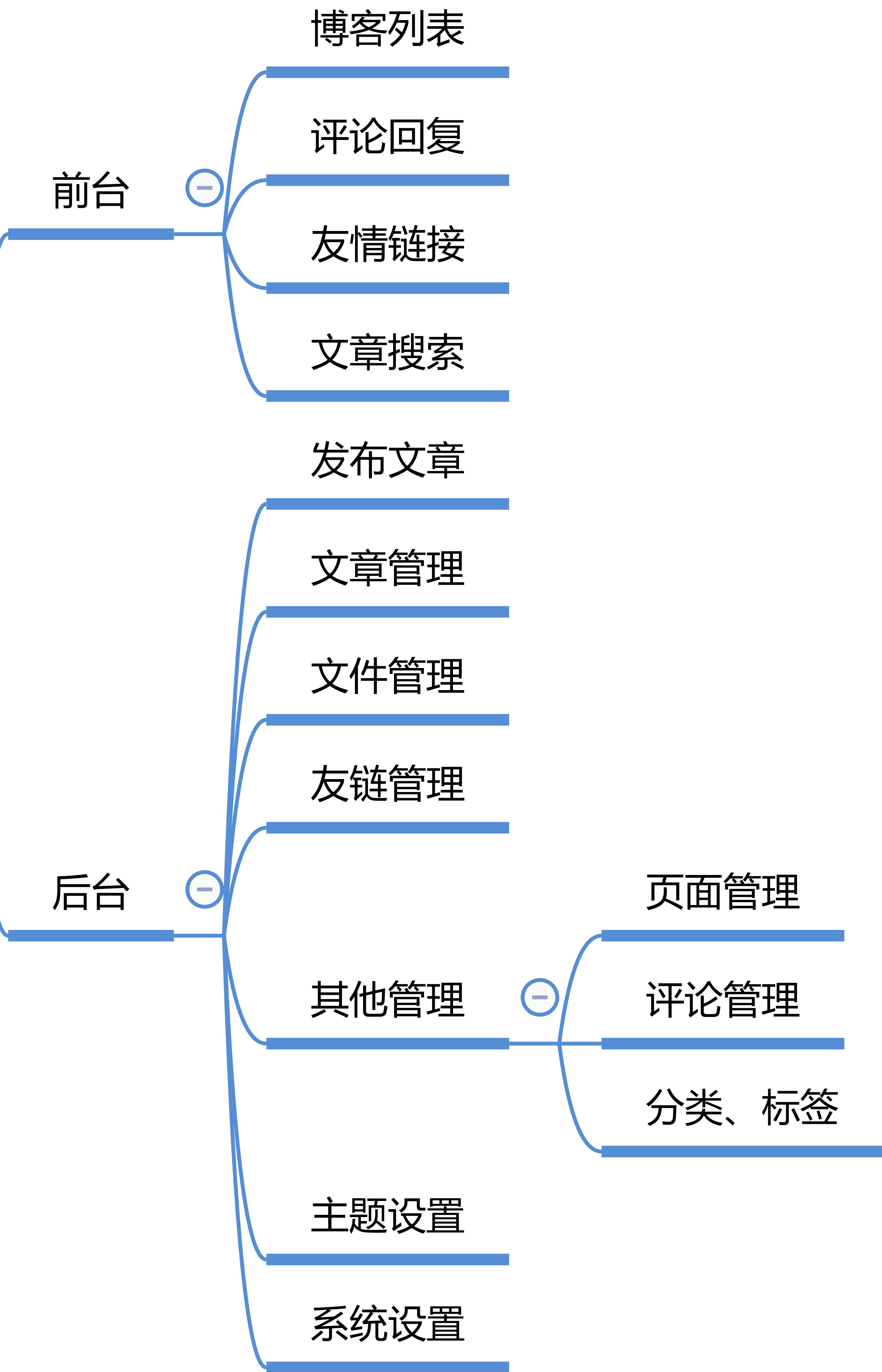


# 😊 1、tale项目简介

📖 微信公众号java-mindmap

## 功能结构



## 你应该学会

- 1 嵌入式SQLite数据库的认识与基本使用
- 2 jetbrick-template页面模板思想，与主流的velocity，freemarker对比
- 3 认识blade mvc框架，对比spring mvc

## 😊 2、技术介绍

📖 微信公众号java-mindmap

### 框架

blade mvc

blade 是一个轻量级，可快速开发的web框架，它内置了IOC管理，拦截器配置，REST API开发等众多主流web特性，集成了模板引擎，缓存插件，数据库操作，邮件发送等常用功能，简洁的源码值得你阅读和学习

### 数据库

SQLite3

是一款轻型的数据库，是遵守ACID的关系型数据库管理系统，它包含在一个相对小的C库中

它的设计目标是嵌入式的，而且目前已经在很多嵌入式产品中使用了它，它占用资源非常的低，在嵌入式设备中，可能只需要几百K的内存就够了

它能够支持Windows/Linux/Unix等等主流的操作系统，同时能够跟很多程序语言相结合，比如 Tcl、C#、PHP、Java等，还有ODBC接口，同样比起Mysql、PostgreSQL这两款开源的世界著名数据库管理系统来讲，它的处理速度比他们都快。

### 页面

jetbrick-template

📖 <http://git.oschin...>

jetbrick-template 是一个新一代 Java 模板引擎，具有高性能和高扩展性。适合于动态 HTML 页面输出或者代码生成，可替代 JSP 页面或者 Velocity 等模板。指令和 Velocity 相似，表达式和 Java 保持一致，易学易用。

#### 特点

支持类似于 Velocity 的多种指令

支持静态编译

支持编译缓存

支持热加载

支持类型推导

支持泛型

支持可变参数方法调用

支持方法重载

支持类似于 Groovy 的方法扩展

支持函数扩展

支持自定义标签 #tag

支持宏定义 #macro

支持布局 Layout

Markdown

Markdown 是一种用来写作的轻量级「标记语言」，它用简洁的语法代替排版，而不像一般我们用的字处理软件 Word 或 Pages 有大量的排版、字体设置。它使我们专心于码字，用「标记」语法，来代替常见的排版格式。

# 😊 3、项目结构

📖 微信公众号java-mindmap






## 😊 4、模板标签与函数

📖 微信公众号java-mindmap

### 标签

for循环 

```
#for(post : articles.list)
$ { post.title }
#end
```

```
#if(post.status == 'publish')
```

```
#elseif(post.status == 'draft')
```

if判断 

```
#end
```

宏定义 

```
#macro pageAdminNav(pageInfo)
```

```
#end
```

引用宏 

```
#call pageAdminNav(attachs)
```

模板引用 

```
#include(' ./header.html', {active:'attach', title:'文件管理'})
```

花括号里面的是json格式参数

变量赋值 

```
#set( comments = comments(6) )
```

前面的comments为变量名，后面的comments扩展函数，参数为6

### 函数

是否为空 

```
is_empty(attachs)
```

返回网站链接下的全址 


```
site_url(String sub)
```

### 自定义函数

AdminCommons.java  后台常用的函数

Commons.java  公共使用的函数

Theme.java  主题定义好的函数，主要用于前台

★ 使用方法 

项目初始化已经加载的自定义扩展函数可以直接在页面模板中使用，和内置的函数使用方法相同使用`${}`来调用，比如`${site_url("/about")}`，就会调用Commons里面的`site_url (String)`方法，并把函数返回的值直接显示在页面模板上。

总结 

其实我觉得和Velocity、Freemarker这些模板技术差不多，可以对比一下哦。

使用模板引擎的技术作为视图的另一好处应该是可以静态化页面

# 😊 5、SQLite介绍

📖 微信公众号java-mindmap

## 概念

📖 htt...

SQLite是一个进程内的库，实现了自给自足的、无服务器的、零配置的、事务性的 SQL 数据库引擎。它是一个零配置的数据库，这意味着与其他数据库一样，您不需要在系统中配置。

就像其他数据库，SQLite 引擎不是一个独立的进程，可以按应用程序需求进行静态或动态连接。SQLite 直接访问其存储文件

## 优点

- 1 不需要一个单独的服务器进程或操作的系统（无服务器的）。
- 2 SQLite 不需要配置，这意味着不需要安装或管理。
- 3 一个完整的 SQLite 数据库是存储在一个单一的跨平台的磁盘文件。意味着数据导入导出备份恢复都是复制文件，维护难度为零
- 4 SQLite 是非常小的，是轻量级的，完全配置时小于 400KiB，省略可选功能配置时小于250KiB。
- 5 SQLite 是自给自足的，这意味着不需要任何外部的依赖。
- 6 读速度快，在数据量不是很大的情况下速度较快，更重要的是：省掉了一次数据库远程链接没有复杂的权限验证，打开就能操作
- 7 SQLite 支持 SQL92（SQL2）标准的大多数查询语言的功能。
- 8 SQLite 可在 UNIX（Linux, Mac OS-X, Android, iOS）和 Windows（Win32, WinCE, WinRT）中运行。

## 缺点

- 1 SQLITE不可储存过多的数据库，它的性能发挥最好只能在存放较小的数据量情况下。不要把它当做MYSQL甚至ORACLE来使用。它只是一个200K的数据库
- 2 在大数据量的情况下表现较差 但是中小站点一般情况下数据量不超过10万
- 3 写入较慢 默认配置下的 SQLite 的写入速度比MySQL慢了很多
- 4 分布式啥的就不用想了



## 😊 6、项目讲解

微信公众号java-mindmap

### 注解

- @Route ⊖ 相当于spring mvc的@RequestMapping
- @JSON ⊖ 相当于spring mvc的@ResponseBody
- @QueryParam ⊖ 相当于spring mvc的@RequestParam
- 其他注解也spring mvc基本相同用法

### 增删改查

所有的增删改查操作都基于com.blade.jdbc.ActiveRecord，所有大部分时间你都不需要些dao层了

方法使用测试ActiveRecordTest.java 

前台主题的连接基本集中在com.tale.controller.IndexController.java类中

### 事务操作

Blade mvc并没有提供事务的注解方式，所以只能在代码里面进行事务操作

```
Tx.begin();  
int c = 1/0;  
activeRecord.update(p1);  
Tx.commit();
```