# 云平台-业务集成概要设计

## 第一章 引言

### 1.1 项目背景

在公司产品的功能开发过程中，许多功能需要调用第三方服务。如人脸识别功能中需要使用在线翻译服务，云相册功能中需要使用图片鉴黄服务等。综合各个产品组的需求，由公共数据平台负责开发，通过将第三方外部服务集成为内部业务，最终为公司项目组提供统一、稳定的服务接口。

### 1.2 编写目的

业务集成即集成第三方服务为内部业务。该项目的设计重点在于：动态配置接口参数、动态切换服务提供方、统一接口参数格式及返回值。本文档将围绕以上三点作概要阐述，旨在梳理流程逻辑，展现设计思路，以供大家观阅及指正。

## 第二章 总体设计

### 2.1 需求概述

#### 灵活性

接口参数可以分为两类：提供方参数、业务参数。提供方参数是指提供服务

的第三方特有的参数如：易源应用ID，或某些基本固定的参数如：URL地址。业务参数是指调用者可自定义参数值的查询参数，如：新闻关键字keyWord。灵活性要求两类参数的值可以在后台动态配置，接口内部第三方服务的调用顺序也可以进行实时切换。

#### 可靠性

同一个业务接口可以通过整合不同的第三方服务来实现，如在线翻译业务既可以使用有道翻译，也可以使用百度翻译或易源翻译来进行实现。当首选的第三方接口调用失败后，自动切换至优先级次之的接口，以保证持续提供可靠的服务。当首选的第三方接口恢复正常后，服务接口需要及时切回，继续以优先级最高且可用的第三方接口作为数据源。

#### 有效性

业务接口可以分为两类：即时更新类、延时更新类。即时更新要求在保证查询效率和吞吐量的前提下尽可能保证返回数据的实时性，如实时天气数据。延时更新不要求数据的实时性，只需定期更新即可，如新闻资讯数据。两类接口都要求一定程度上保证数据的有效性。

#### 可拓展性

可拓展性要求服务接口在不改变参数及返回值格式的前提下，可以拓展整合新的第三方接口。如：今日头条若开放新闻资讯接口，要保证新闻查询业务能够接入头条新闻接口，或整合头条新闻数据。

### 2.2 主要技术

后台服务以Dubbo接口或RESTful接口提供给外部调用。主要使用的技术如下：

* **开发框架：**SpringBoot、MyBatis-Plus、Dubbo、Zookeeper；
* **开发工具：**Spring Tool Suite；
* **数据库：**MySQL；
* **缓存技术：**Redis。

### 2.3 系统总体框架

* **数据存储层：**Mysql作为持久层保存数据，Redis作为中间层缓存数据。通过定期调用第三方接口获取最新数据存入持久层作为数据源。
* **第三方接口：**各种不同来源的第三方服务接口。
* **基础服务层：**所有集成第三方服务后形成的内部业务。业务接口既可以从数据存储层中获取数据，也可以直接调用第三方接口获取数据，将此数据封装之后返回给系统应用层。
* **系统应用层：**提供给用户直接使用的产品或功能服务。系统应用层通过Dubbo或RESTful接口调用基础服务层的服务。
* **客户访问层：**用户可通过机器人或手机等终端访问应用层的应用。



*图2-1 业务集成系统流程图*

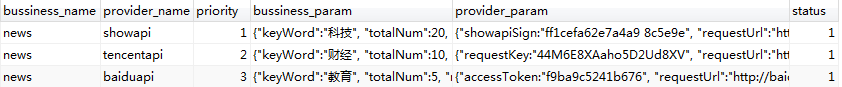
## 第三章 关键流程

### 3.1 实现动态配置参数

表3-1为业务参数表，存储基础服务层中接口的默认调用参数，以及为之提供数据的第三方接口的默认参数及调用信息。图3-2为参数表数据实例。

|  |  |  |
| --- | --- | --- |
| 字段名 | 注释 | 类型 |
| id | 主键id | int |
| bussiness\_name | 业务名称 | varchar（200） |
| provider\_name | 第三方服务提供方名称 | varchar（200） |
| priority | 优先级 | int（2） |
| bussiness\_param | 业务参数 | varchar（1000） |
| provider\_param | 提供方参数 | varchar（1000） |
| status | 状态 | int（1） |

*表3-1 接口参数表*



*图3-2 参数表数据实例*

**说明**

1. 业务名称：基础服务层中业务名称，如：在线翻译业务。
2. 提供方名称：第三方接口中接口名称，如：易源接口。
3. 优先级：调用第三方接口的顺序，值越小，越先调用。
4. 服务参数：用户通过自定义参数值可以获得期望的结果，如：查询条数。所有参数存为一条JSON数据，如：{"keyWord":"科技", "totalNum":20, "needContent":1, "needPic":0}
5. 提供方参数：第三方平台特有的参数或很少变动的参数，如：密钥。所有参数存为一条JSON数据。如：{"showapiSign:"ff1cefa62e7a4a9 8c5e9e", "requestUrl":"http://route.showapi.com/109-35"}
6. 状态：0为未启用，1为已启用。

服务参数和提供方参数都可以通过WEB页面修改参数的值。在程序代码中，通过从Redis动态读取参数值，以此实现动态配置接口参数。图3-3为：用户连续两次操作的时序图。

 *图3-3 用户修改参数并手动刷新页面的时序图*

对于业务参数，后台配置的参数仅是作为默认值。仅当调用者没有传入参数值时，才使用默认值。但对于提供方参数，仅可通过后台进行设置，不能由调用者传入。图3-4为调用业务接口流程图。



*图3-4 调用业务接口流程图*

### 3.2 实现自动切换数据源

#### ZSet简介

Redis有一种存储类型：有序集合ZSet。1个Key对应N个member，1个member对应1个唯一的score。

* **新增member**

zadd(String key, double score, String member)

* **删除member**

zrem(String key, String... members)

* **按score升序取出score介于start与end之间的member**

zrange(String key, long start, long end)

自动切换数据源即：自动切换第三方服务提供方，本质上是：自动切换调用业务接口时传入的providerParam。接下来以调用在线翻译业务为例，来阐述实现接口可靠性的设计思路。

#### 实现细节

1. 假设业务参数及提供方参数均已存入ZSet。bussiness\_name存为key，priority存为score，bussiness\_param和provider\_param存为member。一个第三方服务提供方对应一个member。暂且称之为：**可用资源池**。



1. 从**可用资源池**按score升序取出第一个member1，解析出showApiP1作为调用业务接口的默认参数值，将showApi作为数据源。



1. 业务内部，若调用showApi接口发生异常，需要进行两个操作。
2. 从**可用资源池**移除member1。



1. 将member1存入**等待资源池**。但score和member的值有变动。

**score:** currentTimeMillis + 600000(5min)

**member:** bussinessParam + providerParam + priority



1. 从**可用资源池**中重新取值获取优先级最高的member2，以tencentApi作为数据源。若此接口调用成功，则返回数据。若调用失败，则执行步骤3。
2. 服务内部有一个**定时任务**：每隔1分钟，从**等待资源池**中取出score小于currentTimeMillis的member，检测该接口是否恢复正常。
3. 若可用，则将该member从**等待资源池**中移除，并放回**可用资源池**；
4. 若仍不可用，则更新socore值为：score += 600000，约5分钟后定时程序会自动重新检测该接口可用性。

图3-5为自动切换数据源时序图。



*图3-5 自动切换数据源时序图*

### 3.3 实现策略更新数据源

随着时间推移，一些原本准确的数据可能会失效，如天气预报信息、WiFi定位信息。暂且认为第三方会不断更新及优化接口返回的数据，业务内部通过一定策略来获取并使用第三方接口数据，以此保证数据的有效性。

#### 即时更新类数据

即时更新类数据如：实时天气信息、股票行情信息，通过实时调用第三方接口获取数据并返回，返回数据短暂缓存于Redis。

*实时天气业务*

* **数据存储策略：**不存储于数据库；
* **数据缓存策略：**查询结果缓存于Redis，过期时间30s，以提高响应速度，控制调用第三方接口的频率；
* **第三方调用策略：**直接调用第三方接口，封装接口返回值为统一格式并返回。

#### 延时更新类数据

若从存储方式来看，延时更新数据又可分为两类：保存于数据库、不保存于数据库。不同业务的数据有不同的存储策略、缓存策略及调用策略。以新闻资讯业务和在线翻译业务举例说明。

*新闻资讯业务*

* **数据存储策略：**存储于数据库；通过定时任务，每10分钟分别调用各类第三方新闻资讯接口获取新闻数据存于数据库作为数据源；
* **数据缓存策略：**查询结果缓存于Redis，过期时间10min；
* **第三方调用策略：**从Redis或数据库中获取新闻资讯数据返回给调用者。

*在线翻译业务*

* **数据存储策略：**不存储于数据库；
* **数据缓存策略：**查询结果缓存于Redis，过期时间3天；
* **第三方调用策略：**直接调用第三方接口获取翻译结果，封装后返回给调用者。

#### 数据有效性

不同业务的数据从有效变为无效所需时间各有差异。如：此刻惠风和畅，下一刻就可能乌云压城，天气信息几分钟内就可能失效。如：路由器可能十几年居于一室，也有可能下一刻就随主人乔迁，其对应的WiFi位置信息也随之失效。

在业务设计过程中，开发者将综合所有业务各自不同的属性，不断调整数据源更新策略，力求在：响应速度、存储空间、调用第三方服务接口次数以及数据有效性之间寻求一个最佳的平衡点。

### 3.4 实现横向拓展数据源

一个业务接口可以由N个第三方服务提供为数据源，通过分离提供方参数和业务参数，统一化业务参数格式及接口返回值格式，使得业务接口可以横向拓展第三方数据源。下面通过两个典型例子来展现可拓展性的实现细节。

***新闻资讯业务拓展数据源***

新闻资讯业务接口通过查询数据库获取数据并返回，因此拓展数据源意即：接入新的第三方服务，通过调用接口获取数据并存入新闻数据库；或是接入新的新闻抓取工具，抓取数据并存入数据库。因此实现拓展数据源的关键在于统一数据源输入流格式。

#### 统一输入流格式

基于业务需求，综合第三方新闻接口返回值的内容，设计出新闻数据库表。当接入新的第三方服务，调用接口获取到数据后，遍历每一条新闻数据，对应数据库字段，将每一条数据映射为统一格式的新闻数据存入数据库。服务接入后，数据库新增了一个数据源，而业务接口不会受丝毫影响。

* **定义数据库表结构**

|  |  |  |
| --- | --- | --- |
| 字段 | 含义 | 类型 |
| id | 新闻数据于数据库中主键 | int |
| type | 新闻分类 | varchar（50） |
| title | 新闻标题 | varchar（200） |
| link | 新闻链接 | varchar（200） |
| html | 新闻HTML | text |
| source | 新闻来源 | varchar（100） |
| pubDate | 新闻发布时间 | timestamp |

*表3-6 新闻资讯数据库表结构*

调用第三方接口获取数据并格式化后存入数据库，其流程如下图所示。



*图3-7 保存新闻数据流程说明图*

***在线翻译业务拓展数据源***

在线翻译业务通过在业务内部直接调用第三方接口获取数据并封装后返回给调用者，因此拓展数据源意即：接入新的第三方服务供业务接口内部调用。因此实现拓展数据源的关键在于统一业务接口格式。

#### 统一业务接口格式

统一业务接口格式即统一：业务参数（bussinessParam）格式以及返回值格式。接入新的第三方服务后，该服务会作为数据提供方被放入可用资源池。当该服务被作为数据源使用时，调用者传入统一格式的业务参数调用业务接口，业务参数映射为服务参数以调用第三方服务接口获取数据，再将数据封装为统一格式后返回给调用者。服务接入后，业务接口新增了一个数据源，而业务接口不会受丝毫影响。

* **统一业务参数格式**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 含义 | 类型 | 必填 | 备注 |
| q | 要翻译的文本 | string | 是 |  |
| from | 源语言 | string |  | 默认为auto |
| to | 目标语言 | string |  | 默认为auto |

* **统一返回值格式**

**{**

**"resultCode": 200,**

**"message": "成功",**

**"data": {**

**"translations":**

**[**

**"Red apple"**

**]**

**}**

**}**

图3-8展示了调用者在调用翻译接口后，业务内部的处理过程。



*图3-8 在线翻译业务内部处理机制*

## 第四章 数据库设计

根据目前的业务需求及属性，仅新闻资讯数据需要保存于数据库。

### 4.1 参数表

#### 说明

1. 业务参数：调用业务接口时调用者传入的参数。提供方参数：第三方服务提供方特有的参数，不需要用户传入，由后台配置。
2. 优先级用于标志调用顺序，值越小，越先被调用。
3. 状态值为1的提供方才会出现在可用资源池中。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 含义 | 类型 | 不是null | 默认值 |
| id | 主键id | int | **√** |  |
| bussiness\_name | 业务名称 | varchar（200） |  | Empty String |
| provider\_name | 第三方服务提供方名称 | varchar（200） |  | Empty String |
| priority | 优先级 | int（2） |  | 0 |
| bussiness\_param | 业务参数 | varchar（1000） |  | Empty String |
| provider\_param | 提供方参数 | varchar（1000） |  | Empty String |
| status | 状态 | int（1） |  | 0 |

### 4.2 新闻资讯数据库

#### 说明

用于存储新闻数据，数据来源为：第三方服务接口或新闻抓取工具。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 含义 | 类型 | 不是null | 默认值 |
| id | 主键 | int | **√** |  |
| type | 新闻分类 | varchar（50） |  | Empty String |
| title | 新闻标题 | varchar（200） |  | Empty String |
| link | 新闻链接 | varchar（200） |  | Empty String |
| html | 新闻HTML | text |  | Empty String |
| source | 新闻来源 | varchar（100） |  | Empty String |
| pubDate | 新闻发布时间 | timestamp |  | currentTime |
| heat | 热度 | int |  | 0 |

## 第五章 接口设计

### 返回码

|  |  |
| --- | --- |
| 返回码 | 备注 |
| 200 | 请求成功 |

### 5.2 在线翻译业务

#### 接口说明

1. 请求类型：HTTP POST；
2. 调用场景：需要翻译文本内容时，如：中英互译或其他语言。
3. 接口示例：

<https://bussiness-integration.ubtrobot.com/translation/v1>

**Headers:**

"Content-Type":"application/json;charset=UTF-8"

**Body:**

{

"q": "我是一个红apple",

"to": "EN"

}

#### 参数说明

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 含义 | 类型 | 必填 | 备注 |
| q | 要翻译的文本 | Sring | 是 |  |
| from | 源语言 | Sring |  | 默认为auto |
| to | 目标语言 | Sring |  | 默认为auto |

#### 返回信息

**{**

**"resultCode": 200,**

**"message": "成功",**

**"data": {**

**"translations":**

**[**

**"I am a red apple"**

**]**

**}**

**}**

#### 返回值说明

|  |  |
| --- | --- |
| 字段名 | 说明 |
| resultCode | 返回码 |
| message | 返回码说明 |
| data | 返回数据 |
| translations | 翻译结果 |

### 5.3 新闻资讯业务

#### 接口说明

1. 请求类型：HTTP POST；
2. 调用场景：需要获取新闻资讯时。
3. 接口示例：<https://bussiness-integration.ubtrobot.com/news/v1>

**Headers:**

"Content-Type":"application/json;charset=UTF-8"

**Body:**

{

"fromId": "15926",

"keyWord": "美食"

}

1. 默认按新闻发布时间降序排列返回数据。

#### 参数说明

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 含义 | 类型 | 必填 | 说明 |
| fromId | 起始ID | int |  | 上一次查询结果最后一条新闻的ID，作为下一次查询的查询标志，用于APP端 |
| size | 查询条数 | int |  |  |
| type | 新闻分类 | String |  | 模糊匹配新闻分类type |
| keyWord | 新闻关键字 | String |  | 模糊匹配新闻标题title |
| needHtml | 是否返回HTML | int |  | 默认不返回 |
| heatSort | 是否按热度排名查询 | int |  | 默认按新闻发布时间降序排列 |

#### 返回信息

**{**

**"message": "",**

**"resultCode": 200,**

**"data": {**

**"total": 2,**

**"contentList": [**

**{**

**"id": "15927",**

**"type": "美食频道",**

**"title": "广东美食大盘点",**

**"link": "**[**http://news.timedg.com/2017-09/23/20625310.shtml**](http://news.timedg.com/2017-09/23/20625310.shtml)**",**

**"html": "<p>HTML格式的新闻内容...</p>",**

**"source": "大粤网",**

**"pubDate": "2017-09-20 13:00:00"**

**},**

**{**

**"id": "15928",**

**"type": "美食频道",**

**"title": "吃遍王府井",**

**"link": "**[**http://news.timedg.com/2017-09/23/20625310.shtml**](http://news.timedg.com/2017-09/23/20625310.shtml)**",**

**"html": "<p>HTML格式的新闻内容...</p>",**

**"source": "京城美食网",**

**"pubDate": "2017-09-20 14:00:00"**

**}**

**]**

**}**

**}**

#### 返回值说明

|  |  |
| --- | --- |
| 字段名 | 说明 |
| resultCode | 返回码 |
| message | 返回码说明 |
| data | 返回数据 |
| contentList | 新闻数据列表 |
| total | contentList中新闻总条数 |
| id | 新闻数据于数据库中主键ID |
| type | 新闻分类 |
| title | 新闻标题 |
| link | 新闻链接 |
| html | 新闻HTML |
| source | 新闻来源 |
| pubDate | 新闻发布时间 |