阿里云
表格存储

授权管理

文档版本：20181115

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
	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。	 **禁止：** 重置操作将丢失用户配置数据。
	该类警示信息可能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，恢复业务所需时间约10分钟。
	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。	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。	单击 **确定**。
`courier` 字体	命令。	执行 `cd /d C:/windows` 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。	`swich` *`{stand | slave}`*

目录

1 RAM 和 STS 介绍

阿里云权限管理机制包括访问控制（Resource Access Management，简称 RAM）和安全凭证管理（Security Token Service，简称 STS），可以根据需求使用不同权限的子账号来访问表格存储，也支持为用户提供访问的临时授权。使用RAM和STS能极大地提高管理的灵活性和安全性。

RAM 的主要作用是控制账号系统的权限。通过使用 RAM 可以将在主账号的权限范围内创建子账号，给不同的子账号分配不同的权限，从而达到授权管理的目的。详情请参见访问控制产品帮助文档。

STS 是一个安全凭证（Token）的管理系统，用来授予临时的访问权限，这样就可以通过 STS 来完成对于临时用户的访问授权。

背景介绍

RAM 和 STS 需要解决的一个核心问题是如何在不暴露主账号的 AccessKey 的情况下安全地授权别人访问。因为一旦主账号的 AccessKey 暴露出去，会带来极大的安全风险，别人可以随意操作该账号下所有的资源、盗取重要信息等。

RAM 提供了一种长期有效的权限控制机制，通过分出不同权限的子账号，将不同的权限分给不同的用户，即使子账号泄露也不会造成全局的信息泄露。子账号在一般情况下是长期有效的。因此，子账号的 AccessKey 是不能泄露的。

相对于 RAM 提供的长效控制机制，STS 提供的是一种临时访问授权，通过 STS 可以返回临时的 AccessKey 和 Token，这些信息可以直接发给临时用户用来访问表格存储。一般来说，从 STS 获取的权限会受到更加严格的限制，并且拥有时间限制，因此即使这些信息泄露，对于系统的影响也很小。

基本概念

下表是一些基本概念的简单解释：

基本概念	描述
子账号	从阿里云的主账号中创建出来的子账号，在创建的时候可以分配独立的密码和权限，每个子账号拥有自己的 AccessKey，可以和阿里云主账号一样正常完成有权限的操作。一般来说，这里的子账号可以理解为具有某种权限的用户，可以被认为是一个具有某些权限的操作发起者。

基本概念	描述
角色（Role）	表示某种操作权限的虚拟概念，但是没有独立的登录密码和 AccessKey。子账号可以扮演角色，扮演角色时的权限是该角色自身的权限。
授权策略（Policy）	用来定义权限的规则，比如允许用户读取或者写入某些资源。
资源（Resource）	代表用户可访问的云资源，比如表格存储所有的实例、某个实例或者实例下面的某个表等。

子账号和角色可以类比为某个个人和其身份的关系，某人在公司的角色是员工，在家里的角色是父亲，在不同的场景扮演不同的角色，但是还是同一个人。在扮演不同角色的时候也就拥有对应角色的权限。单独的员工或者父亲概念并不能作为一个操作的实体，只有有人扮演了之后才是一个完整的概念。这里还可以体现一个重要的概念，那就是角色可以被多个不同的个人同时扮演。完成角色扮演之后，该个人就自动拥有该角色的所有权限。

使用示例：

某个阿里云用户，名为 alice，其在表格存储有 alice_a 和 alice_b 两个实例。alice 对这两个实例都拥有完全的权限。

为避免阿里云账号的 AccessKey 泄露而导致安全风险，alice 使用 RAM 创建了两个子账号 bob 和 carol。bob 对 alice_a 拥有读/写权限，carol 对 alice_b 拥有读/写权限。bob 和 carol 都拥有独立的 AccessKey，这样万一泄露了也只会影响其中一个实例，而且 alice 可以很方便地在控制台取消泄露用户的权限。

假设现在需要授权给别人读取 alice_a 中的数据表。这种情况下不应该直接把 bob 的 AccessKey 透露出去，可以新建一个角色，比如 AliceAReader，给这个角色赋予读取 alice_a 的权限。但请注意，这个时候 AliceAReader 还是没法直接用的，因为并不存在对应 AliceAReader 的 AccessKey，AliceAReader 现在仅仅表示一个拥有访问 alice_a 权限的虚拟实体。

为了能获取临时授权，这时可以调用 STS 的 AssumeRole 接口，告诉 STS bob 将要扮演 AliceAReader 这个角色。如果成功，STS 会返回一个临时的 AccessKeyId、AccessKeySecret 和 SecurityToken 作为访问凭证。将这个凭证发给需要访问的临时用户就可以获得访问 alice_a 的临时权限了。凭证的过期时间在调用 AssumeRole 的时候指定。

为什么 RAM 和 STS 这么复杂

RAM 和 STS 的概念之所以复杂，是为了权限控制的灵活性而牺牲了部分的易用性。

将子账号和角色分开，主要是为了将执行操作的实体和代表权限集合的虚拟实体分开。如果用户本身需要的权限很多，比如读/写权限，但是实际上每次操作只需要其中的一部分权限，那么我们就可以创建两个角色，分别具有读和写权限，然后创建一个没有任何权限但是可以拥有扮演这两个角色权限的用户。当用户需要读的时候就可以临时扮演其中拥有读权限的角色，写的时候同理，以降低每次操作中权限泄露的风险。而且通过扮演角色可以将权限授予其他的阿里云用户，更加方便了协同使用。

当然，提供了灵活性并不代表一定要使用全部的功能，应该根据需求来使用其中的一个子集。比如，不需要带过期时间的临时访问凭证的话，完全可以只使用 RAM 的子账号功能而无需使用 STS。

下面的章节会用范例提供一些 RAM 和 STS 的使用指南以及使用上的建议。示例在操作上会尽量使用控制台和命令行等操作方式，减少实际代码使用。如果需要使用代码来实现请参见 *RAM* 和 *STS* 的 API 手册。

2 子账号访问

子账号可以用于登录表格存储控制台，对实例和表进行管理，并且调用 SDK 接口，访问数据类 API 。

登录控制台

1. 使用主账号登录访问控制*RAM*控制台。
2. 单击页面左侧的**用户管理**，进入用户管理页面。

3. 找到需要开通登录控制台权限的子账号，并单击其右侧操作栏下面的**管理**按钮，进入用户详情页面。
4. 在**Web 控制台登录管理**栏中，单击**启用控制台登录**，进入重置密码页面。

5. 为子账号设置登录阿里云云产品控制台的密码，并单击**确定**。
6. 单击页面左侧的**概览**，进入 RAM 概览页面。
7. 单击**RAM 用户登录链接**后面的链接进入 RAM 用户登录页面，并使用步骤 5 中设置的子账号的密码登录阿里云云产品控制台。
8. 登录成功后，单击页面左侧**表格存储**图标，进入表格存储控制台。

启用多因素认证

多因素认证（Multi-Factor Authentication, MFA）是一种简单有效的最佳安全实践方法，它能够在用户名和密码之外再额外增加一层安全保护。启用 MFA 后，用户登录阿里云网站时，系统将要求输入用户名和密码（第一安全要素），然后要求输入来自其 MFA 设备的可变验证码（第二安全要素）。这些多重要素结合起来将为您的账户提供更高的安全保护。

操作步骤如下：

1. 使用主账号登录访问控制*RAM*控制台。
2. 单击页面左侧的**用户管理**，进入用户管理页面。
3. 找到需要开通登录控制台权限的子账号，并单击其右侧操作栏下面的的**管理**按钮，进入用户详情页面。
4. 在**多因素认证设备**栏中，单击**启用虚拟MFA设备**，并按照帮助流程启用 MFA 设备。
5. 成功启用 MFA 设备后，返回用户详情页面，在**Web 控制台登录管理**栏中，开启**必须开启多因素认证**。之后登录控制台时，就需要输入 MFA 设备的可变验证码。

调用 API

为子账号创建 AccessKey，在调用 SDK 接口时传入该 AccessKey，其它使用方式与主账号相同。

3 STS临时授权访问

上面章节只用到了 RAM 的子账号功能，这些子账号都是可以长期正常使用的，发生泄露后如果无法及时解除权限，会非常危险。

当开发者的 app 被用户使用之后，用户的数据要写入 ram-test-dev 这个实例。当 app 的用户数据很多时，要求能够安全地授权给众多的 app 用户上传数据，并且保证多个用户之间存储的隔离。

类似这种场景需要临时访问权限，应该使用 STS 来完成。STS 可以指定复杂的策略来对特定的用户进行限制，仅提供最小的权限。

创建角色

1. 创建一个名为 ram_test_app 的子账号，不需要赋予任何权限，因为在扮演角色的时候会自动获得被扮演角色的所有权限。有关创建RAM用户更多详情，请参见使用示例中的步骤1至步骤7。
2. 创建两个角色，RamTestAppReadOnly 和 RamTestAppWrite。一个用于读取等操作，一个用于上传文件的操作。
   a. 登录*RAM* 控制台。
   b. 选择角色管理 > 新建角色。
   c. 选择角色类型。这里选择用户角色。
   d. 填写类型信息。因为角色是被阿里云账号使用过的，因此选择默认的即可。然后单击下一步。
   e. 配置角色基本信息。本实例中角色名称填写 RamTestAppReadOnly，然后单击创建。
   f. 完成角色创建后，单击关闭。
3. 创建完角色之后，角色是没有任何权限的，因此需要新建一个自定义的授权策略。
   a. 选择策略管理 > 新建授权策略。
   b. 选择空白模板。
   c. 填写授权策略名称。该示例中填写ram-test-app-readonly，策略内容填写如下：

```
{
"Statement": [
 {
   "Effect": "Allow",
   "Action": [
  "ots:BatchGet*",
  "ots:Describe*",
  "ots:Get*",
  "ots:List*"
   ],
   "Resource": [
```

```
  "acs:ots:*:*:instance/ram-test-app",
  "acs:ots:*:*:instance/ram-test-app/table/*"
   ]
 }
],
"Version": "1"
}
```

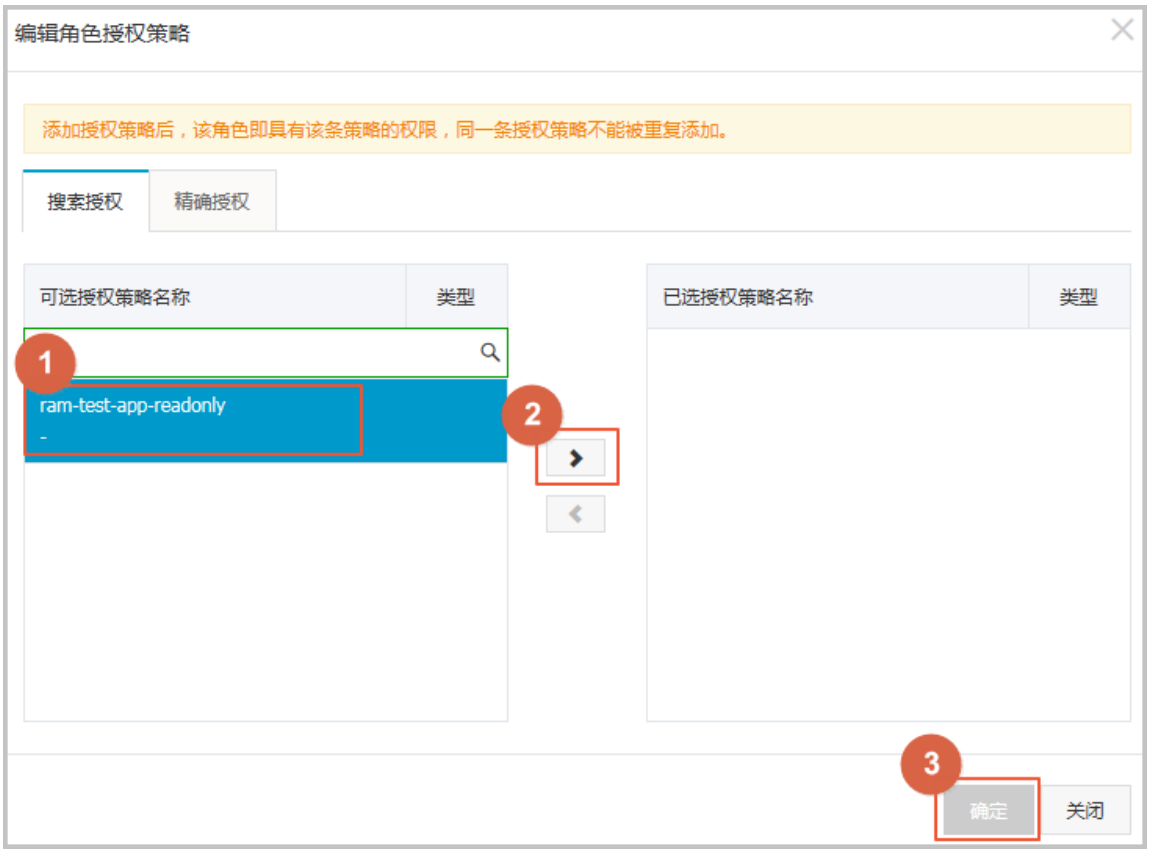
该策略内容表示对 ram-test-app 授予只读权限。

d. 单击**创建授权策略**，然后单击**关闭**。

4. 为 RamTestAppReadOnly 添加上 ram-test-app 的只读授权。

a. 在角色管理页面，单击 RamTestAppReadOnly 右侧操作栏中的**授权**按钮。

b. 将 ram-test-app-readonly 权限添加至右侧栏中。完成给该角色赋予对 ram-test-app 拥有只读的权限。



c. 单击确定。

表示已为 RamTestAppReadOnly 角色授予了 ram-test-app 的只读权限。

5. 参考步骤 2 ~ 步骤 4，建立一个 RamTestAppWrite 的角色，并赋予该角色写 ram-test-app 的自定义授权，步骤 3 中填写的策略内容如下：

```
{
"Statement": [
 {
   "Effect": "Allow",
   "Action": [
  "ots:Create*",
  "ots:Insert*",
  "ots:Put*",
  "ots:Update*",
  "ots:Delete*",
  "ots:BatchWrite*"
   ],
   "Resource": [
  "acs:ots:*:*:instance/ram-test-app",
  "acs:ots:*:*:instance/ram-test-app/table/*"
   ]
 }
],
"Version": "1"
```

```
}
```

在角色管理页面，可以看到已经新建好了 RamTestAppReadOnly 和 RamTestAppWrite 两个角色，分别表示了对于 ram-test-app 的读和写权限。

临时授权访问

创建角色后，就可以使用临时授权来访问表格存储了。

- 准备工作

  在使用 STS 来授权访问前，需要先对子账号进行需扮演角色的授权。若任意子账号都可以扮演这些角色，会带来不可预估的风险，因此有扮演对应角色需求的子账号需要被赋予相应的配置权限。在授权管理策略中新建两个自定义的授权策略并将其赋予 ram_test_app 这个子账号，操作步骤如下：

  1. 单击页面左侧的**策略管理**，进入策略管理页面。
  2. 单击**新建授权策略**按钮，进入创建授权策略的页面。
  3. 选择**空白模板**，进入创建自定义授权策略的页面。
  4. 填写授权策略名称，并将如下内容填写至策略内容栏。

     - AliyunSTSAssumeRolePolicy2016011401

     ```
     {
     "Version": "1",
     "Statement": [
         {
             "Effect": "Allow",
             "Action": "sts:AssumeRole",
             "Resource": "acs:ram:1983407596944237:role/ramtestapp
     readonly"
         }
     ]
     }
     ```

     - AliyunSTSAssumeRolePolicy2016011402

     ```
     {
     "Version": "1",
     "Statement": [
         {
             "Effect": "Allow",
             "Action": "sts:AssumeRole",
             "Resource": "acs:ram:1983407596944237:role/ramtestappwrite
     "
         }
     ]
     ```

```
}
```

> **说明：**
> 下文中 Resource 后面填写的内容表示某个角色 ID，角色的 ID 可以在角色管理页面，单击管理按钮，进入角色详情的页面中找到。

5. 上述的两个授权策略都建成后，单击页面左侧的用户管理，进入用户管理页面。
6. 单击 ram_test_app 右侧操作栏中的授权按钮。进入编辑个人授权策略页面。
7. 搜索AliyunSTSAssumeRolePolicy2016011401和AliyunSTSAssumeRolePolicy2016011402，选中后单击 > 将该权限添加至右侧栏中。
8. 单击确定，完成将这两个授权赋给 ram_test_app 这个账号的操作。

- 使用 STS

  上述准备工作完成后，就可以正式使用 STS 来进行授权访问。这里需要使用 STS 的 Python 命令行工具，该工具安装包的下载地址：*sts.py*。

  具体的调用方法如下，更详细的参数解释可以参考*STS* 帮助文档。

```
$python ./sts.py AssumeRole RoleArn=acs:ram::1983407596944237:role
/ramtestappreadonly RoleSessionName=usr001 Policy='{"Version":"1
","Statement":[{"Effect":"Allow","Action":["ots:ListTable","ots:
DescribeTable"],"Resource":["acs:ots:*:*:ram-test-app","acs:ots:*:*:
ram-test-app/*"]}]}' DurationSeconds=1000 --id=id --secret=secret
```

  参数说明：

参数	说明
RoleArn	指需要扮演的角色 ID。角色的 ID 可以在角色管理页面，单击管理按钮进入角色详情的页面中找到。
RoleSessionName	指临时凭证的名称，一般来说建议使用不同的应用程序用户来区分。
Policy	指在扮演角色时额外加上的一个权限限制。 **说明：** 这里传入的 Policy 是用来限制扮演角色之后的临时凭证的权限。最后临时凭证获得的权限是角色的权限和这里传入的 Policy 的交集。在扮演角色时传入 Policy 是为了灵活

参数	说明
	性，比如上传文件时可以根据不同的用户添加对于上传文件路径的限制。
DurationSeconds	指临时凭证的有效期。单位是秒，最小为 900，最大为 3600。
id and secret	指需要扮演角色的子账号的 AccessKey 的 AccessKeyID 和 AccessKeySecret。

- 测试 STS 的作用

  先在表格存储控制台创建名称为 test_write_read 的表，主键为 name，类型为 string，然后使用 CLI 工具测试读/写操作。

  使用 ram_test_app 这个子账号直接来访问，请将下面的 AccessKey 换成自己测试用的 AccessKey。

```
python2.7 ots_console --url https://TableStoreTest.cn-hangzhou.
ots.aliyuncs.com --id 6iTlV1uhiY71mlRt --key clkkuDiq69IJWJ7PnA9P
XJxhRWMr3P
You cannot access the instance!
ErrorCode: OTSNoPermissionAccess
ErrorMessage: You have no permission to access the requested
resource, please contact the resource owner.
```

  由于 ram_test_app 这个子账号没有访问权限，因此访问失败。

- 使用临时授权写入数据

  使用 STS 来写入数据。这里为了简单，传入的 Policy 和角色的 Policy 一致，过期时间使用默认的 3600s，SessionName 假定为 session001。操作步骤如下：

  1. 使用 STS 来获取临时凭证。

```
python2.7 ./sts.py AssumeRole RoleArn=acs:ram::1983407596944237
:role/ramtestappwrite RoleSessionName=session001 Policy='{"
Statement": [{"Effect": "Allow","Action": ["ots:Create*","ots:
BatchWrite*","ots:Put*","ots:Insert*","ots:Update*","ots:Delete
*"],"Resource": ["acs:ots:*:*:instance/ram-test-app","acs:ots:*:*:
instance/ram-test-app/table/*"]}],"Version": "1"}' --id=6iTlV1uhiY
71mlRt --secret=clkkuDiq69IJWJ7PnA9PXJxhRWMr3P
{
"AssumedRoleUser": {
    "Arn": "acs:ram::1983407596944237:role/ramtestappwrite/
session001",
    "AssumedRoleId": "330629052749595885:session001"
},
"Credentials": {
    "AccessKeyId": "STS.x4gG7KMsfHckQe8nPKLO",
    "AccessKeySecret": "IA6CJh5kE5J5m8mR6aQXWbMemSL63Xh7SIhrEcke
",
```

```
    "Expiration": "2016-01-14T07:58:14Z",
    "SecurityToken": "CAESgAQIARKAATDsbhiBSujhVEHoMKm1i17pyZhP
TCe1BnVF5YzdNyRos4WuQjalxLkOE/hNNxg25vTo9bljKg4VCcrfh6GkJNuj
MMcJ4V1i/0RMDLfXwa0/vOHP9W/oSQpwAD5EaWJfqVY/nxwmJ0aKJDHPmSieWssn
lmocaOZAgHkpCqQSSDA8GhhTVFMueDRnRzdLTXNmSGNrUWU4blBLTE8iEjMz
MDYyOTA1Mjc0OTU5NTg4NSoGdXNyMDAxMPnCkfmjKjoGUnNhTUQ1QuIBCgEx
GtwBCgVBbGxvdxJnCgxBY3Rpb25FcXVhbHMSBkFjdGlvbhpPCgtvdHM6Q3Jl
YXRlKgoPb3RzOkJhdGNoV3JpdGUqCghvdHM6UHV0KgoLb3RzOkluc2VydCoK
C290czpVcGRhdGUqCgtvdHM6RGVsZXRlKhJqCg5SZXNvdXJjZUVxdWFscxII
UmVzb3VyY2UaTgohYWNzOm90czoqOio6aW5zdGFuY2UvcmFtLXRlc3QtYXBw
CilhY3M6b3RzOio6KjppbnN0YW5jZS9yYW0tdGVzdC1hcHAvdGFibGUvKkoQ
MTk4MzQwNzU5Njk0NDIzN1IFMjY4NDJaD0Fzc3VtZWRSb2xlVXNlcmAAahIz
MzA2MjkwNTI3NDk1OTU4ODVyD3JhbXRlc3RhcHB3cml0ZQ==="
},
"RequestId": "5F92B248-F200-40F8-A05A-C9C7D018E351"
}
```

2. 使用 CLI 工具写入数据（版本 V1.2 开始支持 token 参数，待发布）。

```
 python2.7 ots_console --url https://TableStoreTest.cn-hangzhou
.ots.aliyuncs.com --id STS.x4gG7KMsfHckQe8nPKLO --key IA6CJh5kE5
J5m8mR6aQXWbMemSL63Xh7SIhrEcke --token=CAESgAQIARKAATDsbhiB
SujhVEHoMKm1i17pyZhPTCe1BnVF5YzdNyRos4WuQjalxLkOE/hNNxg25vTo
9bljKg4VCcrfh6GkJNujMMcJ4V1i/0RMDLfXwa0/vOHP9W/oSQpwAD5EaWJfqVY
/nxwmJ0aKJDHPmSieWssnlmocaOZAgHkpCqQSSDA8GhhTVFMueDRnRzdLTXNm
SGNrUWU4blBLTE8iEjMzMDYyOTA1Mjc0OTU5NTg4NSoGdXNyMDAxMPnCkfmj
KjoGUnNhTUQ1QuIBCgExGtwBCgVBbGxvdxJnCgxBY3Rpb25FcXVhbHMSBkFj
dGlvbhpPCgtvdHM6Q3JlYXRlKgoPb3RzOkJhdGNoV3JpdGUqCghvdHM6UHV0
KgoLb3RzOkluc2VydCoKC290czpVcGRhdGUqCgtvdHM6RGVsZXRlKhJqCg5S
ZXNvdXJjZUVxdWFscxIIUmVzb3VyY2UaTgohYWNzOm90czoqOio6aW5zdGFu
Y2UvcmFtLXRlc3QtYXBwCilhY3M6b3RzOio6KjppbnN0YW5jZS9yYW0tdGVz
dC1hcHAvdGFibGUvKkoQMTk4MzQwNzU5Njk0NDIzN1IFMjY4NDJaD0Fzc3Vt
ZWRSb2xlVXNlcmAAahIzMzA2MjkwNTI3NDk1OTU4ODVyD3JhbXRlc3RhcHB3cml0ZQ
==

 OTS-TableStoreTest>$ put test_write_read '001' age:integer=30
 A new row has been put in table test_write_read
```

- 使用临时授权读取数据

  使用 STS 来读取数据。这里为了简单，传入的 Policy 和角色的 Policy 一致，过期时间使用默认的 3600s，SessionName 假定为 session002。操作步骤如下：

  1. 使用 STS 来获取临时凭证。

```
python2.7 ./sts.py AssumeRole RoleArn=acs:ram::1983407596944237
:role/ramtestappreadonly RoleSessionName=session002 Policy='{"
Statement": [{"Effect": "Allow","Action": ["ots:BatchGet*","ots
:Describe*","ots:Get*","ots:List*"],"Resource": ["acs:ots:*:*:
instance/ram-test-app","acs:ots:*:*:instance/ram-test-app/table/
*"]}],"Version": "1"}' --id=6iTlV1uhiY71mlRt --secret=clkkuDiq69
IJWJ7PnA9PXJxhRWMr3P
{
"AssumedRoleUser": {
    "Arn": "acs:ram::1983407596944237:role/ramtestappreadonly/
session002",
    "AssumedRoleId": "396025752746614078:session002"
},
"Credentials": {
```

```
    "AccessKeyId": "STS.0qJ2UE8AalcHdQ6n2Q8Q",
    "AccessKeySecret": "pSaUjb8O9mU5M76nkC6FHt6wKwbCVYO27gxSEBAu",
    "Expiration": "2016-01-14T08:14:16Z",
    "SecurityToken": "CAES6wMIARKAAVtHeNgUnhk132OwDfxZTu8gPQCx
fakYLeWha/FxoEYNqBKQTtyI4WPC5mpYuu8+n+yamSYTI2VPQ/z44fcYCNT1
bQ0km87F3nb6EJxVvCdJIPNGVwQBMdQl/FLwBVhEGJ9BIwog4fMzwhERjqnAP8H
bynAIQpG55BHaIXmv53x+GhhTVFMuMHFKMlVFOEFhbGNIZFE2bjJROFEiEjM5
NjAyNTc1Mjc0NjYxNDA3OCoKc2Vzc2lvbjAwMjConMz5oyo6BlJzYU1ENULG
AQoBMRrAAQoFQWxsb3cSSwoMQWN0aW9uRXF1YWxzEgZBY3Rpb24aMwoNb3Rz
OkJhdGNoR2V0KgoNb3RzOkRlc2NyaWJlKgoIb3RzOkdldCoKCW90czpMaXN0
KhJqCg5SZXNvdXJjZUVxdWFscxIIUmVzb3VyY2UaTgohYWNzOm90czoqOio6
aW5zdGFuY2UvcmFtLXRlc3QtYXBwCilhY3M6b3RzOio6KjppbnN0YW5jZS9y
YW0tdGVzdC1hcHAvdGFibGUvKkoQMTk4MzQwNzU5Njk0NDIzN1IFMjY4NDJa
D0Fzc3VtZWRSb2xlVXNlcmAAahIzOTYwMjU3NTI3NDY2MTQwNzhyEnJhbXRl
c3RhcHByZWFkb25seQ=="
},
"RequestId": "EE788165-B760-4014-952C-E58ED229C80D"
}
```

2. 使用 CLI 工具读取数据（版本 V1.2 开始支持 token 参数，待发布）。

```
 python2.7 ots_console --url https://TableStoreTest.cn-hangzhou
.ots.aliyuncs.com --id STS.0qJ2UE8AalcHdQ6n2Q8Q --key pSaUjb8O9m
U5M76nkC6FHt6wKwbCVYO27gxSEBAu --token=CAES6wMIARKAAVtHeNgU
nhk132OwDfxZTu8gPQCxfakYLeWha/FxoEYNqBKQTtyI4WPC5mpYuu8+n+
yamSYTI2VPQ/z44fcYCNT1bQ0km87F3nb6EJxVvCdJIPNGVwQBMdQl/FLwBVhEGJ9
BIwog4fMzwhERjqnAP8HbynAIQpG55BHaIXmv53x+GhhTVFMuMHFKMlVFOEFh
bGNIZFE2bjJROFEiEjM5NjAyNTc1Mjc0NjYxNDA3OCoKc2Vzc2lvbjAwMjCo
nMz5oyo6BlJzYU1ENULGAQoBMRrAAQoFQWxsb3cSSwoMQWN0aW9uRXF1YWxz
EgZBY3Rpb24aMwoNb3RzOkJhdGNoR2V0KgoNb3RzOkRlc2NyaWJlKgoIb3Rz
OkdldCoKCW90czpMaXN0KhJqCg5SZXNvdXJjZUVxdWFscxIIUmVzb3VyY2Ua
TgohYWNzOm90czoqOio6aW5zdGFuY2UvcmFtLXRlc3QtYXBwCilhY3M6b3Rz
Oio6KjppbnN0YW5jZS9yYW0tdGVzdC1hcHAvdGFibGUvKkoQMTk4MzQwNzU5
Njk0NDIzN1IFMjY4NDJaD0Fzc3VtZWRSb2xlVXNlcmAAahIzOTYwMjU3NTI3
NDY2MTQwNzhyEnJhbXRlc3RhcHByZWFkb25seQ==

 OTS-TableStoreTest>: get test_write_read '001'
 age:INTEGER='30'
```

- 使用临时授权访问控制台

  STS 临时授权允许子账户登录表格存储控制台，并管理和查看主账号的实例和表资源。上面的例子中，子账号 ram_test_app 可以扮演 RamTestAppReadOnly 角色，从而拥有查看所有实例和所有表的权限。登录步骤如下：

  1. 使用主账号登录访问控制*RAM*控制台。
  2. 使用主账号登录访问控制RAM控制台，进入概览页面。
  3. 单击**RAM 用户登录链接**后面的链接，进入阿里云 RAM 用户登录页面，输入子账号的 ram_test_app 用户名及密码进行登录。
  4. 登录成功后，单击页面右上角的用户名，然后单击**切换身份**，进入角色切换页面。
  5. 输入企业姓名和角色名，然后单击**切换**。

使用临时授权调用 JAVA SDK

请参考以下方式创建 OTSClient 对象，传入 STS Token 的 AccessKeyId、AccessKeySecret 和 Token 等参数。

```
OTSClient client = new OTSClient(otsEndpoint, stsAccessKeyId,
stsAccessKeySecret, instanceName, stsToken);
```

总结

本章主要介绍了如何使用 STS 来临时授权用户访问表格存储。在典型的移动开发场景中，通过使用 STS，不同的 app 用户需要访问 app 时，可以通过获取到的临时授权来访问表格存储，临时授权可以指定过期时间，因此大大降低了泄露子账号信息的危害。在获取临时授权的时候，可以根据 app 用户的不同，传入不同的授权策略来限制用户的访问权限，比如限制用户访问的表路径，从而达到隔离不同 app 用户的存储空间的目的。

4 自定义权限

Action 定义

Action 是 API 的名称，可以根据 Action 设置开放或限制用户可以访问的 API。在创建表格存储的授权策略时，每个 Action 都需要添加`ots:`前缀，多个 Action 以逗号分隔，并且支持星号通配符（包括前缀匹配和后缀匹配）。

下面是一些典型的 Action 定义：

- 单个 API

```
"Action": "ots:GetRow"
```

- 多个 API

```
"Action": [
"ots:PutRow",
"ots:GetRow"
]
```

- 所有只读 API

```
"Action": [
"ots:BatchGet*",
"ots:Describe*",
"ots:Get*",
"ots:List*",
"ots:ComputeSplitPointsBySize"
]
```

- 所有读写 API

```
"Action": "ots:*"
```

Resource 定义

表格存储的资源由产品、地域、用户 ID、实例名和表名多个字段组成。每个字段支持星号通配符（包括前缀匹配和后缀匹配），格式如下。

```
acs:ots:[region]:[user_id]:instance/[instance_name]/table/[table_name]
```

其中，[xxx] 表示变量，产品固定为 ots，地域为英文缩写（如 cn-hangzhou, 详情请参考地域），用户 ID 为阿里云账号 ID。

说明：

表格存储中实例名称不区分大小写，上述 Resource 资源定义中的 [instance_name] 请用小写表示。

下面是一些典型的 Resource 定义：

- 所有地域的所有用户的所有资源

```
"Resource": "acs:ots:*:*:*"
```

- 华东 1 区域，用户 123456 的所有实例及其下面所有的表

```
"Resource": "acs:ots:cn-hangzhou:123456:instance/*"
```

- 华东 1 区域，用户 123456 的名称为 abc 的实例及其下面所有的表

```
"Resource": [
"acs:ots:cn-hangzhou:123456:instance/abc",
"acs:ots:cn-hangzhou:123456:instance/abc/table/*"
]
```

- 所有以 abc 开头的实例及下面的所有表

```
"Resource": "acs:ots:*:*:instance/abc*"
```

- 所有以 abc 开头的实例下面的所有以 xyz 开头的表（不包括实例资源，不匹配 acs:ots:*:*:instance/abc*）

```
"Resource": "acs:ots:*:*:instance/abc*/table/xyz*"
```

- 所有以 abc 结尾的 Instance 及下面的所有以 xyz 结尾的表

```
"Resource": [
"acs:ots:*:*:instance/*abc",
"acs:ots:*:*:instance/*abc/table/*xyz"
]
```

表格存储的 API 类型

目前表格存储包含以下两类 API：

- 实例读写相关的管理类 API
- 表和行读写相关的数据类 API

各 API 的类别详情如下表所示：

API 名称/Action	API 类别	说明
ListInstance	管理类	获取实例列表，仅控制台调用。

API 名称/Action	API 类别	说明
InsertInstance	管理类	创建实例，仅控制台调用。
GetInstance	管理类	获取实例信息，仅控制台调用。
DeleteInstance	管理类	删除实例，仅控制台调用。
ListTable	数据类	获取表的列表，控制台和 SDK 调用。
CreateTable	数据类	创建表，控制台 和 SDK 调用。
UpdateTable	数据类	更新表信息，控制台和 SDK 调用。
DescribeTable	数据类	获取表信息，控制台和 SDK 调用。
DeleteTable	数据类	删除表，控制台和 SDK 调用。
GetRow	数据类	读取一行数据，仅 SDK 调用。
PutRow	数据类	插入一行数据，仅 SDK 调用。
UpdateRow	数据类	更新一行数据，仅 SDK 调用。
DeleteRow	数据类	删除一行数据，仅 SDK 调用。
GetRange	数据类	范围读取数据，仅 SDK 调用。
BatchGetRow	数据类	批量读取多行数据，仅 SDK 调用。
BatchWriteRow	数据类	批量写入多行数据，仅 SDK 调用。

- 管理类 API 访问的资源

  管理类 API 主要为实例相关的操作，仅由控制台调用。对这类 API 的 Action和 Resource定义，将影响用户使用控制台。下面访问的资源省略了 `acs:ots:[region]:[user_id]:` 前缀，只描述实例和表部分。

API 名称/Action	访问的资源
ListInstance	instance/*
InsertInstance	instance/[instance_name]

API 名称/Action	访问的资源
GetInstance	instance/[instance_name]
DeleteInstance	instance/[instance_name]

- 数据类 API 访问的资源

  数据类 API 主要为表和行相关的操作，控制台和 SDK 都会调用，对这类 API 的 Action 和 Resource 定义，将影响用户使用控制台。下面访问的资源省略了 `acs:ots:[region]:[user_id]:` 前缀，只描述实例和表部分。

API 名称/Action	访问的资源
ListTable	instance/[instance_name]/table/*
CreateTable	instance/[instance_name]/table/[table_name]
UpdateTable	instance/[instance_name]/table/[table_name]
DescribeTable	instance/[instance_name]/table/[table_name]
DeleteTable	instance/[instance_name]/table/[table_name]
GetRow	instance/[instance_name]/table/[table_name]
PutRow	instance/[instance_name]/table/[table_name]
UpdateRow	instance/[instance_name]/table/[table_name]
DeleteRow	instance/[instance_name]/table/[table_name]
GetRange	instance/[instance_name]/table/[table_name]
BatchGetRow	instance/[instance_name]/table/[table_name]
BatchWriteRow	instance/[instance_name]/table/[table_name]

- 常见问题说明
  - Policy 里 Action 和 Resource 通过字符串匹配进行验证的，并且星号通配符区分前缀和后缀匹配。如果 Resource 定义为 acs:ots:*:*:instance/*/，则无法匹配 acs:ots:*:*:instance/abc。如果 Resource 定义为 acs:ots:*:*:instance/abc，则无法匹配 acs:ots:*:*:instance/abc/table/xyz。
  - 登录表格存储控制台管理实例资源，需要授予用户 acs:ots:[region]:[user_id]:instance/* 资源的读取权限，因为控制台需要获取实例的列表。
  - 对于批量操作 API（如 BatchGetRow 和 BatchWriteRow），后端服务会对被访问的每张表分别鉴权，只有所有表都通过鉴权才能执行操作，否则会返回权限错误。

Condition 定义

目前 Policy 支持访问 IP 限制、是否通过 HTTPS 访问、是否通过 MFA（多因素认证）访问和访问时间限制等多种鉴权条件，表格存储所有 API 都已经支持这些条件。

- 访问 IP 限制

  访问控制 RAM 可以限制访问表格存储的源 IP 地址，并且支持根据网段进行过滤。下面是一些典型的使用场景：

  - 限制多个 IP 地址。例如，只允许 IP 地址为 10.101.168.111 和 10.101.169.111 的请求访问。

```
{
"Statement": [
    {
        "Effect": "Allow",
        "Action": "ots:*",
        "Resource": "acs:ots:*:*:*",
        "Condition": {
            "IpAddress": {
                "acs:SourceIp": [
                    "10.101.168.111",
                    "10.101.169.111"
                ]
            }
        }
    }
],
"Version": "1"
}
```

  - 限制单个 IP 地址和 IP 网段。例如，只允许 IP 地址为 10.101.168.111 或 10.101.169.111/24 网段的请求访问。

```
{
"Statement": [
    {
        "Effect": "Allow",
        "Action": "ots:*",
        "Resource": "acs:ots:*:*:*",
        "Condition": {
            "IpAddress": {
                "acs:SourceIp": [
                    "10.101.168.111",
                    "10.101.169.111/24"
                ]
            }
        }
    }
],
"Version": "1"
```

```
}
```

- HTTPS 访问限制

  访问控制可以限制是否通过 HTTPS 访问。

  下面是典型的使用场景，限制请求必须通过 HTTPS 访问。

```
{
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ots:*",
            "Resource": "acs:ots:*:*:*",
            "Condition": {
                "Bool": {
                    "acs:SecureTransport": "true"
                }
            }
        }
    ],
    "Version": "1"
}
```

- MFA 访问限制

  访问控制可以限制是否通过 MFA (多因素认证) 访问。

  下面是典型的使用场景，限制请求必须通过 MFA 访问。

```
{
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ots:*",
            "Resource": "acs:ots:*:*:*",
            "Condition": {
                "Bool": {
                    "acs:MFAPresent ": "true"
                }
            }
        }
    ],
    "Version": "1"
}
```

- 访问时间限制

  访问控制可以限制请求的访问时间，即只允许或拒绝在某个时间点范围之前的请求。下面是典型的使用场景。

  例如，北京时间 2016 年 1 月 1 号凌晨之前用户可以访问，之后就不能再访问。

```
{
    "Statement": [
        {
```

```
            "Effect": "Allow",
            "Action": "ots:*",
            "Resource": "acs:ots:*:*:*",
            "Condition": {
                "DateLessThan": {
                    "acs:CurrentTime": "2016-01-01T00:00:00+08:00"
                }
            }
        }
    ],
    "Version": "1"
}
```

典型使用场景

结合上面对 Action、Resource 和 Condition 的定义，下面列出一些典型使用场景的 Policy 定义和授权方法。

- 多种授权条件

  对于访问 IP 地址为 10.101.168.111/24 网段的用户，可以对所有名称为 online-01 和 online-02 的实例执行读/写操作（包括实例下面的所有表），且要求只能在 2016-01-01 00:00:00 之前访问和通过 HTTPS 访问。

  操作步骤如下：

  1. 使用主账号登录访问控制 *RAM* 管理控制台。（默认已开通访问控制服务）
  2. 单击页面左侧的**策略管理**，进入策略管理页面。
  3. 单击右上角的**新建授权策略**按钮，进入创建授权策略的页面。
  4. 选择**空白模板**，进入创建自定义授权策略的页面。
  5. 填写**授权策略名称**，并将如下内容填写至**策略内容**栏。

```
{
"Statement": [
    {
        "Effect": "Allow",
        "Action": "ots:*",
        "Resource": [
            "acs:ots:*:*:instance/online-01",
            "acs:ots:*:*:instance/online-01/table/*",
            "acs:ots:*:*:instance/online-02",
            "acs:ots:*:*:instance/online-02/table/*"
        ],
        "Condition": {
            "IpAddress": {
                "acs:SourceIp": [
                    "10.101.168.111/24"
                ]
            },
            "DateLessThan": {
                "acs:CurrentTime": "2016-01-01T00:00:00+08:00"
            },
```

```
            "Bool": {
                "acs:SecureTransport": "true"
            }
        }
    }
],
"Version": "1"
}
```

6. 单击**新建授权策略**，授权策略新建成功，然后单击**关闭**。
7. 单击页面左侧的**用户管理**，进入用户管理页面将新建的策略授权给需要的子账号。
8. 找到需要授权的子账号，单击其右侧操作栏下面的**授权**按钮，进入编辑个人授权页面。
9. 搜索刚才新建的策略名称，选中后单击 > 以将该权限添加至已选择授权策略名称栏中，然后单击**确定**，完成对该账号的策略授权。

- 拒绝请求

  对于访问 IP 地址为 10.101.169.111 的用户，拒绝对北京区域，名称以 online 和 product 开头的实例下面的所有表执行写操作（不包括对实例的操作）。

  创建自定义授权策略和给子账号授权的步骤同上，只需将如下内容填写至**策略内容**栏即可。

```
{
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "ots:Create*",
                "ots:Insert*",
                "ots:Put*",
                "ots:Update*",
                "ots:Delete*",
                "ots:BatchWrite*"
            ],
            "Resource": [
                "acs:ots:cn-beijing:*:instance/online*/table/*",
                "acs:ots:cn-beijing:*:instance/product*/table/*"
            ],
            "Condition": {
                "IpAddress": {
                    "acs:SourceIp": [
                        "10.101.169.111"
                    ]
                }
            }
        }
    ],
    "Version": "1"
}
```

5 使用示例

本示例讲解如何利用子账号和授权做到不同账号间读/写权限分离。

当您需要向其他人共享自己账号下某个表格存储实例的数据，但又不希望数据被修改时，可以创建一个具有只读权限的子账号。具体操作步骤如下：

创建子账号

1. 使用主账号登录访问控制 *RAM* 管理控制台。
2. 在左侧导航栏，单击用户管理，进入用户管理页面。
3. 在页面右上角，单击新建用户，进入创建用户页面。
4. 填写账号信息并勾选为该用户自动生成 **AccessKey**，然后单击确定。

说明：

本示例中使用的用户名称是 ram_test。

5. 创建账号后会生成该账号的 AccessKey，单击保存 **AK** 信息。

> **说明：**
>
> AccessKey 创建后，无法再通过控制台查看。请您妥善保存 AccessKey，谨防泄露。

说明：

在用户管理页面，您还可以为该子账号**启用控制台登录**。

为子账号授权

1. 在用户管理页面，找到并单击 **ram_test** ，进入该子账号管理页面。
2. 在左侧导航栏中，单击**用户授权策略**。
3. 在页面右上角，单击**编辑授权策略**。
4. 在弹出的窗口中，搜索表格存储的权限。窗口右侧会出现相应的权限。
5. 选择权限，然后单击**>**，将该权限添加到右侧区域。然后单击**确定**。

说明：

本示例中我们为 ram_test 赋予AliyunOTS ReadOnlyAccess（只读访问表格存储的权限）。

说明：

在用户管理页面，您还可以为该子账号启用控制台登录。

测试示例

使用该账号的 AccessKey 测试创建表和删除表的权限。下面示例中使用了 ram_test 的 AccessKey，试验过程中请替换成您自己获取到的 AccessKey。

```
$python ots_console --url https://TableStoreTest.cn-hangzhou.ots
.aliyuncs.com --id ftWyMEYu1rBYTbWM --key u4qR5IGu5xJsvSO1y8mo
yC6n5vA7af

$OTS-TableStoreTest>: ct test pk1:string,pk2:integer readrt:1 writert:
1
Fail to create table test.

$OTS-TableStoreTest>: dt test
You will delete the table:test!

press Y (confirm) :Y
Fail to delete table test.
```

由上述示例可见，ram_test 没有建表和删表的权限。您可以通过以上方法为您的主账号创建一个只具有只读权限的子账号。