Atitit 接口查询问题与方法总结-----通用数据查询

目录

[1.1. 通用数据查询可以实现的目标 1](#_Toc13549)

[1.2. 总结的查询方案 1](#_Toc20743)

[2. 范例 2](#_Toc17877)

[3. 方式 3](#_Toc4359)

[3.1. 适当抽象一下，与具体业务脱钩，数据标识参数化配置化，避免硬编码 3](#_Toc25554)

[3.2. 如有可能优先使用动态对象 避免硬编码 3](#_Toc31422)

[3.3. 优先使用 动态化的模型json map 等，如有尽可能的避免使用硬编码模式域模型 do vo dto bo， 3](#_Toc17087)

[3.4. 尽可能避免在 代码里面出现表名 字段名等具体硬编码绑定。 3](#_Toc21603)

[4. 通用接口的实现 3](#_Toc29364)

[4.1. 核心查询 3](#_Toc11642)

[4.2. 4.2. 常用接口 聚合sum 4 4](#_Toc19885)

[4.3. 4.3. 常用接口 查询数据与翻页 5 4](#_Toc1648)

[5. Ref 5](#_Toc12094)

[5.1. 赛事资料表 5](#_Toc7683)

[5.2. WordPress 数据表作用介绍 5](#_Toc11992)

[5.3. WordPress 数据库结构及表字段作用解析 - 泪雪博客 6](#_Toc23258)

## 通用数据查询可以实现的目标

少量固定接口（3-5）个，即可以实现对无限业务数据查询，接口与具体业务不绑定，脱钩解耦

跨项目，接口通用与所有项目，跨语言实现（java php python js go c#等）

不只满足现在业务需求，而且尽可能满足未来需求

可以灵活指定接口传输字段数量 哪些要传输哪些不要，特别是包含大字段内容时对性能很有帮助，字段过多时也可以简化提升可读性

后端表 和字段结构变化不影响接口本身，无需调整接口，接口与表结构解耦脱钩

## 总结的查询方案

数据查询语言 GraphQL

Jpql  Java Persistence query language (JPQL

Spel ongl （不适合大量数据）

Json系列查询语言（jpath？？？

http param结构化转mybatis param

LINQ

Backendless架构

# 范例

查询足球赛事

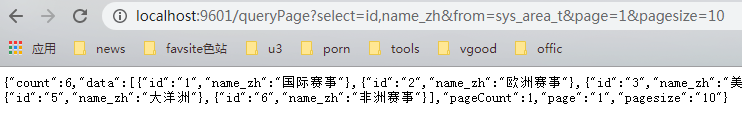
<http://localhost:9601/queryPage?from=football_event_t&page=1&pagesize=10>

篮球赛事（自定义字段

<http://localhost:9601/queryPage?select=id,name_zh&from=basketball_event_t&page=1&pagesize=10>

地区数据

[http://localhost:9601/queryPage?select=\*&from=sys\_area\_t](http://localhost:9601/queryPage?select=*&from=sys_area_t)



Cms wordpress文章

[http://localhost:9601/queryPage?from=wp\_posts](http://localhost:9601/queryPage?select=*&from=sys_area_t)

。。。。。

http接口字段select from where orderby page pagesize limit 等参数 基本免文档说明，和sql一样的概念

排序，翻页，自定义接口返回字段，条件查询等

# 方式

## 适当抽象一下，与具体业务脱钩，数据标识参数化配置化，避免硬编码

## 如有可能优先使用动态对象 避免硬编码

## 优先使用 动态化的模型json map 等，如有尽可能的避免使用硬编码模式域模型 do vo dto bo，

## 尽可能避免在 代码里面出现表名 字段名等具体硬编码绑定。

这样表与字段变化的话都不会有影响现有代码

# 通用接口的实现

## 核心查询

主要就是把req对象接口转换为map接口对象，传入mybatismapper。。。因为mapper不能直接接受req接口，mapper接受map接口作为动态对象

4.1. 核心查询 4

@Autowired

**public** HttpServletRequest req;

@Autowired

MybatisMapper MybatisMapper1;

// 核心接口

@GetMapping("/query")

**public** Object query() **throws** Exception {

Map m = RequestUtil.*getMap*(req);

**return** MybatisQueryUtil.*query*(m, MybatisMapper1);

}

**public** **static** Object query(Map m2, MybatisMapper MybatisMapper1) {

Map m = com.kok.sport.utils.MapUtil.*clone*(m2);

// 简化使用性ux，增加select和limit参数的默认模式，翻页转换page等

*uxEnhance*(m);

*logger*.info(m.toString());

// 安全性检查过滤

sqlutil.*checkSafeForQuery*(m);

// convert mybatis fmt fragment

*paddParam*(m);

List r = MybatisMapper1.query(m);

**return** r;

}

@Mapper

**public** **interface** MybatisMapper {

//

@Select("select ${select} from ${from} ${join} ${where} ${group} ${HAVIN} ${order} ${limit}")

List<LinkedHashMap> query(Map m);

## 4.2. 常用接口 聚合sum 4

## 4.3. 常用接口 查询数据与翻页 5

# Ref

## 赛事资料表

CREATE TABLE `football\_event\_t` (

`id` bigint(19) NOT NULL COMMENT '赛事id',

`season\_id` bigint(19) DEFAULT NULL COMMENT '赛季id',

`area\_id` bigint(19) DEFAULT NULL COMMENT '地区id',

`country\_id` bigint(19) DEFAULT NULL COMMENT '国家id',

`type` tinyint(3) DEFAULT NULL COMMENT '赛事类型 0-未知 1-联赛 2-杯赛 3-友谊赛',

`level` tinyint(3) DEFAULT NULL COMMENT '忽略，兼容使用',

`name\_zh` varchar(100) DEFAULT NULL COMMENT '中文',

`short\_name\_zh` varchar(100) DEFAULT NULL COMMENT '中文缩写',

`name\_zht` varchar(100) DEFAULT NULL COMMENT '粤语',

`short\_name\_zht` varchar(100) DEFAULT NULL COMMENT '粤语缩写',

`name\_en` varchar(100) DEFAULT NULL COMMENT '英文',

`short\_name\_en` varchar(100) DEFAULT NULL COMMENT '英文缩写',

`match\_logo` varchar(100) DEFAULT NULL COMMENT '赛事logo，前缀：http://cdn.sportnanoapi.com/football/competition/',

`delete\_flag` char(1) NOT NULL COMMENT '是否删除(1.已删除0.未删除)',

`create\_time` datetime NOT NULL COMMENT '创建时间',

PRIMARY KEY (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='足球赛事表';

## WordPress 数据表作用介绍

WordPress 所有的数据表

首先分别来先看看 WordPress 所有的数据表都是干什么用的吧，一下是 WordPress 完整的 12 张数据表，当然刚安装好的时候一般是只有 11 张表的，如果还有其它相关的数据库，那么可能就不是 WordPress 本身的数据库，有可能是某些主题或插件需要而创建的，所以一下肯定子凡要介绍的还是 WordPress 本身的数据库表和字段了。

1. wp\_commentmeta：存储 Akismet 或手工审核的评论是否为垃圾评论的判断结果；
2. wp\_comments：存储评论信息，如评论内容、评论所属文章、评论人昵称、邮箱、URL 等；
3. wp\_links：存储友情链接信息，如友链名称、URL、打开方式、描述、是否可见等；
4. wp\_options：存储 WordPress 系统默认及后台系统选项、插件及主题配置信息，包括网站标题、副标题、当前主题等等；
5. wp\_postmeta：存储文章的一些相关信息，如文章附件图片的 alt 信息、文章所在分类的 URL 以及文章自定义的自定义字段，其中可能就有文章访问次数等；
6. wp\_posts：存储文章信息，包括文章标题、正文、摘要、作者、发布时间、访问密码、评论数、修改时间、文章地址等；
7. wp\_termeta：存储对菜单分类的更多设置，属于开发性功能居多，例如分类目录的缩略图、颜色标识等；
8. wp\_terms：存储菜单分类、标签分类名称及 URL 信息；
9. wp\_term\_relationships：存储文章和分类、标签的相互对应关系；
10. wp\_term\_taxonomy：存储分类和标签的描述信息、父子关系、所属包含的文章数等；
11. wp\_usermeta：存储用户的姓名、昵称、权限等信息；
12. wp\_users：存储用户名、密码、昵称、邮箱、注册时间等信息；

## WordPress 数据库结构及表字段作用解析 - 泪雪博客