# Vue学习

<https://cn.vuejs.org/v2/guide/index.html>

# Java项目搭建

## 统一结果

统一结果的一般形式：

* 是否响应成功；
* 响应状态码；
* 状态码描述；
* 响应数据
* 其他标识符

统一结果类

上述第5个属于自定义返回，利用前4者可定义统一返回对象

**注意：**

* 外界只可以调用统一返回类的方法，不可以直接创建，因此构造器私有；
* 内置静态方法，返回对象；
* 为便于自定义统一结果的信息，建议使用链式编程，将返回对象设类本身，即return this;
* 响应数据由于为json格式，可定义为JsonObject或Map形式；

***统一结果类的使用参考了mybatis-plus中R对象的设计***

# 设计模式

## 观察者模式:

定义了对象之间的一对多依赖，这样一来当一个对象改变的时候，他的所有依赖者都会收到通知并自动更新。

## 抽象工厂模式

***抽象工厂模式''（Abstract Factory Pattern)用来创建复杂的对象，这种对象由许多小对象组成，而这些小对象都属于某个特定的“系列"（family）。***

比方说，在GUI系统里可以设计“抽象控件工厂''（abstract widget佰cto），并设计三个 “具体子类工厂" (concrete subclass cto）：MacWidgetFactory、XfcewidgetFactory、 WindowsWidgetFactor,它们都提供创建同一种对象的方法（例如都提供创建按钮的 make button 0方法，都提供创建数值调整框的make\_spinbox 0方法），而具体创建出来的对象的风格则与操作系统平台相符。我们可以编写create\_dia工og 0函数，令其以 “工厂实例"（cto instance)为参数来创建os x、Xfce及Windows风格的对话框，对话框的具体风格取决于传进来的工厂参数

## 工厂方法模式

如果子类的某个方法要根据情况来决定用什么类去实例化相关对象，那么可以考虑工厂方法模式。此模式可单独使用，也可在无法预知对象类型时使用（比方说，待初始化的对象类型要从文件中读入，或是由用户来输入）。

## 代理模式

静态代理

静态代理需要代理对象和被代理对象实现一样的接口

**优点：**可以在不修改目标对象的前提下扩展目标对象的功能

**缺点：**如果需要代理多个类，每个类都会有一个代理类，会导致代理类无限制扩展；如果类中有多个方法，同样的代理逻辑需要反复实现、应用到每个方法上，一旦接口增加方法，目标对象与代理对象都要进行修改

# 常用的认证机制

## HTTP Basic Auth

HTTP Base Auth 在HTTP中，基本认证是一种用来允许Web浏览器或其他客户端程序在请求时提供用户名和口令形式的身份凭证的一种登陆验证方式，通常用户名和密码会通过HTTP头传递。

***优点***：基本认证的一个优点就是基本上所有流行的网页浏览器都支持基本认证

缺点：由于用户名和密码都是base64编码的，base64 编码是可逆的，所以用户名和密码可以认为是明文。所以只有在客户端和服务器主机之间的连接是安全可信的前提下才可以使用

## OAuth

是一个关于授权的开放网络标准。允许用户提供一个令牌，而不是用户名和密码来访问他们存放的特定服务提供着的数据。现在的版本是2.0版。严格来说。OAuth2不是一个标准的协议，而是一个安全的授权框架。它详细描述了系统中的各个角色，用户，服务前端应用（比如APi）以及客户端（比如网站和移动app）之间怎么实现相互认证。

***优点：***

快速开发

实施代码量小

维护工 作减少

如果设计的API要被不同的App使用，并且每个app使用的方式也不一样，使用OAuth2是个不错的选择。

***缺点***：

OAuth2是一个安全框架，描述了在各种不同场景下，多个应用之间的授权问题。有海量的资料需要学习，要完全理解需要花费大量的时间。

OAth2不是一个严格的标准协议，因此在实施过程中更容易出错

## jwt认证

Json web token(jwt),根据官网的定义，是为了在网络应用环境间传递声明而执行的一种基于JSON的开放标准，该token被设计为紧凑且安全的，特别适用于分布式站点的单点登录场（SSO）场景。JWT的声明一般用来在身份提供者和服务提供者间传递被认证的用户身份信息，以便从资源服务器获取资源，也可以增加一些额外的其他逻辑所必须的声明信息，该token也可以直接被用户认证，也可被加密。

***jwt 特点***

* 体积小，因而传输速度快
* 传输方式多样，可以通过URL/POST参数 HTTP头部等方式传输
* 严格的结构化，它自身（payload）就包含了所有与用户相关的验证信息，如用户可访问路由，访问有效期等信息，服务器无需再去连接数据库验证信息的有效性，并且payload支持为你的应用而定制化。
* 支持跨域验证，可以应用于单点登录。

组成结构:

1 Header 由base64算法对称加密 包含 加密算法和类型

2PayLoad 由base64算法对称加密