**手游项目数据  
版本号：Ver1.0**

# 版本说明：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **版本号** | **日期** | **內容** | **作者** | **审核人** |
| **Ver1.0** | 2016-03-04 | 初稿 | 邓世伟 |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# 目录：

[版本说明： 2](#_Toc29787)

[目录： 3](#_Toc22218)

[1. 数据 4](#_Toc21653)

[1.1 解析后端返回数据 4](#_Toc31)

[1.2 数据缓存 4](#_Toc4658)

[1.3 消息通知 4](#_Toc21396)

[1.4 数据获取 4](#_Toc18316)

[2. 生成类工具 4](#_Toc9848)

[2.1 解析\*.md 4](#_Toc28587)

[2.2 解析\*.php 5](#_Toc24285)

[2.3 分析组装 5](#_Toc31448)

[2.4 生成类 5](#_Toc19552)

[3. 时间评估 5](#_Toc208)

[3.1 数据层(27h) 5](#_Toc23573)

[3.2 生成工具(40h) 5](#_Toc17784)

[3.3 开发顺序预估 6](#_Toc10472)

# 数据

## 解析后端返回数据

1. 采用JsonDotNet插件，反序列化解析数据。
2. 需要不同的结构类进行测试。
3. 目前JsonDotNet插件不支持internal的反序列化。

## 数据缓存

1. 采用dictionary的缓存池的方式来缓存。
2. 用表名来当key值，进行缓存。
3. 根据后端返回的msg内容，采用replace的方式
4. 根据后端返回update内容，采用add或update的方式
5. 根据后端返回delete内容，采用delelte的方式 (需要和后端确定)

## 消息通知

1. 采用delegate的方式来触发通知。
2. 提供注册及移除的方式。
3. 根据表内的内容有了修改或者添加或者删除进行派发消息通知。

## 数据获取

1. 根据表名获取数据，返回T
2. 根据表名及Predicatet条件的形式获取数据，返回T
3. 根据表名获取数据，返回List<T>
4. 根据表名及Predicatet条件的形式获取数据，返回List<T>
5. 根据表名获取数据，返回Dictionary<string,T>
6. 根据表名及Predicatet条件的形式获取数据，返回Dictionary<string,T>

# 生成类工具

## 解析\*.md

1. 获取文件夹中的所有.md格式的所有文件
2. 解析md格式的文件，对其进行分解
3. 缓存分解的所有表结构数据

## 解析\*.php

1. 获取orm.config.php的文件
2. 解析orm.config.php的文件
3. 缓存orm.config里面的数据用于查找对应的表名

## 分析组装

1. 对md格式的文件进行逐行分析及分类
2. 缓存分析的表及结构数据

## 生成类

1. 通过缓存的表结构数据进行查找对应的表名
2. 通过缓存的表结构数据进行解析
3. 根据orm.config对应的表结构生成结构体类
4. 采用JsonCSharp插件来生成，目前不支持dictionary

# 时间评估

## 数据层(27h)

1. 解析后端返回数据 （8h）
   1. 直接采用JsonDotNet的来做，排除internal的权限控制。
   2. 不用JsonDotNet插件来做，自己写话时间会比较长。
2. 数据缓存 （8h）
   1. Delete需要后端配合
3. 消息通知 （3h）
4. 数据获取 （8h）

## 生成工具(40h)

1. 解析\*.md （16h）
2. 解析\*.php (8h)
3. 分析组装 (8h)
4. 生成类(8h)
   1. 如果不用JsonCSharp插件的话，自己写时间比较长。

## 开发顺序预估

1. 数据层开发。
2. 生成工具开发。
3. 自己写反序列化插件
4. 自己写生json转换类工具