# 页面风格改版

首页更改背景图片,以及后台更改主题色彩

学习到freeMarker 的一些内容

Freemarker与jsp的区别和联系

# 预约记录统计

分享功能模块任务:预约记录统计

学习sumif(“a==b”,1,0) 从而将行转为多列

# 了解停划算系统开发流程

预约功能流程图

# 学习springcloud

框架结构图

# 将springCloud Demo运行起来

使用eurka和fegin 实现一个简单的demo,理解其流程

# 将预约服务添加到demo中

# 增加一个订单服务，并与预约服务进行交互

原计划是使用httpclient来调用另外一个服务

新方法 服务的注入?

# 项目中config问题

为什么引入config 服务

1. 每一个服务都有配置文件,配置麻烦,有一些共同的内容
2. 服务运行中配置文件修改需要动服务器上的代码

解决方法:

将配置文件的读取也作为一种服务

配置文件默认是放在git上的,服务启动的时候从git上读取,git文件修改后,config服务会通知子服务去修改配置

注意点:配置文件的命名方式有一定的约定条件

# 项目中权限问题

项目中使用的是session中存储的parkid,存放位置redis,其他的解决方案是什么?

项目中的权限是怎么做的?

# 项目中事务问题

事务：DBMS当中用户程序的一次执行，事务是bdms执行的基本单元

一个服务中的两个事务的问题

一个服务中调用另外一个服务的问题—分布式事务

## 四个特性:

**原子性**:要么做，要么不做，支持回滚操作，

实现方式：1.在数据快照上进行，并不修改实际的数据，如果有错并不会提交

2.直接操作实际数据，先预演一遍所有要执行的操作，如果失败则这些操作不会被执行，可以基于日志操作

**隔离性**:并发事务之间相互影响的程度，例如一个事务是否可以读到b事务未提交的数据

**问题**：

脏读：读未提交的数据，如果事务失败，就是脏读了

不可重复读：同一个事务中，对于同一份数据读取到的结果不一致；原因，并发修改记录，读的时候并不会把记录锁掉，解决方法：1.对修改的记录加锁2.mvcc

幻读：同一个事务中，同一个查询读取到的结果不一致（统计）；原因：并发事务增加记录，解决办法：事务串行化，才能避免幻读

第一类丢失：a事务撤销覆盖以及提交的B事务

第二类丢失：A事务覆盖B事务已经提交的数据

**级别**：

读未提交：可以读到另外一个事务未提交的结果

读提交（不可重复读）：只有事务提交后，其更新结果才能被其他事务看见，解决脏读问题

重复读：解决不可重复读的问题，锁或mvcc

解决方法：[数据库的悲观锁和乐观锁](#_数据库乐观锁与悲观锁)来避免问题

[Spring中事务](spring事务.docx)

问题：两个事务中读到的数据是不一致的，为什么？

两个事务中系统版本号是不一样的，操作的是不一致的行记录

如何一致性？

串行化：

**一致性**:系统从一个一致性状态到另外一个一致性状态

强一致性：读操作可以立即读到提交的更新操作

弱一致性：读不到刚提交的更新数据

会话一致性：

**持久性**:：事务提交后，对系统的影响是永久的

Start transaction 开启一项新的事务

Commit rollback 提交与回滚

Set autocommit 0 以后 的操作需要明确的命令进行提交或回滚

# 数据库乐观锁与悲观锁

悲观锁：显式的为数据资源加锁，问题：解决不可重复读问题，影响并发性能

方法：1.sql select .. for update

2.数据库中增加一个表明记录状态的LOCK字段，当取值为Y时，该记录被某个事务锁定

乐观锁：假定当前事务操作数据资源时，不会有其他事务同时访问该数据资源，

方法：应用程序采用版本控制手段（version）来避免可能出现的并发问题；

Inndb 中mvcc 是通过每行记录后面保存两个隐藏的列来实现的，一个是创建的系统版本号，一个是行的（过期/删除）版本号，在重复读和读提交隔离级别下工作

Insert:新插入的行 保存当前系统版本号作为行版本号

Delete:系统版本号作为行删除标示

Update:插入新的一条记录，系统版本号为行版本号，系统版本号带原来行作为行删除标示

Select:系统版本号在行系统版本号和删除版本号之间；