

ALIBABA CLOUD

# 阿里云

IoT安全运营中心  
IoT安全运营中心公共云合集

文档版本：20210222

 阿里云

## 法律声明

阿里云提醒您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。
<b>粗体</b>	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 <code>cd /d C:/window</code> 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	<code>bae log list --instanceid</code> <i>Instance_ID</i>
[ ] 或者 [a b]	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
{ } 或者 {a b}	表示必选项，至多选择一个。	<code>switch {active stand}</code>

# 目录

1.API参考	05
1.1. IoT设备身份认证API	05
1.1.1. API概览	05
1.1.2. Verify	06
1.1.3. VerifyAndEncrypt	08
1.1.4. GetServerRandom	10
1.1.5. GetServerAuthCodeAndEncryptData	11
1.1.6. OtpGetId2	14
1.1.7. 设备端API	16
1.1.8. 错误码	19
1.2. 固件安全检测API	21
1.2.1. API概览	21
1.2.2. CreateScanTask	22
1.2.3. QueryTaskStatus	23
1.2.4. QueryTaskReportUrls	25
1.2.5. QueryTaskReport	26
1.2.6. DeleteScanTask	27
1.2.7. QueryLicense	27
1.2.8. QueryTaskReportUrls	28
1.2.9. 检测结果格式说明	29
1.2.10. 错误码	36
1.2.11. RAM鉴权	37
1.2.12. 获取AccessKey	37
2.了解更多	40
3.相关协议	41

# 1.API参考

## 1.1. IoT设备身份认证API

### 1.1.1. API概览

以下是ID<sup>2</sup>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	ID <sup>2</sup> 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	ID <sup>2</sup> Client SDK初始化，使用ID <sup>2</sup> 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ID <sup>2</sup> 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。

设备端API	描述
id2_client_decrypt	使用ID <sup>2</sup> 解密指定的数据。
id2_client_get_prov_stat	获取设备端ID <sup>2</sup> 的烧录状态。
id2_client_get_otp_auth_code	获取设备端ID <sup>2</sup> 动态下发的认证码。
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](#)。

## 1.1.2. Verify

ID<sup>2</sup> 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~Pf yt01tFpOFFI9HkHYjv qikoZpdch44U22ck mpvuwl1Qfpp00ll GboFmfyhnrX73hG vQ*****tW MFMt bm3t mEqUzf SnYvGAEu/+YcytuZ TKydh9jjLUVeRgKU eS2***** *****	设备端生成的认证码。
Id2	String	是	00AAABBB1112228 1FE15****	ID <sup>2</sup> 标识。
ProductKey	String	是	a1BwAGV****	产品标识，从ID <sup>2</sup> 控制台的产品列表获取。

名称	类型	是否必选	示例值	描述
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。 <ul style="list-style-type: none"> <li>true: 调用成功。</li> <li>false: 调用失败。</li> </ul>

## 示例

### 请求示例

```

http(s)://[Endpoint]/?Action=Verify
&ApiVersion=1.1.2
&AuthCode=3~0~*****~1499753003564~Pfy01tFpOFFl9HkHYjvqikoZpdch44U22ckmpvuwl1QfppOOll
GboFmfyhnrX73hGvQ*****tWMFMt3tmEqUzfSnYvGAEu/+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"
}
```

### 1.1.3. VerifyAndEncrypt

ID<sup>2</sup> 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~Pf yt01tFpOFFI9HkHYjv qikoZpdch44U22ck mpvuwl1Qfpp00ll GboFmfyhnrX73hG vQ*****tW MFMt bm3tmEqUzf SnYvGAEu/+YcytuZ TKydh9jjLUVeRgKU eS2***** *****	设备端生成的认证码。
Data	String	是	6F5DDB5F21C28F06 484A4695FAB9****	待加密的数据。



名称	类型	是否必选	示例值	描述
Id2	String	是	00AAABBB11122281FE15****	ID <sup>2</sup> 标识。
ProductKey	String	是	a1BwAGV****	产品标识，从ID <sup>2</sup> 控制台的产品列表中获得。
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	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。 <ul style="list-style-type: none"> <li>true: 调用成功。</li> <li>false: 调用失败。</li> </ul>

## 示例

### 请求示例

```

http(s)://[Endpoint]/?Action=VerifyAndEncrypt
&ApiVersion=1.1.2
&AuthCode=3~0~*****~1499753003564~Pfy01tFpOFFl9HkHYjvqikoZpdch44U22ckmpvuwl1QfppOOll
GboFmfyhnrX73hGvQ*****tWMFMtbn3tmEqUzfSnYvGAeU/+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"
}
```

## 1.1.4. 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****	ID <sup>2</sup> 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。

名称	类型	示例值	描述
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。 <ul style="list-style-type: none"> <li>true: 调用成功。</li> <li>false: 调用失败。</li> </ul>

## 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"
}
```

## 1.1.5. 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。
DeviceAuthCode	String	是	3~0~1245DC62694 6A9E5~149975300 3564~PfyT01tFpOF fI9HkHYjvqikoZpdc h44U22ckmpvuWl1 Qfpp00IIGboFmfy hnrX73hGvQ5BKzX 1Acie+8MHoKZ64Y 8tWMFMtmb3tmEq UzfSnYvGAEu/+Ycy tuZTKydh9ijjLUVeR gKUeS29q1zj9LOTy GBXOxdesb9n9oQ2 25****	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	00AAABBB1112228 1FE15****	ID <sup>2</sup> 标识。
ProductKey	String	是	a1BwAGV****	产品标识，从ID <sup>2</sup> 控制台的产品列表获取。
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。

名称	类型	是否必选	示例值	描述
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。
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。 <ul style="list-style-type: none"> <li>true: 调用成功。</li> <li>false: 调用失败。</li> </ul>

## 示例

### 请求示例

```

http(s)://[Endpoint]/?Action=GetServerAuthCodeAndEncryptData
&ApiVersion=1.1.2
&Data=abcd1234
&DeviceAuthCode=3~0~1245DC626946A9E5~1499753003564~Pfy01tFpOFFl9HkHYjqikoZpdch44U22ckmpvuw
L1QfppOOIIGboFmfyhnrX73hGvQ5BKzX1Acie+8MHoKZ64Y8tWMFMtbn3tmEqUzfSnYvGAEu/+YcytuZTKydh9ijJ
LUVeRgKUEs29q1zj9LOTyGBXOxdesb9n9oQ225****
&DeviceChallenge=ABCD12348
&Id2=00AAABBB1122281FE15****
&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"
}
```

## 1.1.6. OtpGetId2

ID<sup>2</sup>动态下发接口，用于ID<sup>2</sup>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。

名称	类型	是否必选	示例值	描述
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 <a href="#">设备端API</a> 。

## 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。
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。 <ul style="list-style-type: none"> <li>true: 调用成功。</li> <li>false: 调用失败。</li> </ul>

## 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"
}
```

## 1.1.7. 设备端API

本文描述ID<sup>2</sup>接口在设备端如何使用。ID<sup>2</sup> Client SDK封装了底层对ID<sup>2</sup>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ID<sup>2</sup>服务。

### 初始化

- 函数原型：int id2\_client\_init(void)。
- 功能描述：ID<sup>2</sup> Client SDK初始化。使用ID<sup>2</sup>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ID<sup>2</sup>

- 函数原型：int id2\_client\_get\_id(uint8\_t\* id, uint32\_t\* len)。
- 功能描述：获取ID<sup>2</sup>字符串。
- 参数描述：



参数	参数类型	描述
id	输出参数	存放ID <sup>2</sup> 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ID <sup>2</sup> 的buf长度，输出参数为ID <sup>2</sup> 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ID <sup>2</sup> 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。

参数	参数类型	描述
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：



## ID<sup>2</sup>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ID<sup>2</sup>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ID<sup>2</sup>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ID <sup>2</sup> 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ID<sup>2</sup>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	ID <sup>2</sup> Server颁发的动态下发token，可在ID <sup>2</sup> 控制台的产品详情页获取。
token_len	输入参数	ID <sup>2</sup> 动态下发token的长度，固定值32字节。
auth_code	输出参数	ID <sup>2</sup> 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	ID <sup>2</sup> Server下发的动态下发数据包。
len	输入参数	ID <sup>2</sup> 动态下发数据包的长度。

- 返回值：成功时返回0。

## 1.1.8. 错误码

### 服务端错误码

错误码	错误含义
12	ID <sup>2</sup> 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	ID <sup>2</sup> 不存在

错误码	错误含义
21	认证码无效
32	ID <sup>2</sup> 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	ID <sup>2</sup> 状态无效
44	ID <sup>2</sup> 未完成烧录
45	ID <sup>2</sup> 已暂停使用
46	ID <sup>2</sup> 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	ID <sup>2</sup> 已过期
51	产品不存在
52	厂商未注册
53	ID <sup>2</sup> 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

错误码	错误含义
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 <sup>2</sup> 。

## 1.2. 固件安全检测API

### 1.2.1. 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。

API	描述
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》

## 1.2.2. 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=LTxxxxxxx31Q5&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	dT7HdBSANYc9ch ZRajy84eBCMqAP6 MG	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-B6C5-6AA111111359	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	cfecd128592941f48c 2da09492a47aec	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。

## 1.2.3. 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	cfecd128592941f4 8c 2da0FFFFFFFFFFFF	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-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
ScanStatus	String	SCAN_STATUS_SCAN_CO MPLATE	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。

## 1.2.4. 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/baac03062bbbeab96cf63cccc5f3ddd.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。

## 1.2.5. 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	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。
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。

## 1.2.6. 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。

## 1.2.7. 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）。

## 1.2.8. 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。

## 1.2.9. 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

## 1.2.10. 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	主账户未实名认证
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

## 1.2.11. 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](#)


## 1.2.12. 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。

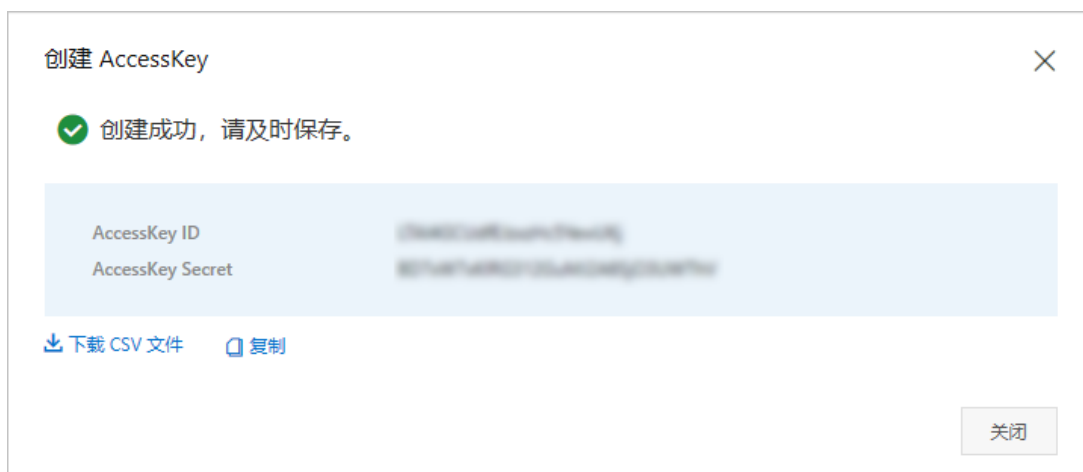
3. 在安全提示页面，选择获取主账号还是子账号的Accesskey。



4. 获取账号Accesskey。

o 获取主账号AccessKey

- 单击继续使用AccessKey。
- 在安全信息管理页面，单击创建AccessKey。
- 在手机验证页面，获取验证码，完成手机验证，单击确定。
- 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。



o 获取子账号AccessKey

- 单击开始使用子用户AccessKey。
- 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。
- 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。
- 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
- 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。



## 2. 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。

□



## 3.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。