoose是node.js 中连接mongoDB的一个类库，可以将mongoDB的数据映射成为node.js中的一个对象，类似于Java中的orm层，使得对象可以方便的进行save, update, remove等一系列操作。  
       
     Mongoose中非常常用的方法population可以将不同的document进行关联，解决NoSQL不能进行类似SQL表关联的操作。在MSE-FOOD中需要对Food和User，Order和Food、User的关联，即使用了population方法。
  3. **d“面向集合”（Collenction-Oriented），意思是数据被分组存储在数据集中，被称为一个集合（Collenction)。每个集合在数据库中都有一个唯一的标识名，并且可以包含无限数目的文档。集合的概念类似关系型数据库（RDBMS）里的表（table），不同的是它不需要定义任何模式（schema)。**
  4. **Fd**
  5. **Fd**
  6. **Fd**
  7. **D**
  8. **Df**

**调试**

**测试**