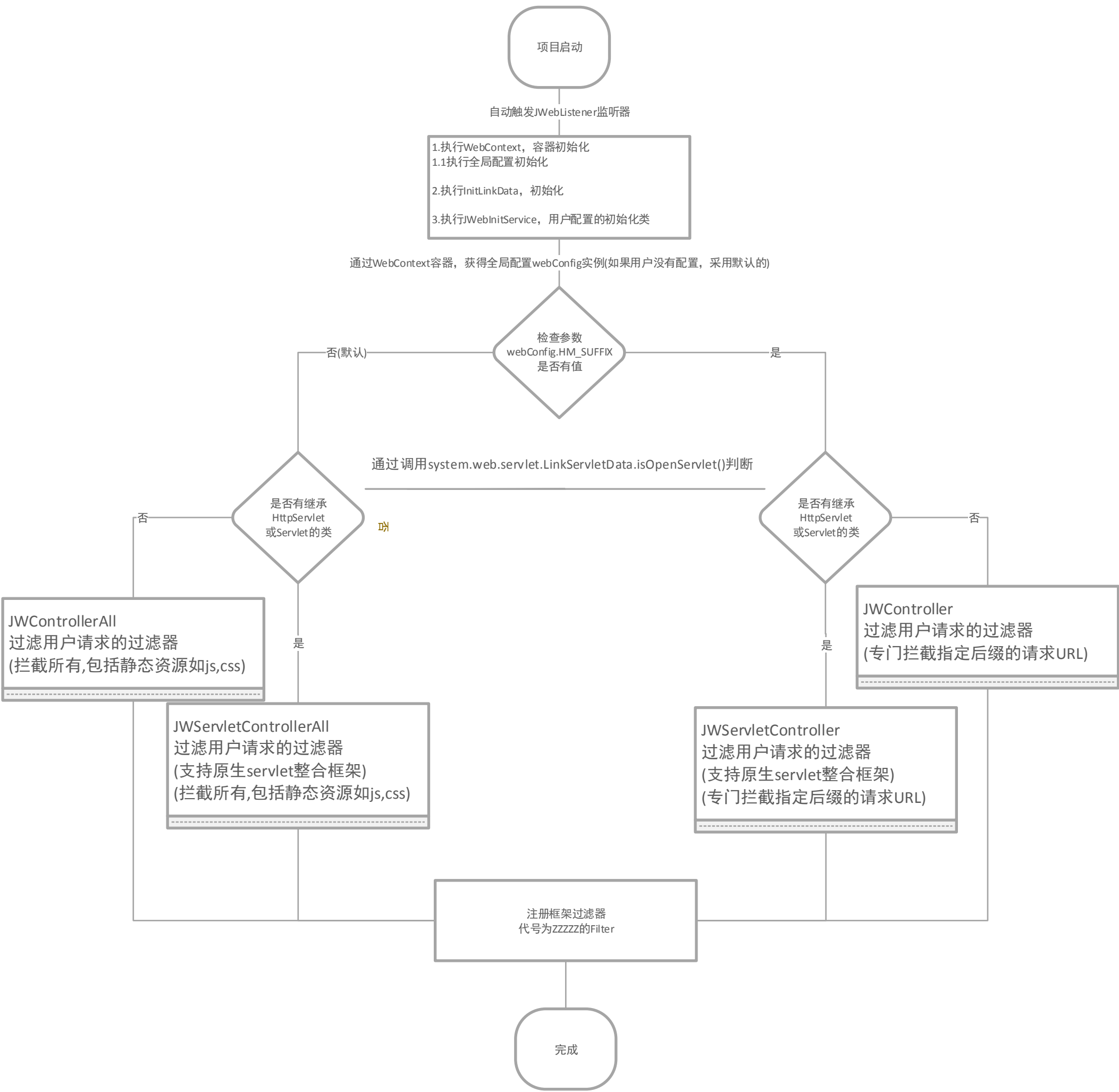


自动注册框架

资源列表

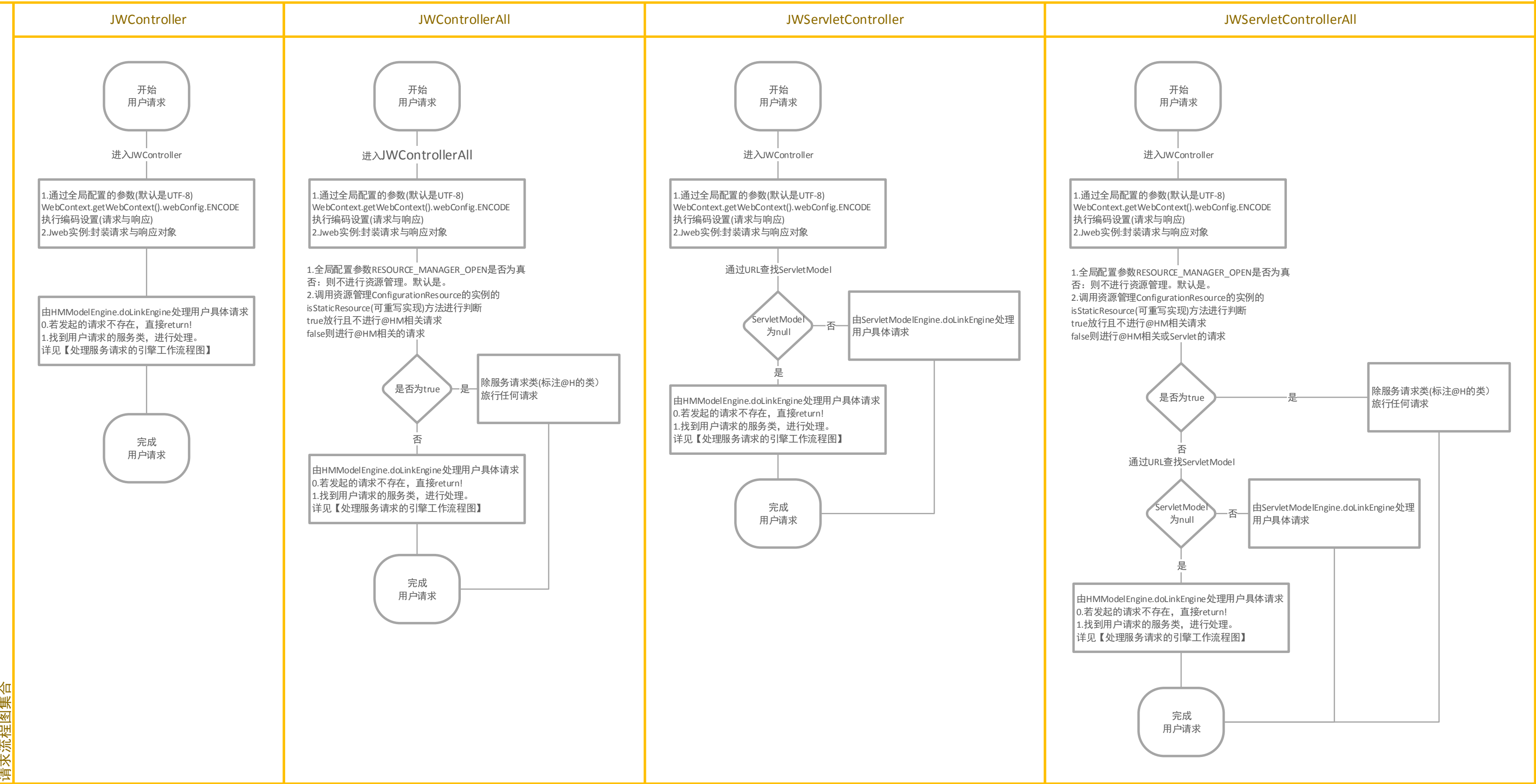
@WebListener public class JWebListener implements ServletContextListener
WebContext 框架的容器
InitLinkData 执行初始化服务请求类
JWebInitService 执行用户配置的初始化
JWServletControllerAll 过滤用户请求的过滤器 (支持原生servlet整合框架) (拦截所有,包括静态资源如js,css)
JWControllerAll 过滤用户请求的过滤器 (拦截所有,包括静态资源如js,css)
JWServletController 过滤用户请求的过滤器 (支持原生servlet整合框架) (专门拦截指定后缀的请求URL)
JWController 过滤用户请求的过滤器 (专门拦截指定后缀的请求URL)

启动流程



不同的过滤器，不同的处理流程  
根据全局配置 自动选择。  
具体 选择原理，请见【自动注册框架流程图】

请求流程图集合



HMMModelEngine.doLinkEngine引擎处理用户请求流程图

执行逻辑阶段

LinkHMMModel类具体

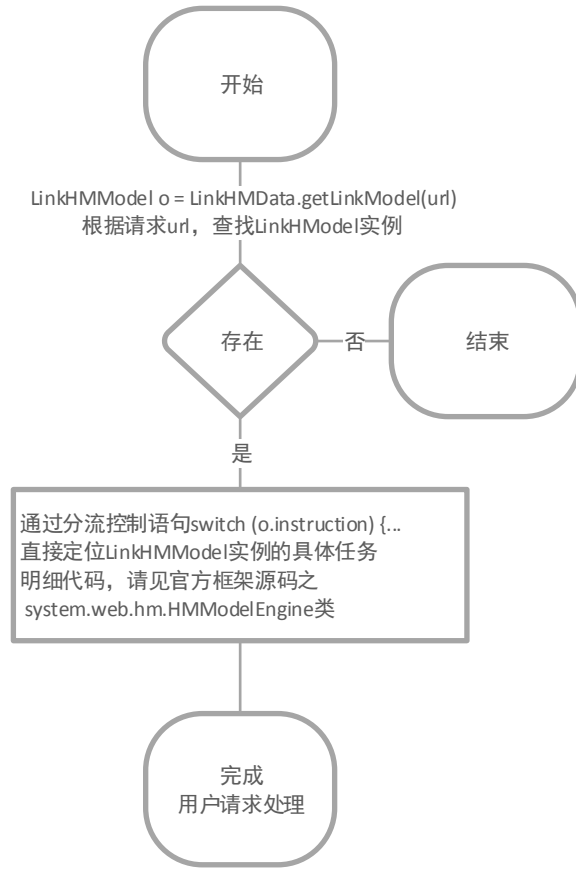
```
LinkHMMModel

//资源指令
public final int instruction;
//执行用户请求关联的类
public final Class hClass;
//执行用户请求的方法
public final Method method;
//1执行顶部过滤链
public final FilterModel[] fmTop;
//2执行校验
public final ValidateModel vm;
//4执行中部过滤链(校验后,@H类前)
public final FilterModel[] fmCenter;
//8执行底部过滤链(@H类后)
public final FilterModel[] fmBottom;
//是否是静态方法
public final boolean isStaticMethod;
```

资源指令是各功能的指令值相加。  
**HMMModelEngine引擎首先从指令进行高速定位。**  
如值为1:  
只会执行fmTop里的过滤链 +处理服务类  
如值为3:  
啞会执行fmTop与vm的资源 +处理服务类  
通过使用数学模型, 请求资源(过滤链\校验)变得  
强大灵活与高速! 没用到的直接看不见!

所有的资源, 在执行注册框架阶段, 就已分类好, 确认好。  
所以对执行引擎来说, 只要根据指令, 就可直接定位处理相关资源! 所以非常  
高效!!

处理流程



ServletModelEngine.doLinkEngine引擎处理用户请求流程图

执行逻辑阶段

ServletModel类具体

ServletModel

//资源指令  
public final int instruction;

//1执行顶部过滤链  
public final FilterModel[] fmTop;

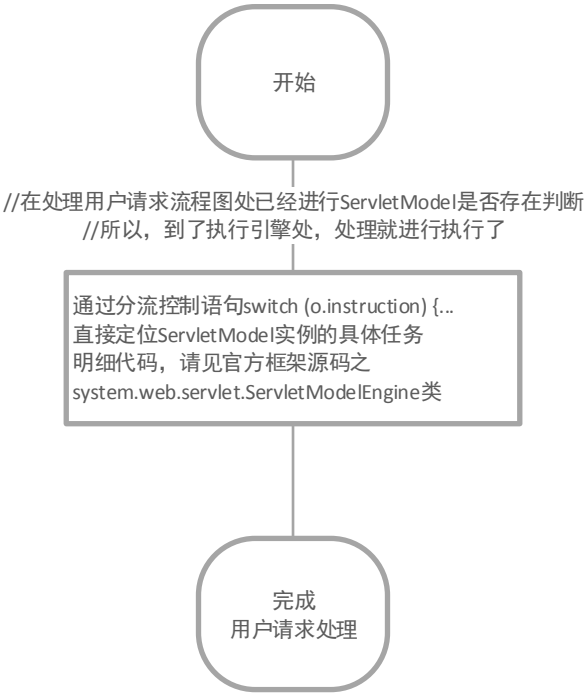
//2执行校验  
public final ValidateModel vm;

//4执行中部过滤链(校验后,@H类前)  
public final FilterModel[] fmCenter;

//8执行底部过滤链(@H类后)  
public final FilterModel[] fmButtom;

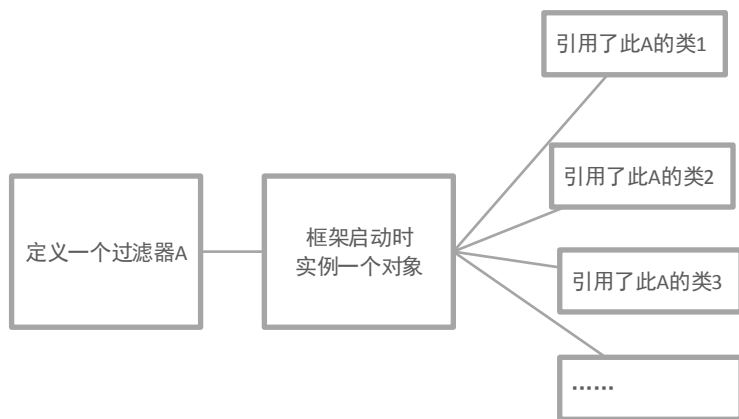
资源指令是各功能的指令值相加。  
**ServletModelEngine擎首先从指令进行高速定位。**  
如值为1:  
只会执行fmTop里的过滤链 +然后转交web容器管控  
如值为3:  
唑会执行fmTop与vm的资源 +然后转交web容器管控  
通过使用数学模型，请求资源(过滤链\校验)变得强大  
灵活与高速！没用到的直接看不见!同时，我们的校  
验、jw过滤器完美与原生Servlet合体！

处理流程



所有的资源，在执行注册框架阶段，就已分类好，确认好。  
所以对执行引擎来说，只要根据指令，就可直接定位处理相关资源！所以非常高效！！

服务模型内的执行顺序
LOCATION_TOP:顶层过滤
Validate校验
LOCATION_CENTER:中间层过滤
处理用户请求的服务类 @H+@M标注的或Servlet
LOCATION_BOTTOM:底层过滤



**1.每一个过滤器都有一个实例。**

如果有多处引用时，都是指向这唯一的实例。

**2.当一个类、方法有重复的配置时，会自动去除重复。**如：多个过滤组配置在同一个方法里。交集的地方，保留最前的，除去后面重复的（过滤器有执行先后顺序）