# 传入参数一定要传入以下几个参数（传入参数都使用字符串，不要使用数字类型）：

**要求传入参数如下：**

|  |  |  |
| --- | --- | --- |
| 传入字段 | 字段名称 | 说明 |
| af | 应用标识  对应原字段appFlag | 0 移动课堂 1 网校项目 2小程序 3妇幼课堂 5 对外 以后新增应用继续增加，pc可以按照模块功能区分（如选课、支付、题库等），具体开发中不断增加。 |
| fs | 分类标识  对应原字段platformSource | 目前分为两级第一级第一位标识，第23位作为二级标识  目前暂定1 安卓 2 ios 3 pc 4 m站 5 小程序 6 未知（注意后续可能出现非数字）  二级标识2、3位，目前暂定01 手机 02 pad 03 TV 如果二级标识没有，传00（注意后续可能出现非数字） |
| ve | 版本 对应原字段version | 版本号 |
| lt | 服务器时间 对应原字段ltime | 13位毫秒串 如：1540455811450 |
| ap | 应用唯一标识 对应原字段appkey | e8ae0de4-fc24-4293-b5d4-f15376cfa5dd |
|  |  |  |

具体如下：

af 0 移动课堂 1 网校项目 2小程序 3妇幼课堂

ve版本号

ap 应用key pc可以传入固定值pc 手机网页调用可以传入mobile 小程序传入：wechat

lt 服务器时间（1519369175288），用来判断时效性，一般人物接口在24小时之内有效，超过24小时无效，这个值一般是将这个服务器时间转化为一个token值，这个值参与加密，产生token的规则按照如下方式处理（String strMd5=Md5.Md5(nowTime+privateKey,32).substring(0,10)，其中的nowTime就是传入的服务器时间）;

fs：目前分为两级第一级第一位标识，第23位作为二级标识

目前暂定1 安卓 2 ios 3 pc 4 m站 5 小程序 6 未知（注意后续可能出现非数字）

二级标识2、3位，目前暂定01 手机 02 pad 03 TV 如果二级标识没有，传00（注意后续可能出现非数字）

上面参数几个值都会传入的，可通过上面几个值来判断是什么端来调取的接口。也可以通过上面几个参数来做个性化开发（调接口时需要注意的上面参数必须传入）

**例如 安卓的网校app调用接口，传入参数如下：**

member.chinaacc.com/mapi/school/courCenter/order/getMobileCart76?userID=49931021&time=2018-10-25 16%3A29%3A13&isFront=0&selectCourse=&isCart=1&siteID=1&pkey=7a3f471273be61537ef2de8e14939bb0&fs=101&ss=1&ltime=1540455811450&productids=&ve=7.7&isProductNum=0&uid=49931021&AppPage=com.cdel.accmobile.shopping.activities.PromptActivty& ap=e8ae0de4-fc24-4293-b5d4-f15376cfa5dd&af=1

# 2、返回值和提示语：

目前的返回值都是采用json方式返回，必须返回字段 success，errorCode，errorMsg返success为true代表数据正常返回， 返回false为失败。失败要有各种返回值返回。（各个业务组要注意的，返回值要确定保证按照这种方式开发。）

**例如：**

{

"result": {

"demoList":[],

"demoDetail":{}

},

"success": true,

"errorCode": null,

"errorMsg": null

}

上面result内部放的是正式的业务数据，上面只是一个样例，并不是要求输入demoLits或者demoDetail，业务组可以根据自己的需要放各种集合对象。

具体返回值如下：

拼接的提示语的规则为：异常码=业务码+分类标识+应用标识

比如安卓的会计网校接口加密出现问题时提示语为：请求异常（异常码：enc1011），请与客服联系！

|  |  |  |
| --- | --- | --- |
| Code值 | 业务码 | 说明 |
| 1 | suc | 数据返回成功(网关) |
| -1 | par | 请求参数错误(网关) |
| -2 | unk | 未知请求资源，无法找到对应接口(网关) |
| -3 | ove | 请求超时(网关) |
| -4 | vis | 接口访问错误(网关) |
| -5 | inn | 未知内部错误(网关) |
| -6 | puk | 请求公钥已过期(网关) |
| -7 | log | 需要登录(网关) |
| -8 | gra | 需要被授权(网关) |
| -9 | fro | 请求数据格式非法(网关) |
| -10 | nde | 资源未定义(网关) |
| -11 | req | 超过请求数，请稍后再试(网关) |
| -12 | enc | 不符合要求的密钥(网关) |
| -13 | beo | 请求已过期(网关) |
| -14 | con | 需要网开教务配置显示，但是网开教务忘记配置导致数据显示异常(网关) |
| -1000 | senc | 加密错误时（服务端） |
| -1001 | snec | 必要性验证时（服务端） |
| -1002 | scon | 需要网开教务配置显示，但是网开教务忘记配置导致数据显示异常（服务端） |
| -1003 | saut | 用户就是没有权限使用这个功能（服务端） |
| -1004 | slim | 出现限制性错误使用此功能，例如这个功能其他人正在用被锁定，等解锁后可以使用此功能（服务端） |
| -1005 | sdat | 出现数据库异常时（服务端） |
| -1006 | sove | 当出现时间超时异常（服务端） |
| -1008 | sinv | sid失效（服务端） |
| -1009 | sunk | 未知异常（服务端） |
|  |  |  |

实现例子：

其他错误值返回值具体如下，各种错误返回值对应各种错误，部分例子如下：

/\*\*加密错误时 \*/

public static final Integer CODE\_0 =- 1000;

/\*\*必要性验证时 \*/

public static final Integer CODE\_NEG\_1 = -1001;

/\*\*需要网开教务配置显示，但是网开教务忘记配置导致数据显示异常时 \*/

public static final Integer CODE\_NEG\_2 = -1002;

/\*\*用户就是没有权限使用这个功能时 \*/

public static final Integer CODE\_NEG\_3 = -1003;

/\*\*出现限制性错误使用此功能，例如这个功能其他人正在用被锁定，等解锁后可以使用此功能 \*/

public static final Integer CODE\_NEG\_4 = -1004;

/\*\*当出现数据库异常时\*/

public static final Integer CODE\_NEG\_5 = -1005;

/\*\*当出现时间超时异常时\*/

public static final Integer CODE\_NEG\_6 = -1006;

/\*\*sid失效 \*/

public static final Integer CODE\_NEG\_8 = -1008;

/\*\*未知异常 \*/

public static final Integer CODE\_NEG\_9 = -1009;

下面都是以安卓手机的网校应用的提示语

/\*\*加密错误时，使用此提示语 \*/

public static final String CODE\_MSG\_0 = "请求异常（异常码：senc1011），请与客服联系！";

/\*\*必要性验证时，使用此提示语 \*/

public static final String CODE\_MSG\_NEG\_1 = "请求异常（异常码：snec1011），请与客服联系！";

/\*\*需要网开教务配置显示，但是网开教务忘记配置导致数据显示异常时，使用此提示语 \*/

public static final String CODE\_MSG\_NEG\_2 = "请求异常（异常码：scon1011），请与客服联系！";

/\*\*用户就是没有权限使用这个功能时，使用此提示语 \*/

public static final String CODE\_MSG\_NEG\_3 = "请求异常（异常码：saut1011），请与客服联系！";

/\*\*出现限制性错误使用此功能，例如这个功能其他人正在用被锁定，等解锁后可以使用此功能，使用此提示语 \*/

public static final String CODE\_MSG\_NEG\_4 = "请求异常（异常码：slim1011），请与客服联系！";

/\*\*当出现数据库异常时使用此提示语 \*/

public static final String CODE\_MSG\_NEG\_5 = "请求异常（异常码：sdat1011），请与客服联系！";

/\*\*当出现时间超时异常时使用此提示语 \*/

public static final String CODE\_MSG\_NEG\_6 = "请求异常（异常码：sove1011），请与客服联系！";

/\*\*当出现时间超时异常时使用此提示语 \*/

public static final String CODE\_MSG\_NEG\_8 = "请求异常（异常码：sinv1011），请与客服联系！";

/\*\*未知异常 \*/

public static final String CODE\_MSG\_NEG\_9 = "请求异常（异常码：sunk1011），请与客服联系！";

/\*\*程序正常返回使用此提示语 \*/

public static final String CODE\_MSG\_1 = "成功";

各个业务组开发不需要有各种安全加密的处理，只处理业务逻辑即可，加密处理的相关，是由整体网关进行处理的。

# 3、接口开发流程：

3-1业务部门下需求

3-2经过和业务部门、前端讨论 。

3-3接口业务定义接口。

3-4接口方填写mock数据（业务复杂接口需要后端补充mock后，接口才能提供。）以及后端填写接口，mock数据。注意mock数据要提供完善。否则前端无法开发。

3-5后台提供接口，前端在这段时间内使用mock数据开发，接口方书写调用架子，测试种类。

3-6接口对业务部门提供的接口进行验证。

3-7后端提供接口并验证通过后，将接口提供给前端，前端使用正式接口调用。

3-8前端开发完成，后端和接口上线。

# 4、数据验证

接口对各个业务组提供的数据接口验证如下：

4-1如果必要参数全部不传的情况返回结果是否正常

4-2如果必要参数全部传空字端穿返回结果是否正常

4-3 部分必要参数不传入的验证

4-4 前部参数全部传入，没有返回数据的时候的验证情况

4-5对正常业务进行测试，验证数据是否正常，有一种逻辑会验证一次

4-6 对返回数据进行检查，看mock数据是否与填入的返回一致，看返回的结果是否按照规则返回。

4-7 检查接口中mock规则是否符合，如果本来是一个list，返回的结果是一个字符串，这种情况认为是不符合规则的。

如果提供接口超过30%验证不通过，打回各业务组，由业务组进行重新调整。

例如提供的一批接口中如果提供了10个接口，但是其中有3个以上（不包含三个）有问题，认为业务组是有问题的，需要重新调整。

注：返回的json要符合规则，字段中如果这个值为空，也要返回，不能key、value都没有了。阿里的json转化有一个问题就是如果这个值为空，连对应的key都没有了，这点要注意。