提供一个经典的sql拼装

$result = DB::*table*(**'advertisement\_relationship as atd'**)
 ->join(**'department as dt'**, **'atd.departmentid'**, **'='**, **'dt.id'**)
 ->select([**'atd.hospitalid'**, **'atd.global\_id'**, **'atd.departmentid'**, **'atd.id'**, **'atd.sort'**, **'dt.name'**])
 ->where(**function** ($query) **use** ($departmentid) {
 **if** ($departmentid) {
 $query->where([**'atd.departmentid'** => $departmentid]);
 }
 })
 ->whereIn(**'atd.hospitalid'**, *array\_column*($hospital, **'hospitalid'**))->groupBy([**'departmentid'**])
 ->get()->map(**function** ($value) {
 **return** (array)$value;
 })->toArray();

特别复杂的连表查询，就不使用关联模型查询了。

$result = DB::*table*(**'advertisement\_hospital\_department as ahd'**)
 ->join(**'department as dt'**, **'ahd.departmentid'**, **'='**, **'dt.id'**)
 ->join(**'advertisement\_hospital as ahl'**, **'ahd.hospitalid'**, **'='**, **'ahl.id'**)
 ->select(**'ahd.status'**, **'ahd.hospitalid'**, **'ahd.global\_id'**, **'ahd.departmentid'**, **'ahd.id'**, **'dt.name'**, **'ahl.hospital\_name'**)
 ->where([**'hospitalid'** => $hospitalid])->orderBy(**'ahd.gmt\_created'**, **'desc'**)->limit($pageSize)
 ->offset($offset)->get();

**class** ResourceModel **extends** Model
{
 */\*\** ***@var*** *string $table 数据表 \*/* **protected $table** = **'department\_resources'**;

 */\*\** ***@var*** *array 定义返回字段数据类型 \*/* **protected $casts** = [
 **'global\_id'** => **'string'**,
 **'department\_global\_id'** => **'string'** ];

 **const *CREATED\_AT*** = **'gmt\_created'**;
 **const *UPDATED\_AT*** = **'gmt\_modified'**;

 */\*\*
 \* add 新增数据
 \*
 \** ***@param*** *array $data
 \*
 \** ***@return*** *int
 \*
 \*/* **public function** add(**array** $data)
 {
 **return self**::*query*()->insertGetId($data);
 }

}

UserController.php 登陆

**public function** login(Request $request)
{
 $credentials = $request->only(**'username'**, **'password'**);
 $rules = [
 **'username'** => **'required'**,
 **'password'** => **'required'**,
 ];
 $messages = [
 **'username.required'** => **'用户名必须填写'**,
 **'password.required'** => **'密码必须填写'**,
 ];
 $validator = Validator::*make*($credentials, $rules, $messages);
 **if** ($validator->fails()) { *//失败生成错误信息* **throw new** ApiException($validator->errors()->first());
 }
 $token = auth(**'api'**)->attempt($credentials);
 **if** (!$token) {
 **return** response()->json([**'code'** => 1, **'message'** => **'用户名/密码有误，请重新输入'**], 401);
 }

 $tokenInfo[**'username'**] = $credentials[**'username'**];
 $tokenInfo[**'access\_token'**] = **'Bearer '** . $token;
 $tokenInfo[**'token\_type'**] = **'Bearer'**;
 $tokenInfo[**'expires\_in'**] = auth(**'api'**)->factory()->getTTL() \* 60;
 **throw new** ApiException(**'登录成功'**, 0, $tokenInfo);
}

**public function** logout()
{
 auth(**'api'**)->logout();
 **throw new** ApiException(**'登出成功'**, 0);
}

获取用户信息 通过id

**public function** detail(Request $request)
{
 $params = $request->all();
 $rules = [
 **'id'** => **'required|int'**,
 ];
 $messages = [
 **'id.required'** => **'用户Id必填'**,
 **'id.int'** => **'参数错误'**,
 ];
 $validator = Validator::*make*($params, $rules, $messages);
 **if** ($validator->fails()) {
 **throw new** ApiException($validator->errors()->first());
 }
 $columns = [**'global\_id'**, **'username'**, **'contact'**, **'mobile'**, **'status'**, **'gmt\_modified'**, **'gmt\_created'**];
 $user = UserModel::*query*()->where([**'global\_id'** => $params[**'id'**]])->select($columns)->first();
 **if** (**empty**($user)) {
 **throw new** ApiException(**'不存在的用户信息'**);
 }

 **throw new** ApiException(**'查询成功'**, 0, $user->toArray());
}