

1. 协议标准

1) 请求方式

1.

// 为统一的请求方式，除非特殊接口（例如登录），否则全部使用post方式，参数全部使用JSON传输
2.

POST

2) Http 请求头参数

请求参数	作用域	参数值	描述
Authorization	登录	Basic dGVybWluYWw6dGVybWluYWxfc2VjcmV0	Basic开头 + 一个空格 + 考试云分配给终端的客户密钥（经过了base64编码）
Authorization	API请求	Bearer 8f52db34-0228-4a47-8cf7-d9977668a190	Bearer开头 + 一个空格 + token

注意：考试云必须遵守此规定，必须在Http请求头带参数，否则会拒绝请求

3) 请求体

```
1. bizType:
2.
3.
4. 十六进制表示，高位的1个字节为标志位，低位的3个字节为业务位
5. String str = "10000FA1";
6. System.out.println(0x02000FA1 & 0x10000000);
7. Integer a = Integer.parseInt(str, 16);
8. System.out.println(a);

1. {
2.     // 业务类型（必须的）：int型，每一个接口为一个业务类型，每个业务类型的数据体可能不一样，int型，具体项目具体自行定义
3.     "bizType": 4001,
4.     // 数据体：具体传送过来的数据，可以是任意类型，根据具体业务定义，此消息体必须加密
5.     "data": "748e01498c9e132e46205b0cac25526c2bbd7ace50e9d2a7bd5d37e025df390554d3ac29e9657f206387c4d284605dd455570127f8a8e7a75a0a063497f26636",
6.     // 时间戳（毫秒），数值型
7.     "requestTime": 1562643180408,
8.     // 签名：具体请看“签名生成”
9.     "sign": "0e0f3c09qhbzyrajx436862e666c4833e4dc4a39f244a3c78",
10.    // 设备ID - 可表示哪一方调用，具体值为【max地址，IMEI，CPU，网卡芯片号】
11.    "msEquipment": "02:00:00:00:00:00,865640025134468, MT8735",
12.    // 请求的token，用于安全验证，长度200以内，考试云在请求头设置token，此处token也会设置，但是考试云不取此处的token
13.    "token": "573001a9-18f3-4cf3-970f-9dcefed4bdd9"
14. }
```

4) 标准下发格式

```
1. {
2.     "code": 1, // 错误码 ：数字类型，0 表示成功，1 表示失败，其它项目可以自定义错误码
3.     "msg": "考生成绩查询失败", // 返回消息：字符串类型
4.     "data": null, // 根据接口自定义类型，可以是对象、字符串、数字等
5.     "sign": null, // 返回数据签名（请参考 4.签名生成与内容加密）
6.     "encrypt": 0 // 是否加密，0: 不加密 1: 加密
7. }
```

5)签名生成与内容加密

- 例如：现有一个请求，内容如下：
- 原始内容

```
1. {
2.     "bizType": 1001,
3.     "data": {
4.         "examSite": 15,
```

```

5.     "siteSchedule": 2018080001,
6.     "school": "南山",
7.     "other": null
8. },
9.     "requestTime": 1540881713822,
10.    "sign": "",
11.    "msEquipment": "02:00:00:00:00:00,865640025134468, MT8735,",
12.    "token": "573001a9-18f3-4cf3-970f-9dcefed4bdd9"
13. }

```

■ 需要引入加解密包

```

1.  JAVA
2.  <!-- https://mvnrepository.com/artifact/commons-codec/commons-codec -->
3.  <dependency>
4.      <groupId>commons-codec</groupId>
5.      <artifactId>commons-codec</artifactId>
6.      <version>1.11</version>
7.  </dependency>
8.
9.  C++ # Crypto++® Library 8.2 [https://www.cryptopp.com/]
10.  请参考其API进行加密，或咨询谢安勇（已实现）

```

■ 处理签名

```

1.  签名内容包含：data的值（转为JSON字符串，且去掉为空的属性）
2.  字段1： 1001
3.  字段2：
4.      {"examSite":15,"siteSchedule":2018080001,"school":"南山","other":null}
5.      去掉空值的属性
6.      {"examSite":15,"siteSchedule":2018080001,"school":"南山"}
7.
8.  将进行签名的字符串： {"examSite":15,"siteSchedule":2018080001,"school":"南山"}
9.  最小的签名的字符串： {}
10.
11.  使用JAVA进行签名：
12.  String str = "{\"examSite\":15,\"siteSchedule\":2018080001,\"school\":\"南山\"}";
13.  String hexStr = org.apache.commons.codec.digest.DigestUtils.sha1Hex(str);
14.  签名后的sign为： 0e0f3c09436862e666c4833e4dc4a39f244a3c78

```

■ 生成加密key

```

1.  // key使用固定的6个字符+10个随机字符串
2.
3.  // 以下为生成方法
4.  private static final String BASESTRING = "0123456789abcdefghijklmnopqrstuvwxyz";
5.
6.  public static ThreadLocalRandom getRandom() {
7.      return ThreadLocalRandom.current();
8.  }
9.
10.
11.  public static String randomString(String baseString, int length) {
12.      final StringBuilder sb = new StringBuilder();
13.
14.      if (length < 1) {
15.          length = 1;
16.      }
17.      int baseLength = baseString.length();
18.      for (int i = 0; i < length; i++) {
19.          int number = getRandom().nextInt(baseLength);
20.          sb.append(baseString.charAt(number));
21.      }
22.      return sb.toString();
23.  }

```

```
24.  
25. // 1. 固定字符串: fp2018  
26. // 2. 获取随机的10个字符串: qhbzyraijx  
27. // 3. 将随机字符串添加到签名的第8个位置后面:  
28. // 0e0f3c09 + qhbzyraijx + 436862e666c4833e4dc4a39f244a3c78
```

■ 处理加密

```
1. /**  
2. *使用AES加密, AES/ECB/PKCS5Padding(加密内容必须转为UTF8) 例如:  
3. **/  
4. String AES_KEY = "fp2018qhbzyraijx"  
5.  
6.  
7. AES加密后的字符串为: 748e01498c9e132e46205b0cac25526c2bbd7ace50e9d2a7bd5d37e025df390554d3ac29e9657f206  
387c4d284605dd455570127f8a8e7a75a0a063497f26636  
8.  
9. JAVA 加密与解密  
10. // 加密  
11. SecretKey secretKey = new SecretKeySpec(AES_KEY.getBytes(), "AES");  
12. Cipher cipher = Cipher.getInstance("AES");  
13. cipher.init(Cipher.ENCRYPT_MODE, secretKey);  
14. byte [] encoderStr = cipher.doFinal("{\"examSite\":\"15,\\\"siteSchedule\\\":2018080001,\\\"school\\\":\\\"南山\\\"\"}.getBytes("UTF-8")  
);  
15. String hexE = new String(Hex.encodeHex(encoderStr));  
16. System.out.println(hexE);  
17.  
18. // 解密  
19. Cipher cipher1 = Cipher.getInstance("AES");  
20. cipher1.init(Cipher.DECRYPT_MODE, secretKey);  
21. byte [] decodeStr = cipher1.doFinal(Hex.decodeHex(hexE.toCharArray()));  
22. System.out.println(new String(decodeStr, "UTF-8"));  
23.  
24. > 注意: UTF8编码
```

<http://tool.chacuo.net/cryptaes>

AES加密模式: ECB 填充: pkcs5padding 数据块: 128位 密码: fp2018qhbzyraijx 偏移量: iv偏移量, ecb模式不用 输出: hex 字符集: utf8

待加密、解密的文本: {"examSite":15,"siteSchedule":2018080001,"school":"南山"}

↑ 将你电脑文件直接拖入试试 ^_^

AES加密 AES解密

AES加密、解密转换结果(base64了): 748e01498c9e132e46205b0cac25526c2bbd7ace50e9d2a7bd5d37e025df390554d3ac29e9657f206387c4d284605dd455570127f8a8e7a75a0a063497f26636

■ 最终结果

```
1. {  
2.   "bizType": 1001,  
3.   "data": "748e01498c9e132e46205b0cac25526c2bbd7ace50e9d2a7bd5d37e025df390554d3ac29e9657f206387c4d2846  
05dd455570127f8a8e7a75a0a063497f26636",  
4.   "requestTime": 1540881713822,  
5.   "sign": "0e0f3c09qhbzyraijx436862e666c4833e4dc4a39f244a3c78",  
6.   "msEquipment": "02:00:00:00:00:00,865640025134468, MT8735",  
7.   "token": "573001a9-18f3-4cf3-970f-9dcefed4bdd9"  
8. }
```

2.用户操作接口

1) 用户登录接口（考试体侧通用）

业务类型：1001

请求

```
1. url: /auth/terminal/token?username=nskdtest@2nskdtest&password=123456
```

参数

```
1. "username": "", //用户名 规则：用户名@机器码 例如：nskdtest@2nskdtest
2. "password": "" //密码
```

返回数据

```
1. {
2.   "code": 0,
3.   "msg": "登录成功",
4.   "data": {
5.     "examName": "2019深圳中考",
6.     "siteId": "1146332063859740674",
7.     "typeValue": "4000",
8.     "type": "考点测试",
9.     "token": "d97edc60-56d5-438d-a50c-4a0f19129eae"
10.  },
11.   "sign": null,
12.   "encrypt": 0
13. }
```

2) 修改密码接口（考试专用）

业务类型：1002

请求

```
1. url: /run/testuser/modifyPassword
```

上传参数

```
1. {
2.   "bizType": 1002, // 业务类型（必须的）：每一个接口为一个业务类型，每个业务类型的数据体可能不一样，int型
3.   "data": {
4.     "oldPassword": "", //原密码，
5.     "newPassword": "" //新密码
6.   }
7.   "msEquipment": "", // 测试设备信息: 记录日志使用，请求方传入，string类型包括max地址，IMEI，CPU，网卡芯片号
8.   "requestTime": 0, // 时间戳，13位例如 1540881713822
9.   "sign": "string", // 签名：具体请看“签名生成”
10.  "token": "string" // 请求的token，安全验证，长度200以内
11. }
```

返回数据格式

```
1. {
2.   "code": 0, // 错误码：数字类型，0 表示成功，1 表示失败，其它项目可以自定义错误码
3.   "data": {},
4.   "msg": "string", // 返回消息：本次请求返回的消息，例如：success
5.   "sign": null,
6.   "encrypt": 0
7. }
```

3) 退出登录接口（考试体侧通用）

业务类型：1003

请求

```
1. url: /auth/token/logout
```

上传参数

```
1. {
2.   "bizType": 1003, // 业务类型（必须的）：每一个接口为一个业务类型，每个业务类型的数据体可能不一样，int型
3.   "data": {},
4.   "msEquipment": "", // 测试设备信息: 记录日志使用，请求方传入，string类型包括max地址，IMEI，CPU，网卡芯片号
5.   "requestTime": 0, // 时间戳，13位例如 1540881713822
6.   "sign": "string", // 签名：具体请看“签名生成”
7.   "token": "string" // 请求的token，安全验证，长度200以内
8. }
```

返回数据格式

```
1. {
2.   "code": 0, // 错误码：数字类型，0表示成功，1表示失败，其它项目可以自定义错误码
3.   "data": {},
4.   "msg": "string", // 返回消息：本次请求返回的消息，例如：success
5.   "sign": null,
6.   "encrypt": 0
7. }
```

4) 设备绑定接口（体侧专用）

业务类型：1000

请求

```
1. url: /run/deviceBind
```

上传参数

```
1. {
2.   "bizType": 1000, // 业务类型（必须的）：每一个接口为一个业务类型，每个业务类型的数据体可能不一样，int型
3.   "data": [{
4.     "deviceNo": "", // 日程编号
5.     "hostNumber": "", // 主机序号
6.     "machineCode": "", // 机器码
7.   }], // 此处展示完整结构，实际是加密数据
8.   "msEquipment": "", // 测试设备信息: 记录日志使用，请求方传入，string类型包括max地址，IMEI，CPU，网卡芯片号
9.   "requestTime": 0, // 时间戳，13位例如 1540881713822
10.  "sign": "string", // 签名：具体请看“签名生成”
11.  "token": "string" // 请求的token，安全验证，长度200以内
12. }
```

返回数据格式与上传相同（回传数据为失败数据）

```
1. {
2.   "code": 0, // 错误码：0表示成功，其它为HTTP状态错误码或自定义错误码
3.   "data": {
4.     "token": "d97edc60-56d5-438d-a50c-4a0f19129eae"
5.   }, // 此处展示完整结构，实际是加密数据
6.   "msg": "string", // 返回消息：本次请求返回的消息，例如：success
7.   "sign": null,
8.   "encrypt": 1
9. }
```

```
9. }
```

3.数据下发接口

1) 下发考点项目信息（考试体侧通用）

业务类型：2001

请求

```
1. url: /run/downItemInfo
```

上传参数

```
1. {
2.   "bizType": 2001, // 业务类型（必须的）：每一个接口为一个业务类型，每个业务类型的数据体可能不一样，int型
3.   "data": {}
4.   "msEquipment": "", // 测试设备信息: 记录日志使用，请求方传入，string类型包括max地址，IMEI，CPU，网卡芯片号
5.   "requestTime": 0, // 时间戳，13位例如 1540881713822
6.   "sign": "string", // 签名：具体请看“签名生成”
7.   "token": "string" // 请求的token，安全验证，长度200以内
8. }
```

返回数据格式

```
1. {
2.   "code": 0, // 错误码：0表示成功，其它为HTTP状态错误码或自定义错误码
3.   "data": [
4.     "itemName": "", // 项目名称
5.     "examItemCode": "", // 考试项目代码
6.     "resultUnit": "", // 成绩单位
7.     "itemType": 1, // 体侧项目类型：0: 身高、体重, 1: 肺活量, 2: 耐力, 3: 柔韧, 4: 速度, 5: 视力
8.     "testType": 1, // 测量方式, 计时，计数，远度，力量
9.     "resultTestNum": 1, // 测量次数
10.    "decimalDigits": 0, // 小数位数
11.    "carryMode": 0, // 进位方式（1.四舍五入 2.舍位 3.非零进位）
12.    "minResult": "", // 最小成绩
13.    "maxResult": "", // 最大成绩
14.    "lastResultMode": 0, // 最终成绩选择模式（1.最后成绩, 2.补考成绩, 3.最好）
15.    "machineCode": "" // 机器码
16.  ], // 此处展示完整结构，实际是加密数据
17.   "msg": "string", // 返回消息：本次请求返回的消息，例如：success
18.   "sign": null,
19.   "encrypt": 1
20. }
```

2) 下发考点日程信息（考试体侧通用）

业务类型：2002

请求

```
1. url: /run/downSiteScheduleInfo
```

上传参数

```
1. {
2.   "bizType": 2002, // 业务类型（必须的）：每一个接口为一个业务类型，每个业务类型的数据体可能不一样，int型
3.   "data": {}
4.   "msEquipment": "", // 测试设备信息: 记录日志使用，请求方传入，string类型包括max地址，IMEI，CPU，网卡芯片号
5.   "requestTime": 0, // 时间戳，13位例如 1540881713822
6.   "sign": "string", // 签名：具体请看“签名生成”
```

```
7.  "token": "string" // 请求的token，安全验证，长度200以内
8.  }
```

返回数据格式

```
1.  {
2.  "code": 0,    // 错误码：0表示成功，其它为HTTP状态错误码或自定义错误码
3.  "data": {
4.    "siteName": "", // 考点名称
5.    "examType": "", // 考试模式(0:个人模式 1:分组模式)
6.    "siteScheduleInfoVOList": [{ // 考点日程信息，List
7.      "scheduleNo": "", // 考点日程编号 日程编号就是场次
8.      "beginTime": 0, // 开始时间 long类型 时间戳
9.      "endTime": 0, // 结束时间 long类型 时间戳
10.     "examItemVOList": [{ // 考点日程项目信息
11.       "examItemCode": "", // 考试项目代码
12.     }]
13.   }]
14. }, // 此处展示完整结构，实际是加密数据
15. "msg": "string", // 返回消息：本次请求返回的消息，例如：success
16. "sign": null,
17. "encrypt": 1
18. }
```

3) 下发考生信息（考试体侧通用）

业务类型：2003

④根据考点日程项目下发考生

请求

```
1.  url: /run/downSiteScheduleItemStudent
```

上传参数

```
1.  {
2.  "bizType": 2003, // 业务类型（必须的）：每一个接口为一个业务类型，每个业务类型的数据体可能不一样，int型
3.  "data": {
4.    "scheduleNo": "", // 日程编号 可为空，下发考点项目考生
5.    "examItemCode": "", // 考试项目代码 可为空
6.    "batch": 1, // 分批查询，批号
7.    "batchTotal": 1, // 分批查询，总批数（总页数）--可为空
8.    "examType": 0 // 考生考试状态
9.    "uploadTime": 0 // 最后更新时间
10.  }, // 此处展示完整结构，实际是加密数据
11. "msEquipment": "", // 测试设备信息：记录日志使用，请求方传入，string类型包括max地址，IMEI，CPU，网卡芯片号
12. "requestTime": 0, // 时间戳，13位例如 1540881713822
13. "sign": "string", // 签名：具体请看“签名生成”
14. "token": "string" // 请求的token，安全验证，长度200以内
15. }
```

**** 返回数据格式****

```
1.  {
2.  "code": 0,    // 错误码：0表示成功，其它为HTTP状态错误码或自定义错误码
3.  "data": {
4.    "batch": 1, // 分批查询，批号
5.    "batchTotal": 1, // 分批查询，总批数（总页数）
6.    "examType": 0, // 考生考试状态
7.    "dataInfo": [{
8.      "schoolName": "", // 学校名称
9.      "sortName": "", // 组别名称
10.     "className": "", // 班级名称

```

```

11.     "studentType": 0, //考生类型（0.正常，1.择考，2.免考）
12.     "studentCode": "", //准考证号
13.     "examNo": "", //考试编号
14.     "registeredNo": "", //报名号
15.     "idCard": "", //身份证号
16.     "studentName": "", //姓名
17.     "gender": 0, //性别(0:男 1:女)
18.     "examType": 0, //考试类型 0.正常 1.补考，2.缓考
19.     "examItemCode": "", //项目代码
20.     "scheduleNo": "", //日程编号
21.     "machineCode": "", //机器码
22.     "icCard": "", //ic卡
23.     "gradeName": "", //年级名称
24.     "subject": "", //专业或科目
25.     "deptName": "", //院系或部门
26.     "downloadTime": 0, //时间戳
27.     "examItemName": "" //项目名称
28.   }]
29. }, // 此处展示完整结构，实际是加密数据
30. "msg": "string", // 返回消息：本次请求返回的消息，例如：success
31. "sign": null,
32. "encrypt": 1
33. }

```

4）下发分组信息（考试专用）

①根据日程项目下发分组信息

业务类型：2004

请求

```
1. url: /run/downSiteScheduleItemGroup
```

上传参数

```

1.  {
2.     "bizType": 2004, // 业务类型（必须的）：每一个接口为一个业务类型，每个业务类型的数据体可能不一样，int型
3.     "data":{
4.         "scheduleNo": "", //日程编号 可为空
5.         "examItemCode": "", //考试项目id
6.         "batch": 1, //分批查询，批号
7.         "batchTotal": 1, //分批查询，总批数
8.         "examType": 0 //考生考试状态
9.     }, // 此处展示完整结构，实际是加密数据
10.    "msEquipment": "", // 测试设备信息: 记录日志使用，请求方传入，string类型包括max地址，IMEI，CPU，网卡芯片号
11.    "requestTime": 0, // 时间戳，13位例如 1540881713822
12.    "sign": "string", // 签名：具体请看“签名生成”
13.    "token": "string" // 请求的token，安全验证，长度200以内
14.  }

```

**** 返回数据格式****

```

1.  {
2.     "code": 0, // 错误码：0表示成功，其它为HTTP状态错误码或自定义错误码
3.     "data":{
4.         "batch": 1, //分批查询，批号
5.         "batchTotal": 1, //分批查询，总批数
6.         "examStatus": 0, //考生考试状态
7.         "dataInfo":[{
8.             "sortName": "", //组别
9.             "groupNo": "", //分组
10.            "groupType": 0, //分组类别（0.男子 1.女子 2.混合）
11.            "beginTime": 0, //开始时间，Long类型，时间戳 日程开始时间
12.            "endTime": 0, //结束时间，Long类型，时间戳 日程开始时间

```



```
13.     "scheduleNo": "", //考点日程编号    日程编号就是场次
14.     "examType": 0, //考试类型 0:正常 1.补考, 2.缓考
15.     "studentCodeList": [{
16.         "studentCode": "" //准考证号
17.         "trackNo": "" //道号
18.     }]
19. }
20. }, // 此处展示完整结构, 实际是加密数据
21. "msg": "string", // 返回消息: 本次请求返回的消息, 例如: success
22. "sign": null,
23. "encrypt": 1
24. }
```

②根据日程项目下发分组信息-单独一组

业务类型: 2005

请求

```
1. url: /run/downSiteScheduleItemGroupByGroupNo
```

上传参数

```
1.  {
2.     "bizType": 2004, // 业务类型 (必须的): 每一个接口为一个业务类型, 每个业务类型的数据体可能不一样, int型
3.     "data": {
4.         "scheduleNo": "" //日程编号 可为空
5.         "examItemCode": "" //考试项目编码
6.         "sortName": "" //组别
7.         "groupNo": "" //分组
8.         "groupType": "" //分组类别 (0: 男子, 1: 女子, 2: 混合)
9.     }, // 此处展示完整结构, 实际是加密数据
10.    "msEquipment": "", // 测试设备信息: 记录日志使用, 请求方传入, string类型包括max地址, IMEI, CPU, 网卡芯片号
11.    "requestTime": 0, // 时间戳, 13位例如 1540881713822
12.    "sign": "string", // 签名: 具体请看 “签名生成”
13.    "token": "string" // 请求的token, 安全验证, 长度200以内
14. }
```

**** 返回数据格式****

```
1.  {
2.     "code": 0,    // 错误码 : 0表示成功, 其它为HTTP状态错误码或自定义错误码
3.     "data": [{
4.         "sortName": "", //组别
5.         "groupNo": "", //分组
6.         "groupType": "", //分组类别 (0.男子 1.女子 2.混合)
7.         "beginTime": 0, //开始时间, Long类型, 时间戳 日程开始时间
8.         "endTime": 0, //结束时间, Long类型, 时间戳 日程开始时间
9.         "scheduleNo": "", //考点日程编号    日程编号就是场次
10.        "examType": 0, //考试类型 0:正常 1.补考, 2.缓考
11.        "studentCodeList": [{
12.            "studentCode": "" //准考证号
13.            "trackNo": "" //道号
14.        }]
15.    }], // 此处展示完整结构, 实际是加密数据
16.    "msg": "string", // 返回消息: 本次请求返回的消息, 例如: success
17.    "sign": null,
18.    "encrypt": 1
19. }
```

4.检录信息接口---暂未使用

1) 入口检录---暂未使用

业务类型：3001

请求

```
1. url: /run/uploadCheckEntrance
```

上传参数

```
1. {
2.   "bizType": 3001, // 业务类型（必须的）：每一个接口为一个业务类型，每个业务类型的数据体可能不一样，int型
3.   "data": {[
4.     "scheduleNo": "", //日程编号
5.     "studentCode": "", //准考证号
6.     "checkType": 1 //检录状态 0:检入 1: 检出
7.   ]}, // 此处展示完整结构，实际是加密数据
8.   "msEquipment": "", // 测试设备信息: 记录日志使用，请求方传入，string类型包括max地址，IMEI，CPU，网卡芯片号
9.   "requestTime": 0, // 时间戳，13位例如 1540881713822
10.  "sign": "string", // 签名：具体请看“签名生成”
11.  "token": "string" // 请求的token，安全验证，长度200以内
12. }
```

返回数据格式与上传相同（回传数据为失败数据）

```
1. {
2.   "code": 0, // 错误码：0表示成功，其它为HTTP状态错误码或自定义错误码
3.   "data": [
4.     "scheduleNo": "", //日程编号
5.     "studentCode": "", //准考证号
6.     "checkType": 1 //检录状态 1:检入 2: 检出
7.   ], // 此处展示完整结构，实际是加密数据
8.   "msg": "string", // 返回消息：本次请求返回的消息，例如：success
9.   "sign": null,
10.  "encrypt": 1
11. }
```

2) 项目检录---暂未使用

业务类型：3002

请求

```
1. url: /run/uploadCheckItem
```

上传参数

```
1. {
2.   "bizType": 3002, // 业务类型（必须的）：每一个接口为一个业务类型，每个业务类型的数据体可能不一样，int型
3.   "data": {[
4.     "scheduleNo": "", //日程编号;
5.     "studentCode": "", //准考证号;
6.     "itemCode": "", //项目代码;
7.     "groupNo": "", //组号;
8.     "trackNo": "", //道号;
9.     "examState": 0, //考试状态（0.正常，1.缓考，2.补考）
10.    "rollCallTime": 0, //检录时间，long类型，时间戳
11.    "userInfo": "", //用户信息
12.    "msEquipment": "" //检录设备信息
13.  ]}, // 此处展示完整结构，实际是加密数据
14.  "msEquipment": "", // 测试设备信息: 记录日志使用，请求方传入，string类型包括max地址，IMEI，CPU，网卡芯片号
15.  "requestTime": 0, // 时间戳，13位例如 1540881713822
```

```
16.  "sign": "string", // 签名：具体请看“签名生成”
17.  "token": "string" // 请求的token，安全验证，长度200以内
18.  }
```

返回数据格式与上传相同（回传数据为失败数据）

```
1.  {
2.  "code": 0,    // 错误码：0表示成功，其它为HTTP状态错误码或自定义错误码
3.  "data": [{
4.    "scheduleNo": "", //日程编号;
5.    "studentCode": "", //准考证号;
6.    "itemCode": "", //项目代码;
7.    "groupNo": "", //组号;
8.    "trackNo": "", //道号;
9.    "examState": 0, //考试状态（0.正常, 1.缓考, 2.补考）
10.   "rollCallTime": 0, //检录时间, long类型, 时间戳
11.   "userInfo": "", //用户信息
12.   "msEquipment": "" //检录设备信息
13. }], // 此处展示完整结构，实际下发的是加密数据
14. "msg": "string" // 返回消息：本次请求返回的消息，例如：success
15. "sign": null,
16. "encrypt": 1
17. }
```

5.成绩上传接口（考试体侧通用）

业务类型：4001

请求

```
1.  url: /run/uploadStudentResult
```

上传参数

```
1.  {
2.  // 业务类型（必须的:每一个接口为一个业务类型，每个业务类型的数据体可能不一样，int型
3.  "bizType": 4001,
4.  "data": [{
5.    "siteScheduleNo": "", //日程编号
6.    "examItemCode": "", //项目代码
7.    "studentCode": "", //准考证号
8.    "testNum": "", //测试次数
9.    "groupNo": "", //组号
10.   "groupType": "", //0:男子, 1: 女子, 2: 混合
11.   "sortName": "", // 组别
12.   "hostNumber": "", //主机序号
13.   "machineCode": "", //机器码
14.   "examItemName": "", //项目名称
15.   "result": "", // 最终成绩
16.   "result2": "", //第二个成绩，体侧的体重或者右眼视力
17.   "testTime": "", //时间戳
18.   "resultStatus": 0, // 成绩状态, 0:未检录 1:正常 2:犯规 3:中退 4:弃权 5:测试，将最终成绩的isfour填入
19.   "roundResultList": [{ //轮次成绩
20.     "trackNo": "", // 道号
21.     "roundNo": "", // 轮次号
22.     "penalty": "", //判罚值
23.     "isFoul": 0, //是否犯规 0:未检录 1:正常 2:犯规 3:中退 4:弃权 5:测试
24.     "resultStatus": 0, // 成绩状态, 0:未检录 1:正常 2:犯规 3:中退 4:弃权 5:测试，isfour填入即可
25.     "result": "", //成绩=机器成绩+判罚值
26.     "result2": "", //第二个成绩，体侧的体重或者右眼视力
27.     "machineResult": "", //机器成绩
```

```
28.     "resultType": 0, //是否最终决定成绩 0:不是 1: 是
29.     "examState": 0, //考试状态 (0.正常, 1.缓考, 2.补考)
30.     "testTime": "", //测试时间
31.     "printTime": "", //打印时间
32.     "userInfo": "", //测试用户信息
33.     "msEquipment": "", //测试设备信息
34.     "mtEquipment": "", //监控设备信息
35.     "uploadTime": "", //上传时间
36.     "stumbleCount": 0, //绊绳次数, 仅跳绳项目要有
37.     "leftLens": 0, //左眼串镜, 1:正片上升、负片下降, -1:正片下降、负片上升, 2:其他, 9:未测
38.     "rightLens": 0, //右眼串镜, 同左眼串镜一致
39.     "leftRefractive": 0, //左眼屈光不正, 0:正常 1:近视 2:远视 3:其他 9:未测
40.     "rightRefractive": 0, //右眼屈光不正, 同左眼屈光不正一致
41.     "remark": "" // 备注
42. },
43. }, // 此处展示完整结构, 实际上传的是加密数据
44. "msEquipment": "", // 测试设备信息: 记录日志使用, 请求方传入, string类型包括max地址, IMEI, CPU, 网卡芯片号
45. "requestTime": 0, // 时间戳, 13位例如 1540881713822
46. "sign": "string", // 签名: 具体请看 “签名生成”
47. "token": "string" // 请求的token, 安全验证, 长度200以内
48. }
```

返回数据格式

```
1.  {
2.    "code": 0, // 错误码 : success(0,"全部成功"),failure(1,"失败"),partFail(2,"部分失败");
3.    "data": []
4.    "msg": "string", // 返回消息: 本次请求返回的消息, 例如: success
5.    "sign": null,
6.    "encrypt": 1
7.  }
```