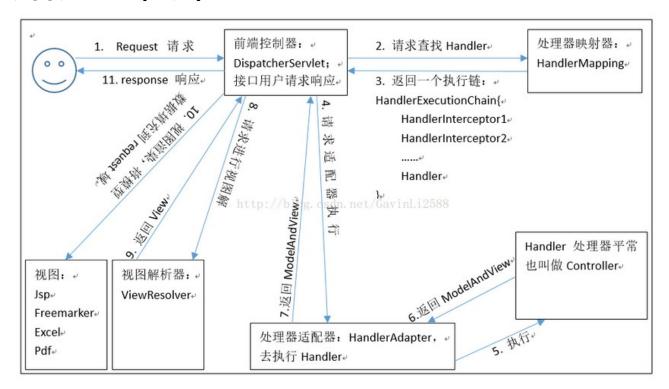
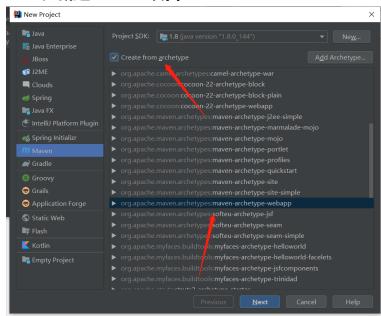
# 环境搭建和入门案例

## 项目流程图 (重点)

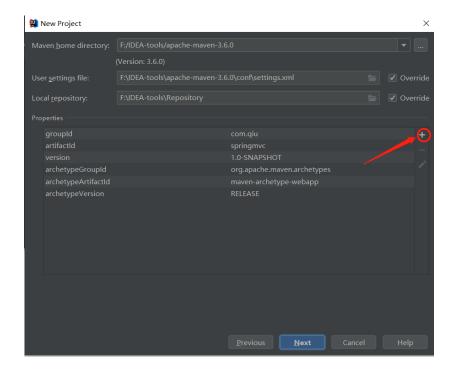


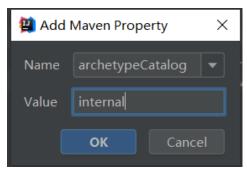
### 一、环境搭建

#### 1、新建Maven项目



2、添加 archetypeCatalog internal





### 3、配置pom.xml (更改properties和dependencies)

- cproperties>
- project.build.sourceEncoding>
- <maven.compiler.source>1.8</maven.compiler.source>
- <maven.compiler.target>1.8</maven.compiler.target>
- <spring.version>5.0.2.RELEASE</spring.version>
- </properties>
- <dependencies>
  - <dependency>
  - <groupId>org.springframework</groupId>
  - <artifactId>spring-context</artifactId>
  - <version>\${spring.version}</version>
  - </dependency>
  - <dependency>
    - <groupId>org.springframework</groupId>
    - <artifactId>spring-web</artifactId>
  - <version>\${spring.version}</version>
  - </dependency>
  - <dependency>
  - <groupId>org.springframework</groupId>
  - <artifactId>spring-webmvc</artifactId>
  - <version>\${spring.version}</version>

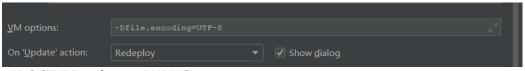
- </dependency>
  <dependency>
  <groupId>javax.servlet</groupId>
  <artifactId>servlet-api</artifactId>
  <version>2.5</version>
  <scope>provided</scope>
  </dependency>
  <dependency>
  <groupId>javax.servlet.jsp</groupId>
  <artifactId>jsp-api</artifactId>
  <version>2.0</version>
  <scope>provided</scope>
  </dependency>
- </dependencies>

#### 4、添加文件夹,并使用



5、 在web.xml配置核心前端控制器- 一个 Servlet 配置filter时要注意位置,不然</web-app>会报错,没事,可以正常运行

如果依旧中文乱码,则修改tomcat的 添加: -Dfile.encoding=UTF-8 过滤器filter的路径要加/\*,如果是/的话他不会过滤jsp文件



- <!DOCTYPE web-app PUBLIC
- "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
- "http://java.sun.com/dtd/web-app 2 3.dtd" >
- <web-app>
  - <display-name>Archetype Created Web Application</display-name>
- <!--配置解决中文乱码的过滤器-->
- <filter>
  - <filter-name>CharacterEncodingFilter</filter-name>
  - <filter-class>org.springframework.web.filter.CharacterEncodingFilter</filter-class>
  - <init-param>
    - <param-name>encoding</param-name>
    - <param-value>UTF-8</param-value>
  - </init-param>
  - <init-param>
    - <param-name>forceEncoding</param-name>

```
<param-value>true</param-value>
  </init-param>
</filter>
<filter-mapping>
  <filter-name>CharacterEncodingFilter</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
 <!--配置前端控制器-->
 <servlet>
  <servlet-name>dispatcherServlet</servlet-name>
  <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
  <init-param>
   <param-name>contextConfigLocation</param-name>
   <param-value>classpath:springmvc.xml</param-value>
  </init-param>
  <load-on-startup>1</load-on-startup>
 </servlet>
 <servlet-mapping>
  <servlet-name>dispatcherServlet</servlet-name>
  <url-pattern>/</url-pattern>
 </servlet-mapping>
</web-app>
```

#### 6、创建配置文件xxx.xml

# 二、开始入门:

```
springMVCText01 F:\java\IDEA-Worksapce\SpringM\
🕨 🖿 .idea
▼ 🖿 src
  ▼ I main
    ▼ Iijava
       ▼ 🖿 com
          🔻 陆 qiu
            ▼ controller
                 HelloController
          提 springmvc.xml
     webapp
       ▼ WEB-INF
          ▼ lages
               🚜 success.jsp
            🚜 web.xml
         🚜 index.jsp
target
  m pom.xml
```

### 1、配置springmvc.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"</pre>
   xmlns:mvc="http://www.springframework.org/schema/mvc"
   xmlns:context="http://www.springframework.org/schema/context"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="
    http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans.xsd
    http://www.springframework.org/schema/mvc
    http://www.springframework.org/schema/mvc/spring-mvc.xsd
    http://www.springframework.org/schema/context
    http://www.springframework.org/schema/context/spring-context.xsd">
  <!-- 开启注解扫描 -->
  <context:component-scan base-package="com.giu"/>
  <!-- 视图解析器对象 -->
  <bean id="internalResourceViewResolver"</pre>
class="org.springframework.web.servlet.view.InternalResourceViewResolver">
    property name="prefix" value="/WEB-INF/pages/"/>
    cproperty name="suffix" value=".jsp"/>
  </bean>
  <!-- 开启SpringMVC框架注解的支持 -->
  <mvc:annotation-driven/>
</beans>
2、配置web.xml
<!DOCTYPE web-app PUBLIC
"-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
"http://java.sun.com/dtd/web-app 2 3.dtd" >
<web-app>
 <display-name>Archetype Created Web Application</display-name>
 <!--配置前端控制器-->
 <servlet>
  <servlet-name>dispatcherServlet</servlet-name>
  <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
  <init-param>
   <param-name>contextConfigLocation</param-name>
   <param-value>classpath:springmvc.xml;</param-value>
  </init-param>
  <load-on-startup>1</load-on-startup>
 </servlet>
 <servlet-mapping>
  <servlet-name>dispatcherServlet</servlet-name>
  <url-pattern>/</url-pattern>
 </servlet-mapping>
</web-app>
```

## 3、配置好jsp文件

# 4、测试

```
@Controller
public class HelloController {
    @RequestMapping(path="/hello")

public String sayHello(){
    System.out.println("Hello StringMVC");
    return "success";
    }
}
```