

阿里云 全局事务服务

API 参考

文档版本：20200512

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. 阿里云文档中所有内容，包括但不限于图片、架构设计、页面布局、文字描述，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。
	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。
	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。	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 Instance_ID</code>
[]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

法律声明.....	I
通用约定.....	I
1 使用 API 开启事务.....	1

1 使用 API 开启事务

GTS 提供一种非 Spring 框架下直接使用 API 开启事务的方式。

获取单例的全局事务操作对象

```
TxcTransaction txcTransaction = TxcTransaction.getInstance(vgroup, ak, sk)
```

其中 **TxcTransaction** 为单例对象，**vgroup** 为客户申请的事务分组，**ak/sk** 为用户的阿里云账号的 Access Key ID 和 Access Key Secret。

开启分布式事务

`txcTransaction.begin(timeout)`: **timeout** 为设置事务超时时间，单位毫秒。开启的事务上下文会绑定到当前线程中。

提交分布式事务

`txcTransaction.commit()`: 根据当前线程中的事务上下文提交分布式事务。

回滚分布式事务

`txcTransaction.rollback()`: 根据当前线程中的事务上下文回滚分布式事务。

另一组 API 在事务进行中操作事务上下文



说明:

可以参考 `txc-yun-sample` 中的 `#unique_4` 和 `#unique_5` 了解 API 的使用方式。

1. 使用 `String xid = TxcContext.getCurrentXid()`; 获取当前全局事务 XID。
2. 使用 `TxcContext.bind(xid, null)`; 手工将事务上下文绑定到当前线程中。
3. 使用 `TxcContext.unbind()`; 从当前线程中清除事务上下文，与 `bind` 操作对应。
4. 暂停及恢复全局事务传播。
 - `xid = TxcContext.suspendTxcTransaction()`: 临时挂起当前线程里的事务上下文。
 - `TxcContext.resumeTxcTransaction(xid)`: 将事务上下文恢复到当前线程，与 `suspendTxcTransaction` 操作对应。

```
@TxcTransaction
public void businessA()
{
    String xid = TxcContext.suspendTxcTransaction();
    // 调用 B
    TxcContext.resumeTxcTransaction(xid);
}
```

```
}
```

5. 完全手工控制全局事务边界。

```
public void myBusinessWithGlobalTransaction() {
    // 创建一个全局事务实例
    TxcTransaction tx = TxcTransaction.getInstance("myTxcVgroup", "myAK", "mySK");

    String xid = null;
    try {
        // 全局事务开启
        tx.begin(60000);
        // 建议把 xid 打印到应用日志中，方便问题排查
        xid = TxcContext.getCurrentXid();
        logger.info("GTS Tx Begins: " + xid);

    } catch (TxcException ex) {
        // 开启事务失败
        throw new MyBusinessException(ex);
    }

    try {
        // 业务逻辑

        // 全局事务提交
        tx.commit();
        logger.info("GTS Committed: " + xid);
    } catch (Throwable ex) {
        try {
            // 全局事务回滚
            tx.rollback();
            logger.info("GTS Rollbacked: " + xid);
        } catch (Throwable rollbackFailed) {
            // 打印回滚异常并记录 xid 以备数据核验
            logger.error("Failed to rollback: " + xid, rollbackFailed);
        }
    }

}
}
```