

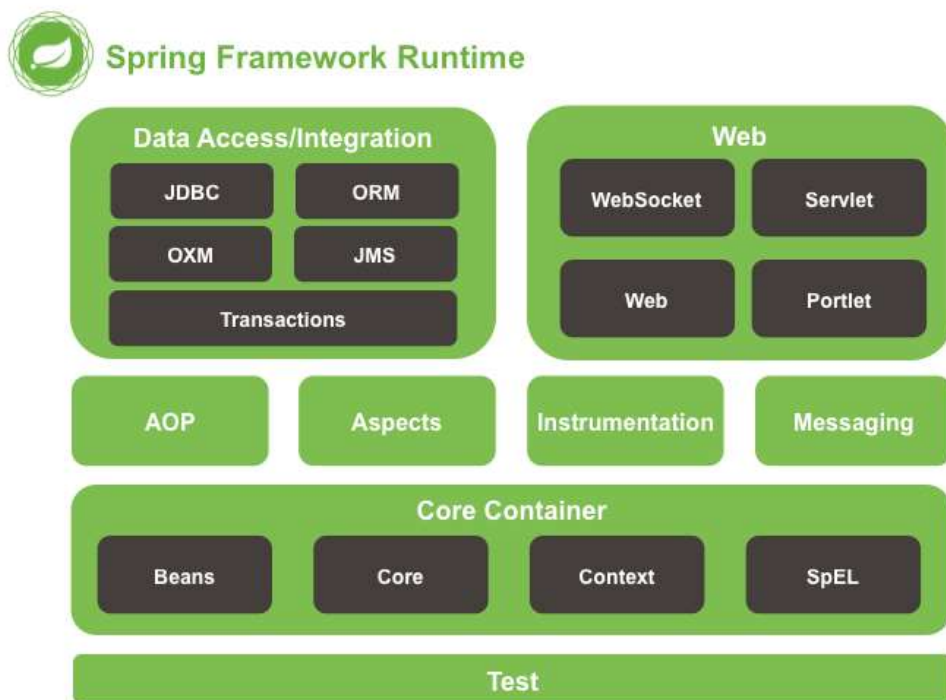
## 1、SpringMVC简介

笔记本： spring

创建时间： 2022/4/4 17:24

作者： 雷丰阳

1、Spring实现web模块的；简化web开发的；



SpringMVC==Spring的web模块；

MVC：

MVC：新的软件架构模式

M：Model模型：封装和映射数据（javaBean）

V：View视图：界面显示工作（.jsp）

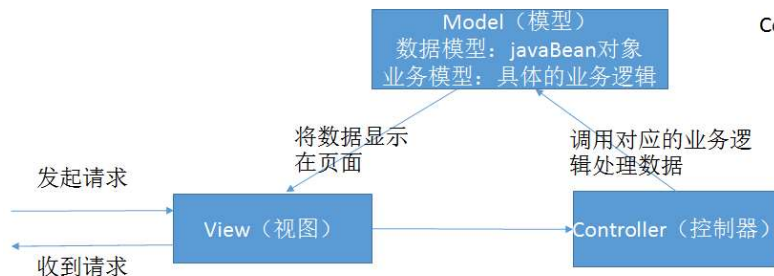
C：Controller控制器：控制整个网站的跳转逻辑（Servlet）

Com.atguigu.entities(实体)

Com.atguigu.dao

Com.atguigu.service

Com.atguigu.controller



MVC提倡：每一层只编写自己的东西，不写任何其他的代码

控制器{

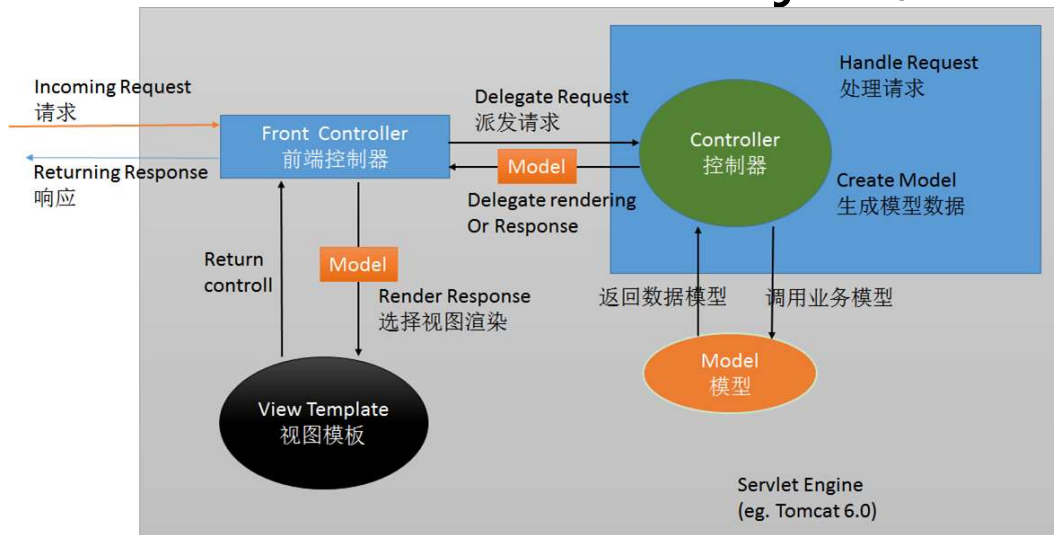
调用业务逻辑处理

调整到某个页面

}

分层就为了解耦，解耦为了维护方便和分工合作

# POJO: Plain Old Java Object;



SpringMVC的MVC实现思想

HelloWorld:

0) 、

New Dynamic Web Project

**Dynamic Web Project**

Create a standalone Dynamic Web project or add it to a new or existing Enterprise Application.

Project name: 1.SpringMVC\_helloworld

Project location

☒ Use default location

Location: F:\0325EE\1.SpringMVC\_helloworld Browse...

Target runtime

Apache Tomcat v7.0 New Runtime...

Dynamic web module version

3.0 推荐选2.5

Configuration

<custom> Modify...

Hint: Get started quickly by selecting one of the pre-defined project configurations.

EAR membership

☐ Add project to an EAR

EAR project name: 1.SpringMVC\_helloworldEAR New Project...

Working sets

☒ Add project to working sets

Working sets: SpringMVC Select...

? < Back Next > Finish Cancel

1) 、导包

1)、SpringMVC是Spring的web模块；所有模块的运行都是依赖核心模块（IOC模块）

#### 1)、核心容器模块：

```
commons-logging-1.1.3.jar
spring-aop-4.0.0.RELEASE.jar
spring-beans-4.0.0.RELEASE.jar
spring-context-4.0.0.RELEASE.jar
spring-core-4.0.0.RELEASE.jar
spring-expression-4.0.0.RELEASE.jar
```

#### 2)、web模块

```
spring-web-4.0.0.RELEASE.jar
spring-webmvc-4.0.0.RELEASE.jar
```

### 2)、写配置

#### 1)、web.xml

```
<!-- SpringMVC思想是有一个前端控制器能拦截所有请求，并智能派发；
    这个前端控制器是一个servlet；应该在web.xml中配置这个servlet来拦截所有请求
-->

<!-- The front controller of this Spring Web application,
    responsible for handling all application requests -->
<servlet>
    <servlet-name>springDispatcherServlet</servlet-name>
    <servlet-
class>org.springframework.web.servlet.DispatcherServlet</servlet-class>

    <init-param>
        <!-- contextConfigLocation:指定SpringMVC配置文件位置 -->
        <param-name>contextConfigLocation</param-name>
        <param-value>classpath:springmvc.xml</param-value>
    </init-param>
    <!-- servlet启动加载，servlet原本是第一次访问创建对象；
    load-on-startup:服务器启动的时候创建对象；值越小优先级越高，越先创建对象；
    -->
    <load-on-startup>1</load-on-startup>
</servlet>

<!-- Map all requests to the DispatcherServlet for handling -->
<servlet-mapping>
    <servlet-name>springDispatcherServlet</servlet-name>
    <!--
        /*和/都是拦截所有请求； /: 会拦截所有请求，但是不会拦截*.jsp；能保证
        jsp访问正常；
        /*的范围更大；还会拦截到*.jsp这些请求；一但拦截jsp页面就不能显示了；
        -->
    <url-pattern>/</url-pattern>
</servlet-mapping>
```

#### 2)、SpringMVC配置文件

```
<!-- 扫描所有组件 -->
<context:component-scan base-package="com.atguigu"></context:component-
scan>

<!-- 配置一个视图解析器；能帮我们拼接页面地址-->
<bean
class="org.springframework.web.servlet.view.InternalResourceViewResolver">
    <property name="prefix" value="/WEB-INF/pages/"></property>
    <property name="suffix" value=".jsp"></property>
</bean>
```

### 3)、测试

```

/**
 * 1、告诉SpringMVC这是一个处理器，可以处理请求
 *   @Controller: 标识哪个组件是控制器
 *   @author lfy
 */
@Controller
public class MyFirstController {

    /**
     *   /代表从当前项目下开始；处理当前项目下的hello请求
     */
    @RequestMapping("/hello")
    public String myfirstRequest(){
        System.out.println("请求收到了....正在处理中");
        //视图解析器自动拼串；
        // <property name="prefix" value="/WEB-INF/pages/"></property>
        // <property name="suffix" value=".jsp"></property>
        //   (前缀)/WEB-INF/pages/+返回值(success)+后缀(.jsp)
        return "success";
    }
}

```