

# Web MVC 视图应用

---

## 模板引擎

---

### 新一代服务端模板引擎



参考资源：<https://www.thymeleaf.org/doc/tutorials/3.0/usingthymeleaf.html>

### 模板类型

- HTML
- XML
- TEXT
- JAVASCRIPT
- CSS
- RAW

### Thymeleaf 语法

```
<!DOCTYPE html>

<html xmlns:th="http://www.thymeleaf.org">

  <head>
    <title>Good Thymes Virtual Grocery</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <link rel="stylesheet" type="text/css" media="all"
          href="../../css/gtvg.css" th:href="@{/css/gtvg.css}" />
  </head>

  <body>

    <p th:text="#{home.welcome}">Welcome to our grocery store!</p>

  </body>

</html>
```

## 核心要素

### 资源定位（模板来源）

- 通用资源抽象
  - 文件资源：File
  - ClassPath资源：ClassLoader
  - 统一资源：URL
  - Web资源：ServletContext
- Spring 资源抽象：
  - Spring 资源：Resource

### 渲染上下文（变量来源）

- 不同的实现
  - Context：Thymeleaf 渲染上下文
  - Model：Spring Web MVC 模型
  - Attribute：Servlet 上下文

### 模板引擎（模板渲染）

- ITemplateEngine 实现
  - TemplateEngine：Thymeleaf 原生实现
  - SpringTemplateEngine：Spring 实现
  - SpringWebFluxTemplateEngine：Spring WebFlux 实现

### 示例：使用 Thymeleaf API 渲染内容

```
// 构建引擎
SpringTemplateEngine templateEngine = new SpringTemplateEngine();
// 创建渲染上下文
Context context = new Context();
context.setVariable("message", "Hello,World");
// 模板的内容
String content = "<p th:text=\"${message}\">!!!</p>";
// 渲染（处理）结果
String result = templateEngine.process(content, context);
// 输出渲染（处理）结果
System.out.println(result);
```

### 示例：Thymeleaf 与 Spring 资源整合

- Spring 资源
  - ResourceLoader 与 Resource
- 目的
  - 理解 Spring Resource 抽象

## 视图处理

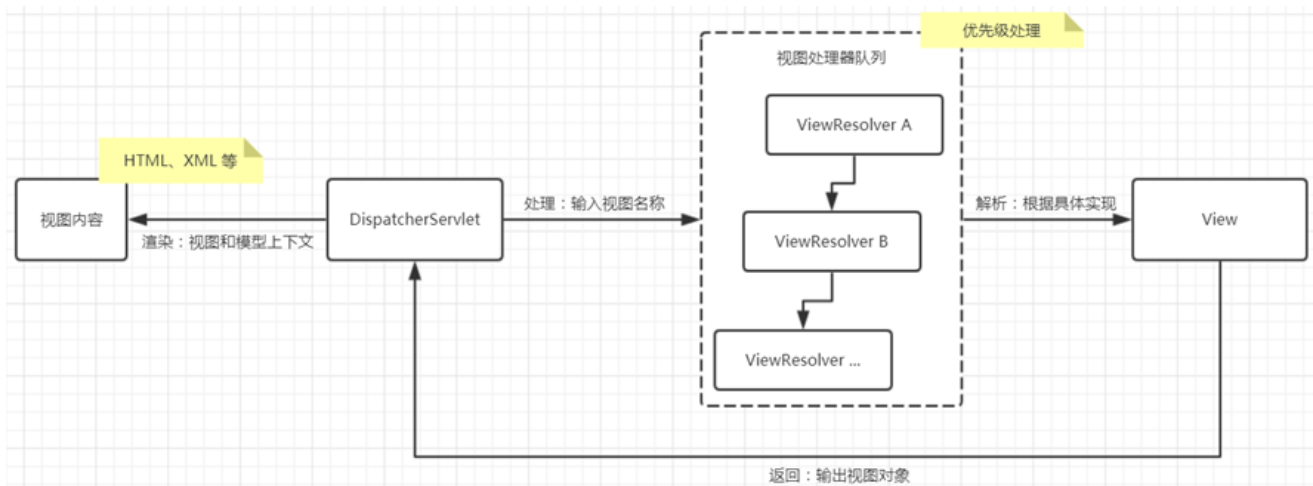
### Spring Web MVC 视图组件

- ViewResolver : 视图解析器
- View : 视图组件
- DispatcherServlet : 总控

### Thymeleaf 整合 Spring Web MVC

- ViewResolver : ThymeleafViewResolver
- View : ThymeleafView
- ITemplateEngine : SpringTemplateEngine

### 交互流程



### 示例：多视图处理器并存

- 视图处理器
  - ThymeleafViewResolver
  - InternalResourceViewResolver
- 目的
  - 理解 ViewResolver Order

- 理解 `ViewResolver` 模板资源查找
- 自定义 `ViewResolver` `Order`

`InternalResourceViewResolver` -> forward -> `JspServlet`

E:\temp\tomcat-docbase.4244314024417281980.8080

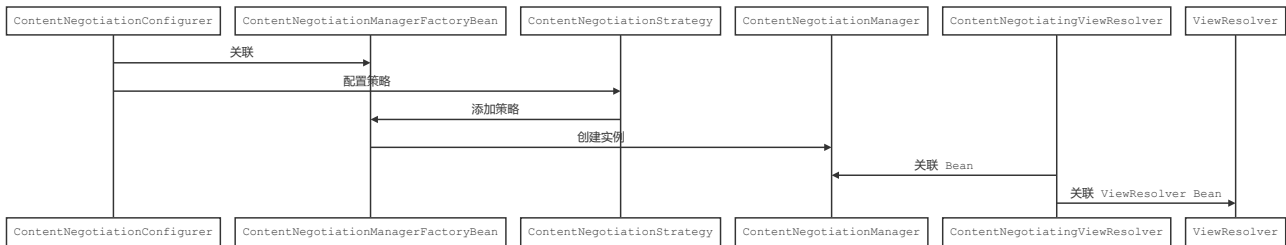
E:\temp\tomcat-docbase.4244314024417281980.8080

# 视图内容协商

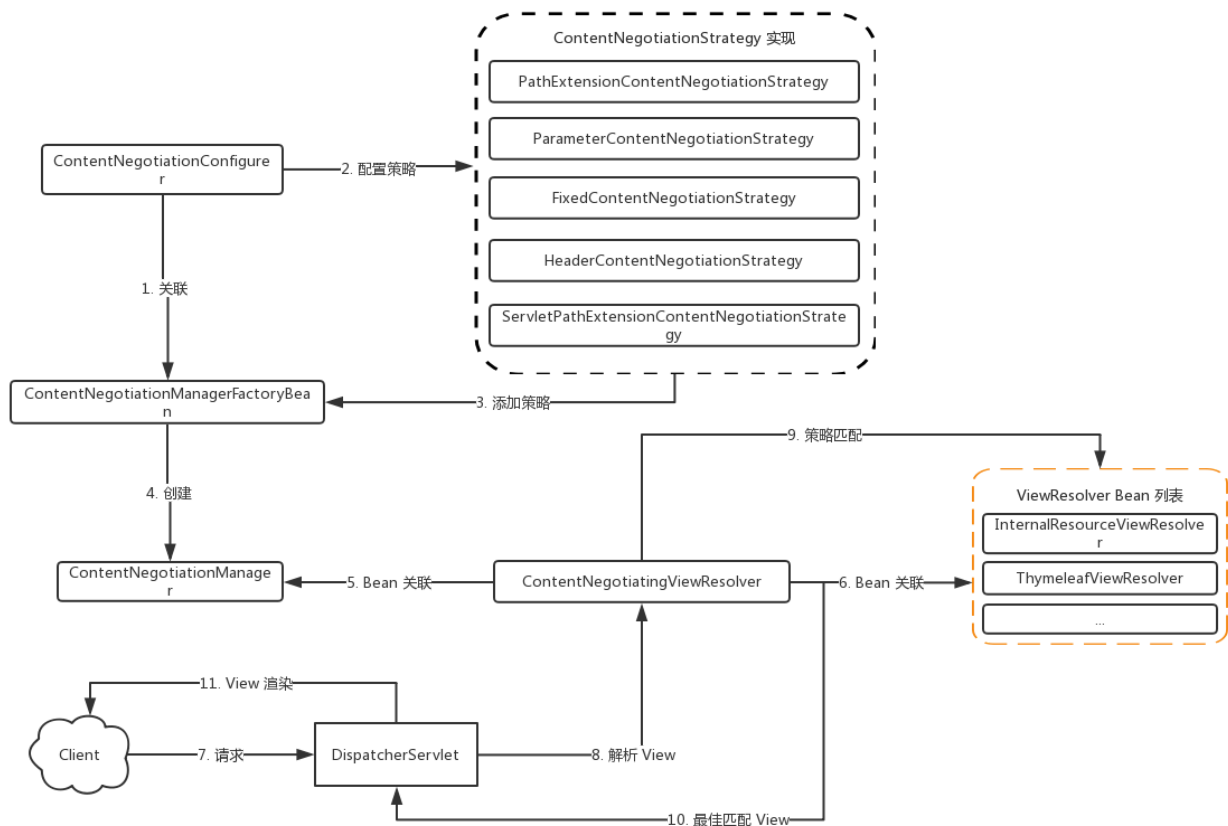
## 核心组件

- 视图解析
  - `ContentNegotiatingViewResolver`
    - `InternalResourceViewResolver`
    - `BeanNameViewResolver`
    - `ThymeleafViewResolver`
- 配置策略
  - 配置 Bean : `WebMvcConfigurer`
  - 配置对象 : `ContentNegotiationConfigurer`
- 策略管理
  - Bean : `ContentNegotiationManager`
  - `FactoryBean` : `ContentNegotiationManagerFactoryBean`
- 策略实现
  - `ContentNegotiationStrategy`
    - 固定 `MediaType` : `FixedContentNegotiationStrategy`
    - "Accept" 请求头 : `HeaderContentNegotiationStrategy`
    - 请求参数 : `ParameterContentNegotiationStrategy`
    - 路径扩展名 : `PathExtensionContentNegotiationStrategy`

## 序列图



## 交互图



## 示例：多视图处理器内容协商

- 视图处理器协商

- ContentNegotiatingViewResolver**

- BeanNameViewResolver**
    - InternalResourceViewResolver** Content-Type : text/xml;charset=UTF-8
    - ThymeleafViewResolver** Content-Type : text/html;charset=UTF-8

- 目的

- 理解 **BeanNameViewResolver**
  - 理解 HTTP Accept 请求头 与 **View** Content-Type 匹配
  - 理解最佳 **View** 匹配规则
    - ViewResolver** 优先规则
      - 自定义 **InternalResourceViewResolver**
      - ThymeleafViewResolver**
      - 默认 **InternalResourceViewResolver**
    - MediaType** 匹配规则
      - Accept 头策略
      - 请求参数策略

Accept	text/html,application/xhtml+xml,application/xml;q=0.9,;q=0.8
1	text/html
2	application/xhtml+xml
3	application/xml
4	<div>*/*</div>

# 视图组件自动装配

## 自动装配 Bean

### 视图处理器

- InternalResourceViewResolver
- BeanNameViewResolver
- ContentNegotiatingViewResolver
- ViewResolverComposite
- ThymeleafViewResolver ( Thymeleaf 可用)

### 内容协商

- ContentNegotiationManager

### 外部化配置

- WebMvcProperties
- WebMvcProperties.Contentnegotiation
- WebMvcProperties.View