

Spring学习笔记

基本概念

- | | |
|--|------------|
| 轻量级、非侵入式 | 对现有的类库没有影响 |
| 可以建立复杂逻辑、如事务管理、WS等 | |
| AOP的良好支持，方便进行功能增强，使得业务系统松耦合 | |
| 对主流的框架提供了很好的集成支持，如hibernate、Struts、JPA等，提供一个统一入口，把一些好的框架放在一起方便使用 | |
| 使用Spring的IOC容器，将对象之间的依赖关系交给Spring，降低低层之间的耦合性，让我们更关注于业务 | |
| Spring DI机制降低了业务对象依赖的复杂性，可以自由选择Spring部分或全部 | |
| Spring的框架可以开放性，并不强制依赖于Spring，可以自由选择Spring部分或全部 | |
| 缺少一个公用控制器 | |
| 没有SpringBoot了 | |
| Spring像一个胶水，而框架聚在一起，后期再拆分就容易拆分了 | |

4

- [illegible]

- | | |
|---|---|
| <p>方式</p> <p>4. 配置方式</p> <p>通过 <code>java</code> 类配置</p> <p>通过 <code>Configuration</code> 类创建 <code>Bean</code> 这两个注解实现的</p> <p>核心功能就是功能功能能分离运行开发，然后把功能功能和核心业务功能“捆”在一起，这样 <code>EAPC</code> 上运行代码与业务代码分离</p> <p>可以运行管理功能开发，开发业务功能都有业务代码实现，再在运行管理功能上通过 <code>Bean</code> 注入功能开发</p> <p>功能管理：日志、事务管理、权限控制</p> | <p>（<code>Configuration</code> 作用于此上，相当于一个 <code>xml</code> 配置文件）</p> <p>（<code>Bean</code> 作用于此方法上，相当于 <code>xml</code> 配置里的 <code>beans</code>）</p> |
|---|---|

事务管理

- [illegible]

扫码回复【脑图】
下载120张超清晰脑图高清文件



ApplicationContext

- | Filesystem\Xml\ApplicationContext | App |
|-----------------------------------|-----|
| ClassPath\Xml\ApplicationContext | |
| Web\Xml\ApplicationContext | |
-
1. 用户发送请求至前端控制器DispatchServlet
 DispatcherServlet会检查是否使用HandlerMapping处理逻辑。
 控制器则根据请求url找到对应的处理逻辑，生成对
 象及与处理逻辑相关的数据并返回视图——并关闭该
 DispatcherServlet。
 DispatcherServlet通过HandlerAdapter处理逻辑并
 返回给浏览器。
 2. 我们使用Controller，也就是控制器部分。
 4. Controller执行完成返回ModelAndView
 7. HandlerAdapter返回Handler执行结果
 ModelAndViewAndView返回给DispatcherServlet
 DispatcherServlet返回ModelAndView给浏
 览器View来解释和展示
 9. View返回数据并返回给浏览器
 DispatcherServlet将View进行渲染逻辑（将模型数
 据填充为页面内容）
 10. DispatcherServlet返回用户

SpringMVC

-
- ```

graph LR
 A[1. 加载和实例化] --- B[]
 B --- C[2. 初始化]
 C --- D[3. 请求处理]
 B --- E[get]

```

- 
- ```

graph LR
    A[1. 加载和实例化] --- B[ ]
    B --- C[2. 初始化]
    C --- D[ ]
    D --- E[3. 请求处理]
    B --- F[ ]
    F --- G[get]
  
```

- [illegible]

IOC

- 由于这个是在Bean初始化结束的时候调用的方法，也可以

知识	
----	--

- [illegible]

[illegible]

- [illegible]

流程

- [illegible]