

Spring+Mybatis快速开发

使用SM+Servlet开发人员信息管理系统

课程结构

- 课程介绍

- 案例展示

- 项目搭建

- 功能实现

- 课程总结

课程介绍

- **案例**

- 人员信息管理系统

- **前置条件**

- Mybatis、Spring、Servlet
- Mysql

- **主要技术**

- Servlet、Jsp
- Spring IOC、Spring AOP
- Mybatis
- Mybatis+Spring整合



案例展示



开发环境

- 操作系统

- Win7

- Web容器

- Tomcat-8.5

- 数据库

- Mysql-5.7

- 开发工具

- IntelliJ IDEA

相关资料

- **Spring**
– 4.0.2.RELEASE

- **Mybatis**
– 3.4.4

- **文档**
 - 配置文件模板
 - 页面原型

项目结构

- **三层架构**

- 持久层--Mybatis
- 表现层--Servlet+Jsp
- Spring--管理对象、切面处理

- **基于MVC模式**

- 视图--Jsp
- 模型--JavaBean
- 控制器--Servlet+JavaBean

关于配置与注解

- **配置**

- 非Java代码
- 便于后期维护

- **注解**

- 灵活、精简

- **应用场合**

- 需要变动的信息
- 一处配置与多处注解
- 代码精简
- 注解需要复杂的非Java代码

数据库设计

部门		
编号	int(0)	<pk>
名称	varchar(20)	
位置	varchar(100)	

日志	
操作时间	datetime
类型	varchar(10)
操作人	varchar(20)
模块	varchar(20)
操作	varchar(20)
结果	varchar(100)

员工		
编号	int	<pk>
账户	varchar(20)	
密码	varchar(20)	
状态	varchar(20)	
所属部门	int	<fk>
姓名	varchar(20)	
性别	varchar(20)	
身份证号码	char(18)	
入职时间	datetime	
离职时间	datetime	
出生日期	date	
备注	varchar(200)	

创建项目

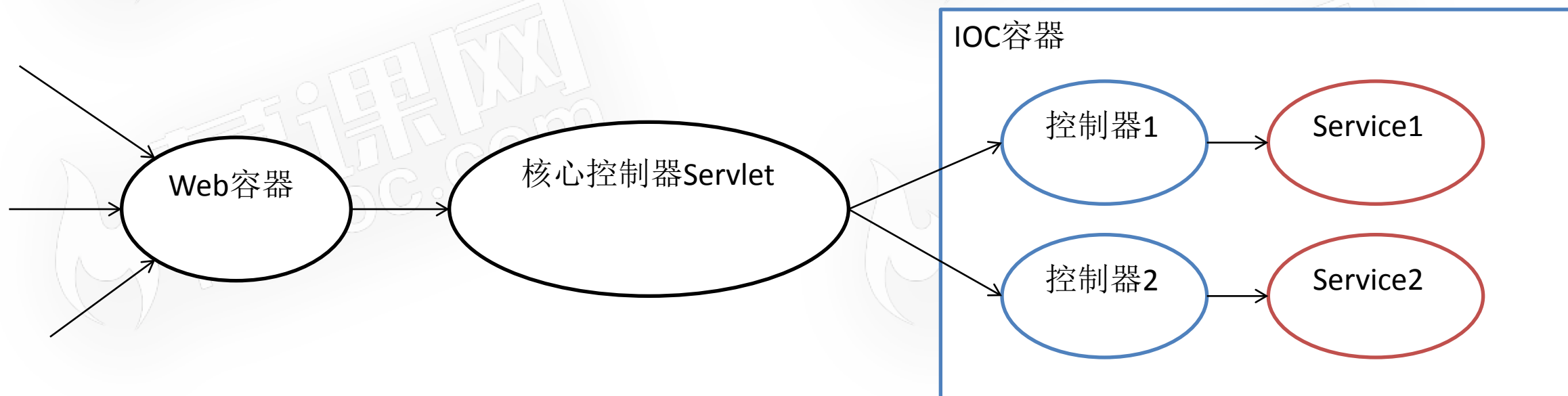
- **sm**
 - 父module
 - 全局定义与组织
- **sm_service**
 - 持久层、业务层
 - Mybatis依赖、Spring依赖
- **sm_web**
 - 表现层
 - Servlet依赖

Spring全局配置

- **Mybatis自动映射**
 - dataSource
 - sqlSessionSessionFactory
 - MapperScannerConfigurer
- **声明式事务**
 - transactionManager
 - txAdvice
- **全局扫描**

工具类

- 编码过滤器
- 核心控制器
 - Servlet对象由Web容器管理、Service对象由IOC容器管理



功能实现

- 部门管理

- 员工管理

- 登陆及个人中心

- 日志处理

部门管理

- 部门信息的增删改查
- 步骤
 - 实体类
 - Dao接口与Sql映射文件
 - Service接口与其实现类
 - 控制器
 - 页面

员工管理

- 员工信息的增删改查
- 关注点
 - 员工与部门的关联关系
 - 默认登陆密码

登陆与个人中心

- 登陆、退出、个人信息、修改密码
- 关注点
 - Session操作
 - 登陆过滤器

日志处理

- 日志查看、自动记录
- 关注点
 - 业务功能设计
 - 通知处理

课程总结

- **项目结构**

- 按层划分
- 按模块划分

- **CURD**

- 持久化操作
- 业务功能设计
- 页面安排

- **技术点**

- Mybatis自动映射
- Spring AOP应用场合
- Spring整合Mybatis
- 核心控制器意义