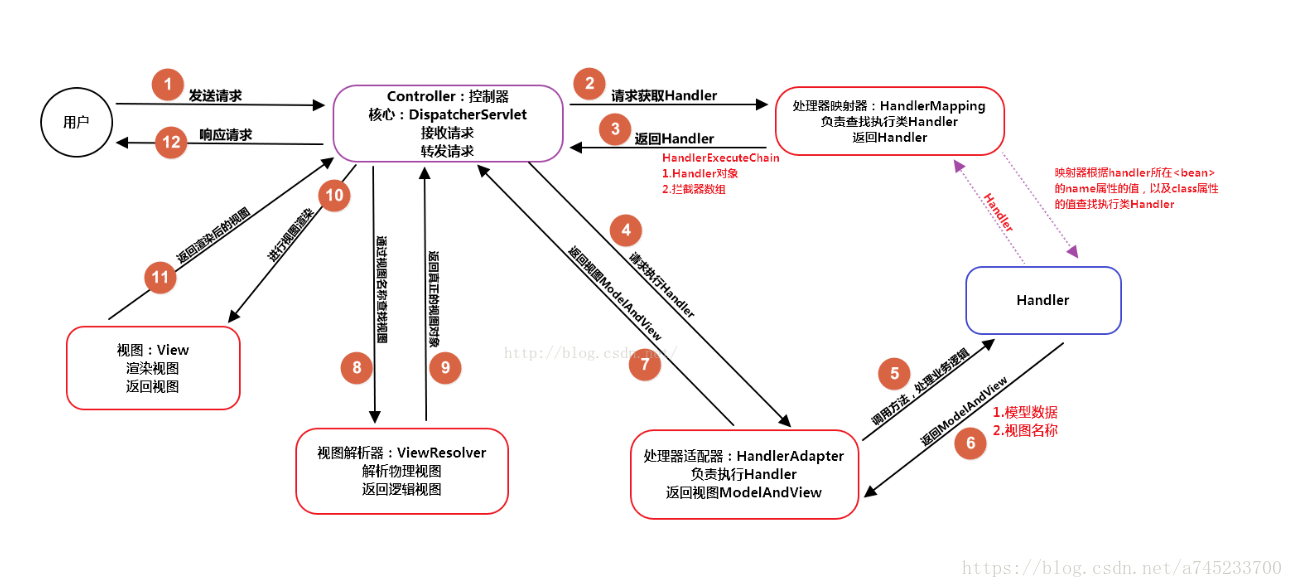
Spring MVC: M model，dao层的数据模型，Java bean数据模型。V view就是视图层，通过视图渲染，来展示页面。

用户发起请求，前端控制器（接受请求，调用业务，派发页面）。Model（service，dao,entity）。View视图层

用户浏览器访问：

1. 发出request请求：来到前端控制器
2. 查找处理器handler：来到处理器映射器
3. 处理器映射器返回一个处理器的执行链：来到前端控制器
4. 前端控制器将这个执行连给处理器适配器，处理器适配器找到相应的处理器，
5. 处理器执行，将结果ModelandView返回给返回给处理器适配器，
6. 处理器适配器将结果返回给前端控制器。
7. 前端控制器将model和view给视图解析器
8. 视图解析器将view返回给处理器
9. 处理器将试图渲染，同时将response相应返回给用户。



通过前端控制器，接受请求，响应结果，相当于是一个转发器，减少其他组件之间的耦合度

处理器映射器：根据请求的URL来查找我们的处理器（handler），就是我们的control层。

处理器适配器：负责调用处理器，

视图解析器：将逻辑view和model解析成真正的视图

MVC的主要组件：

前端控制器，处理器映射器，处理器，处理器适配器，视图解析器，视图（就是一个接口，）

@RequestMapping：用于表示访问的路径。

@RequestBody：接受http种的json数据，并将json数据转换成Java对象。

@ResponseBody: 将controller方法返回结果转换成json数据

@Controller+@ResponseBody=@RestController

对于请求乱码：1.post请求，在web的xml配置文件中添加CharacterEncodingFilter：设置为utf-8

<filter>

<filter-name>CharacterEncodingFilter</filter-name>

<filter-class>org.springframework.web.filter.CharacterEncodingFilter</filter>

<init-param>

<param-name>encoding</param-name>

<param-value>utf-8</param-value>

</init-taram>

</filter>

<filter-mapping>

<filter-name>CharacterEncodingFilter</filter-name>

<url-pattern>/\*</url-pattern>

</filter-mapping>

如果是get方法乱码：修改tomcat的配置文件，将tomcat的编码改成和工程一样的编码。

事务并发出现的问题：脏读、幻读、丢失修改、不可重复。

事务隔离级别 丢失修改 脏读 不可重复读 幻读

未提交读 解决 出现 出现 出现

已提交读 解决 解决 出现 出现

可重复读 解决 解决 解决 出现

串行化 解决 解决 解决 解决

Mysql默认是可重复读。

Springboot:

对spring中的大部分配置采用默认值，简化了spring的难度，实现了开箱即用，采用约定也是大于配置。

Springboot配置的加载顺序：properties文件，yaml，系统的环境变量，命令行

Yaml文件相对于属性文件，更加简洁，层次清晰。

Spring Data JPA（Java Persisstence API Java持久层api） 是spring框架对jpa的整合。减少（访问数据层）dao层的开发，

定义数据访问层：定义一个接口，让这个接口继承JpaRepository就可以

Public inerface Student Repository extends JpaRepository<Person, Long> {

}

Java data jpa 相当于是其他ORM框架的一种抽象。