

# 阿里云 数据传输

## API概览

文档版本：20190918

# 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。	 <b>禁止：</b> 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。	 <b>警告：</b> 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。	 <b>说明：</b> 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
<b>粗体</b>	表示按键、菜单、页面名称等UI元素。	单击 <b>确定</b> 。
<code>courier</code> 字体	命令。	执行 <code>cd /d C:/windows</code> 命令，进入Windows系统文件夹。
<code>##</code>	表示参数、变量。	<code>bae log list --instanceid</code> <code>Instance_ID</code>
<code>[ ]</code> 或者 <code>[a b]</code>	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
<code>{ }</code> 或者 <code>{a b}</code>	表示必选项，至多选择一个。	<code>swich {stand   slave}</code>

# 目录

---

法律声明.....	I
通用约定.....	I
1 API概览.....	1
2 公共参数.....	3
3 数据订阅.....	5
3.1 配置订阅通道.....	5
3.2 启动订阅通道.....	8
3.3 新增消费组.....	10
3.4 修改消费组密码.....	11
3.5 查询消费组详情.....	13
3.6 删除消费组.....	15
4 数据同步.....	17
4.1 更改同步作业的数据库连接信息.....	17
4.2 查询更改数据库连接信息的任务执行状态.....	19
5 监控报警.....	21
5.1 配置迁移任务的监控报警.....	21
5.2 查询迁移任务的监控报警设置.....	23
5.3 配置订阅通道的监控报警.....	25
5.4 查询订阅通道的监控报警设置.....	27
5.5 配置同步作业的监控报警.....	29
5.6 查询同步作业的监控报警设置.....	31

# 1 API概览

数据传输服务DTS提供以下API接口。

## 数据迁移

API	描述
<a href="#">CreateMigrationJob</a>	购买数据迁移实例
<a href="#">ConfigureMigrationJob</a>	配置数据迁移任务
<a href="#">StartMigrationJob</a>	启动或重启数据迁移任务
<a href="#">SuspendMigrationJob</a>	暂停数据迁移任务
<a href="#">StopMigrationJob</a>	结束数据迁移任务
<a href="#">DescribeMigrationJobStatus</a>	查询数据迁移任务状态
<a href="#">DescribeMigrationJobDetail</a>	查询数据迁移任务详情
<a href="#">DeleteMigrationJob</a>	删除数据迁移实例

## 数据订阅

API	描述
<a href="#">CreateSubscriptionInstance</a>	购买订阅实例
<a href="#">ConfigureSubscriptionInstance</a>	配置订阅通道
<a href="#">StartSubscriptionInstance</a>	启动订阅通道
<a href="#">ModifySubscriptionObject</a>	修改订阅对象
<a href="#">ModifyConsumptionTimestamp</a>	修改消费时间点
<a href="#">CreateConsumerGroup</a>	新增消费组
<a href="#">ModifyConsumerGroupPassword</a>	修改消费组密码
<a href="#">DescribeConsumerGroup</a>	查询消费组详情
<a href="#">DescribeSubscriptionInstances</a>	查询订阅实例列表
<a href="#">DescribeSubscriptionInstanceStatus</a>	查询订阅通道状态
<a href="#">DeleteConsumerGroup</a>	删除消费组
<a href="#">DeleteSubscriptionInstance</a>	释放订阅实例

## 数据同步

API	描述
<a href="#">CreateSynchronizationJob</a>	购买同步实例
<a href="#">ConfigureSynchronizationJob</a>	配置同步作业
<a href="#">StartSynchronizationJob</a>	启动同步作业
<a href="#">SuspendSynchronizationJob</a>	暂停同步作业
<a href="#">DescribeSynchronizationJobs</a>	查询同步作业列表
<a href="#">DescribeSynchronizationJobStatus</a>	查询同步作业状态
<a href="#">ModifySynchronizationObject</a>	修改同步对象
<a href="#">DescribeSynchronizationObjectModifyStatus</a>	查询同步对象变更状态
<a href="#">DeleteSynchronizationJob</a>	删除同步实例
<a href="#">SwitchSynchronizationEndpoint</a>	更改同步作业的数据库连接信息
<a href="#">DescribeEndpointSwitchStatus</a>	查询更改数据库连接信息的任务执行状态

## 监控报警

API	描述
<a href="#">ConfigureMigrationJobAlert</a>	配置迁移任务的监控报警
<a href="#">DescribeMigrationJobAlert</a>	查询迁移任务的监控报警设置
<a href="#">ConfigureSubscriptionInstanceAlert</a>	配置订阅通道的监控报警
<a href="#">DescribeSubscriptionInstanceAlert</a>	查询订阅通道的监控报警设置
<a href="#">ConfigureSynchronizationJobAlert</a>	配置同步作业的监控报警
<a href="#">DescribeSubscriptionInstanceAlert</a>	查询同步作业的监控报警设置

## 更多信息

- [API资源导航](#)
- [API错误中心](#)

## 2 公共参数

公共参数是指所有接口在调用时都需要用到的参数，包含公共请求参数和公共返回参数。

### 公共请求参数

名称	类型	是否必选	描述
Format	String	否	返回值的类型，支持JSON和XML，默认为XML。
Version	String	是	OpenAPI版本号，版本号为日期格式：YYYY-MM-DD。当前版本号为2018-08-01。
AccessKeyID	String	是	阿里云颁发给用户的访问服务所用的密钥ID。
Signature	String	是	签名结果串。
SignatureMethod	String	是	签名方式，目前仅支持HMAC-SHA1。
Timestamp	String	是	请求的时间戳，格式为yyyy-MM-dd'THH:mm:ssZ（UTC时间）。
SignatureVersion	String	是	签名算法版本，目前版本为1.0。
SignatureNonce	String	是	唯一随机数，用于防止网络重放攻击。不同的请求需使用不同的随机数值。

### 公共返回参数

用户发送的每次接口调用请求，无论成功与否，系统都会返回一个唯一识别码（RequestId）给用户，可用于跟踪请求的执行情况。

### 返回示例

DTS OpenAPI使用标准的HTTP状态码表示请求处理的结果，具体状态码如下所示：

- 返回的HTTP状态码为2xx，代表调用成功。
- 返回的HTTP状态码为4xx或5xx，代表调用失败。

数据格式主要有XML和JSON两种，默认为XML格式，外部系统可以在请求时指定返回的数据格式。为便于查看，本文档中的返回示例做了格式化处理，实际返回结果是没有进行换行、缩进等处理的。

### 成功结果

- XML示例

```
<?xml version="1.0" encoding="utf-8"?>
<!--结果的根结点-->
<接口名称+Response>
  <!--返回请求标签-->
  <RequestId>4C467B38-3910-447D-87BC-AC049166F216</RequestId>
  <!--返回结果数据-->
</接口名称+Response>
```

- JSON示例

```
{
  "RequestId": "4C467B38-3910-447D-87BC-AC049166F216",
  /* 返回结果数据 */
}
```

### 错误结果

调用出错后，HTTP请求会返回一个4xx或5xx的HTTP状态码。返回的消息中包含具体的错误代码及错误信息，还包含一个全局唯一的请求ID（RequestId）和一个本次请求访问的站点ID（HostId）。



#### 说明:

您可以根据返回的错误码来定位错误原因，详情请参见[错误码](#)。若您无法定位错误原因，请联系阿里云客服人员，并提供调用出错返回的HostId和RequestId，以便我们能尽快帮您解决问题。

- XML示例

```
<?xml version="1.0" encoding="UTF-8"?>
<Error>
  <RequestId>8906582E-6722-409A-A6C4-0E7863B733A5</RequestId>
  <HostId>dts.aliyuncs.com</HostId>
  <Code>UnsupportedOperation</Code>
  <Message>The specified action is not supported.</Message>
</Error>
```

- JSON示例

```
{
  "RequestId": "7463B73D-35CC-4D19-A010-6B8D65D242EF",
  "HostId": "dts.aliyuncs.com",
  "Code": "UnsupportedOperation",
  "Message": "The specified action is not supported."
}
```

## 3 数据订阅

### 3.1 配置订阅通道

调用ConfigureSubscriptionInstance接口配置订阅通道。

调用本接口时，订阅通道需处于未配置状态。



说明:

如您暂未拥有订阅通道，您可以通过调用CreateSubscriptionInstance接口创建。

#### 注意事项

- 在成功调用StartSubscriptionInstance接口启动订阅通道之前，您可以重复调用本接口修改订阅通道的配置。
- 如果传入了SubscriptionInstanceNetworkType参数则订阅通道被定义为新版，如果不传入该参数则订阅通道被定义为旧版。



说明:

- 旧版订阅通道支持订阅RDS for MySQL、DRDS的数据。
- 新版订阅通道支持订阅RDS for MySQL、POLARDB for MySQL和Oracle的数据。

#### 请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	ConfigureSubscriptionInstance	要执行的操作，取值：ConfigureSubscriptionInstance。
SubscriptionInstanceId	String	是	dtsxxxxxxxx	订阅实例ID。
SourceEndpointInstanceId	String	是	rm-mxxxxxxxx	待订阅的实例ID。

名称	类型	是否必选	示例值	描述
SubscriptionObjects	List	是	<pre> [{"DBName":"dtstestdata",   "TableIncludes":   [{"TableName":"customer"}]}] </pre>	订阅对象，格式为JSON串，例如： <pre> [   {     "DBName":"dtstestdata",     "TableIncludes":[       {         "TableName":"customer"       }     ]   } ] </pre>
SubscriptionInstanceName	String	否	测试MySQL	订阅通道名称。
SourceEndpointInstanceType	String	否	RDS	数据源的实例类型，取值： <ul style="list-style-type: none"> <li>· RDS：RDS实例</li> <li>· POLARDB：POLARDB for MySQL集群</li> </ul> <div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 5px; margin-top: 10px;">  说明： 默认取值为RDS。         </div>
SubscriptionInstanceNetworkType	String	否	vpc	订阅通道的网络类型，取值： <ul style="list-style-type: none"> <li>· classic：经典网络</li> <li>· vpc：专有网络</li> </ul> <div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 5px; margin-top: 10px;">  说明：           <ul style="list-style-type: none"> <li>· 如果传入了本参数则订阅通道被定义为新版，如果不传入该参数则订阅通道被定义为旧版。</li> <li>· 旧版订阅通道支持订阅RDS for MySQL、DRDS和Oracle的数据，新版订阅通道支持订阅RDS for MySQL和Oracle的数据。</li> <li>· 本参数传入的值为vpc时，您还需要传入SubscriptionInstance.VPCId和SubscriptionInstance.VSwitchID参数。</li> </ul> </div>

名称	类型	是否必选	示例值	描述
SubscriptionInstance.VPCId	String	否	vpc-bpxxxxxxx	订阅通道的专有网络（VPC）ID。   说明： 当SubscriptionInstance.NetworkType参数值为vpc时，本参数才可用。
SubscriptionInstance.VSwitchID	String	否	vsw-bpxxxxxxx	订阅通道的虚拟交换机ID。   说明： 当SubscriptionInstance.NetworkType参数值为vpc时，本参数才可用。
SubscriptionDataType.DML	Boolean	否	True	是否订阅DML类型的数据，取值： <ul style="list-style-type: none"> <li>· True：订阅DML类型的数据</li> <li>· False：不订阅DML类型的数据</li> </ul>  说明： 默认取值为True。

### 返回参数

名称	类型	示例值	描述
RequestId	String	36923CC2-DDAB-4B48-A144-DA92C1E19537	请求ID。

### 示例

```
public void configureSubscriptionJob(String subscriptionJobId,
String jobName, String region, String rdsInstanceId, String rdsUsername,
String rdsPassword, String netType) throws Exception {
    ConfigureSubscriptionInstanceRequest request = new ConfigureSubscriptionInstanceRequest();

    // The subscription jobid created before
    request.setSubscriptionInstanceId(subscriptionJobId);

    // The subscription job name
    request.setSubscriptionInstanceName(jobName);

    request.setSourceEndpointInstanceType("RDS");
    request.setSourceEndpointUserName(rdsUsername);
    request.setSourceEndpointPassword(rdsPassword);
    request.setSourceEndpointInstanceID(rdsInstanceId);

    // Subscription DDL update
    request.setSubscriptionDataTypeDDL(Boolean.TRUE);
    // Subscription DML update
```

```

        request.setSubscriptionDataTypeDML(Boolean.TRUE);

        request.setSubscriptionObject("[{\"DBName\":\"portal_test\", \"
NewDBName\":\"portal_test\", \"TableIncludes\": [{\"TableName\":\"test
\", \"NewTableName\":\"test\"}]}]");

        request.setSubscriptionInstanceNetworkType(netType);

        ConfigureSubscriptionInstanceResponse response = client.
getAcqsResponse(request);
        if (!"true".equals(response.getSuccess())) {
            throw new Exception(response.getErrorMessage() + " requestId
=" + response.getErrCode());
        }
    }
}

```

### 错误码

错误码	错误信息	描述
InvalidParameter	The Input Parameter\${parameter} is not valid.	输入的参数不合法，请检查输入的参数是否正确。
InvalidParameterValid	The Value of Input Parameter \${parameter} is not valid.	参数值校验不通过，请检查是否缺失所提示的参数以及用户AK是否正确。
MissingParameter	The parameter "<parameter name>" is needed but no provided.	缺少参数，请检查是否缺失所提示的参数以及用户AK是否正确，具体请参照DTS OpenAPI文档。
Forbidden.InstanceNotFound	The specified Instance Not Found.	RDS实例不存在或当前用户没有该实例的操作权限，请检查实例ID或权限是否正确。
OperationDenied.DBInstanceType	The operation is not permitted due to the instance is Read only.	RDS实例是只读实例，不允许执行当前操作。
OperationDenied.DBInstanceStatus	The operation is not permitted due to status of instance.	实例状态不允许执行当前操作，请检查当前的实例状态是否正确。

## 3.2 启动订阅通道

调用StartSubscriptionInstance接口启动订阅通道。

调用本接口时，订阅通道需处于未启动、启动失败或暂停状态。

## 请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	StartSubscriptionInstance	要执行的操作，取值：StartSubscriptionInstance。
SubscriptionInstanceId	String	是	dtsxxxxxxxxx	订阅实例ID。

## 返回参数

名称	类型	示例值	描述
RequestId	String	36923CC2-DDAB-4B48-A144-DA92C1E19537	请求ID。

## 错误码

错误码	错误信息	描述
InvalidParameter	The Input Parameter\${parameter} is not valid.	输入的参数不合法，请检查输入的参数是否正确。
InvalidParameterValid	The Value of Input Parameter \${parameter} is not valid.	参数值校验不通过，请检查是否缺失所提示的参数以及用户AK是否正确。
MissingParameter	The parameter "<parameter name>" is needed but no provided.	缺少参数，请检查是否缺失所提示的参数及用户AK是否正确，具体请参照DTS OpenAPI文档。
OperationDenied.JobStatus	The operation is not permitted due to Migration job status is \${status}.	任务处于提示中状态时，不允许操作，请检查任务状态，预检查通过状态下才允许其他操作。
InvalidInstanceType	The type of RDS is \${instanceType}. Data Transmission can not support it.	RDS实例是只读实例，不允许当前操作。
Forbidden.InstanceNotFound	Instance is not found or does not belong to this account.	实例不存在，或者当前用户没有该实例的操作权限。

### 3.3 新增消费组

调用CreateConsumerGroup接口为订阅通道新增消费组。

- 本接口仅适用于新版订阅通道。



说明:

在调用ConfigureSubscriptionInstance接口时，需要传入SubscriptionInstance NetworkType参数，如果不传入该参数则订阅通道被定义为旧版。

- 只有订阅通道配置完成后，才能调用本接口。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	CreateConsumerGroup	要执行的操作，取值：CreateConsumerGroup。
SubscriptionInstanceId	String	是	dtsxxxxxxxxx	订阅通道ID。
ConsumerGroupName	String	是	测试订阅组	消费组名称。建议配置具有业务意义的名称，便于后续识别。
ConsumerGroupName	String	是	dtstest	消费组的账号。 · 由大写字母、小写字母、数字、下划线中的任意一种或多种组成。 · 最长16个字符。
ConsumerGroupPassword	String	是	Test123456	该消费组账号对应的密码。 · 由大写字母、小写字母、数字、特殊字符中的任意两种或两种以上组成。 · 密码长度为8-32个字符。

返回参数

名称	类型	示例值	描述
RequestId	String	36923CC2-DDAB-4B48-A144-DA92C1E19537	请求ID。
ConsumerGroupID	String	dtsxxxxxxxxx	消费组ID。

## 示例

```

public String createConsumerGroup(String instanceId, String
consumerGroupName, String consumerGroupName, String consumerGr
oupPassword) throws Exception {
    CreateConsumerGroupRequest request = new CreateConsumerGroupR
equest();
    request.setSubscriptionInstanceId(instanceId);
    request.setConsumerGroupName(consumerGroupName);
    request.setConsumerGroupName(consumerGroupName);
    request.setConsumerGroupPassword(consumerGroupPassword);

    CreateConsumerGroupResponse response = client.getAcsResponse(
request);
    return response.getConsumerGroupID();
}

```

## 错误码

错误码	错误信息	描述
InvalidParameter	The Input Parameter\${parameter} is not valid.	输入的参数不合法，请检查输入的参数是否正确。
InvalidParameterValid	The Value of Input Parameter \${parameter} is not valid.	参数值校验不通过，请检查是否缺失所提示的参数以及用户AK是否正确。
MissingParameter	The parameter "<parameter name>" is needed but no provided.	缺少参数，请检查是否缺失所提示的参数以及用户AK是否正确，具体请参照DTS OpenAPI文档。
Forbidden.InstanceNotFound	Instance is not found or does not belong to this account.	实例不存在，或者当前用户没有该实例的操作权限。

## 3.4 修改消费组密码

调用ModifyConsumerGroupPassword接口修改消费组密码。

- 本接口仅适用于新版订阅通道。



说明:

在调用ConfigureSubscriptionInstance接口时，需要传入SubscriptionInstance NetworkType参数，如果不传入该参数则订阅通道被定义为旧版。

- 调用本接口时，订阅通道需处于未启动、启动失败、正常、异常状态。

## 请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	ModifyConsumerGroupPassword	要执行的操作，取值：ModifyConsumerGroupPassword。
SubscriptionInstanceId	String	是	dtsxxxxxxxxx	订阅实例ID。
ConsumerGroupId	String	是	dtsxxxxxxxxx	消费组ID。
ConsumerGroupPassword	String	是	Test123456	消费组当前的密码。
ConsumerGroupNewPassword	String	是	Test654321	消费组的新密码。 · 密码由大写字母、小写字母、数字、特殊字符中的任意两种或两种以上组成。 · 密码长度为8-32个字符。
ConsumerGroupName	String	否	测试消费组	消费组名称。
ConsumerGroupName	String	否	dtstest	消费组的账号。

## 返回参数

名称	类型	示例值	描述
RequestId	String	36923CC2-DDAB-4B48-A144-DA92C1E19537	请求ID。

## 示例

```
public void modifyConsumerGroup(String instanceId, String consumerGroupId, String consumerGroupName, String consumerGroupName, String consumerGroupPassword, String consumerGroupNewPassword) throws Exception {
    ModifyConsumerGroupPasswordRequest request = new ModifyConsumerGroupPasswordRequest();
    request.setSubscriptionInstanceId(instanceId);
    request.setConsumerGroupId(consumerGroupId);
    request.setConsumerGroupName(consumerGroupName);
    request.setConsumerGroupName(consumerGroupName);
    request.setConsumerGroupName(consumerGroupName);
    request.setConsumerGroupPassword(consumerGroupPassword);
    request.setConsumerGroupNewPassword(consumerGroupNewPassword);
}
```

```

ModifyConsumerGroupPasswordResponse response = client.
getAcqsResponse(request);

    if (!"true".equals(response.getSuccess())) {
        throw new Exception(response.getErrorMessage() + " requestId
=" + response.getRequestId());
    }
}

```

### 错误码

错误码	错误信息	描述
InvalidParameter	The Input Parameter\${parameter} is not valid.	输入的参数不合法，请检查输入的参数是否正确。
InvalidParameterValid	The Value of Input Parameter \${parameter} is not valid.	参数值校验不通过，请检查是否缺失所提示的参数以及用户AK是否正确。
MissingParameter	The parameter "<parameter name>" is needed but no provided.	缺少参数，请检查是否缺失所提示的参数以及用户AK是否正确，具体请参照DTS OpenAPI文档。
MatchParameter	The Input Parameter\${parameter} is not Inconsistent.	密码错误或密码不一致。

## 3.5 查询消费组详情

调用DescribeConsumerGroup接口查询DTS订阅实例中的消费组详情。

- 本接口仅适用于新版订阅通道。



说明:

在调用ConfigureSubscriptionInstance接口时，需要传入SubscriptionInstance NetworkType参数，如果不传入该参数则订阅通道被定义为旧版。

- 只有订阅通道配置完成后，才能调用本接口。

### 请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	DescribeConsumerGroup	要执行的操作，取值：DescribeConsumerGroup。

名称	类型	是否必选	示例值	描述
SubscriptionInstanceId	String	是	dtsxxxxxxxxx	订阅实例ID。
pageSize	Integer	否	30	每页显示的消费组数量，默认值为30。
pageNum	Integer	否	1	页码。 大于0，且不超过Integer的最大值，默认值为1。

## 返回参数

名称	类型	示例值	描述
RequestId	String	36923CC2-DDAB-4B48-A144-DA92C1E19537	请求ID。
DescribeConsumerChannel	List	不涉及	消费组列表。
└ConsumerGroupId	String	dtsxxxxxxxxx	消费组ID。
└ConsumerGroupName	String	测试消费组	消费组名称。
└ConsumptionCheckpoint	String	2019-06-02T06:33:07Z	订阅通道的下游消费时间点，格式为yyyy-MM-ddTHH:mm:ssZ（UTC时间）。
└UnconsumedData	Long	500	堆积的记录数。
└MessageDelay	Long	15	消息延迟时间，单位为秒。
└ConsumerGroupNameUserName	String	dtstest	当前消费组的账号。
PageRecordCount	Integer	1	本页消费组数量。
PageNumber	Integer	1	页码。
TotalRecordCount	Integer	1	消费组总数。

## 示例

```
public DescribeConsumerGroupResponse describeConsumerGroup(String
instanceId) throws Exception {
    DescribeConsumerGroupRequest request = new DescribeCo
nsumerGroupRequest();
    request.setSubscriptionInstanceId(instanceId);
```

```

        DescribeConsumerGroupResponse response = client.getAcsResponse
(request);
        System.out.println(response.getConsumerChannels().size());
        return response;
    }

```

### 错误码

错误码	错误信息	描述
InvalidParameter	The Input Parameter\${parameter} is not valid.	输入的参数不合法，请检查输入的参数是否正确。
InvalidParameterValid	The Value of Input Parameter \${parameter} is not valid.	参数值校验不通过，请检查是否缺失所提示的参数以及用户AK是否正确。
MissingParameter	The parameter "<parameter name>" is needed but no provided.	缺少参数，请检查是否缺失所提示的参数以及用户AK是否正确，具体请参照DTS OpenAPI文档。
Forbidden.ConsumerGroupNotFound	ConsumerGroup is not found.	消费组不存在。

## 3.6 删除消费组

调用DeleteConsumerGroup接口删除DTS订阅实例中的消费组。

本接口仅适用于新版订阅通道。



说明:

在调用ConfigureSubscriptionInstance接口时，需要传入SubscriptionInstance NetworkType参数，如果不传入该参数则订阅通道被定义为旧版。

### 请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	DeleteConsumerGroup	要执行的操作，取值：DeleteConsumerGroup。
SubscriptionInstanceId	String	是	dtsxxxxxxxxx	订阅实例ID。
ConsumerGroupID	String	是	dtsxxxxxxxxx	消费组ID。

## 返回参数

名称	类型	示例值	描述
RequestId	String	36923CC2-DDAB-4B48-A144-DA92C1E19537	请求ID。

## 示例

```

public void deleteConsumerGroup(String instanceId, String
consumerGroupId) throws Exception {
    DeleteConsumerGroupRequest request = new DeleteConsumerGroupR
equest();
    request.setSubscriptionInstanceId(instanceId);
    request.setConsumerGroupID(consumerGroupId);

    DeleteConsumerGroupResponse response = client.getAcsResponse(
request);

    if (!"true".equals(response.getSuccess())) {
        throw new Exception(response.getErrorMessage() + " requestId
=" + response.getRequestId());
    }
}

```

## 错误码

错误码	错误信息	描述
InvalidParameter	The Input Parameter\${parameter} is not valid.	输入的参数不合法，请检查输入的参数是否正确。
InvalidParameterValid	The Value of Input Parameter \${parameter} is not valid.	参数值校验不通过，请检查是否缺失所提示的参数以及用户AK是否正确。
MissingParameter	The parameter "<parameter name>" is needed but no provided.	缺少参数，请检查是否缺失所提示的参数以及用户AK是否正确，具体请参照DTS OpenAPI文档。
Forbidden.InstanceNotFound	ConsumerGroup is not found	消费组不存在。

## 4 数据同步

### 4.1 更改同步作业的数据库连接信息

当数据同步作业的源数据库或目标数据库发生了主备切换，您可以调用SwitchSynchronizationEndpoint接口，将主备切换后新数据库的连接信息传给DTS，DTS将从断点处重新启动数据同步。

#### 前提条件

数据同步作业的源数据库或目标数据库发生了主备切换，导致连接信息发生变化。



说明：

数据库类型需为MySQL。

#### 请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	SwitchSynchronizationEndpoint	要执行的操作，取值：SwitchSynchronizationEndpoint。
SynchronizationJobId	String	是	dtsxxxxxxxx	数据同步实例ID。
EndpointType	String	是	Source	待调整连接信息的数据库，取值： <ul style="list-style-type: none"> <li>Source：源数据库。</li> <li>Destination：目标数据库。</li> </ul>
EndpointInstanceType	String	是	ECS	新数据库所属的实例类型，取值： <ul style="list-style-type: none"> <li>ECS：ECS上的自建数据库。</li> <li>Express：通过专线接入的本地数据库。</li> </ul>

名称	类型	是否必选	示例值	描述
EndpointInstanceId	String	是	i-bpxxxxxxx	ECS实例ID。   <b>说明:</b> 当EndpointInstanceType取值为ECS时，本参数才可用且必须传入。
EndpointPort	String	是	3306	新数据库的数据库服务端口。
EndpointIP	String	否	172.16.xx.xx	新数据库的IP地址。   <b>说明:</b> 当EndpointInstanceType取值为Express时，本参数才可用且必须传入。
SynchronizationDirection	String	否	Forward	同步方向，取值： · Forward：单向同步。 · Reverse：双向同步。   <b>说明:</b> 默认取值为Forward。

返回参数

名称	类型	示例值	描述
RequestId	String	36923CC2-DDAB-4B48-A144-DA92C1E19537	请求ID。
TaskId	String	1xxxxxx	任务ID，您可以根据此返回值调用 <a href="#">DescribeEndpointSwitchStatusRequest</a> 接口查询该任务的执行状态。
Success	String	true	是否调用成功，取值true或false。

示例

```
public String endpointSwitch(String synchronizationJobId, String
    synchronizationDirection, String endpointType, String endpointInstanceType,
    String endpointInstanceId,
    String endpointIp, String endpointPort)
{
```

```

SwitchSynchronizationEndpointRequest request = new SwitchSynchronizationEndpointRequest();

request.setSynchronizationJobId(synchronizationJobId);
request.setSynchronizationDirection(synchronizationDirection);
request.setEndpointType(endpointType);
request.setEndpointInstanceType(endpointInstanceType);
request.setEndpointInstanceId(endpointInstanceId);
request.setEndpointIP(endpointIp);
request.setEndpointPort(endpointPort);

try
{
    SwitchSynchronizationEndpointResponse response = client.getAcSResponse(request);
    String taskId = response.getTaskId();
    return taskId;
}
catch (Exception e)
{
    System.out.println("Switch synchronization endpoint failed");
    System.out.println(e.toString());
}
return null;
}

```

#### 错误码

访问[错误码中心](#)查看更多错误码信息。

## 4.2 查询更改数据库连接信息的任务执行状态

当调用SynchronizationEndpointSwitch接口修改数据库连接信息后，您可以调用DescribeEndpointSwitchStatus接口查看任务的执行状态。

#### 前提条件

已调用过[SwitchSynchronizationEndpoint](#)接口。

#### 请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	DescribeEndpointSwitchStatus	要执行的操作，取值：DescribeEndpointSwitchStatus。
TaskId	String	是	1xxxxxx	任务ID，调用 <a href="#">SwitchSynchronizationEndpoint</a> 接口后会返回该值。

## 返回参数

名称	类型	示例值	描述
Status	String	Finished	任务执行状态，返回值为： <ul style="list-style-type: none"> <li>NotStarted: 未启动。</li> <li>Initializing: 初始化中。</li> <li>Switching: 切换中。</li> <li>Failed: 执行失败。</li> <li>Finished: 执行完毕。</li> </ul>
ErrorMessage	String	demo	主备切换失败返回的错误信息。  <b>说明：</b> 当Status的返回值为Failed时，才会返回本参数。

## 示例

```
public void describeEndpointSwitchStatus(String taskId) throws
ServerException, ClientException
{
    DescribeEndpointSwitchStatusRequest request = new DescribeEn
dpointSwitchStatusRequest();

    request.setTaskId(taskId);

    try
    {
        DescribeEndpointSwitchStatusResponse response = client.
getAcsResponse(request);
        System.out.println("Status:" + response.getStatus() + ";
ErrorMessage:" + response.getErrorMessage());
    }
    catch (Exception e)
    {
        System.out.println("Describe endpoint switch status failed");
        System.out.println(e.toString());
    }
}
```

## 错误码

访问[错误码中心](#)查看更多错误码信息。

## 5 监控报警

### 5.1 配置迁移任务的监控报警

DTS支持监控迁移任务的延迟和状态，您可以调用ConfigureMigrationJobAlert接口配置监控报警。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	ConfigureMigrationJobAlert	要执行的操作，取值：ConfigureMigrationJobAlert。
MigrationJobId	String	是	dtsxxxxxxxx	数据迁移实例ID。
DelayAlertStatus	String	否	enable	设置延迟监控项的开关，取值： <ul style="list-style-type: none"> <li>enable：启用。</li> <li>disable：禁用。</li> </ul> <div style="border: 1px solid #ccc; background-color: #f0f0f0; padding: 5px; margin-top: 10px;"> <p> 说明：</p> <ul style="list-style-type: none"> <li>默认取值为enable。</li> <li>本参数和ErrorAlertStatus参数两者中必须传入一个。</li> </ul> </div>
DelayAlertPhone	String	否	136xxxxxxxx, 137xxxxxxxx	延迟监控项的报警联系人手机号码，多个手机号码以英文逗号（,）分隔。
DelayOverSeconds	String	否	5	延迟监控项的报警阈值，单位为秒。 <div style="border: 1px solid #ccc; background-color: #f0f0f0; padding: 5px; margin-top: 10px;"> <p> 说明：</p> <p>当DelayAlertStatus取值为enable时，本参数必须传入。</p> </div>

名称	类型	是否必选	示例值	描述
ErrorAlertStatus	String	否	enable	设置迁移状态监控项的开关，取值： <ul style="list-style-type: none"> <li>enable：启用。</li> <li>disable：禁用。</li> </ul> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  <b>说明：</b> <ul style="list-style-type: none"> <li>默认取值为enable。</li> <li>本参数和DelayAlertStatus参数两者中必须传入一个。</li> <li>开启迁移状态监控项后，当检测到迁移状态为异常时，则触发报警。</li> </ul> </div>
ErrorAlertPhone	String	否	136xxxxxxx, 137xxxxxxx	迁移状态监控项的报警联系人手机号码，多个手机号码以英文逗号(,)分隔。

## 返回参数

名称	类型	示例值	描述
RequestId	String	36923CC2-DDAB-4B48-A144-DA92C1E19537	请求ID。

## 示例

```

public void configureMigrationJobAlert(String MigrationJobId, String
    delayAlertStatus, String delayAlertPhone, String delayOverSeconds,
    String errorAlertStatus, String errorAlertPhone) throws Exception
    {
        ConfigureMigrationJobAlertRequest request = new ConfigureM
            igrationJobAlertRequest();
        request.setMigrationJobId(MigrationJobId);

        request.setDelayAlertStatus(delayAlertStatus);
        request.setDelayAlertPhone(delayAlertPhone);
        request.setDelayOverSeconds(delayOverSeconds);
        request.setErrorAlertPhone(errorAlertPhone);
        request.setErrorAlertStatus(errorAlertStatus);

        ConfigureMigrationJobAlertResponse response = client.
            getAcResponse(request);

        try {
            response = client.getAcResponse(request);

            System.out.println("Configure Migration Job Alert Succeed!");
        } catch (Exception e) {
            // TODO: handle exception
        }
    }

```

```

        System.out.println("Configure Migration Job Alert Failed");
        System.out.println(e.toString());
    }
}

```

### 错误码

访问[错误码中心](#)查看更多错误码信息。

## 5.2 查询迁移任务的监控报警设置

调用DescribeMigrationJobAlert接口查询迁移任务的监控报警设置详情。

### 请求参数

名称	类型	是否必选	实例值	描述
Action	String	是	DescribeMigrationJobAlert	要执行的操作，取值：DescribeMigrationJobAlert。
MigrationJobId	String	是	dtsxxxxxxxx	数据迁移实例ID。

### 返回参数

名称	类型	示例值	描述
MigrationJobId	String	tvhmxxxxxxxx	数据迁移实例ID。
MigrationJobName	String	POLARDB_TO_MySQL	数据迁移任务名称。
DelayAlertStatus	String	enable	延迟监控项的开关状态，返回值为： <ul style="list-style-type: none"> <li>enable：启用。</li> <li>disable：禁用。</li> </ul>
DelayAlertPhone	String	136xxxxxxxx, 137xxxxxxxx	延迟监控项的报警联系人手机号码。 <div style="border: 1px solid #ccc; padding: 5px; margin-top: 5px;">  说明： 如果有多个手机号码，将以英文逗号（,）分隔。           </div>
DelayOverSeconds	String	20	延迟监控项的报警阈值，单位为秒。

名称	类型	示例值	描述
ErrorAlert Status	String	enable	迁移状态监控项的开关状态，返回值为： <ul style="list-style-type: none"> <li>· enable：启用。</li> <li>· disable：禁用。</li> </ul>
ErrorAlert Phone	String	138xxxxxxxx, 139xxxxxxxx	迁移状态的报警联系人手机号码。  说明： 如果有多个手机号码，将以英文逗号 (,) 分隔。
Success	Boolean	true	接口是否调用成功，返回值为： <ul style="list-style-type: none"> <li>· true：成功。</li> <li>· flase：失败。</li> </ul>

### 示例

```
public DescribeMigrationJobAlertResponse describeMigrationJobAlert(
String MigrationJobId) throws Exception
{
    DescribeMigrationJobAlertRequest request = new DescribeMi
migrationJobAlertRequest();
    request.setMigrationJobId(MigrationJobId);

    DescribeMigrationJobAlertResponse response = client.getAcSResp
onse(request);

    if (!"true".equals(response.getSuccess()))
    {
        throw new Exception(response.getErrMessage() + " requestId=" +
response.getRequestId());
    }
    return response;
}
```

### 错误码

访问[错误码中心](#)查看更多错误码信息。

## 5.3 配置订阅通道的监控报警

DTS支持监控订阅通道的延迟和状态，您可以调用ConfigureSubscriptionInstanceAlert接口配置监控报警。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	ConfigureSubscriptionInstanceAlert	要执行的操作，取值：ConfigureSubscriptionInstanceAlert。
SubscriptionInstanceId	String	是	dtsxxxxxxxx	数据订阅实例ID。
DelayAlertStatus	String	否	enable	设置订阅延迟监控项的开关，取值： <ul style="list-style-type: none"> <li>enable：启用。</li> <li>disable：禁用。</li> </ul> <div style="background-color: #f0f0f0; padding: 5px; margin-top: 10px;">  <b>说明：</b> <ul style="list-style-type: none"> <li>默认取值为enable。</li> <li>本参数和ErrorAlertStatus参数两者中必须传入一个。</li> </ul> </div>
DelayAlertPhone	String	否	136xxxxxxxx, 137xxxxxxxx	订阅延迟监控项的报警联系人手机号码，多个手机号码以英文逗号（,）分隔。
DelayOverSeconds	String	否	5	订阅延迟监控项的报警阈值，单位为秒。 <div style="background-color: #f0f0f0; padding: 5px; margin-top: 10px;">  <b>说明：</b>           当DelayAlertStatus取值为enable时，本参数必须传入。           </div>

名称	类型	是否必选	示例值	描述
ErrorAlertStatus	String	否	enable	设置订阅状态监控项的开关，取值： <ul style="list-style-type: none"> <li>enable：启用。</li> <li>disable：禁用。</li> </ul> <div style="border: 1px solid #ccc; background-color: #f0f0f0; padding: 5px; margin-top: 10px;">  <b>说明：</b> <ul style="list-style-type: none"> <li>默认取值为enable。</li> <li>本参数和DelayAlertStatus参数两者中必须传入一个。</li> <li>开启订阅状态监控项后，当检测到订阅状态为异常时，则触发报警。</li> </ul> </div>
ErrorAlertPhone	String	否	136xxxxxxxx, 137xxxxxxxx	订阅状态监控项的报警联系人手机号码，多个手机号码以英文逗号(,)分隔。

## 返回参数

名称	类型	示例值	描述
RequestId	String	36923CC2-DDAB-4B48-A144-DA92C1E19537	请求ID。

## 示例

```

public void configureSubscribeJobAlert(String MigrationJobId, String
    delayAlertStatus, String delayAlertPhone, String delayOverSeconds,
    String errorAlertStatus, String errorAlertPhone) throws Exception
    {
        ConfigureSubscriptionInstanceAlertRequest request = new
        ConfigureSubscriptionInstanceAlertRequest();
        request.setSubscriptionInstanceId(MigrationJobId);

        request.setDelayAlertStatus(delayAlertStatus);
        request.setDelayAlertPhone(delayAlertPhone);
        request.setDelayOverSeconds(delayOverSeconds);
        request.setErrorAlertPhone(errorAlertPhone);
        request.setErrorAlertStatus(errorAlertStatus);

        ConfigureSubscriptionInstanceAlertResponse response = client.
        getAcResponse(request);

        try {
            response = client.getAcResponse(request);

            System.out.println("Configure Subscription Job Alert Succeed!");
        } catch (Exception e) {
            // TODO: handle exception
        }
    }

```

```

        System.out.println("Configure Subscription Job Alert Failed");
        System.out.println(e.toString());
    }
}

```

### 错误码

访问[错误码中心](#)查看更多错误码信息。

## 5.4 查询订阅通道的监控报警设置

调用DescribeSubscriptionInstanceAlert接口查询订阅通道的监控报警设置详情。

### 请求参数

名称	类型	是否必选	实例值	描述
Action	String	是	DescribeSubscriptionInstanceAlert	要执行的操作，取值：DescribeSubscriptionInstanceAlert
SubscriptionInstanceId	String	是	dtsxxxxxxxx	数据订阅实例ID。

### 返回参数

名称	类型	示例值	描述
SubscriptionInstanceName	String	MySQL	数据订阅通道名称。
DelayAlertStatus	String	enable	订阅延迟监控项的开关状态，返回值为： <ul style="list-style-type: none"> <li>· enable：启用。</li> <li>· disable：禁用。</li> </ul>
DelayAlertPhone	String	136xxxxxxxx, 137xxxxxxxx	订阅延迟监控项的报警联系人手机号码。 <div style="border: 1px solid #ccc; padding: 5px; margin-top: 5px;">  说明： 如果有多个手机号码，将以英文逗号(,)分隔。         </div>
DelayOverSeconds	String	20	订阅延迟监控项的报警阈值，单位为秒。

名称	类型	示例值	描述
ErrorAlert Status	String	enable	订阅状态监控项的开关状态，返回值为： <ul style="list-style-type: none"> <li>· enable：启用。</li> <li>· disable：禁用。</li> </ul>
ErrorAlert Phone	String	138xxxxxxxx, 139xxxxxxxx	订阅状态监控项的报警联系人手机号码。  说明： 如果有多个手机号码，将以英文逗号 (,) 分隔。
Success	Boolean	true	接口是否调用成功，返回值为： <ul style="list-style-type: none"> <li>· true：成功。</li> <li>· flase：失败。</li> </ul>

### 示例

```
public DescribeSubscriptionInstanceAlertResponse describeSubscribeJob
Alert(String MigrationJobId) throws Exception
{
    DescribeSubscriptionInstanceAlertRequest request = new DescribeSu
bscriptionInstanceAlertRequest();
    request.setSubscriptionInstanceId(MigrationJobId);

    DescribeSubscriptionInstanceAlertResponse response = client.
getAcpsResponse(request);

    if (!"true".equals(response.getSuccess()))
    {
        throw new Exception(response.getErrorMessage() + " requestId=" +
response.getRequestId());
    }
    return response;
}
```

### 错误码

访问[错误码中心](#)查看更多错误码信息。

## 5.5 配置同步作业的监控报警

DTS支持监控同步作业的延迟和状态，您可以调用ConfigureSynchronizationJobAlert接口配置监控报警。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	ConfigureSynchronizationJobAlert	要执行的操作，取值：ConfigureSynchronizationJobAlert。
SynchronizationJobId	String	是	dtsxxxxxxxx	数据同步实例ID。
SynchronizationDirection	String	否	Forward	同步方向，取值： <ul style="list-style-type: none"> <li>Forward：单向同步。</li> <li>Reverse：双向同步。</li> </ul> <div style="background-color: #f0f0f0; padding: 5px; margin-top: 10px;">  说明： 默认取值为Forward。         </div>
DelayAlertStatus	String	否	enable	设置同步延迟监控项的开关，取值： <ul style="list-style-type: none"> <li>enable：启用。</li> <li>disable：禁用。</li> </ul> <div style="background-color: #f0f0f0; padding: 5px; margin-top: 10px;">  说明：           <ul style="list-style-type: none"> <li>默认取值为enable。</li> <li>本参数和ErrorAlertStatus参数两者中必须传入一个。</li> </ul> </div>
DelayAlertPhone	String	否	136xxxxxxxx, 137xxxxxxxx	同步延迟监控项的报警联系人手机号码，多个手机号码以英文逗号（,）分隔。
DelayOverSeconds	String	否	5	同步延迟监控项的报警阈值，单位为秒。 <div style="background-color: #f0f0f0; padding: 5px; margin-top: 10px;">  说明： 当DelayAlertStatus取值为enable时，本参数必须传入。         </div>

名称	类型	是否必选	示例值	描述
ErrorAlertStatus	String	否	enable	设置同步状态监控项的开关，取值： <ul style="list-style-type: none"> <li>enable：启用。</li> <li>disable：禁用。</li> </ul> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  <b>说明：</b> <ul style="list-style-type: none"> <li>默认取值为enable。</li> <li>本参数和DelayAlertStatus参数两者中必须传入一个。</li> <li>开启同步状态监控项后，当检测到同步状态为异常时，则触发报警。</li> </ul> </div>
ErrorAlertPhone	String	否	136xxxxxxxx, 137xxxxxxxx	同步状态监控项的报警联系人手机号码，多个手机号码以英文逗号(,)分隔。

## 返回参数

名称	类型	示例值	描述
RequestId	String	36923CC2-DDAB-4B48-A144-DA92C1E19537	请求ID。

## 示例

```

public void configureSyncJobAlert(String MigrationJobId, String
delayAlertStatus, String delayAlertPhone, String delayOverSeconds,
String errorAlertStatus, String errorAlertPhone) throws Exception
{
    ConfigureSynchronizationJobAlertRequest request = new
ConfigureSynchronizationJobAlertRequest();
    request.setSynchronizationJobId(MigrationJobId);

    request.setDelayAlertStatus(delayAlertStatus);
    request.setDelayAlertPhone(delayAlertPhone);
    request.setDelayOverSeconds(delayOverSeconds);
    request.setErrorAlertPhone(errorAlertPhone);
    request.setErrorAlertStatus(errorAlertStatus);

    ConfigureSynchronizationJobAlertResponse response = new ConfigureS
ynchronizationJobAlertResponse();

    try {
        response = client.getAcsResponse(request);

        System.out.println("Configure Sync Job Alert Succeed!");
    } catch (Exception e) {
        // TODO: handle exception
    }
}

```

```

        System.out.println("Configure Sync Job Alert Failed");
        System.out.println(e.toString());
    }
}

```

### 错误码

访问[错误码中心](#)查看更多错误码信息。

## 5.6 查询同步作业的监控报警设置

调用DescribeSubscriptionInstanceAlert接口查询同步作业的监控报警设置详情。

### 请求参数

名称	类型	是否必选	实例值	描述
Action	String	是	DescribeSynchronizat ionJobAlert	要执行的操作，取值：DescribeSynchronizat ionJobAlert
Synchroniz ationJobId	String	是	dtsxxxxxxxx	数据同步实例ID。

### 返回参数

名称	类型	示例值	描述
Subscripti onInstance Name	String	RDS_TO_POLARDB	数据同步作业名称。
Synchroniz ationJobId	String	dtsxxxxxxxx	数据同步实例ID。
DelayAlert Status	String	enable	同步延迟监控项的开关状态，返回值为： <ul style="list-style-type: none"> <li>enable：启用。</li> <li>disable：禁用。</li> </ul>
DelayAlert Phone	String	136xxxxxxxx, 137xxxxxxxx	同步延迟监控项的报警联系人手机号码。 <div style="border: 1px solid #ccc; background-color: #f0f0f0; padding: 5px; margin-top: 5px;">  <b>说明：</b> 如果有多个手机号码，将以英文逗号（,）分隔。 </div>
DelayOverS econds	String	20	同步延迟监控项的报警阈值，单位为秒。

名称	类型	示例值	描述
ErrorAlert Status	String	enable	同步状态监控项的开关状态，返回值为： <ul style="list-style-type: none"> <li>· enable：启用。</li> <li>· disable：禁用。</li> </ul>
ErrorAlert Phone	String	138xxxxxxx, 139xxxxxxx	同步状态监控项的报警联系人手机号码。  说明： 如果有多个手机号码，将以英文逗号 (,) 分隔。
Success	Boolean	true	接口是否调用成功，返回值为： <ul style="list-style-type: none"> <li>· true：成功。</li> <li>· flase：失败。</li> </ul>

### 示例

```
public DescribeSynchronizationJobAlertResponse describeSyncJobAlert(
String MigrationJobId) throws Exception
{
    DescribeSynchronizationJobAlertRequest request = new DescribeSynchronizationJobAlertRequest();
    request.setSynchronizationJobId(MigrationJobId);

    DescribeSynchronizationJobAlertResponse response = client.getAcsResponse(request);

    if (!"true".equals(response.getSuccess()))
    {
        throw new Exception(response.getErrMessage() + " requestId=" + response.getRequestId());
    }
    return response;
}
```

### 错误码

访问[错误码中心](#)查看更多错误码信息。