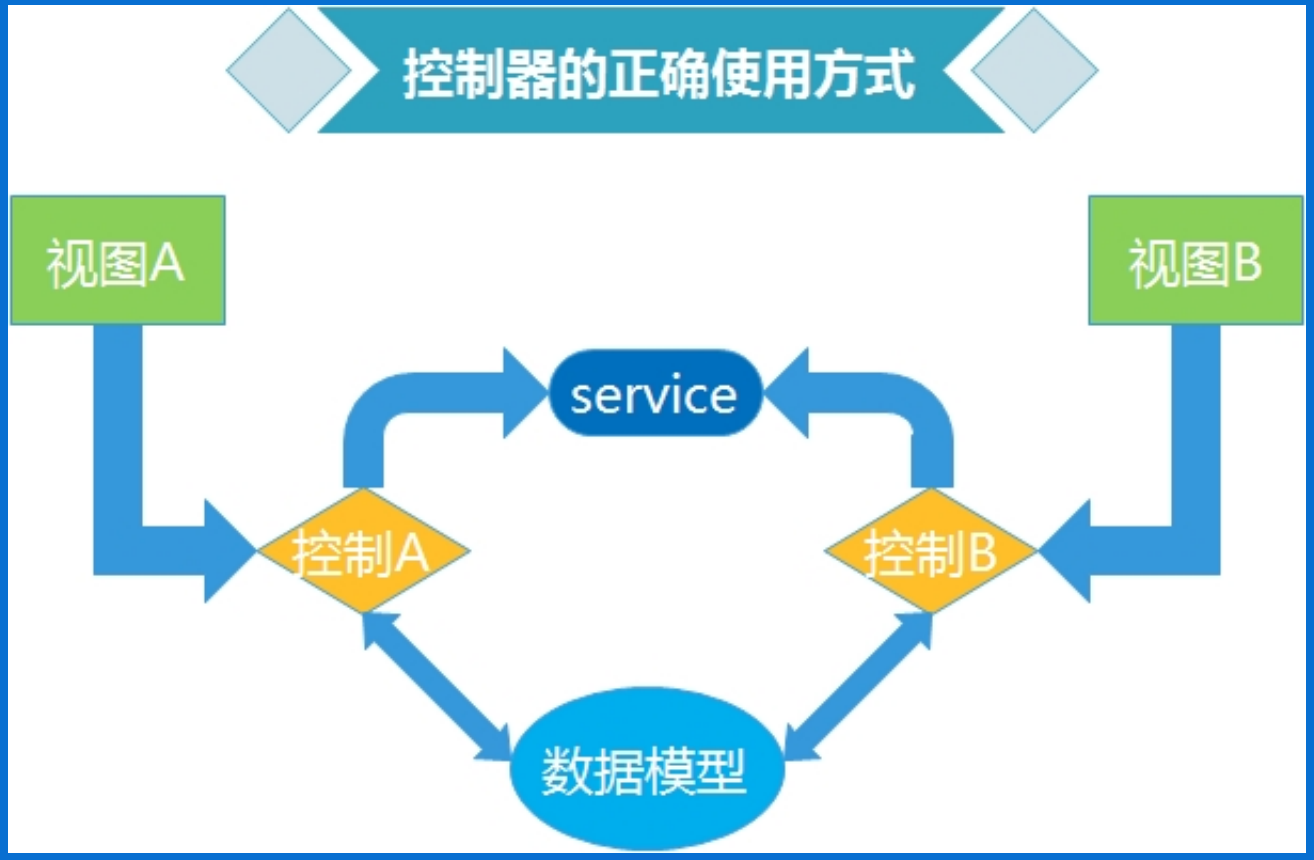
为什么需要MVC

- 项目规模越来越大，需要 **切分模块** 和职责
- 消除重复代码，实现 **复用**
- 方便 **重构** ，修改一个地方不影响其它功能
- 核心目标是 **模块化** 和 **复用**

MVC的实现



Model模型使用

- MVC借助于 **$scope** 实现
- $scope是一个 **基本** javascript对象
- $scope是一个 **树型** 结构，与DOM标签平行
- 子$scope对象遵循原型继承,会 **继承** 父$scope上的属性和方法
- 每一个angular应用只有一个 **根** $rootScope对象(ng-app)上
- $scope是MVC和 **双向数据绑定** 的基础
- $scope是表达式的执行 **上下文环境**
- 有自己的生命周期 创建->注册监控者->模型变化->检测变化->销毁
- $scope提供一些工具方法 **$watch/$apply**
- $scope可以 **传播事件** ，类似DOM事件，可以向上也可以向下传播

控制器

- ng-controller指令就是用来定义应用程序控制器的，并且同时创建了一个新的作用域关联到相应的DOM元素上
- 不要 **复用** Controller,一个控制器只负责一个视图
- 不要在控制器中操作DOM，使用 **指令**
- 不要在Controller里做数据 **过滤和格式化** ，使用filter过滤器
- 控制器之间不要互相直接调用，控制器之间的交互通过 **事件** 交互

- 一切是从模块开始，module就是容器,其它的元素都挂在module里面.
- 模块是一些功能的集合，如控制器、服务、过滤器、指令等子元素组成的整体

模块之间如何依赖依赖注入

```
//创建、获取angular中的模块
    angular.module();

    // 传递参数多于一个表示新建模块;空数组代表该模块不依赖其他模块
    var createModule = angular.module("myModule", []);

    // 只有一个参数(模块名),代表获取模块,如果模块不存在,angular框架会抛异常
    var getModule = angular.module("myModule");

    该函数既可以创建新的模块，也可以获取已有模块，是创建还是获取，通过参数的个数来区分。

    angular.module(name, [requires], [configFn]);

    - name：字符串类型，代表模块的名称；
    - requires：字符串的数组，代表该模块依赖的其他模块列表，如果不依赖其他模块，用空数组即可；
    - configFn：用来对该模块进行一些配置。
```