ALIBABA CLOUD

阿里云

IoT安全中心

IoT安全运营中心公共云合集

文档版本：20210330

阿里云

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。

2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。

3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。

4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。

5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。

6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	危险 重置操作将丢失用户配置数据。
警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	警告 重启操作将导致业务中断，恢复业务时间约十分钟。
注意	用于警示信息、补充说明等，是用户必须了解的内容。	注意 权重设置为0，该服务器不会再接受新请求。
说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击**设置**> **网络**> **设置网络类型**。
粗体	表示按键、菜单、页面名称等UI元素。	在**结果确认**页面，单击**确定**。
`Courier字体`	命令或代码。	执行 `cd /d C:/window` 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	`bae log list --instanceid` *Instance_ID*
[] 或者 [a|b]	表示可选项，至多选择一个。	`ipconfig [-all|-t]`
{} 或者 {a|b}	表示必选项，至多选择一个。	`switch {active|stand}`

目录

1.API参考 06
1.1. 获取AccessKey 06
1.2. IoT设备身份认证API 07
1.2.1. API概览 07
1.2.2. Verify 09
1.2.3. VerifyAndEncrypt 11
1.2.4. GetServerRandom 12
1.2.5. GetServerAuthCodeAndEncryptData 14
1.2.6. OtpGetId2 17
1.2.7. 设备端API 19
1.2.8. 错误码 21
1.3. 固件安全检测API 23
1.3.1. API概览 23
1.3.2. RAM鉴权 24
1.3.3. CreateScanTask 24
1.3.4. QueryTaskStatus 26
1.3.5. QueryTaskReportUrls 27
1.3.6. QueryTaskReport 28
1.3.7. DeleteScanTask 29
1.3.8. QueryLicense 29
1.3.9. QueryTaskReportUrls 30
1.3.10. 检测结果格式说明 31
1.3.11. 错误码 38
1.4. 安全RFID标签 API 39
1.4.1. 安全RFID标签-网页跳转 39
1.4.2. 安全RFID标签-手机API 40

1.4.3. 安全RFID标签-超高频读写器 42
1.4.4. 错误码 46
2.IoT安全合作伙伴 49
2.1. 成为IoT安全合作伙伴 49
3.了解更多 50
4.相关协议 51

1.API参考

1.1. 获取AccessKey

您可以为阿里云主账号和子账号创建一个访问密钥（AccessKey）。在调用阿里云API时您需要使用AccessKey完成身份验证。

背景信息

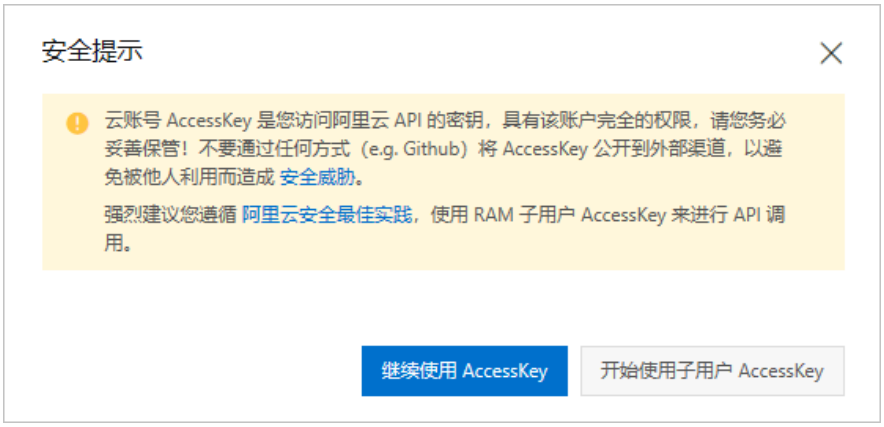
AccessKey包括AccessKey ID和AccessKey Secret。

- AccessKeyId：用于标识用户。
- AccessKeySecret：用于验证用户的密钥。AccessKeySecret必须保密。

> **警告** 主账号Accesskey泄露会威胁您所有资源的安全。建议使用子账号（RAM用户）Accesskey进行操作，可以有效降低Accesskey泄露的风险。

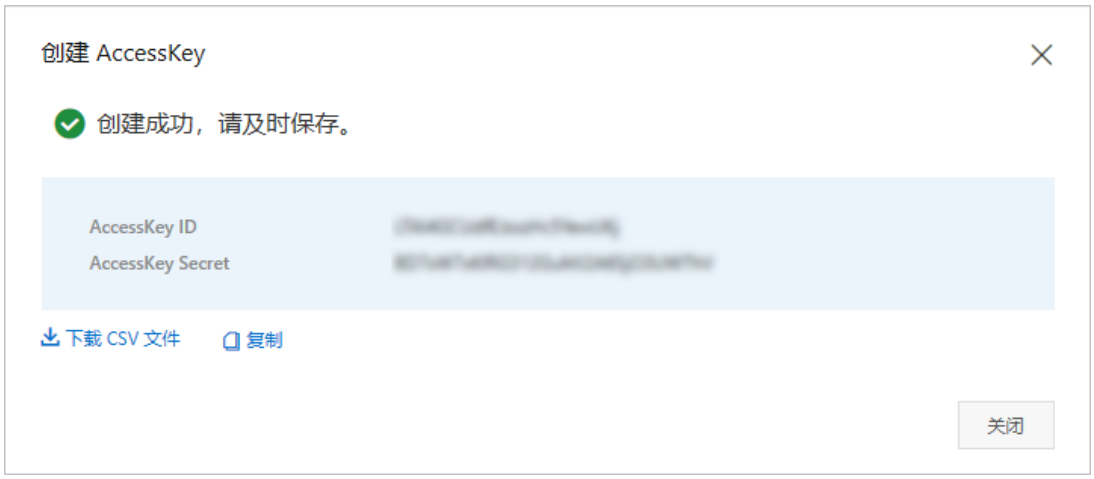
操作步骤

1. 以主账号登录阿里云管理控制台。
2. 将鼠标置于页面右上方的账号图标，单击**accesskeys**。
3. 在**安全提示**页面，选择获取主账号还是子账号的Accesskey。



4. 获取账号Accesskey。
   - 获取主账号AccessKey
     a. 单击**继续使用AccessKey**。
     b. 在**安全信息管理**页面，单击**创建AccessKey**。
     c. 在**手机验证**页面，获取验证码，完成手机验证，单击**确定**。

d. 在**新建用户AccessKey**页面，展开**AccessKey详情**，查看AccessKeyId和AccessKeySecret。可以单击**保存AK信息**，下载AccessKey信息。



- 获取子账号AccessKey
  a. 单击**开始使用子用户AccessKey**。
  b. 如果未创建RAM用户，在系统跳转的RAM访问控制台的**新建用户**页面，创建RAM用户。如果是获取已有RAM用户的Accesskey，则跳过此步骤。
  c. 在RAM访问控制台的左侧导航栏，选择**人员管理 > 用户**，搜索需要获取AccessKey的用户。
  d. 单击用户登录名称，在用户详情页**认证管理**页签下的**用户AccessKey**区域，单击**创建新的AccessKey**
  e. 在**手机验证**页面，获取验证码，完成手机验证，单击**确定**。
  f. 在**创建AccessKey**页面，查看AccessKeyId和AccessKeySecret。可以单击**下载CSV文件**，下载AccessKey信息或者单击**复制**，复制AccessKey信息。

1.2. IoT设备身份认证API

1.2.1. API概览

以下是ID²服务端和设备端的API列表。

服务端API列表

服务端API	描述
Verify	服务端认证接口。
VerifyAndEncrypt	服务端认证并加密接口。
GetServerRandom	获取服务端随机数接口。
GetServerAuthCodeAndEncryptData	获取服务端认证码接口。
OtpGetId2	ID²动态下发接口。

设备端API

详情请查看设备端API文档

设备端API	描述
id2_client_init	ID² Client SDK初始化，使用ID²设备端其它API之前，需要先调用该API进行初始化操作。
id2_client_get_id	获取ID²字符串。
id2_client_get_challenge_auth_code	基于挑战应答模式生成设备端认证码，可选携带额外数据extra，携带的extra数据参与设备认证码的签名运算。
id2_client_get_timestamp_auth_code	基于时间戳模式生成设备端认证码，可选携带额外数据extra，携带的extra数据参与设备认证码的签名运算。时间戳可以由设备端RTC生成，或者由SP携带发送给设备端。
id2_client_decrypt	使用ID²解密指定的数据。
id2_client_get_prov_stat	获取设备端ID²的烧录状态。
id2_client_get_otp_auth_code	获取设备端ID²动态下发的认证码。

设备端API	描述
id2_client_load_otp_data	烧录动态下发数据到设备中。

错误码

详情请查看错误码。

1.2.2. Verify

ID² 运行时认证接口。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	Verify	系统规定参数，取值：Verify。
ApiVersion	String	是	1.1.2	API版本号，当前取值1.1.2。
AuthCode	String	是	3~0~**********~1499753003564~Pfyt01tFpOFfI9HkHYjvqikoZpdch44U22ckmpvuwL1QfppOOIIGboFmfyhnrX73hGvQ**********tWMFMtbm3tmEqUzfSnYvGAEu/+YcytuZTKydh9ijJLUVeRgKUeS2**********	设备端生成的认证码。
Id2	String	是	00AAABBB11122281FE15****	ID²标识。
ProductKey	String	是	a1BwAGV****	产品标识，从ID²控制台的产品列表中获取。
Extra	String	否	digest1234	与认证码关联的辅助认证数据。
ProductSource	Integer	否	1	产品来源，当前取值为1，也可以不填。

返回数据

名称	类型	示例值	描述
Code	Integer	200	调用失败时会返回错误码。
Data	Boolean	true	返回数据。
RequestId	String	F6AFB45A-0FD1-405E-AD2A-C50E34C429E5	阿里云为该请求生成的唯一标识符。
Success	Boolean	true	是否调用成功。 • true：调用成功。 • false：调用失败。

示例

请求示例

```
http(s)://[Endpoint]/?Action=Verify
&ApiVersion=1.1.2
&AuthCode=3~0~**************~1499753003564~Pfyt01tFpOFfI9HkHYjvqikoZpdch44U22ckmpvuwL1QfppOOIIGboFmfyhnrX73hGvQ**************tWMFMtbm3tmEqUzfSnYvGAEu/+YcytuZTKydh9ijJLUVeRgKUeS2**********
&Id2=00AAABBB11122281FE15****
&ProductKey=a1BwAGV****
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<VerifyResponse>
 <RequestId>F6AFB45A-0FD1-405E-AD2A-C50E34C429E5</RequestId>
 <Data>true</Data>
 <Code>200</Code>
 <Success>true</Success>
</VerifyResponse>
```

`JSON` 格式

```
{
  "RequestId": "F6AFB45A-0FD1-405E-AD2A-C50E34C429E5",
  "Data": "true",
  "Code": "200",
  "Success": "true"
}
```

1.2.3. VerifyAndEncrypt

ID² 运行时认证并加密接口。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	VerifyAndEncrypt	系统规定参数，取值：VerifyAndEncrypt。
ApiVersion	String	是	1.1.2	API版本号，当前取值1.1.2。
AuthCode	String	是	3~0~**********~1499753003564~Pfyt01tFpOFfI9HkHYjvqikoZpdch44U22ckmpvuwL1QfppOOIIGboFmfyhnrX73hGvQ**********tWMFMtbm3tmEqUzfSnYvGAEu/+YcytuZTKydh9ijLUVeRgKUeS2**********	设备端生成的认证码。
Data	String	是	6F5DDB5F21C28F06484A4695FAB9****	待加密的数据。
Id2	String	是	00AAABBB11122281FE15****	ID²标识。
ProductKey	String	是	a1BwAGV****	产品标识，从ID²控制台的产品列表中获取。
Extra	String	否	digest1234	与认证码关联的辅助认证数据。
ProductSource	Integer	否	1	产品来源，当前取值为1，也可以不填。

返回数据

名称	类型	示例值	描述
Code	Integer	200	调用失败时会返回错误码。

名称	类型	示例值	描述
Data	String	MIGfM****DAQAB	返回的经Base64编码后的加密数据。
RequestId	String	F6AFB45A-0FD1-405E-AD2A-C50E34C429E5	阿里云为该请求生成的唯一标识符。
Success	Boolean	true	表示是否调用成功。 • true：调用成功。 • false：调用失败。

示例

请求示例

```
http(s)://[Endpoint]/?Action=VerifyAndEncrypt
&ApiVersion=1.1.2
&AuthCode=3~0~**************~1499753003564~Pfyt01tFpOFfI9HkHYjvqikoZpdch44U22ckmpvuwL1QfppOOIIGboFmfyhnrX73hGvQ**************tWMFMtbm3tmEqUzfSnYvGAEu/+YcytuZTKydh9ijJLUVeRgKUeS2**********
&Data=6F5DDB5F21C28F06484A4695FAB9****
&Id2=00AAABBB11122281FE15****
&ProductKey=a1BwAGV****
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<VerifyAndEncryptResponse>
 <RequestId>F6AFB45A-0FD1-405E-AD2A-C50E34C429E5</RequestId>
 <Data>MIGfM****DAQAB</Data>
 <Code>200</Code>
 <Success>true</Success>
</VerifyAndEncryptResponse>
```

`JSON` 格式

```
{
  "RequestId": "F6AFB45A-0FD1-405E-AD2A-C50E34C429E5",
  "Data": "MIGfM****DAQAB",
  "Code": "200",
  "Success": "true"
}
```

1.2.4. GetServerRandom

获取服务端随机数接口。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	GetServerRandom	系统规定参数，取值：GetServerRandom。
ApiVersion	String	是	1.1.2	API版本号，当前取值1.1.2。
Id2	String	是	00FFFF00FFFFFF1DB47A****	ID²标识。

返回数据

名称	类型	示例值	描述
Code	Integer	200	调用失败时会返回错误码。
Data	String	6F5DDB5F21C28F06484A4695FAB9****	返回的随机数。
RequestId	String	F6AFB45A-0FD1-405E-AD2A-C50E34C429E5	阿里云为该请求生成的唯一标识符。
Success	Boolean	true	是否调用成功。 • true：调用成功。 • false：调用失败。

示例

请求示例

```
http(s)://[Endpoint]/?Action=GetServerRandom
&ApiVersion=1.1.2
&Id2=00FFFF00FFFFFF1DB47A****
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<GetServerRandomResponse>
 <RequestId>F6AFB45A-0FD1-405E-AD2A-C50E34C429E5</RequestId>
 <Data>6F5DDB5F21C28F06484A4695FAB9****</Data>
 <Code>200</Code>
 <Success>true</Success>
</GetServerRandomResponse>
```

JSON 格式

```
{
  "RequestId": "F6AFB45A-0FD1-405E-AD2A-C50E34C429E5",
  "Data": "6F5DDB5F21C28F06484A4695FAB9****",
  "Code": "200",
  "Success": "true"
}
```

1.2.5. GetServerAuthCodeAndEncryptData

验证设备认证码有效性，获取服务端认证码，并加密数据。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	GetServerAuthCodeAndEncryptData	系统规定参数，取值：GetServerAuthCodeAndEncryptData。
ApiVersion	String	是	1.1.2	API版本号，当前取值1.1.2。
Data	String	是	abcd1234	待加密数据。

名称	类型	是否必选	示例值	描述
DeviceAuthCode	String	是	3~0~1245DC626946A9E5~1499753003564~Pfyt01tFpOFfI9HkHYjvqikoZpdch44U22ckmpvuwL1QfppOOIlGboFmfyhnrX73hGvQ5BKzX1Acie+8MHoKZ64Y8tWMFMtbm3tmEqUzfSnYvGAEu/+YcytuZTKydh9ijJLUVeRgKUeS29q1zj9LOTyGBXOxdesb9n9oQ225****	设备端生成的认证码。
DeviceChallenge	String	是	ABCD12348	设备挑战字，用于生成服务端认证码。
Id2	String	是	00AAABBB11122281FE15****	ID²标识。
ProductKey	String	是	a1BwAGV****	产品标识，从ID²控制台的产品列表中获取。
DeviceExtra	String	否	digest1234	与设备认证码关联的辅助认证数据。
ServerExtra	String	否	1234abcfe	与服务端认证码关联的辅助认证数据。
ProductSource	Integer	否	1	产品来源，当前取值为1，也可以不填。

返回数据

名称	类型	示例值	描述
Code	Integer	200	调用失败时会返回错误码。
Data	Struct		返回数据。

名称	类型	示例值	描述
RequestId	String	F6AFB45A-0FD1-405E-AD2A-C50E34C429E5	阿里云为该请求生成的唯一标识符。
Success	Boolean	true	是否调用成功。 • true：调用成功。 • false：调用失败。

示例

请求示例

```
http(s)://[Endpoint]/?Action=GetServerAuthCodeAndEncryptData
&ApiVersion=1.1.2
&Data=abcd1234
&DeviceAuthCode=3~0~1245DC626946A9E5~1499753003564~Pfyt01tFpOFfI9HkHYjvqikoZpdch44U22ckmpvuw
L1QfppOOIIGboFmfyhnrX73hGvQ5BKzX1Acie+8MHoKZ64Y8tWMFMtbm3tmEqUzfSnYvGAEu/+YcytuZTKydh9ijJ
LUVeRgKUeS29q1zj9LOTyGBXOxdesb9n9oQ225****
&DeviceChallenge=ABCD12348
&Id2=00AAABBB11122281FE15****
&ProductKey=a1BwAGV****
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<GetServerAuthCodeAndEncryptDataResponse>
 <RequestId>F6AFB45A-0FD1-405E-AD2A-C50E34C429E5</RequestId>
 <Data>
    <EncryptData>3s+wT***x4=</EncryptData>
    <ServerAuthCode>10~2~3DDFA7A45590CF12~QaB/DeZhx4KpBahW***gAZ5Q==</ServerAuthCode>
 </Data>
 <Code>200</Code>
 <Success>true</Success>
</GetServerAuthCodeAndEncryptDataResponse>
```

`JSON` 格式

```
{
  "RequestId": "F6AFB45A-0FD1-405E-AD2A-C50E34C429E5",
  "Data": {
    "EncryptData": "3s+wT***x4=",
    "ServerAuthCode": "10~2~3DDFA7A45590CF12~QaB/DeZhx4KpBahW***gAZ5Q=="
  },
  "Code": "200",
  "Success": "true"
}
```


1.2.6. OtpGetId2

ID²动态下发接口，用于ID²及其密钥的动态下发。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	OtpGetId2	系统规定参数，取值：OtpGetId2。
ApiVersion	String	是	1.1.2	API版本号，当前取值1.1.2。
DeviceAuthCode	String	是	0~0~******~********** ****123F872731B71 12387CDF~H6snGUr SJ**********rxPvm 6RzeNqYk/JtFLgLE yWf9XoR2X5IwJTN WWpZ8vot********** ****g6Bnjda0Zq61 +qnf6IR********** xIRHKhfnPY70/Tjlo 3cvJdmQm2eVU4P TrpISjYcYw91ULrO OUFM********** **********	设备认证码，由设备端SDK基于动态下发Token生成，具体生成方式，请参考 设备端API 。

返回数据

名称	类型	示例值	描述
Code	Integer	200	调用失败时会返回错误码。

名称	类型	示例值	描述
Data	Struct		返回数据。
RequestId	String	F6AFB45A-0FD1-405E-AD2A-C50E34C429E5	阿里云为该请求生成的唯一标识符。
Success	Boolean	true	是否调用成功。 • true：调用成功。 • false：调用失败。

示例

请求示例

```
http(s)://[Endpoint]/?Action=OtpGetId2
&ApiVersion=1.1.2
&DeviceAuthCode=0~0~******~************123F872731B7112387CDF~H6snGUrSJ************rxPvm6RzeNqYk
/JtFLgLEyWf9XoR2X5IwJTNWWpZ8vot************g6Bnjda0Zq61+qnf6IR************xIRHKhfnPY70/Tjlo3cvJd
mQm2eVU4PTrpISjYcYw91ULrOOUFM************************
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<OtpGetId2Response>
 <RequestId>F6AFB45A-0FD1-405E-AD2A-C50E34C429E5</RequestId>
 <Data>
    <Remaining>100</Remaining>
    <ProvisionData>ATAww****nNfro=</ProvisionData>
 </Data>
 <Code>200</Code>
 <Success>true</Success>
</OtpGetId2Response>
```

`JSON` 格式

```
{
  "RequestId": "F6AFB45A-0FD1-405E-AD2A-C50E34C429E5",
  "Data": {
    "Remaining": "100",
    "ProvisionData": "ATAww****nNfro="
  },
  "Code": "200",
  "Success": "true"
}
```

1.2.7. 设备端API

本文描述ID²接口在设备端如何使用。ID² Client SDK封装了底层对ID²载体和接口的操作细节，应用只需调用该接口进行相关操作，即可访问设备端的ID²服务。

初始化

- 函数原型：int id2_client_init(void)。
- 功能描述：ID² Client SDK初始化。使用ID²设备端其它API之前，需先调用该API进行初始化操作。
- 参数描述：无。
- 返回值：成功时返回0；其他返回值，请查看错误码。

获取ID²

- 函数原型：int id2_client_get_id(uint8_t* id, uint32_t* len)。
- 功能描述：获取ID²字符串。
- 参数描述：

参数	参数类型	描述
id	输出参数	存放ID²字符串的起始地址，长度不小于24个字节。
len	输入参数和输出参数	输入参数为ID²的buf长度，输出参数为ID²字符串的实际长度。

- 返回值：成功时返回0。

以挑战应答模式获取设备认证码

- 函数原型：int id2_client_get_challenge_auth_code(const char* challenge, const uint8_t* extra, uint32_t extra_len, uint8_t* auth_code, uint32_t* auth_code_len)。
- 功能描述：基于挑战应答模式生成设备端认证码。可选择携带额外数据extra，携带的extra数据参与设备认证码的签名运算。
- 参数描述：

参数	参数类型	描述
challenge	输入参数	挑战字起始地址，由SP Server从ID² Server获取。
extra	输入参数	可选参数。额外数据起始地址，由Device和SP Server根据某种约定分别生成。
extra_len	输入参数	额外数据长度，最大长度512字节。
auth_code	输出参数	设备认证码起始地址，长度不小于256字节。
auth_code_len	输入参数和输出参数	输入参数为authCode的buf长度，输出参数为设备认证码的实际长度。

- 返回值：成功时返回0。
- 交互流程图：

以时间戳模式获取设备认证码

- 函数原型：int id2_client_get_timestamp_auth_code(const char* timestamp, const uint8_t* extra, uint32_t extra_len, uint8_t* auth_code, uint32_t* auth_code_len)。
- 功能描述：基于时间戳模式生成设备端认证码，可选择携带额外数据extra，携带的extra数据参与设备认证码的签名运算。时间戳可以由设备端RTC生成，或者由SP Server携带发送给设备端。
- 参数描述：

参数	参数类型	描述
timestamp	输入参数	当前系统时间（从1970年1月1日午夜开始经过的毫秒数），时间戳格式，如1500954672653。
extra	输入参数	可选参数。额外数据起始地址，由Device和SP Server根据某种约定分别生成。
extra_len	输入参数	额外数据长度，最大长度512字节。
auth_code	输出参数	设备认证码起始地址，长度不小于256字节。
auth_code_len	输入参数和输出参数	输入参数为authCode的buf长度，输出参数为设备认证码的实际长度。

- 返回值：成功时返回0。
- 交互流程图：

ID²解密

- 函数原型：int id2_client_decrypt(const uint8_t* in, uint32_t in_len, uint8_t* out, uint32_t* out_len)。
- 功能描述：使用ID²解密指定的数据。
- 参数描述：

参数	参数类型	描述
in	输入参数	待解密的数据起始地址。数据如果是base64编码，需要先进行base64解码。
in_len	输入参数	待解密的数据长度，不超过512字节。
out	输出参数	解密后的数据起始地址。
out_len	输入参数和输出参数	输入参数为参数out的buf长度，输出参数为解密后的数据长度。

- 返回值：成功时返回0。

获取设备烧录状态

- 函数原型：irot_result_t id2_client_get_prov_stat(bool *is_prov)。
- 功能描述：获取设备端ID²的烧录状态。
- 参数描述：

参数	参数类型	描述
is_prov	输出参数	布尔类型，存放ID²的烧录状态。

- 返回值：成功时返回0。

获取设备动态下发认证码

- 函数原型：irot_result_t id2_client_get_otp_auth_code(const uint8_t *token, uint32_t token_len, uint8_t *auth_code, uint32_t *len)。
- 功能描述：获取设备端ID²动态下发的认证码。
- 参数描述：

参数	参数类型	描述
token	输入参数	ID² Server颁发的动态下发token，可在ID²控制台的产品详情页获取。
token_len	输入参数	ID²动态下发token的长度，固定值32字节。
auth_code	输出参数	ID²动态下发认证码的内存，长度小于256字节。
len	输入参数和输出参数	输入参数为authCode的内存大小，输出参数为认证码的实际长度。

- 返回值：成功时返回0。

烧录动态下发数据

- 函数原型：irot_result_t id2_client_load_otp_data(const uint8_t *otp_data, uint32_t len)。
- 功能描述：烧录动态下发数据到设备中。
- 参数描述：

参数	参数类型	描述
otp_data	输入参数	ID² Server下发的动态下发数据包。
len	输入参数	ID²动态下发数据包的长度。

- 返回值：成功时返回0。

1.2.8. 错误码

服务端错误码

错误码	错误含义
12	ID²已激活、已烧录
13	芯片不存在
14	签名验签、解密加密错误
15	烧录许可证不存在
16	配额不足
18	ID²不存在
21	认证码无效
32	ID²未激活
33	时间戳过期
34	时间戳认证码重复
35	挑战字无效
36	data无效
37	api版本无效
38	服务端认证码extra无效
39	设备端认证码extra无效
40	设备端挑战字无效
41	使用授权码无效
42	不支持该功能
43	ID²状态无效
44	ID²未完成烧录
45	ID²已暂停使用
46	ID²已废弃
47	用户无权操作
48	厂商编码无效
49	产品编码无效
50	ID²已过期

错误码	错误含义
51	产品不存在
52	厂商未注册
53	ID[2]下载数量受限
54	系统繁忙
55	空发未使能
56	无效指纹
57	重复的设备指纹
200	成功
500	参数错误
700	系统错误

设备端错误码

设备端错误码	返回值	描述
IROT_ERROR_GENERIC	-1	通用错误。
IROT_ERROR_BAD_PARAMETERS	-2	输入参数错误。
IROT_ERROR_SHORT_BUFFER	-3	数据缓冲长度不够。
IROT_ERROR_EXCESS_DATA	-4	数据长度超出请求。
IROT_ERROR_OUT_OF_MEMORY	-5	无法分配内存。
IROT_ERROR_COMMUNICATION	-7	通讯错误。
IROT_ERROR_NOT_SUPPORTED	-8	不支持的操作。
IROT_ERROR_NOT_IMPLEMENTED	-9	对应的操作没有实现。
IROT_ERROR_TIMEOUT	-10	通信超时。
IROT_ERROR_ITEM_NOT_FOUND	-11	没有找到ID[2]。

1.3. 固件安全检测API

1.3.1. API概览

IoT固件安全检测（FSS）提供的API包括提交检测任务、获取检测状态、获取检测报告、删除检测任务、查询授权信息、查询检测结果等。

IoT固件安全检测

API	描述
CreateScanTask	创建检测任务
QueryTaskStatus	查询任务检测状态，如果使用检测状态通知机制，可不调用该接口。
QueryTaskReport	查询任务检测结果。
DeleteScanTask	删除任务。
QueryLicense	查询当前用户的授权信息。
QueryTaskReportUrls	查询任务检测结果。

更多API参考：
1，API调用请参考《调用方式说明》
2，解析检测结果请参考《检测结果解析》
3，使用子账号调用API请参考《RAM鉴权》

1.3.2. RAM鉴权

在使用RAM账号调用 IoT固件安全检测（FSS）API前，需要主账号通过创建授权策略对RAM账号进行授权。在授权策略中，使用资源描述符（Alibaba Cloud Resource Name, ARN）指定授权资源。

可授权的资源类型

在进行RAM子账号授权时，FSS的权限描述如下：

权限策略名称	备注
AliyunFSSFullAccess	管理IoT固件安全检测（FSS）的权限。
AliyunFssReadOnlyAccess	只读访问IoT固件安全检测（FSS）的权限。

关于创建RAM用户的具体操作步骤，请参见创建RAM用户

1.3.3. CreateScanTask

创建一个固件安全检测任务。

请求参数

名称	类型	是否必选	示例值	说明
FwName	String	否	摄像头	固件名称。最大长度为50字节。
FwVersion	String	否	1.0	固件版本号。最大长度为50字节。
FwUrl	String	是	https://fss-store-bins.oss-cn-shanghai.aliyuncs.com/18/91fd9e86f53b4429b68b5266a9cebc78?Expires=1572506923&OSSAccessKeyId=LTxxxxxxxb31QS&Signature=f2deFrds/Aefd	固件文件存储的URL，供FSS检测时下载，目前FSS只支持FwUrl为阿里云OSS存储的链接地址。
Email	String	否	<用户名>@<公司域名>	用于接收固件检测完成通知的邮箱地址。最大长度为128字节。
FwId	String	否	dT7HdBSANYYc9chZRajy84eBCMQAP6MG	用户自定义固件ID，用于标识一个检测任务。可以为空。最大长度为64字节。
DetectionRuleCodes	String	否	CPoC2_3	检测规则，取值可为空值、“BASIC”、“CPoC2_2”、“CPoC2_3”。当不填写该字段或设置为空值时，系统默认以“BASIC”处理。“BASIC”表示基础检测）、“CPoC2_2”表示等保2.0二级检测、“CPoC2_3”表示等保2.0三级检测。

返回参数

名称	类型	示例值	说明
Code	Integer	200	错误码
Message	String	成功	返回提示信息
Data	Struct		返回数据

RequestId	String	A07E3902-B92E-44A6-B6C5-6AA1111111359	为公共参数，每个请求的ID都是唯一的，可用于排查和定位问题

Data结构说明

名称	类型	示例值	说明
TaskId	String	cfecd128592941f48c2da09492a47aec	检测任务唯一ID。

返回值

当Code为200时，表示检测任务创建成功。

1.3.4. QueryTaskStatus

用于查询检测任务当前的状态。该接口会逐步弃用，建议使用QueryTaskReportUrls接口获取检测报告。

请求参数

名称	类型	是否必选	示例值	说明
TaskId	String	是	cfecd128592941f48c2da0FFFFFFFFFFF	检测任务唯一ID。

返回参数

名称	类型	示例值	说明
Code	Integer	200	错误码。
Message	String	成功	返回提示信息。
Data	Struct	NULL	返回数据。
RequestId	String	A07E3902-B92E-44A6-B6C5-FFFFFFFFFFF	为公共参数，每个请求的ID都是唯一的，可用于排查和定位问题。

Data结构说明

名称	类型	示例值	说明
ScanStatus	String	SCAN_STATUS_SCAN_COMPLETE	检测状态名称。
Description	String		检测状态描述。
ScannedRate	String	91.3	已完成检测比例。

检测状态说明

状态值	说明
SCAN_STATUS_QUEUING	检测排队中。
SCAN_STATUS_DOWNLOADING	固件下载中。
SCAN_STATUS_DOWNLOAD_FAILED	固件下载失败，为检测结束状态。
SCAN_STATUS_PARSING	固件解析中。
SCAN_STATUS_PARSE_COMPLETE	固件解析完成。
SCAN_STATUS_SCANNING	固件检测中。
RPT_GENERATING	检测报告生成中。
RPT_GENERATE_FAILED	检测报告生成失败，为检测结束状态。
SCAN_STATUS_SCAN_EXCEPTION	固件检测失败，检测出现异常，为检测结束状态。
SCAN_STATUS_SCAN_COMPLETE	固件检测成功，检测完成，为检测结束状态。

说明 当Code为200时，表示查询成功。

当ScanStatus为“SCAN_STATUS_SCAN_COMPLETE”时，调用QueryTaskReport接口获取完整检测报告。

1.3.5. QueryTaskReportUrls

查询任务检测结果。

请求参数

名称	类型	是否必选	示例值	说明
TaskId	String	是	cfecd128592941f48c2da09492a47aec	检测任务唯一ID。

返回参数

名称	类型	示例值	说明
Code	Integer	200	错误码
Message	String	成功	返回提示信息

Data	Struct		返回数据
RequestId	String	A07E3902-B92E-44A6-B6C5-6AA1111111359	为公共参数，每个请求的ID 都是唯一的，可用于排查和定位问题

Data结构说明：

名称	类型	示例值	说明
PdfURL	String	http://report.oss-cn-shanghai.aliyuncs.com/baaac03062bbbeab96cf63cccc5f3ddd.pdf?Expires=1597659875	PDF检测报告URL地址链接。
JsonURL	String	http://report.oss-cn-shanghai.aliyuncs.com/baaac03062bbbeab96cf63cccc5f3ddd.json?Expires=1597659875	Json检测报告URL地址链接。
HtmlURL	String	https://fss.iot.aliyun.com/reporthtml/cfecd128592941f48c2da09492a47aec	HTML检测报告URL地址链接。

当Code为200时，表示查询成功。

1.3.6. QueryTaskReport

用于查询检测任务的结果。

请求参数

名称	类型	是否必选	示例值	说明
TaskId	String	是	cfecd128592941f48cFFFFFFFFFFFFF	检测任务唯一ID。

返回参数

名称	类型	示例值	说明
Code	Integer	200	错误码。返回200表示成功。

Message	String	成功	返回提示信息。
Data	Struct		返回数据。
RequestId	String	A07E3902-B92E-44A6-B6C5-FFFFFFFFFFFF	为公共参数，每个请求的ID 都是唯一的，可用于排查和定位问题。

Data结构说明：

名称	类型	示例值	说明
ReportJson	String	参考“检测结果格式”	检测结果，以JSON格式存储。详细格式见“检测结果格式”一节。
ReportUrl	String	https://fss.iot.aliyun.com/reporthtml/FFFFFFFFFFFF	检测报告URL地址。

说明 当Code为200时，表示查询成功。

1.3.7. DeleteScanTask

用于删除指定TaskId的检测任务。

请求参数

名称	类型	是否必选	示例值	说明
TaskId	String	是	cfecd128592941f48c2da09492a47aec	检测任务唯一ID。

返回参数

名称	类型	示例值	说明
Code	Integer	200	错误码。返回200表示成功；返回304表示未找到该任务。
Message	String	成功	返回提示信息。
RequestId	String	A07E3902-B92E-44A6-B6C5-6AA1111111359	为公共参数，每个请求的ID 都是唯一的，可用于排查和定位问题。

1.3.8. QueryLicense

查询主账号的授权信息。

请求参数

无。

返回参数

名称	类型	示例值	说明
Code	Integer	200	错误码。返回200表示成功。
Message	String	成功	返回提示信息。
Data	Struct		返回数据。
RequestId	String	A07E3902-B92E-44A6-B6C5-FFFFFFFFFFFF	为公共参数，每个请求的 ID 都是唯一的，可用于排查和定位问题。

Data结构说明：

名称	类型	示例值	说明
UsedCount	Integer	10	已使用检测授权数量。
MaxCount	Integer	100	总授权检测授权数量。
ScanningCount	Integer	1	正在检测中的任务数量（包括正在排队中的任务）。

1.3.9. QueryTaskReportUrls

查询任务检测结果。

请求参数

名称	类型	是否必选	示例值	说明
TaskId	String	是	cfecd128592941f48c2da09492a47aec	检测任务唯一ID。

返回参数

名称	类型	示例值	说明
Code	Integer	200	错误码

Message	String	成功	返回提示信息
Data	Struct		返回数据
RequestId	String	A07E3902-B92E-44A6-B6C5-6AA1111111359	为公共参数，每个请求的ID 都是唯一的，可用于排查和定位问题

Data结构说明：

名称	类型	示例值	说明
PdfURL	String	http://report.oss-cn-shanghai.aliyuncs.com/baaac03062bbbeab96cf63cccc5f3ddd.pdf?Expires=1597659875	PDF检测报告URL地址链接。
JsonURL	String	http://report.oss-cn-shanghai.aliyuncs.com/baaac03062bbbeab96cf63cccc5f3ddd.json?Expires=1597659875	Json检测报告URL地址链接。
HtmlURL	String	https://fss.iot.aliyun.com/reporthtml/cfecd128592941f48c2da09492a47aec	HTML检测报告URL地址链接。

当Code为200时，表示查询成功。

1.3.10. 检测结果格式说明

检测结果分为四部分固件基本信息、检测结果统计、基础检测风险列表、检测规则的检测结果。以JSON格式存储。总体格式如下：

```
{
  "fwBasicInfo": { //固件基本信息
    "fwFileSize": 0,
    "arch": "",
    "fwSha256": "",
    "filesystem": "",
    "scanStartTime": 0
  },
  "riskStatistics": { //检测结果统计
    "severityStatistics": [ //风险等级漏洞数统计
      { "name": "严重", "count": 0 },
      { "name": "高危", "count": 0 },
      { "name": "中危", "count": 0 },
      { "name": "低危", "count": 0 },
      { "name": "通告", "count": 0 }
    ],
    "riskBrief": [ //按风险等级及分类统计
      {
        "risks": [{"riskType": "", "suggestion": "", "count": 0, "description": ""}],
        "name": "严重", "count": 0
      }
    ]
  },
  "risks": [ //检测风险列表
    {
      "name": "", "code": "", "enCategory": "", "category": "",
      "description": "", "suggestion": "",
      "vulnInfos": [
        {
          "severity": "", "filename": "",
          "detail": [],
          "cve": {}
        }
      ]
    }
  ],
  "detectionRuleResult": [ //检测规则（等保2.0）的检测结果
    {
      "detectionUnitName": "", //控制点
      "attackSurface": "",    //控制面
      "category": "",       //分类
      "categoryCode": "",    //分类Code
      "detectionRuleName": "", //合规分级名称，如等保2.0三级
      "classification": "",   //合规分级Code
      "description": "",     //描述
      "severity": "",       //风险等级
      "detail": "",        //风险详情
      "solution": "",       //改进建议
      "status": ""         //合规检测状态。0表示不通过，1表示通过。
    }
  ]
}
```

固件基本信息

基本信息的key为“fwBasicInfo”，类型为JSONObject。格式为：

```
"fwBasicInfo": {
  "fwFileSize": 0,
  "arch": "",
  "fwSha256": "",
  "filesystem": "",
  "scanStartTime": 0
}
```

字段定义参考：

字段名	类型	说明
fwFileSize	long	固件文件大小
arch	String	固件的CPU体系
fwSha256	String	固件文件的SHA256
filesystem	String	固件的文件系统
scanStartTime	Date	检测时间

检测结果统计

检测结果统计的key为“riskStatistics”，类型为JSONObject。格式为：

```
"riskStatistics": {
  "severityStatistics": [ //风险等级漏洞数统计
    { "name": "严重", "count": 0 },
    { "name": "高危", "count": 0 },
    { "name": "中危", "count": 0 },
    { "name": "低危", "count": 0 },
    { "name": "通告", "count": 0 }
  ],
  "riskBrief": [ //按风险等级及分类统计
    {
      "risks": [{"riskType": "", "suggestion": "", "count": 0, "description": ""}],
      "name": "严重", "count": 0
    }
  ]
}
```

检测结果统计分为2部分：

1）风险等级漏洞数统计

按风险等级统计的当前固件风险结果，风险等级分为严重、高危、中危、低危和通告。

key为severityStatistics，类型为JSONArray，每个item格式为：

字段名	类型	说明
name	String	风险等级名
count	int	风险数量

2）按风险等级及分类统计

按风险等级及分类统计的当前固件风险结果，每个风险等级下检测到了哪些风险类型及风险数量等。

key为riskBrief，类型为JSONArray，每个item格式为：

字段名	类型	说明
name	String	风险等级名
count	int	风险数量
risks	JSONArray	风险列表

risks格式为：

字段名	类型	说明
riskType	String	风险类型名
count	int	风险数量
description	String	风险描述
suggestion	String	描述修复紧急程度

基础检测风险列表

检测结果统计的key为“risks”，类型为JSONArray，格式如下：

```
"risks": [
  {
    "name": "", "code": "", "enCategory": "", "category": "",
    "description": "", "suggestion": "",
    "vulnInfos": [
      {
        "severity": "", "filename": "",
        "detail": [],
        "cve": {}
      }
    ]
  }
]
```

字段定义如下：

字段	类型	描述
name	String	风险类型中文名称
code	String	风险类型英文名称或风险类型代码
category	String	风险类型分组中文名称
enCategory	String	风险类型分组英文名称
description	String	风险描述
suggestion	String	修复建议
vulnInfos	JSONArray<VulnInfo>	风险详细情况列表

VulnInfo定义如下：

字段	类型	描述
filename	String	受影响的文件名
severity	String	风险安全等级
detail	JSONArray	风险详细描述，不同风险描述也不相同
cve	JSONObject	CVE漏洞列表，只有当前分组为CVE漏洞(CveVuln)，该字段才有效。

1）detail字段

保存检测到的非CVE漏洞的详细信息，定义如下：

字段	类型	描述
cnName	String	详细信息的中文描述名
enName	String	详细信息的英文描述名
value	JSONArray<String>	风险详细信息，以JSONArray<String>存储。

2）cve字段

保存检测到的CVE漏洞的详细信息，定义如下：

字段	类型	描述
cveId	String	CVE漏洞标识
cweId	String	CVE漏洞类型标识

impact	JSONObject	定义漏洞攻击向量、风险等级等。优先使用cvssV3，其中vectorString为攻击向量、severity为风险等级。
cpe	JSONObject	为Common Platform Enumeration的缩写，描述软硬件产品的唯一标识符。 参考下面的cpe结构说明。
description	String	漏洞描述信息。
references	JSONArray<JSONObject>	参考信息。其中referenceType为参考类型，referenceSource为参考源，referenceUrl为参考链接地址。

cpe结构说明：

```
"cpe": {
 "cpeName": "",
 "cpeType": "",
 "allFixedVersion": {
  "expression": "",
  "version": ""
 },
 "highCriticalFixedVersion": {
  "expression": "",
  "version": ""
 }
}
```

其中cpeName为cpe名称，cpeType为cpe类型，

allFixedVersion为修复当前版本所有漏洞，需要升级到的版本建议。其中version表示版本号，expression表示version包含关系，如果为“>”表示需要升级到在大于version的版本；如果为”>=“表示需要升级到在大于或等于version的版本。如果version为""，表示目前还没有可用的修复版本号。

highCriticalFixedVersion，为修复当前版本中严重、高危漏洞，需要升级到的版本建议。

检测规则的检测结果

检测结果统计的key为“risks”，类型为JSONArray，格式如下：

```
"detectionRuleResult": [ //检测规则（等保2.0）的检测结果
  {
    "detectionUnitName": "", //控制点
    "attackSurface": "",    //控制面
    "category": "",        //分类
    "categoryCode": "",    //分类Code
    "detectionRuleName": "", //合规分级名称，如等保2.0三级
    "classification": "",   //合规分级Code
    "description": "",     //描述
    "severity": "",        //风险等级
    "detail": "",         //风险详情
    "solution": "",        //改进建议
    "status": ""          //合规检测状态。0表示不通过，1表示通过。
  }
]
```

字段定义如下：

字段	类型	描述
detectionUnitName	String	控制点
attackSurface	String	控制面
category	String	分类
categoryCode	String	分类Code
detectionRuleName	String	合规分级名称，如等保2.0三级
classification	String	合规分级Code
description	String	描述
severity	String	风险等级
detail	String	风险详情，为Json字符串。
solution	String	改进建议
status	String	合规检测状态。0表示不通过，1表示通过。

其中detail字段为为Json字符串，可解析为Json数组，数组每个成员结构为：

字段	类型	描述
riskName	String	检测到的风险名称
filename	String	受影响文件
riskDetail	String	检测到的风险详细

1.3.11. 错误码

IoT固件安全检测API对应的错误码：

错误码	描述
0	未分配
200	成功
300	没有可用的检测数量
307	文件大小超出最大限制
303	此文件已存在
304	未找到该任务
308	任务创建失败
350	账户状态异常
352	主账户未实人认证
356	不支持的账号类型
357	账号没有操作权限
450	无效的链接
453	任务检测未完成
500	服务异常
600	请求参数错误
601	不支持的文件类型

603	邮箱格式不正确
606	不支持的操作
900	未知错误

1.4. 安全RFID标签 API

1.4.1. 安全RFID标签-网页跳转

适用于安全RFID标签：Ntag424。

Ntag424目前仅支持NFC NDEF网页交互模式，使用支持NFC的手机靠近标签后会自动弹出网页。

数据交互说明

- 概述：用户在安全中心开通项目时填写要跳转的网址，产线生产时会写入标签中。用户手机端无需做任何开发集成，只需打开NFC，靠近RFID标签即可弹出网页。
- 跳转网址：详细要求参加下个章节。
- 服务端：用户需自行开发服务端，暴露Restful接口，确保在安全中心注册的网址可以正常访问。用户服务端对应接口的详细接口要求，参见下面章节。

跳转网址要求

- 流程：在NDEF模式下，手机与标签交互后，首先由阿里云服务端通过一些列密钥运算对标签进行安全认证，然后会进行重定向至客户注册的URL接口，并在session中携带相应数据。
- 用户服务端接口需确保在http session中接收参数。若是Java后台，需要用@ModelAttribute修饰，参见以下示例代码。
- 用户Restful接口对应的接收参数格式：

名称	类型	是否必选	示例值	描述
code	String	是	"00"	结果码。"00":成功；其他参照错误码
msg	String	是	"success"	结果描述。
tagId	String	是	"148EF48A4C3425"	标签的唯一ID
serviceId	String	是	"1234567890"	该项目的serviceId
authState	String	是	"00"	"00":认证成功；其他参照错误码

tokenValue	String	否	F8FA4F62BF2545E0ABEF9B1401D8D8CD	动态Token，每次认证后返回。可以在一定时间内通过Token查询该标签最近一次认证的结果。

用户的服务端接口示例代码

```
  @RequestMapping(path = "/index", method = RequestMethod.GET)
  public String index(
      @ModelAttribute("code") String code,
      @ModelAttribute("msg") String msg,
      @ModelAttribute("tagId") String tagId,
      @ModelAttribute("serviceId") String serviceId,
      @ModelAttribute("authState") String authState,
      @ModelAttribute("tokenValue") String tokenValue
  ) {
    return "code: " + code + ", msg: " + msg
        + "success: tagId: "  + tagId + ", serviceId: " + serviceId
        + ", authState: " + authState + ", tokenValue: " + tokenValue;
  }
}
```

1.4.2. 安全RFID标签-手机API

适用于安全RFID标签：EM4425芯片。

数据交互说明

- 交互链路：RFID -> NFC手机 <-> 客户服务端 <-> 阿里云
- 服务端：详细接口参考下面的章节。

> **说明**
>
> 手机端：近期会上线支付宝小程序插件，已支付宝小程序模式交互，可同时兼容安卓和IOS。

手机端接口

- 手机端提供的支付宝小程序插件即将上线，敬请期待。

服务端接口

- 接口形式：Restful
- 入口地址：
  https://iotx-trusted-tag-vpc-sh.iot.aliyun.com
- 接口地址：/em4425/svr/std/auth
- 功能描述：对阿里云发行的RFID标签（针对EM4425芯片）做安全认证
- 请求方式：POST
- 参数格式：application/json

- 参数说明：

名称	类型	是否必选	取值要求	示例值	描述
customerId	String	是	16位字符	"1434964027179725"	阿里云账号UID
sid	String	是	10位字符	"1234567890"	IoT安全中心创建的项目serviceId
uid	String	是	16位字符	"BEA0B1302D6D7CD7"	EM4425芯片的UID，为16位十六进制字符
challenge	String	是	20位字符	"73628192734959607082"	20位随机数字字符手机端请求芯片认证时输入的随机数。（会参与芯片的密钥安全运算）
cipherText	String	是	32位字符	"8C03F5587BD4082010A9BAE0B6965B19"	32位十六进制字符手机端请求芯片认证时，从芯片内返回的密文运算值。
info	String	否	无要求	"标签信息XXX"	此标签需上传的信息，如商品编号等。此字段由客户决定，不强制。

- 输入参数示例

```
{
  "customerId": "1434964027179725",
  "sid": "1234567890",
  "uid": "1EFE1508017816E0",
  "challenge": "73628192734959607082",
  "cipherText": "8C03F5587BD4082010A9BAE0B6965B19",
  "info": "test"
}
```

- 返回参数示例

```
{
  "code": "00", //结果码， "00"表示成功，其他值参照错误码表
  "msg": "success", //结果消息
  "data": {     // 当code为"00"时，data对象不为空
    "uid": "1EFE1508017816E0", //标签唯一ID，与请求参数中的uid一致。
    "authState": "00",       //认证结果，"00"表示成功，其他值参见错误码表
    "otp": "D1203447"        //一次性动态密码。可在一次性登陆芯片的安全隐私区，当手机远离标签后，该动态密码失效，需重新认证后获得
  },
  "failed": false,
  "success": true
}
```

1.4.3. 安全RFID标签-超高频读写器

适用于超高频芯片的通用接口，不局限于某款芯片。

数据交互说明

- 1. 超高频RFID只能与特殊的超高频读写设备交互，无法通过手机识别。
- 2. 阿里云提供服务端接口，由超高频读写器厂商进行集成和适配，已完成适配的读写器型号会陆续更新。
- 3. 提供四个能力：
  - 发行安全EPC
  - 认证安全EPC
  - 发行安全标识码
  - 认证安全标识码

设备端接口

无（由读写器设备商自行集成）

服务端接口

1）发行安全EPC

- 接口形式：Restful
- 入口地址：https://iotx-trusted-tag-vpc-sh.iot.aliyun.com
- 接口地址：/uhf/1/issue/96epc
- 功能描述：发行阿里云标准96bit 超高频RFID安全EPC，用于写入芯片的EPC区域
- 请求方式：POST
- 参数格式：application/json
- 参数说明：

名称	类型	是否必选	取值要求	示例值	描述

customerId	String	是	16位字符	"1434964027179725"	阿里云账号UID
serviceId	String	是	10位字符	"1234567890"	IoT安全中心创建的项目serviceId
tagId	String	是	24位字符	"D854A8559F7046D4BEF0432D"	超高频RFID的唯一ID（96bit），为24位十六进制字符

- 输入参数示例

```
{
  "customerId": "1114975502172915",
  "serviceId": "9623656710",
  "tagId": "E280B11C211078010815E297",
}
```

- 返回参数示例

```
{
  "code": "00",    //结果码， "00"表示成功，其他值参照错误码表
  "msg": "success", //结果消息
  "data": {       // 当code为"00"时，data对象不为空
    "encSessionKeyBase64": "vwm/hoaQ4ZaNCdr9fCdwWL2xHI0TUOkdgc2upi4vSBfO52dE8RBxy9J+rFVpsMIVdlSpd2w70dIVRinaet+V8OXIThTGkd1/ezr0D4FO2MPgVVNp1gAj0FzpYoRsuR/8ose88brzyy15pLcf41P6/jXIQaLGqG0TOn1LnYArJWU=", //被公钥经RSA1024算法加密后的session key
    "encDataBase64": "vCCaehK8kfopfKNARxe46z2zxWzBbYEWunHlr5Hswnm2cXHeU7lm7ckcn9zOT6Gq3yOlDLZWcuXE2kyGydnkfJiLPxcKZUvZrTS2llwwq4M4IE6UuZa/jQsCZWsr7pHja3qoGWgOJIzdLngps3XrvA==" //被session key经AES128算法加密后的EPC值
  },
  "failed": false,
  "success": true
}
```

2）认证安全EPC

- 接口形式：Restful
- 入口地址：https://iotx-trusted-tag-vpc-sh.iot.aliyun.com
- 接口地址：/uhf/1/auth/96epc
- 功能描述：对阿里云发行的标准96bit 超高频RFID安全EPC进行认证
- 请求方式：POST
- 参数格式：application/json
- 参数说明：

名称	类型	是否必选	取值要求	示例值	描述
customerId	String	是	16位字符	"1434964027179725"	阿里云账号UID
serviceId	String	是	10位字符	"1234567890"	IoT安全中心创建的项目serviceId
tagId	String	是	24位字符	"ACB0B7302D6C7CD7010A9BAE"	超高频芯片的UID，为24位十六进制字符
epc	String	是	24位字符	"A58C0BE6D3234386A8B9C76D"	阿里云发行的安全EPC，为24位十六进制数

- 输入参数示例

```
{
  "customerId": "1114975502172915",
  "serviceId": "9623656710",
  "tagId": "E280B11C211078010815E297",
  "epc": "1809EC8C211078010815E297",
}
```

- 返回参数示例

```
{
  "code": "00",     //结果码， "00"表示成功，其他值参照错误码表
  "msg": "success",    //结果消息
  "data": {         // 当code为"00"时，data对象不为空
    "tagId": "E280B11C211078010815E297", // 与请求参数中的tagId一致
    "epc": "1809EC8C211078010815E297"    // 与请求参数中的epc一致
  },
  "failed": false,
  "success": true
}
```

3）发行安全标识码

- 接口形式：Restful
- 入口地址：https://iotx-trusted-tag-vpc-sh.iot.aliyun.com
- 接口地址：/uhf/1/issue/32sc
- 功能描述：发行阿里云超高频RFID安全码，用于写入RFID芯片的user区域
- 请求方式：POST
- 参数格式：application/json

- 参数说明：

名称	类型	是否必选	取值要求	示例值	描述
customerId	String	是	16位字符	"1434964027179725"	阿里云账号UID
serviceId	String	是	10位字符	"1234567890"	IoT安全中心创建的项目serviceId
tagId	String	是	24位字符	"ACB0B7302D6C7CD7010A9BAE"	超高频芯片的UID，为24位十六进制字符

- 输入参数示例

```
{
  "customerId": "1114975502172915",
  "serviceId": "9623656710",
  "tagId": "E280B11C211078010815E297",
}
```

- 返回参数示例

```
{
  "code": "00",    //结果码，"00"表示成功，其他值参照错误码表
  "msg": "success",   //结果消息
  "data": {     // 当code为"00"时，data对象不为空
    "encSessionKeyBase64": "n52HM55wHJ97fz2I9iLUkFa9Cbf30lGClMOG/aw2Z+FUdDyBOtGLc47Ar+iGFn4wYlqX1k0MzK7BB80Tj61SB/PCcjjsFriM4qIKR4NNZqVnyL+IM3vNMcKhddFRILM5lrdNADtRPWpntupfZT+xNUIhjH5n0/V25YGY4hHyeFg=",  //被公钥经RSA1024算法加密后的session key
    "encDataBase64": "nLsZoR6GdjPmUw0vpzFhBTX+d8w71K3BDsgPgUFsuTcuO0e3PFdeLZ2ZRmPsOMRm88Cth1cNQi0u23TsC4lh19VQK89uo8ZzFWDhxrE6j7O8mvEy9Tw8uN0MsOm2ERZuTGEJrfcuq+ZQXr8TAQfxgQ==" //被session key经AES128算法加密后的secure code
  },
  "success": true,
  "failed": false
}
```

4）认证标识

接口形式：Restful

- 入口地址：

  https://iotx-trusted-tag-vpc-sh.iot.aliyun.com

- 接口地址：/uhf/1/auth/32sc
- 功能描述：对阿里云发行的标准超高频RFID安全码进行认证
- 请求方式：POST

- 参数格式：application/json
- 参数说明：

名称	类型	是否必选	取值要求	示例值	描述
customerId	String	是	16位字符	"1434964027179725"	阿里云账号UID
serviceId	String	是	10位字符	"1234567890"	IoT安全中心创建的项目serviceId
tagId	String	是	24位字符	"ACB0B7302D6C7CD7010A9BAE"	超高频芯片的UID，为24位十六进制字符
secureCode	String	是	8位字符	"1809EC8C"	阿里云发行的安全码，为8位十六进制数

- 输入参数示例

```
{
  "customerId": "1114975502172915",
  "serviceId": "9623656710",
  "tagId": "E280B11C211078010815E297",
  "secureCode": "1809EC8C"
}
```

- 返回参数示例

```
{
  "code": "00",
  "msg": "success",
  "data": {
    "tagId": "E280B11C211078010815E297",
    "secureCode": "1809EC8C"
  },
  "success": true,
  "failed": false
}
```

1.4.4. 错误码

本文介绍了安全RFID标签的错误码以及错误码的定义。

错误码	描述
1000	失败
1001	请求参数为空或空字符串
1002	返回值为空
1003	内部错误
1004	JSON解析错误
1005	超时
1006	license错误
1007	系统繁忙
1008	参数长度错误
2001	项目权限错误
3001	RFID芯片密钥类型不支持
3002	RFID芯片安全类型不支持
3003	项目服务已到期
3004	超过项目限制的使用次数上限
3005	超过单个标签使用次数限制的上限
3006	项目serviceId不存在
3007	标签tagId不存在
3008	customerId无权访问

3009	超过该项目设定的总标签个数上限
3031	数据库操作失败
3101	RFID标签认证失败
3102	Token方式认证失败
3103	Token过期，请重新认证并获取有效token
3104	token长度错误
3105	芯片指令解析错误
3106	动态密码长度错误（仅针对EM4425芯片）

2.IoT安全合作伙伴

2.1. 成为IoT安全合作伙伴

本文介绍了IoT安全的合作伙伴计划。申请成为合作伙伴安全RFID标签合作安全芯片合作

前提条件

IoT安全面向物联网企业开通合作伙伴申请，您的阿里云账号必须通过阿里云的“企业实名认证”。

> 说明
>
> 如果您是海外的企业，请联系我们

IoT安全合作伙伴

1，登录IoT安全中心控制台

2，单击右上角的“成为合作伙伴”。

3，成为合作伙伴后，您可以在一个或多个安全元器件上与IoT安全进行合作，IoT安全会根据实际项目的需求向阿里云的客户推荐您的安全元器件。

4，按照阿里云客户的需求，需要您在安全元器件中烧录IoT安全的数据（涉及到开发对接）。

安全RFID标签合作

安全RFID标签合作需要您提供安全RFID标签的产品规格说明书、完成与安全RFID标签服务的对接。

> 说明
>
> 如果您希望通过安全RFID标签进行合作，请联系我们

安全芯片合作

安全芯片合作需要您提供安全芯片的产品规格说明书、完成IoT安全产线工具的对接。

> 说明
>
> 如果您希望通过安全RFID标签进行合作，请联系我们

3.了解更多

钉钉扫码加群获得更多IoT安全产品、解决方案的信息。

4.相关协议

服务条款

- IoT设备身份认证的《通用服务条款》
- IoT安全运营中心的《通用服务条款》

服务等级协议

- IoT设备身份认证的《服务等级协议》适用于：安全防护-设备身份认证相关的功能。