## 数据库后台开发指引

### 一、数据库访问

host: 192.168.3.188

port: 33306

account: developer

password: 3rhva5

database: ai\_rom

该用户在ai\_rom库下的表,具有SELECT ,INSERT,UPDATE,CREATE ,ALTER的权限

### 二、数据表结构

USE ai\_rom;

#### 用户信息表

CREATE TABLE IF NOT EXISTS `user\_info`(

`user\_id` INT UNSIGNED AUTO\_INCREMENT,

`user\_account` VARCHAR(24) NOT NULL,

`user\_pwd` VARCHAR(100) NOT NULL,

`user\_class` INT NOT NULL,

`register\_time` DATE,

PRIMARY KEY ( `user\_id` ),

UNIQUE(`user\_account`)

)ENGINE=InnoDB DEFAULT CHARSET=utf8;

这张表用于存放用户信息，主要在用户注册和登录验证时使用。

#### 用户操作日志表

CREATE TABLE IF NOT EXISTS `user\_operation\_log`(

`operation\_id` INT UNSIGNED AUTO\_INCREMENT,

`user\_id` INT NOT NULL,

`module\_id` INT NOT NULL,

`task\_id` INT NOT NULL,

`operate\_time` DATETIME,

PRIMARY KEY ( `operation\_id` )

)ENGINE=InnoDB DEFAULT CHARSET=utf8;

这张表用于存放用户在不同模块时进行操作的记录，主要记录的模块见下表。

#### 模块信息表

CREATE TABLE IF NOT EXISTS `module\_info`(

`module\_id` INT UNSIGNED AUTO\_INCREMENT,

`module\_def` VARCHAR(24) NOT NULL,

PRIMARY KEY ( `module\_id` )

)ENGINE=InnoDB DEFAULT CHARSET=utf8;

insert into task\_status\_info (task\_status\_id,task\_status\_def) values (1,'sign');

insert into task\_status\_info (task\_status\_id,task\_status\_def) values (2,'change password');

insert into task\_status\_info (task\_status\_id,task\_status\_def) values (3,'view task result');

#### 任务信息表

CREATE TABLE IF NOT EXISTS `task\_info`(

`task\_id` INT UNSIGNED AUTO\_INCREMENT,

`user\_id` INT NOT NULL,

`task\_class\_id` INT NOT NULL,

`doctor\_id` INT ,

`create\_time` DATE,

`task\_status\_id` INT,

`tmp\_file\_name` VARCHAR(300),

PRIMARY KEY ( `task\_id` )

)ENGINE=InnoDB DEFAULT CHARSET=utf8;

这张表用于存放用户发起的任务信息，其中主键task\_id会作为后续一系列流程的唯一凭证，在用户成功创建任务后生成，不能成功创建的任务不会被记录。

#### 任务状态表

CREATE TABLE IF NOT EXISTS `task\_status\_info`(

`task\_status\_id` INT UNSIGNED AUTO\_INCREMENT,

`task\_status\_def` VARCHAR(24) NOT NULL,

PRIMARY KEY ( `task\_status\_id` )

)ENGINE=InnoDB DEFAULT CHARSET=utf8;

insert into task\_status\_info (task\_status\_id,task\_status\_def) values (1,'to be analyzed');

insert into task\_status\_info (task\_status\_id,task\_status\_def) values (2,'finished');

insert into task\_status\_info (task\_status\_id,task\_status\_def) values (3,'canceled');

#### 任务类型表

CREATE TABLE IF NOT EXISTS `task\_class\_info`(

`task\_class\_id` INT UNSIGNED AUTO\_INCREMENT,

`task\_class\_def` VARCHAR(24) NOT NULL,

`result\_table\_name` VARCHAR(36) NOT NULL,

PRIMARY KEY ( `task\_class\_id` )

)ENGINE=InnoDB DEFAULT CHARSET=utf8;

insert into task\_class\_info (task\_class\_id,task\_class\_def,result\_table\_name) values (1,'拇指桡侧外展测量','shoulder\_thumb\_result');

insert into task\_class\_info (task\_class\_id,task\_class\_def,result\_table\_name) values (2,'正面右侧肩关节外展测量','shoulder\_thumb\_result');

这张表中，result\_table\_name存放的是任务对应的结果记录表的名字，然后可以通过记录表的名字在相关表中进行结果记录的插入和查询。

#### 任务结果表

##### shoulder\_thumb\_result

CREATE TABLE IF NOT EXISTS `shoulder\_thumb\_result`(

`task\_id` INT NOT NULL,

`frame` INT NOT NULL,

`body\_part` INT NOT NULL,

`angle` float(6,2) NOT NULL,

`cross\_point` VARCHAR(100) NOT NULL,

`line1` VARCHAR(400) NOT NULL,

`line2` VARCHAR(400) NOT NULL

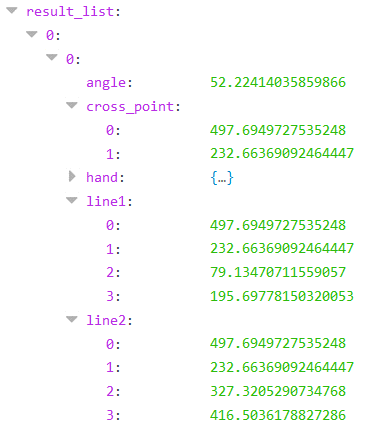
)ENGINE=InnoDB DEFAULT CHARSET=utf8;

这张表记录截取后的用户任务结果。

其中frame是指任务的图像序列(图像任务该项为0，视频任务对应的是帧的序列(也是从0开始)。

body\_part部位，只有一个部位的时候该项为0，多部位的时候根据结果的jason文件的顺序进行记录。

API返回的jason格式关键部分如下：



angle是角度，对jason文件的数值保留两位小数后记录到本表中。

cross\_point是交叉点坐标，对jason文件的数值保留两位小数后记录到本表中，将所有的values以“,”进行分隔之后拼成一个字符串记录在本表中。

line1和line2是交叉点延长直线坐标，对jason文件的数值保留两位小数后记录到本表中，将所有的values以“,”进行分隔之后拼成一个字符串记录在本表中。

### 主键values

需求上提到过一个系统设置页面，可以定义user\_id、operation\_id和task\_id的生成方式和起始范围。原则上要求他们的数据格式和范围要显著不一样，但由于这三个键在对应的表设计中，都已经设置为会自增的序列类型(AUTO\_INCREMENT)，能适应主键values自动生成的要求。不过在建表的时候我并没有明确指定每个AUTO\_INCREMENT的起始值，这点交由系统设置页面维护。前段发起指令之后，在数据库后端实际上执行的命令是：

ALTER TABLE table\_name AUTO\_INCREMENT = XXXX

同样地，在对表user\_info/user\_operation\_log /task\_info进行插入时，是不需要显式在代码中写出主键的value，如：

insert user\_info (user\_account,user\_pwd,user\_class,register\_time) values ('sys\_test',MD5('abc123'),0,'2020-08-13')

但是之后的一系列操作都需要获取当前的主键，尤其是task\_id，因为需要获取当前的task\_id来以此创建目录时，可以在insert之后，使用以下语句来获取：

SELECT `AUTO\_INCREMENT`

FROM INFORMATION\_SCHEMA.TABLES

WHERE TABLE\_SCHEMA = 'ai\_rom' AND TABLE\_NAME = 'task\_ info';