Atitit enhance dev eff read req提升开发效率 可读性规范

目录

[1. 提升效率的俩大原则 1](#_Toc32289)

[2. 命名规范 见名字知道意思 1](#_Toc26561)

[3. 层次结构缩减 单层 vs 双层 2](#_Toc8709)

[3.1. Static 函数 vs oop 2](#_Toc13633)

[3.2. 代码结构扁平化 项目 模块模式 2](#_Toc32553)

[4. 缩减项目文件数量与代码数量 2](#_Toc31116)

[4.1. 使用简洁模式 减少不必要的文件 2](#_Toc6463)

[4.2. 使用dsl 4gl、 sql 模式大力缩减代码 2](#_Toc2050)

[4.3. 代码业务抽象通用化 库表查询 操作 2](#_Toc8915)

[5. 不要使用重量级模式 3](#_Toc27392)

[5.1. Dp的实现 servicelocator模式也很好，由于Ioc实现模式 3](#_Toc7883)

[5.2. 使用易于理解的模式 3](#_Toc7142)

[5.3. Vue里面使用js模式，不要npm模式 3](#_Toc25291)

[6. Other 3](#_Toc31851)

[6.1. 尽可能使用默认模式 3](#_Toc10702)

[6.2. 使用gradle代替maven 3](#_Toc9041)

[7. 存在的问题 3](#_Toc8307)

[7.1. 不要使用webpacket打包模式麻烦 3](#_Toc24570)

[7.2. Springboot、 ioc的实先 3](#_Toc24786)

[7.3. 太多泛滥的接口 4](#_Toc13168)

[7.4. 层次过多哦 4](#_Toc16853)

[7.5. Gradle自定义过多复杂 4](#_Toc13018)

# 提升效率的俩大原则

Kiss原则简单原则

可读性原则

# 命名规范 见名字知道意思

大力使用中文提升可读性

# 层次结构缩减 单层 vs 双层

## Static 函数 vs oop

## 代码结构扁平化 项目 模块模式

不要树形嵌套过深。。一般项目双层足够

# 缩减项目文件数量与代码数量

## 使用简洁模式 减少不必要的文件

## 使用dsl 4gl、 sql 模式大力缩减代码

## 代码业务抽象通用化 库表查询 操作

# 不要使用重量级模式

## Dp的实现 servicelocator模式也很好，由于Ioc实现模式

## 使用易于理解的模式

## Vue里面使用js模式，不要npm模式

# Other

## 尽可能使用默认模式

## 使用gradle代替maven

# 存在的问题

## 不要使用webpacket打包模式麻烦

## Springboot、 ioc的实先

可以使用springboot的web rest接口。。但不要ioc。。使用简单的new serviceloctor模式就解决了dp问题

## 太多泛滥的接口

## 层次过多哦

## Gradle自定义过多复杂