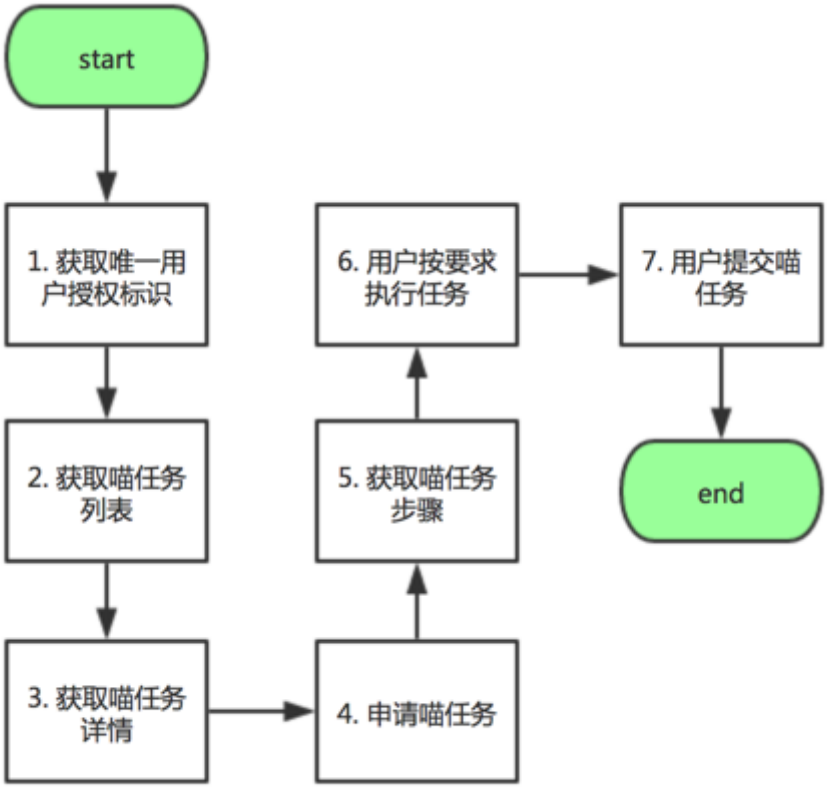
1 业务流程



第一步，通过`3.8 喵任务用户授权服务`获取唯一用户授权标识`auth_id`
第二步，通过`3.1 获取喵任务列表`获取可执行的喵任务列表
第三步，通过`3.2 获取喵任务详情`获取喵任务详情信息
第四步，通过`3.4 申请喵任务`申请喵任务
第五步，通过`3.3 获取喵任务步骤`获取对应喵任务的执行步骤说明
第六步，用户按照任务步骤说明进行执行(当步骤中要求上传图片时，需要调用`3.7 上传图片服务`)
第七步，将用户的任务步骤执行数据通过`3.5 用户提交喵任务`提交
第八步，喵任务执行流程结束。

2 API规范

2.1 通讯协议

统一采用`HTTP`协议的`POST`方法进行通讯

2.2 字符编码

统一为`UTF-8`

2.3 数据格式

统一为`json`字符串

2.4 请求

2.4.1 参数说明

参数	说明
clientId	客户端标识
name	服务名称，详见`3 服务`中定义的服务名称
version	服务版本，详见`3 服务`中定义的服务版本
content	业务数据(json字符串)的加密字符串, 业务数据明文见`3 服务`中的`请求`部分, 具体加密算法详见`2.8 加密方式`
signature	签名字符串, 具体签名方式和验签方式见`2.6 签名方式`和`2.7 验签方式`

2.4.2 请求示例

```
{
    "clientId": "JW0000001",
    "name" : "mewService",
    "version" : "1.0",
    "content" : "",
    "signature": ""
}
```

2.5 响应

2.5.1 参数说明

参数	说明
clientId	客户端标识
name	服务名称，详见`3 服务`中定义的服务名称
version	服务版本，详见`3 服务`中定义的服务版本
uid	唯一请求标识
content	业务数据(json字符串)的加密字符串, 业务数据明文见`3 服务`的`响应`部分, 具体解密算法详见`2.9 解密方式`
signature	签名字符串, 具体验签方式见`2.7 验签方式`
respCode	响应码, `0000`表示请求成功, 其他响应码定义见`附录 5.1 响应码`
respMsg	响应码说明

2.5.2 响应示例

```
{
    "clientId": "JW0000001",
    "name" : "mewService",
    "version" : "1.0",
    "uid" : "",
    "content" : "",
    "signature": "",
    "respCode": "0000",
    "respMsg" : "请求成功"
}
```

2.6 签名方式

将业务数据(json字符串)通过`SHA-256`算法生成签名字符串（结果为小写十六进制字符串），即`signture`字段内容

2.6.1 java示例代码

```
import java.nio.charset.Charset;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;

/**
 * SHA-256生成签名示例代码
 *
 * @date    Sep 26, 2016 3:42:07 PM <br/>
 * @version
 * @since   JDK 1.7
 */
public class SignatureTest {

  public static void main(String[] args) throws NoSuchAlgorithmException {
    String msg = "Hello, JiuWei!";
    String key = "458fa84deb98f2133f0d94a2cd173244"; // 秘钥
    String signature = ""; // 签名字符串

    signature = signature(key + msg + key);
    System.out.println(">> SHA-256生成签名: " + signature);
  }

  public static String signature(String message) throws NoSuchAlgorithmException
{
    MessageDigest mdInst = MessageDigest.getInstance("SHA-256");
    mdInst.update(message.getBytes(Charset.forName("UTF-8")));
    return byte2Hex(mdInst.digest());
  }

  private static String byte2Hex(byte[] bytes) {
    StringBuilder hexs = new StringBuilder();
    String s = "";
    for (int i = 0; i < bytes.length; i++) {
      s = Integer.toHexString(bytes[i] & 0XFF);
      if (s.length() == 1){
        hexs.append("0");
      }
      hexs.append(s);
```

```
    }

    return hexs.toString();
  }

}
```

2.6.2 php示例代码

```php
<?php
class AES {
    public static function encrypt($input, $key) {
        $data = openssl_encrypt($input, 'AES-128-ECB', hex2bin($key),
OPENSSL_RAW_DATA);
        $data = bin2hex($data);
        return $data;
    }

    public static function decrypt($sStr, $sKey) {
        $decrypted = openssl_decrypt(hex2bin($sStr), 'AES-128-ECB',
hex2bin($sKey), OPENSSL_RAW_DATA);
        return $decrypted;
    }
}

// 秘钥
$key = "";
// 明文
$data = "Hello, JiuWei!";

// 签名字符串
$signature = hash('sha256', utf8_encode($key.$data.$key));

// 加密
$encryptedText = AES::encrypt($data, $key);

// 解密
$decryptedText = AES::decrypt($encryptedText , $key);

print '<b>Plaintext</b>:  ' . $data .'<br/>';
print '<b>Signature</b>:  ' . $signature .'<br/>';
print '<b>Encrypted-Text</b>:  ' . $encryptedText .'<br/>';
print '<b>Decrypted-Text</b>:  ' . $decryptedText .'<br/>';
?>
```

2.7 验签方式

按照`2.6 签名方式`生成的签名字符串与请求参数中的`signature`进行比较, 不一致则验签失败.

2.8 加密方式

将业务数据(json字符串)通过指定秘钥`key` （128位），通过`AES/ECB/PKCS5Padding`加密算法生成加密字符串(结果进行十六进制编码), 即`content`字段内容。

2.8.1 java示例代码

```java
import java.nio.charset.Charset;
import javax.crypto.Cipher;
import javax.crypto.spec.SecretKeySpec;

/**
 * AES加密示例代码
 *
 * @date    Sep 26, 2016 3:56:29 PM <br/>
 * @version
 * @since   JDK 1.7
 */
public class AESEncryptor {

  public static void main(String[] args) throws Exception {
    String msg = "Hello, JiuWei!"; // 具体业务数据, json字符串
    String key = "32decfe75d701e79bbed4a757548b8ad"; // 密钥
    String encryptMsg = "";

    encryptMsg = encrypt(msg, key);

    System.out.println(">> AES加密前: " + msg);
    System.out.println(">> AES加密后: " + encryptMsg);
  }

  private static String encrypt(String msg, String key) throws Exception {
    SecretKeySpec keySpec = new
SecretKeySpec(hex2Byte(key.getBytes(Charset.forName("UTF-8"))), "AES");
    Cipher cipher = Cipher.getInstance("AES");
    cipher.init(Cipher.ENCRYPT_MODE, keySpec);

    return byte2Hex(cipher.doFinal(msg.getBytes(Charset.forName("UTF-8"))));
  }

  private static String byte2Hex(byte[] bytes) {
    StringBuilder hexs = new StringBuilder();
    String s = "";

    for (int i = 0; i < bytes.length; i++) {
      s = Integer.toHexString(bytes[i] & 0XFF);
      if (s.length() == 1){
        hexs.append("0");
      }

      hexs.append(s);
    }

    return hexs.toString();
  }


 private static byte[] hex2Byte(byte[] b) {
    byte[] b2 = new byte[b.length / 2];
    for (int n = 0; n < b.length; n += 2) {
        String item = new String(b, n, 2);
        b2[n/2] = (byte) Integer.parseInt(item, 16);
```

```
    }

    return b2;

  }

}
```

2.8.2 php示例代码

见`2.6.2 php示例代码`

2.9 解密方式

通过指定秘钥`key`和`AES`算法将请求中的`content`字段内容进行解密得到具体业务数据(json字符串)。

2.9.1 java示例代码

```
import javax.crypto.Cipher;
import javax.crypto.spec.SecretKeySpec;

/**
 * AES解密示例代码
 *
 * @date    Sep 26, 2016 4:12:56 PM <br/>
 * @version
 * @since   JDK 1.7
 */
public class AESDecryptor {

  public static void main(String[] args) throws Exception {
    String encryptMsg = "9661157bb3cab94b664eb3bfb2874803"; // 已加密的字符串
    String key = "32decfe75d701e79bbed4a757548b8ad"; // 密钥
    String msg = ""; // 具体业务数据, json字符串

    msg = decrypt(encryptMsg, key);

    System.out.println(">> AES解密前: " + encryptMsg);
    System.out.println(">> AES解密后: " + msg);
  }

  private static String decrypt(String msg, String key) throws Exception {
    SecretKeySpec secretKey = new
SecretKeySpec(hex2Byte(key.getBytes(Charset.forName("UTF-8"))), "AES");

    Cipher cipher = Cipher.getInstance("AES");
    cipher.init(Cipher.DECRYPT_MODE, secretKey);

    return new String(cipher.doFinal(hex2Byte(msg.getBytes(Charset.forName("UTF-
8")))), Charset.forName("UTF-8"));
  }

  private static byte[] hex2Byte(byte[] b) {
    byte[] b2 = new byte[b.length / 2];
```

```
    for (int n = 0; n < b.length; n += 2) {
        String item = new String(b, n, 2);
        b2[n/2] = (byte) Integer.parseInt(item, 16);

    }

    return b2;
  }

}
```

2.9.2 php示例代码

见`2.6.2 php示例代码`

2.10 参数类型

标识符	说明
STRING	字符类型, 字母, 数字, 中文以及特殊字符
ARRAY	数组
INT	数值类型

2.11 参数状态

标识符	说明
M	必须
O	可选
C	满足条件时为必须

2.12 其他约定

- **时间戳** 指格林威治时间1970年01月01日00时00分00秒(北京时间1970年01月01日08时00分00秒)起至现在的**总毫秒数**

3 服务

3.1 获取喵任务列表

3.1.1 getMewTasks(1.0)

3.1.1.1 请求

名称	类型	长度	状态	备注
task_province	STRING	16	M	喵任务限制的省, 所有省详见附录
task_city	STRING	16	M	喵任务限制的市, 所有市详见附录
task_type	INT		O	喵任务类型, 数据字典: 0-不限, 默认; 1-分享; 2-互动; 3-体验; 4-其他
task_sort_type	INT		O	喵任务列表排序类型, 数据字典: 0-最新发布, 默认; 1-单价最高; 2-最快审核
task_platform	INT		M	喵任务限制的执行平台, 数据字典: 0-不限, 默认; 1-Android; 2-IOS
page_num	INT		M	页码
page_size	INT		O	每页数据大小, 最大30, 默认值20

3.1.1.2 响应

数据类型: 数组

名称	类型	长度	状态	备注
task_id	STRING	32	M	喵任务唯一标识
task_title	STRING	64	M	喵任务标题
task_price	INT		M	用户指导价, 单位分
task_create_time	INT		M	喵任务发布时间, 时间戳
task_check_time	INT		M	企业审核喵任务时间, 单位小时
task_repeat_hours	INT		M	喵任务执行周期, 单位小时, 0表示不限
task_repeat_num	INT		M	喵任务在 `task_repeat_hours` 小时内可以重复执行次数, 0表示不限
task_platform	INT		M	喵任务限制的执行平台,数据字典: 0-不限; 1-Android; 2-IOS
task_surplus	INT		M	喵任务剩余量
task_type	INT		M	喵任务类型, 数据字典: 1-分享; 2-互动; 3-体验; 4-其他

> 只会返回的状态正常的任务

3.2 获取喵任务详情

3.2.1 getMewTaskDetail(1.0)

3.2.1.1 请求

名称	类型	长度	状态	备注
task_id	STRING	32	M	喵任务唯一标识
auth_id	STRING	32	O	用户授权唯一标识, 通过 `3.8 喵任务用户授权服务` 获取

3.2.1.2 响应

数据类型: 对象

名称	类型	长度	状态	备注
task_id	STRING	32	M	喵任务唯一标识
task_title	STRING	64	M	喵任务标题
task_price	INT		M	用户指导价, 单位分
task_imgs	ARRAY		M	喵任务图示地址
task_desp	STRING	128	M	任务描述
task_create_time	INT		M	喵任务开始时间
task_end_time	INT		M	喵任务结束时间
task_check_time	INT		M	企业审核喵任务时间, 单位小时
task_repeat_type	INT		M	任务重复类型, 数据字典: 0-不可重复执行; 1-可重复执行
task_surplus	INT		M	喵任务剩余量
task_status	INT		M	喵任务状态, 数据字典: 0-正常; 1-已过期; 2-已下架; 3-已暂停
user_task_status	INT		C	用户喵任务状态, 数据字典: 0-未申请; 1-已申请, 用户满足去重条件, 即相同唯一设备号或手机号码之一有申请或完成该喵任务，且喵任务有限制条件时出现; 2-已完成(等同于审核通过)，用户已完成该任务，且任务相对用户来说不可继续申请执行时出现; 3-进行中，用户已申请该任务，但未提交任务; 4-审核中，用户已提交任务，企业审核中; 5-审核通过，用户提交任务，且企业已审核通过; 6-已下架，表明任务不可申请/提交; 7-驳回，用户提交任务，但企业审核不通过; 8-已过期：表示用户申请过该任务，但是在任务的执行时间内没有提交任务。
user_task_count_down	INT		M	用户喵任务剩余执行时间, 单位秒

> `task_status`为**0-正常**且`user_task_status`为**0-未申请**时，表明用户可以申请任务。注意：`auth_id`没有传值时，`user_task_status`也为**0-未申请**，但不一定表明用户可以申请任务

3.2.2 getMewTaskDetail(2.0)

3.2.2.1 请求

名称	类型	长度	状态	备注
task_id	STRING	32	M	喵任务唯一标识
auth_id	STRING	32	O	用户授权唯一标识, 通过 `3.8 喵任务用户授权服务` 获取

3.2.1.2 响应

数据类型: 对象

名称	类型	长度	状态	备注
task_id	STRING	32	M	喵任务唯一标识
task_title	STRING	64	M	喵任务标题
task_price	INT		M	用户指导价, 单位分
task_desp	STRING	128	M	任务描述
task_create_time	INT		M	喵任务开始时间
task_end_time	INT		M	喵任务结束时间
task_check_time	INT		M	企业审核喵任务时间, 单位小时
task_repeat_type	INT		M	任务重复类型, 数据字典: 0-不可重复执行; 1-可重复执行
task_surplus	INT		M	喵任务剩余量
task_status	INT		M	喵任务状态, 数据字典: 0-正常; 1-已过期; 2-已下架; 3-已暂停
user_task_status	INT		C	用户喵任务状态, 数据字典: 0-未申请; 1-已申请, 用户满足去重条件, 即相同唯一设备号或手机号码之一有申请或完成该喵任务，且喵任务有限制条件时出现; 2-已完成(等同于审核通过)，用户已完成该任务，且任务相对用户来说不可继续申请执行时出现; 3-进行中，用户已申请该任务，但未提交任务; 4-审核中，用户已提交任务，企业审核中; 5-审核通过，用户提交任务，且企业已审核通过; 6-已下架，表明任务不可申请/提交; 7-驳回，用户提交任务，但企业审核不通过; 8-已过期：表示用户申请过该任务，但是在任务的执行时间内没有提交任务。
user_task_count_down	INT		M	用户喵任务剩余执行时间, 单位秒

> `task_status`为**0-正常**且`user_task_status`为**0-未申请**时，表明用户可以申请任务。注意：`auth_id`没有传值时，`user_task_status`也为**0-未申请**，但不一定表明用户可以申请任务

3.3 获取喵任务步骤

3.3.1 getMewTaskSteps(1.0)

3.3.1.1 请求

名称	类型	长度	状态	备注
task_id	STRING	32	M	喵任务唯一标识

3.3.1.2 响应

返回类型: 数组

名称	类型	长度	状态	备注
task_id	STRING	32	M	喵任务唯一标识
task_step_imgs	ARRAY		M	喵任务步骤图示地址
task_step_url	ARRAY	128	M	喵任务步骤中的链接地址
task_step_no	INT		M	喵任务步骤序号
task_step_type	INT		M	喵任务步骤类型, 数据字典: 0-图文说明页; 1-链接打开页; 2-短文本提交页; 3-长文本提交页; 4-图片提交页
task_step_desp	STRING		M	喵任务步骤描述
task_step_demo_imgs	ARRAY		M	喵任务步骤示例图片
task_step_demo_text	STRING		M	喵任务步骤示例文本

3.4 申请喵任务

3.4.1 applyMewTask(1.0)

3.4.1.1 请求

名称	类型	长度	状态	备注
auth_id	STRING	32	M	用户授权唯一标识, 通过`3.8 喵任务用户授权服务获取`
task_id	STRING	32	M	喵任务唯一标识
platform	STRING	64	M	系统平台, 0-其他; 1-Android; 2-iOS
device_id	STRING	64	C	唯一的设备号, 当`platform`不为`0`时, 该值为必须

3.4.1.2 响应

数据类型: 对象

名称	类型	长度	状态	备注
user_task_status	INT		M	用户喵任务状态, 数据字典: 0-未申请, 申请失败; 1-已申请, 用户满足去重条件, 即相同唯一设备号或手机号码之一有申请或完成该喵任务，且喵任务有限制条件时出现; 2-已完成, 用户已完成喵任务, 无法继续申请该任务; 3-进行中, 表示申请成功; 4-审核中, 表示用户已申请该任务且任务已提交审核; 5-审核通过; 6-已下架; 7-驳回; 8-已过期.
task_apply_id	INT		C	用户申请喵任务的唯一标识

> 用户申请成功时，`user_task_status`为**3-进行中**

3.5 用户提交喵任务

3.5.1 submitMewTask(1.0)

3.5.1.1 请求

名称	类型	长度	状态	备注
auth_id	STRING	32	M	用户授权唯一标识, 通过`3.8 喵任务用户授权服务`获取
task_apply_id	INT		M	用户申请喵任务的唯一标识
user_task_steps	ARRAY		M	用户提交喵任务步骤数据

user_task_steps

名称	类型	长度	状态	备注
task_step_no	INT		M	喵任务步骤序号
task_step_type	INT		M	喵任务步骤类型, 数据字典: 0-图文说明页; 1-链接打开页; 2-短文本提交页; 3-长文本提交页; 4-图片提交页
task_content	STRING		C	`task_step_type` 为3时填用户输入的内容
task_imgs	ARRAY		C	`task_step_type` 为4时填图片标识(数组, 图片标识通过 `3.7` 获取)

3.5.1.2 响应

数据类型: 对象

名称	类型	长度	状态	备注
user_task_status	INT		M	用户喵任务状态, 数据字典: 0-未申请; 1-已申请; 2-已完成; 3-进行中; 4-审核中; 5-审核通过; 6-已下架; 7-驳回; 8-已过期.

- 用户提交成功时为 `4-审核中`

3.6 查询用户喵任务

3.6.1 queryUserMewTask(1.0)

3.6.1.1 请求

名称	类型	长度	状态	备注
auth_id	STRING	32	M	用户授权唯一标识, 通过`3.8 喵任务用户授权服务`获取
task_apply_id	INT		O	用户申请喵任务的唯一标识
user_task_type	INT		O	用户喵任务分类, 数据字典: 0-全部, 返回用户所有状态的喵任务; 1-进行中, 返回用户进行中的喵任务; 2-审核中, 返回用户审核中的喵任务; 3-已结束, 返回审核通过、已下架、驳回和已过期的任务.
page_num	INT		C	页码, `task_apply_id`没有传值时，为必须
page_size	INT		C	每页数据大小, 最大30, 默认值20, `task_apply_id`没有传值时，为必须

3.6.1.2 响应

返回类型: 数组

名称	类型	长度	状态	备注
task_id	STRING	32	M	喵任务唯一标识
task_apply_id	INT		C	用户申请喵任务的唯一标识
task_price	INT		M	用户指导价, 单位分
task_title	STRING	64	M	喵任务标题
task_type	INT		M	喵任务类型, 数据字典: 0-不限, 默认; 1-分享; 2-互动; 3-体验; 4-其他
task_create_time	INT	32	M	喵任务发布时间, 时间戳
task_apply_time	INT	32	M	申请喵任务时间, 时间戳
task_submit_time	INT	32	C	提交喵任务时间, 时间戳
task_end_time	INT	32	M	喵任务结束时间, 时间戳
task_check_time	INT		M	企业审核喵任务时间, 单位小时
task_surplus	INT		M	喵任务剩余量
user_task_status	INT		C	用户喵任务状态, 数据字典: 3-进行中; 4-审核中; 5-审核通过; 6-已下架; 7-驳回; 8-已过期
user_task_count_down	INT		M	用户喵任务剩余执行时间, 单位秒
remark	STRING	32		`user_task_status`为7时，为驳回原因
user_task_steps	ARRAY		C	用户提交喵任务步骤数据

user_task_steps

名称	类型	长度	状态	备注
task_step_no	INT		M	喵任务步骤序号
task_step_type	INT		M	喵任务步骤类型, 数据字典: 0-图文说明页; 1-链接打开页; 2-短文本提交页; 3-长文本提交页; 4-图片提交页
task_content	STRING		C	用户提交喵任务步骤中内容, `task_step_type`为3时有值
task_imgs	ARRAY		C	用户提交喵任务的图片地址, `task_step_type`为4时有值

3.7 上传图片服务

3.7.1 uploadImg(1.0)

3.7.1.1 请求

名称	类型	长度	状态	备注
auth_id	STRING	32	M	用户授权唯一标识, 通过`3.8 喵任务用户授权服务`获取
img	STRING		M	图片的base64编码字符串, 限制2M内

> 注意：图片的base64编码的字符串不能被编码，否则图片会无法显示，企业审核不通过

3.7.1.2 响应

数据类型: 对象

名称	类型	长度	状态	备注
img_id	STRING		M	图片的唯一标识, 该标识在2小时(喵任务执行时效)有效

3.8 喵任务用户授权服务

3.8.1 userAuth(1.0)

3.8.1.1 请求

名称	类型	长度	状态	备注
user_phone	STRING	16	M	用户手机号码
user_id	STRING	32	M	用户的唯一标识

- 需要确保请求参数`user_id`是唯一用户的标识，即相同的user_id, 即使手机号码不一样（用户变更手机号码的情况）也会返回相同的auth_id

3.8.1.2 响应

数据类型: 对象

名称	类型	长度	状态	备注
auth_id	STRING	32	M	喵任务用户授权的唯一标识

3.9 查询喵任务用户提交内容

3.9.1 queryUserMewTaskSteps(2.0)

3.9.1.1 请求

名称	类型	长度	状态	备注
task_id	STRING	32	M	喵任务唯一标识
page_num	INT		M	页码
page_size	INT		O	每页数据大小, 最大30, 默认值20

3.9.1.2 响应

数据类型: 对象

名称	类型	长度	状态	备注
auth_id	STRING	32	M	用户授权唯一标识, 通过`3.8 喵任务用户授权服务`获取
task_id	STRING	32	M	喵任务唯一标识
task_apply_id	INT		C	用户申请喵任务的唯一标识
task_price	INT		M	用户指导价, 单位分
task_title	STRING	64	M	喵任务标题
task_create_time	STRING	32	M	喵任务发布时间, 时间戳
task_apply_time	STRING	32	M	申请喵任务时间, 时间戳
task_submit_time	STRING	32	C	提交喵任务时间, 时间戳
task_end_time	STRING	32	M	喵任务结束时间
task_check_time	INT		M	企业审核喵任务时间, 单位小时
task_surplus	INT		M	喵任务剩余量
user_task_status	INT		C	用户喵任务状态, 数据字典: 3-进行中; 4-审核中; 5-审核通过; 6-已下架; 7-驳回; 8-已过期
user_task_count_down	INT		M	用户喵任务剩余执行时间, 单位秒
remark	STRING	32		`user_task_status`为7时，为驳回原因
user_task_steps	ARRAY		C	用户提交喵任务步骤数据

user_task_steps

名称	类型	长度	状态	备注
task_step_no	INT		M	喵任务步骤序号
task_step_type	INT		M	喵任务步骤类型, 数据字典: 0-图文说明页; 1-链接打开页; 2-短文本提交页; 3-长文本提交页; 4-图片提交页
task_content	STRING		C	用户提交喵任务步骤中内容, `task_step_type`为3时有值
task_imgs	ARRAY		C	用户提交喵任务的图片地址, `task_step_type`为4时有值

4. 回调

4.1 用户喵任务状态回调

数据类型: 数组

名称	类型	长度	状态	备注
auth_id	STRING	32	M	用户授权唯一标识
task_id	STRING	32	M	喵任务唯一标识
task_title	STRING	64	M	喵任务标题
task_price	INT		M	用户指导价, 单位分
task_apply_id	INT		M	用户申请喵任务的唯一标识
task_apply_time	STRING	32	M	申请喵任务时间, 时间戳
task_submit_time	STRING	32	M	提交喵任务时间, 时间戳
user_task_status	INT		M	用户喵任务状态, 数据字典: 5-审核通过; 6-已下架; 7-驳回; 8-已过期
remark	STRING	32	C	`user_task_status`为7时，为驳回原因

注意

- 当渠道方接收到回调通知后，需要返回`OK`字符串，否则九尾喵任务系统会以一定策略(通知频率为：1m/2m/5m/10m/60m/120m/300m/600m)，不断重发回调通知，直到收到响应的`OK`字符串或回调通知次数达到8次。
- 同样的通知结果可能会发送多次，渠道方需要能处理重复回调的情况。
- 渠道方收到回调后的响应信息（`OK`字符串）直接返回明文即可

5. 附录

5.1 响应码

响应码	说明
0000	请求成功
9999	请求失败，请稍后再试
FFFF	未知异常
F000	字段格式不正确
F001	未发现可用的服务列表
F002	未发现名称[%s]及其版本[%s]的服务
F003	服务内部处理错误
F004	暂时无法访问名称[%s]及版本[%s]的服务，请稍后再试
F005	暂时无法访问名称[%s]及版本[%s]的服务，请稍后再试
F006	请求内容不能为空
F007	请求内容格式不正确
F008	IP[%s]禁止访问
F009	解密失败
F010	验签失败
1001	非法的client_id
1002	content格式错误
1003	请勿重复请求
2001	任务不存在
2002	非法用户授权标识
2003	已申请该任务
2004	任务状态异常
2005	用户喵任务状态异常
2006	用户提交喵任务内容不能为空
2007	用户提交喵任务文本类型内容不能为空
2008	用户提交喵任务图片类型内容不能为空
2007	手机号码格式不正确
2009	BASE64图片字符串不能为空