# 登录注册逻辑

参考<https://juejin.cn/post/7114293013794422820>这篇文档

数据库用户创建以及权限设置

目的是开放所有权限给为数据库课设专门开设的账号

GRANT ALTER,ALTER ROUTINE,CREATE,CREATE ROUTINE,CREATE TEMPORARY TABLES,CREATE USER,CREATE VIEW,DELETE,DROP,EVENT,EXECUTE,GRANT OPTION,INDEX,INSERT,LOCK TABLES,PROCESS,REFERENCES,RELOAD,REPLICATION CLIENT,REPLICATION SLAVE,SELECT,SHOW DATABASES,SHOW VIEW,TRIGGER,UPDATE ON \*.\* TO 'LiHongJun'@'%'

创建用户表信息sql代码

CREATE TABLE `user` (

`id` int (11) NOT NULL,

`email` varchar (150) NULL DEFAULT '',

`password` varchar (150) CHARACTER SET `utf8mb4` COLLATE `utf8mb4\_0900\_ai\_ci` NULL DEFAULT '',

PRIMARY KEY (`id`)

) ENGINE = innodb DEFAULT CHARACTER SET = "utf8mb4" COLLATE = "utf8mb4\_0900\_ai\_ci"

可以通过sql变更预览去看

插入数据也可以sql语句查看

INSERT INTO `user` (`id`, `email`, `password`) VALUES (1, '123@123.net', '测试密码');

服务端开发遇到的问题和解决方式

解决 sqlalchemy报Textual SQL expression should be explicitly declared as text的错误 ——https://blog.csdn.net/vonhehe/article/details/95756567

sqlalchemy.exc.OperationalError: (pymysql.err.OperationalError) (1049, "Unknown database 'social'")问题解决

Exception raised for errors that are related to the database’s operation and not necessarily under the control of the programmer, e.g. an unexpected disconnect occurs, the data source name is not found, a transaction could not be processed, a memory allocation error occurred during processing, etc.

This error is a [DBAPI Error](https://docs.sqlalchemy.org/en/20/errors.html#error-dbapi) and originates from the database driver (DBAPI), not SQLAlchemy itself.

The OperationalError is the most common (but not the only) error class used by drivers in the context of the database connection being dropped, or not being able to connect to the database. For tips on how to deal with this, see the section [Dealing with Disconnects](https://docs.sqlalchemy.org/en/20/core/pooling.html#pool-disconnects).**与数据库操作相关且不一定在程序员控制之下的错误引发的异常，例如，意外断开连接、找不到数据源名称、无法处理事务、处理过程中出现内存分配错误等。 此错误是一个DBAPI错误，源自数据库驱动程序(DBAPI)，而不是SQLAlchemy本身。 OperationalError是驱动程序在数据库连接被删除或无法连接到数据库的上下文中使用的最常见(但不是唯一)的错误类。有关如何处理这种情况的提示，请参见处理断开连接的部分。**

**因此目前问题定位在云数据库上，最后定位到安全组忘了开了**

遇到问题TypeError: cannot convert dictionary update sequence element #0 to a sequence

问题核心在于db.session.execute(text(" select \* from user ")).fetchall()返回的数据是list类型，但是每一个元素都是sqlalchemy.engine.row.Row

接口文档

详见test文件

开发流程

1. 需求分析
2. 功能设计
3. 技术架构设计
4. Ui设计、数据库设计
5. 前端开发
6. 服务端设计接口文档
7. 。。。。

数据库相关创建过程

新建user\_info表

CREATE TABLE `user\_info` (

`user\_id` bigint (11) NOT NULL COMMENT '用户编号（手机号）',

`user\_name` char (20) NOT NULL DEFAULT '' COMMENT '用户昵称',

`user\_password` char (20) NOT NULL DEFAULT '' COMMENT '用户密码',

`user\_permission` tinyint (1) NOT NULL COMMENT '用户权限（普通用户0，管理员1）',

PRIMARY KEY (`user\_id`)

) ENGINE = innodb DEFAULT CHARACTER SET = "utf8mb4" COLLATE = "utf8mb4\_0900\_ai\_ci"

后端user\_info测试用例插入

INSERT INTO `user\_info` (`user\_id`, `user\_name`, `user\_password`, `user\_permission`) VALUES ('18031589519', '李虹均', '123', 1);

新建news\_info

CREATE TABLE `news\_bs4` (

`news\_id` int (0) NOT NULL COMMENT '新闻编号（通过uuid算法生成全局唯一ID，18位）',

`news\_title` varchar (200) NOT NULL DEFAULT '' COMMENT '新闻标题（使用 varchar 的原因是因为我在看新闻时发现看到的文章标题最长有70多个字，也就是140多个字节，但是有的就5个字，差别比较大，使用char浪费的空间较多）',

`news\_source` varchar (100) NOT NULL DEFAULT '' COMMENT '新闻来源url',

`news\_time` date NOT NULL COMMENT '新闻发布时间（由于有日报没有时分秒，因此就使用DATE）',

`news\_content` text NOT NULL COMMENT '新闻正文（以今日头条为例，有关政治的文章字数一般都超万字，因此需要TEXT）',

PRIMARY KEY (`news\_id`)

) ENGINE = innodb DEFAULT CHARACTER SET = "utf8mb4" COLLATE = "utf8mb4\_0900\_ai\_ci"

修改news\_info表结构

ALTER TABLE

`news\_bs4`

MODIFY

COLUMN `news\_source` char (20) CHARACTER SET `utf8mb4` COLLATE `utf8mb4\_0900\_ai\_ci` NOT NULL DEFAULT '' COMMENT '新闻来源的报刊或者网站名'

AFTER

`news\_title`,

ADD

COLUMN `news\_link` varchar (100) NOT NULL DEFAULT '' COMMENT '此新闻真实的url'

问题

sqlalchemy.exc.DataError: (pymysql.err.DataError) (1406, "Data too long for column 'news\_url' at row 1")

位置不够，需要扩大

由于id生成算法的重写，因此更改news\_id的长度为固定的8

ALTER TABLE

`news\_bs4`

MODIFY

COLUMN `news\_id` char (8) CHARACTER SET `utf8mb4` COLLATE `utf8mb4\_0900\_ai\_ci` NOT NULL COMMENT '新闻编号（通过uuid算法生成全局唯一ID，18位）' FIRST

新建news\_classification表

CREATE TABLE `news\_classification` (

`classification\_id` tinyint NOT NULL COMMENT '新闻分类编号（一共只有26种粗粒度分类）',

`news\_classification` char (20) NOT NULL DEFAULT '' COMMENT '新闻分类（通过新闻内容对新闻进行分析从而分类）',

PRIMARY KEY (`classification\_id`)

) ENGINE = innodb DEFAULT CHARACTER SET = "utf8mb4" COLLATE = "utf8mb4\_0900\_ai\_ci"

新建news\_source表

CREATE TABLE `news\_source` (

`news\_source\_name` char (20) NOT NULL DEFAULT '' COMMENT '新闻来源的报刊或者网站名',

`news\_source\_link` varchar (300) NOT NULL DEFAULT '' COMMENT '这个报刊主页的新闻链接',

PRIMARY KEY (`news\_source\_name`)

) ENGINE = innodb DEFAULT CHARACTER SET = "utf8mb4" COLLATE = "utf8mb4\_0900\_ai\_ci"

新建keyWord\_and\_news表

CREATE TABLE `keyWord\_and\_news` (

`news\_id` char (8) NOT NULL DEFAULT '' COMMENT '新闻编号（通过我的算法生成全局唯一ID）',

`keyWord\_id` int NOT NULL COMMENT '关键词编号（每次有新的新闻，就删除上一次的，编号也从0开始进行）'

) ENGINE = innodb DEFAULT CHARACTER SET = "utf8mb4" COLLATE = "utf8mb4\_0900\_ai\_ci"

News\_classification根据分类的总情况填写

INSERT INTO `news\_classification` (`classification\_id`, `news\_classification`) VALUES (1, '国际');

INSERT INTO `news\_classification` (`classification\_id`, `news\_classification`) VALUES (2, '体育');

INSERT INTO `news\_classification` (`classification\_id`, `news\_classification`) VALUES (3, '娱乐');

INSERT INTO `news\_classification` (`classification\_id`, `news\_classification`) VALUES (4, '社会');

INSERT INTO `news\_classification` (`classification\_id`, `news\_classification`) VALUES (5, '财经');

INSERT INTO `news\_classification` (`classification\_id`, `news\_classification`) VALUES (6, '时事');

INSERT INTO `news\_classification` (`classification\_id`, `news\_classification`) VALUES (7, '科技');

INSERT INTO `news\_classification` (`classification\_id`, `news\_classification`) VALUES (8, '情感');

INSERT INTO `news\_classification` (`classification\_id`, `news\_classification`) VALUES (9, '汽车');

INSERT INTO `news\_classification` (`classification\_id`, `news\_classification`) VALUES (10, '教育');

INSERT INTO `news\_classification` (`classification\_id`, `news\_classification`) VALUES (11, '时尚');

INSERT INTO `news\_classification` (`classification\_id`, `news\_classification`) VALUES (12, '游戏');

INSERT INTO `news\_classification` (`classification\_id`, `news\_classification`) VALUES (13, '军事');

INSERT INTO `news\_classification` (`classification\_id`, `news\_classification`) VALUES (14, '旅游');

INSERT INTO `news\_classification` (`classification\_id`, `news\_classification`) VALUES (15, '美食');

INSERT INTO `news\_classification` (`classification\_id`, `news\_classification`) VALUES (16, '文化');

INSERT INTO `news\_classification` (`classification\_id`, `news\_classification`) VALUES (17, '健康养生');

INSERT INTO `news\_classification` (`classification\_id`, `news\_classification`) VALUES (18, '搞笑');

INSERT INTO `news\_classification` (`classification\_id`, `news\_classification`) VALUES (19, '家居');

INSERT INTO `news\_classification` (`classification\_id`, `news\_classification`) VALUES (20, '动漫');

INSERT INTO `news\_classification` (`classification\_id`, `news\_classification`) VALUES (21, '宠物');

INSERT INTO `news\_classification` (`classification\_id`, `news\_classification`) VALUES (22, '母婴育儿');

INSERT INTO `news\_classification` (`classification\_id`, `news\_classification`) VALUES (23, '星座运势');

INSERT INTO `news\_classification` (`classification\_id`, `news\_classification`) VALUES (24, '历史');

INSERT INTO `news\_classification` (`classification\_id`, `news\_classification`) VALUES (25, '音乐');

INSERT INTO `news\_classification` (`classification\_id`, `news\_classification`) VALUES (26, '综合');