

SE-Android

# SchooledInApp

服务端概要设计文档

冯伟赞，朱修羽

2017-7-14



## 1. 引言

### 1.1 背景

SchooledInApp 是一个以兴趣和问题驱动的知识咨询共享服务平台，该 App 由个人用户和工作室成员参与互动，从而让工作室可以为个人用户提供知识咨询和共享服务, 该应用采用 C/S 架构。

本篇文档是该应用的服务端的设计文档，主要介绍服务端的代码架构设计。

### 1.2 相关文档

## 2. 系统要求

### 1.1 操作系统要求

64 位 Linux，或 64 为 Windows

### 1.2 支撑软件要求

JAVA: jdk8

Web 容器: Apache Tomcat-8.0

数据库: MySQL5

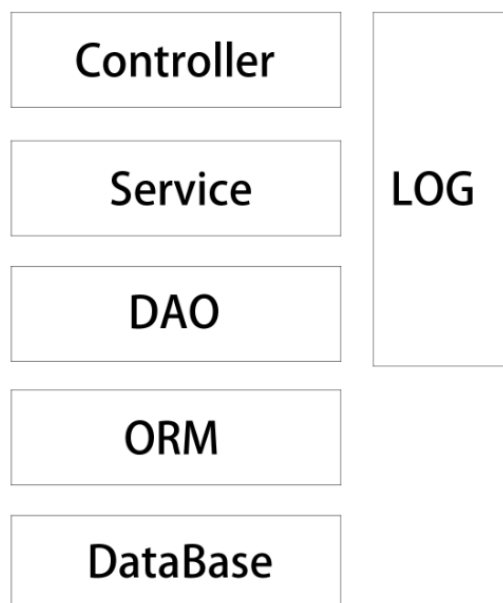
自动化构建工具: Gradle3

## 3. 系统设计

### 1.1 概述

服务端采用分层架构设计，把代码分为几个层次，实现各个层次的松耦合。

架构概要图如下：



**DataBase:** 使用 MySQL Server 作为数据库;

**ORM:** 使用 Hibernate5 把数据库中的关系表映射为 Java 实体类对象;

**DAO:** 对 Java 实体类进行增删改查操作;

**Service:** 包括对 Java 实体类的操作, 消息推送服务, 对某些数据进行倒排索引的建立, 更新和删除;

**Controller:** 使用 SpringMVC 框架接收请求, 并调用 Service 层的接口进行处理, 返回 Json 格式的请求体;

**Log:** 使用 Log4j 进行日志记录, 日志记录到文件;

## 1.2 数据库设计

实体与属性:

1. 用户:

| Field           | Type         | Null | Key | Default | Extra          |
|-----------------|--------------|------|-----|---------|----------------|
| id              | int(11)      | NO   | PRI | NULL    | auto_increment |
| age             | int(11)      | NO   |     | NULL    |                |
| answersNumber   | int(11)      | NO   |     | NULL    |                |
| avatar          | int(11)      | NO   |     | NULL    |                |
| bio             | varchar(255) | YES  |     | NULL    |                |
| department      | varchar(255) | YES  |     | NULL    |                |
| email           | varchar(255) | YES  |     | NULL    |                |
| location        | varchar(255) | YES  |     | NULL    |                |
| password        | varchar(255) | YES  |     | NULL    |                |
| phone           | varchar(255) | YES  |     | NULL    |                |
| questionsNumber | int(11)      | NO   |     | NULL    |                |
| sex             | varchar(255) | YES  |     | NULL    |                |
| username        | varchar(255) | YES  |     | NULL    |                |

2. 工作室:

| Field           | Type         | Null | Key | Default | Extra          |
|-----------------|--------------|------|-----|---------|----------------|
| id              | int(11)      | NO   | PRI | NULL    | auto_increment |
| answersNumber   | int(11)      | NO   |     | NULL    |                |
| avatar          | int(11)      | NO   |     | NULL    |                |
| bio             | varchar(255) | YES  |     | NULL    |                |
| essaysNumber    | int(11)      | NO   |     | NULL    |                |
| membersNumber   | int(11)      | NO   |     | NULL    |                |
| name            | varchar(255) | YES  |     | NULL    |                |
| questionsNumber | int(11)      | NO   |     | NULL    |                |
| topEssayId      | int(11)      | NO   |     | NULL    |                |
| topQuestionId   | int(11)      | NO   | MUL | NULL    |                |
| manager_id      | int(11)      | YES  |     | NULL    |                |

3. 问题

| Field          | Type         | Null | Key | Default | Extra          |
|----------------|--------------|------|-----|---------|----------------|
| id             | int(11)      | NO   | PRI | NULL    | auto_increment |
| answersNumber  | int(11)      | NO   |     | NULL    |                |
| content        | longtext     | NO   |     | NULL    |                |
| createdAt      | bigint(20)   | NO   |     | NULL    |                |
| heat           | int(11)      | NO   |     | NULL    |                |
| supportsNumber | int(11)      | NO   |     | NULL    |                |
| title          | varchar(255) | YES  |     | NULL    |                |
| type           | int(11)      | NO   |     | NULL    |                |
| directed_to_id | int(11)      | YES  | MUL | NULL    |                |
| asker_id       | int(11)      | YES  | MUL | NULL    |                |

4. 回答

| Field          | Type       | Null | Key | Default | Extra          |
|----------------|------------|------|-----|---------|----------------|
| id             | int(11)    | NO   | PRI | NULL    | auto_increment |
| commentsNumber | int(11)    | NO   |     | NULL    |                |
| content        | longtext   | NO   |     | NULL    |                |
| createdAt      | bigint(20) | NO   |     | NULL    |                |
| type           | int(11)    | NO   |     | NULL    |                |
| answerer_id    | int(11)    | YES  | MUL | NULL    |                |
| question_id    | int(11)    | YES  | MUL | NULL    |                |

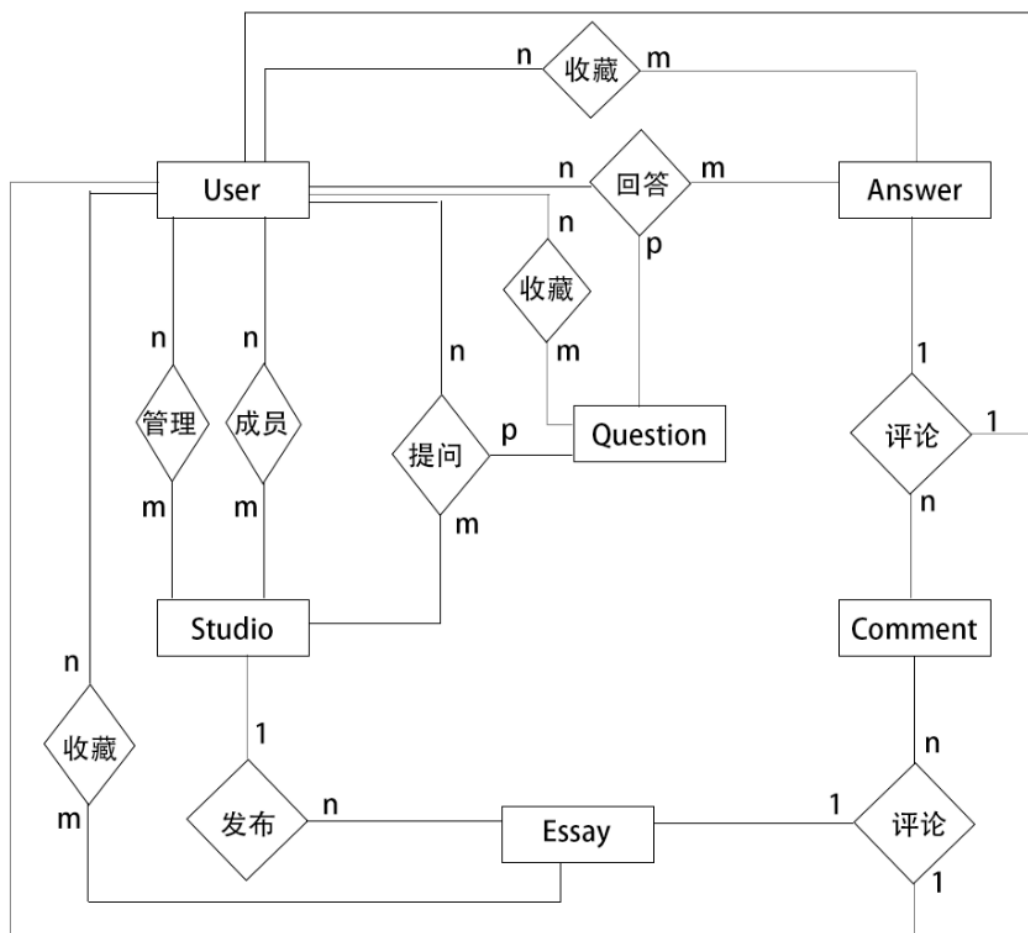
5. 文章

| Field            | Type         | Null | Key | Default | Extra          |
|------------------|--------------|------|-----|---------|----------------|
| id               | int(11)      | NO   | PRI | NULL    | auto_increment |
| commentsNumber   | int(11)      | NO   |     | NULL    |                |
| content          | longtext     | NO   |     | NULL    |                |
| createdAt        | bigint(20)   | NO   |     | NULL    |                |
| heat             | int(11)      | NO   |     | NULL    |                |
| supportsNumber   | int(11)      | NO   |     | NULL    |                |
| title            | varchar(255) | YES  |     | NULL    |                |
| type             | int(11)      | NO   |     | NULL    |                |
| author_studio_id | int(11)      | YES  | MUL | NULL    |                |

## 6. 评论

| Field             | Type         | Null | Key | Default | Extra          |
|-------------------|--------------|------|-----|---------|----------------|
| id                | int(11)      | NO   | PRI | NULL    | auto_increment |
| content           | varchar(255) | YES  |     | NULL    |                |
| createdAt         | bigint(20)   | NO   |     | NULL    |                |
| target            | int(11)      | NO   |     | NULL    |                |
| comment_answer_id | int(11)      | YES  | MUL | NULL    |                |
| comment_essay_id  | int(11)      | YES  | MUL | NULL    |                |
| commenter_id      | int(11)      | YES  | MUL | NULL    |                |

ER 图：（省略了属性的表示）



文件的存储:

文件存在操作系统的文件系统下，相关实体类的属性记录其文件路径;

比如 `essay.content`, `answer.content`, `question.content` 等，当它们为富文本或语音时，它们可能会包含文件路径;

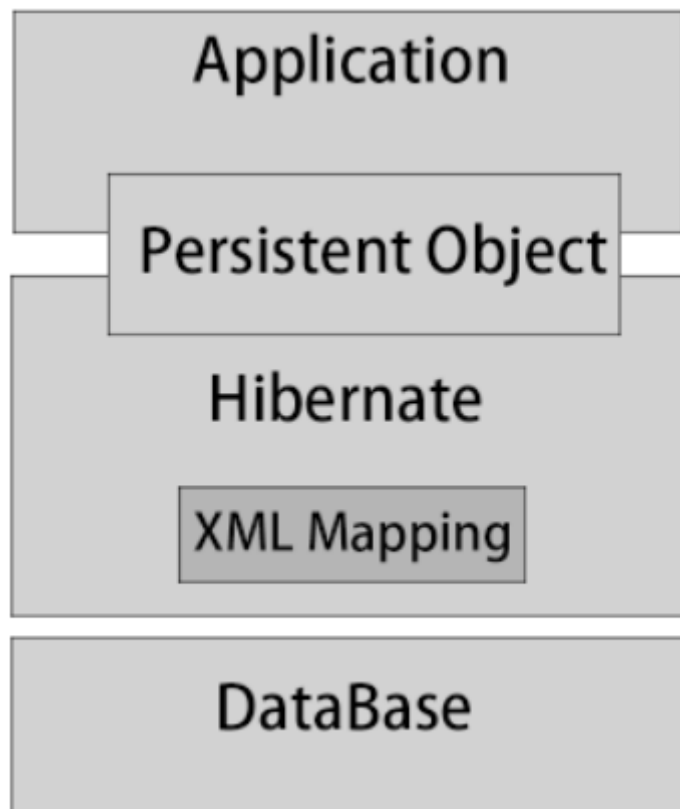
### 1.3 ORM 与 DAO

对象关系映射 (Object Relational Mapping), 是一种程序技术, 用于实现面向对象编程语言里不同类型系统的数据之间的转换。从效果上说, 它其实是创建了一个可在编程语言里使用的——“虚拟对象数据库”。

在服务端上, 使用 Hibernate 框架实现 ORM; Hibernate 不仅管理 Java 类到数据库表的映射 (包括从 Java 数据类型到 SQL 数据类型的映射), 还提供数据查询和获取数据的方法, 可以大幅度减少开发时人工使用 SQL 和 JDBC 处理数据的时间。这正是它的设计目标, 即将软件开发人员从大量相同的数据持久层相关编程工作中解放出来。

通过 Hibernate，服务端没有必要通过写 SQL 语句对数据库里的关系表进行操作，建表和表的约束操作由 Hibernate 自动完成，增删改查等操作通过对 Java 对象的操作来完成；

本服务端的 ORM-Hibernate 整体体系如下所示：



Hibernate 的配置文件在 `/src/main/resources/hibernate.cfg.xml`，该配置文件描述了 Hibernate 所需的配置以及对象关系的映射；

DAO (Data Access Object) 数据访问对象是一个面向对象的数据库接口，服务端编写了一系列 DAO 的 API 供上层使用，在 DAO 中使用了 Hiberante API 对数据库进行增删改查的操作。

## 1.4 Service

Service 层调用了 DAO 层的 API 对数据库进行增删改查的操作，另外在 Service 层进行了消息推送和倒排索引的建立和更改的操作。

### 1. 消息推送

消息推送使用了极光推送 (JPUSH)，JPush 是经过考验的大规模 APP 推送平台，每天推送消息数超过 5 亿条。开发者集成 SDK 后，可以通过调用 API 推送消息。同时，JPush 提供可视化的 web 端控制台发送通知，统计分析推送效果。JPush 全面支持 Android, iOS, Winphone 三大手机平台。JPush 提供四种消息形式：通知，自定义消息，富媒体和本地通知。本项目采用 Jpush 的自定义消息



当新增一条评论，问题，回答时，调用 JPush 的 API 对相应的用户进行推送，服务端推送的过程是：构造并发送请求至极光的服务器，再由极光根据请求对相应的用户进行推送；

## 2. 倒排索引

索引的建立，使用了 Lucene 框架，Lucene 是 apache 软件基金会 4 jakarta 项目组的一个子项目，是一个开放源代码的全文检索引擎工具包，但它不是一个完整的全文检索引擎，而是一个全文检索引擎的架构，提供了完整的查询引擎和索引引擎，部分文本分析引擎（英文与德文两种西方语言）。Lucene 的目的是为软件开发人员提供一个简单易用的工具包，以方便的在目标系统中实现全文检索的功能，或者是以此为基础建立起完整的全文检索引擎。

当新增、删除或修改一条用户，问题，工作室，文章时，调用 Lucene 的 API 对索引进行增加、删除、修改；

## 1.5 Controller

使用 SpringMVC 对请求进行处理，Spring 框架提供了构建 Web 应用程序的全功能 MVC 模块。Controller 里每个方法通过 @RequestMapping 注解对请求进行路由，通过 @RequestBody 注解对绑定请求体到相应的对象；在每个方法中，调用 Service 层的 API 进行业务的处理，并通过构造 ResponseEntity 对象，返回需求的 Json 格式响应体和 Http 状态码；

