

阿里云

# 分布式任务调度 SchedulerX

## Scheduling 和 Triggers

文档版本：20200417

# 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、特殊性或惩罚性的损害，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。未经阿里云和/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。
	该类警示信息可能会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。	<b>注意：</b> 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。	<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。	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 <b>确定</b> 。
Courier字体	命令。	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

---

<b>法律声明</b>	1
<b>通用约定</b>	1
<b>1 定时调度</b>	1
1.1 Cron	1
1.2 Fixed rate	5
1.3 Second delay	7
<b>2 工作流调度</b>	10
<b>3 API 调度</b>	13
<b>4 重刷数据</b>	14

# 1 定时调度

## 1.1 Cron

Cron 是一款类 Unix 的操作系统下的基于时间的任务管理系统。您可以通过 Cron 在固定时间、日期间隔下运行定时任务（可以是命令和脚本）。

### Cron 表达式

Cron 的表达式为：**秒分时 日月周[年]**

为了帮助您理解，下面介绍一些常用的 Cron 表达式示例。

Contab 表达式	说明
0 */1 * * * ?	每隔 1 分钟触发一次
0 0 5-15 * * ?	每天 5-15 点整点触发
0 0/3 * * * ?	每隔 3 分钟触发一次
0 0-5 14 * * ?	每天 14:00 ~ 14:05 期间每隔 1 分钟触发一次
0 0/5 14 * * ?	每天 14:00 ~ 14:55 期间每隔 5 分钟触发一次
0 0/5 14,18 * * ?	每天 14:00 ~ 14:55 和 18:00 ~ 18:55 两个时间段内每 5 分钟触发一次
0 0/30 9-17 * * ?	每天 9:00 ~ 17:00 内每半小时触发一次
0 0 10,14,16 * * ?	每天 10:00、14:00 和 16:00 触发
0 0 12 ? * WED	每周三 12:00 触发
0 0 17 ? * TUES,THUR,SAT	每周二、周四、周六 17:00 触发
0 10,44 14 ? 3 WED	每年 3 月的每周三的 14:10 和 14:44 触发
0 15 10 ? * MON-FRI	周一至周五的上午 10:15 触发
0 0 23 L * ?	每月最后一天 23:00 触发
0 15 10 L * ?	每月最后一天 10:15 触发
0 15 10 ? * 6L	每月最后一个周五 10:15 触发
0 15 10 * * ? 2005	2005 年的每天 10:15 触发
0 15 10 ? * 6L 2002-2005	2002 年 ~ 2005 年的每月的最后一个周五上午 10:15 触发
0 15 10 ? * 6#3	每月的第三个周五 10:15 触发

## Cron 定时调度配置示例

在创建调度任务时可以使用 Cron 定时调度，创建调度任务的详细操作步骤请参见[#unique\\_5/unique\\_5\\_Connect\\_42\\_section\\_ww9\\_0re\\_gta](#)。本文仅介绍在创建调度任务时如何配置 Cron 定时调度。

1. 进入[定时配置](#)页签，详情请参见[#unique\\_5/unique\\_5\\_Connect\\_42\\_section\\_ww9\\_0re\\_gta](#)。

2. 在创建任务配置向导的**定时配置**页签中设置定时调度参数，然后单击**下一步**。

← 创建任务

1 基本配置 2 定时配置 3 报警配置

\* 时间类型: cron

\* cron表达式: 请输入对应的时间表达式

使用生成工具 验证cron

高级配置

时间偏移: 数据时间相对调度时间偏移量，默认0

时区: 请选择时区

上一步 下一步

a) 在**时间类型**右侧的列表选择**cron**。

b) 设置**cron表达式**。

如果定时规则比较简单，建议使用工具生成，本文也会介绍如何使用工具生成 Cron 表达式。

如果规则负责，工具无法生成，可以输入 Cron 表达式，相关示例请参见[Cron 表达式](#)。

A. 单击**使用生成工具**。

B. 在弹出的列表中选择时间周期、时间等规则，例如每天 3:00 执行，然后单击**确定**。

← 创建任务 ×

1 基本配置 2 定时配置 3 报警配置

\* 时间类型: cron

\* cron表达式: 请输入对应的时间表达式

每 天 03:00:00

**确定** **验证cron**

高级配置 >

时	分	秒
3	0	0
4	1	1
5	2	2
6	3	3
7	4	4
8	5	5
9	6	6

C. 单击**验证cron**，查看规则最近 5 次触发时间。

← 创建任务 ×

1 基本配置 2 定时配置 3 报警配置

\* 时间类型: cron

0 0 3 \* \* ?

**使用生成工具** **验证cron**

最近5次触发时间

2020-04-03 03:00:00
2020-04-04 03:00:00
2020-04-05 03:00:00
2020-04-06 03:00:00
2020-04-07 03:00:00

**确定** **取消**

c) (可选) 如果需要，在高级设置区域还可以设置**时间偏移**和**时区**。

当时间类型选择 Cron 后，可以进行高级配置。高级配置参数说明如下：

配置名称	意义	默认值
时间偏移	数据时间相对于调度时间的偏移，可以在调度时从上下文获取该值。	无
时区	可以根据实际情况选择不同时区，包括一些常用国家或地区，也包括标准的 GMT 表达方式。	无

## 1.2 Fixed rate

由于 Crontab 必须被 60 整除，如果需要每隔 40 分钟执行一次调度，则 Cron 无法支持。Fixed rate 专门用来做定期轮询，表达式简单，不支持秒级别。

### Fixed rate 定时配置示例

在创建调度任务时可以使用 Fixed rate 定时调度，创建调度任务的详细操作步骤请参见[#unique\\_5/unique\\_5\\_Connect\\_42\\_section\\_ww9\\_0re\\_gta](#)。本文仅介绍在创建调度任务时如何配置 Fixed rate 定时调度。

1. 进入定时配置页签，详情请参见[#unique\\_5/unique\\_5\\_Connect\\_42\\_section\\_ww9\\_0re\\_gta](#)。

2. 在创建任务配置向导的**定时配置**页签中设置定时调度参数，然后单击**下一步**。

← 创建任务

1 基本配置 2 定时配置 3 报警配置

\* 时间类型: cron

\* cron表达式: 请输入对应的时间表达式

使用生成工具 验证cron

高级配置

时间偏移: 数据时间相对调度时间偏移量，默认0

时区: 请选择时区

上一步 下一步

a) 在**时间类型**右侧的列表中选择 **fixed\_rate**。

b) 在**固定频率**右侧输入执行间隔。



**说明:**

固定频率必须高于 60 秒。

## 1.3 Second delay

Second delay 即秒级别调度，适用于对实时性要求比较高的业务，例如需要不停做轮询的准实时业务。

### 背景信息

Second delay 支持 1~60 秒间隔的秒级延迟调度，即每次任务执行完成后，间隔秒级时间再次触发调度。

Second delay 具有以下优势：

- 高可靠：SchedulerX 的秒级别任务具有高可靠的特性，如果某台机器宕机了，可以在 30 秒内在另一台机器上重新拉起。
- 丰富的任务类型：SchedulerX 的秒级别任务属于定时调度类型，可以适用于所有的任务类型和执行方式。不但简单 Java 任务支持秒级别调度，分布式 Java 任务以及脚本任务同样适用。通过内存网格和秒级别调度，可以不停地处理海量的数据。
- 统计信息：SchedulerX 还提供了秒级别任务的统计信息，例如当天执行了多少次，成功的次数和失败的次数，最近 10 次运行结果等。

### Second delay 定时配置示例

在创建调度任务时可以使用 Second delay 定时调度，创建调度任务的详细操作步骤请参见[#unique\\_5/unique\\_5\\_Connect\\_42\\_section\\_ww9\\_0re\\_gta](#)。本文仅介绍在创建调度任务时如何配置Second delay 定时调度。

1. 进入定时配置页签，详情请参见[#unique\\_5/unique\\_5\\_Connect\\_42\\_section\\_ww9\\_0re\\_gta](#)。

2. 在创建任务配置向导的**定时配置**页签中设置定时调度参数，然后单击**下一步**。

← 创建任务

1 基本配置 2 定时配置 3 报警配置

\* 时间类型: second\_delay

\* 固定延迟: 40

上一步 下一步

- a) 在**时间类型**右侧的列表中选择 **second\_delay**。
- b) 在**固定延迟**右侧输入固定延迟，单位为秒，范围为 1 秒~60 秒。

#### 查看秒级别任务的统计信息

Second delay 任务创建完成后，在**任务实例详情**页面会多一个**秒级任务统计详情**页签，展示如下信息：

- 当天任务实例运行结果
- 当天任务实例运行结果

- 最近 10 次运行结果

← 任务实例详情

基本信息 历史执行记录 执行日志

当天任务实例运行结果

统计截止时间	总量	池子	运行	成功	失败
2019-05-19 00:00:08	6557	0	1	6557	0

昨天任务实例运行结果

统计截止时间	总量	池子	运行	成功	失败
2019-05-18 00:00:01	33956	0	1	33956	0

> 第952150次循环，耗时：5.05s，开始时间：10:42:39，结束时间：10:42:44

> 第952151次循环，耗时：7.83s，开始时间：10:42:45，结束时间：10:42:53

> 第952152次循环，耗时：4.08s，开始时间：10:42:57，结束时间：10:43:01

> 第952153次循环，耗时：8.12s，开始时间：10:43:05，结束时间：10:43:13

> 第952154次循环，耗时：3.94s，开始时间：10:43:17，结束时间：10:43:21

> 第952155次循环，耗时：4.73s，开始时间：10:43:25，结束时间：10:43:29

> 第952156次循环，耗时：4.02s，开始时间：10:43:33，结束时间：10:43:37

> 第952157次循环，耗时：4.08s，开始时间：10:43:42，结束时间：10:43:46

> 第952158次循环，耗时：4.76s，开始时间：10:43:50，结束时间：10:43:55

确定 取消

## 2 工作流调度

通过可视化的工作流进行任务编排，支持 Cron 表达式和 API。

### 背景信息

工作流中的 Job 没有独立调度时间，跟随工作流的时间开始调度。

支持上下游数据传输。详情请参见[#unique\\_9](#)。

工作流调度至少要有 2 个 Job，且有依赖关系。如果只有一个 job，请直接使用任务管理。

### 创建工作流

您可以创建工作流调度任务。



#### 说明:

目前工作流调度仅支持 Cron 表达式。

1. 在左侧导航栏单击**流程管理**。
2. 在**流程管理**页面的**所属命名空间**列表选择具体的命名空间，然后单击**创建工作流**。

3. 在创建工作流（流程画布）页面设置工作流的**名称**、**描述**、**应用ID**（任务分组）和**时间类型**（包括 Cron 和 API），然后单击**确定**。

创建工作流

\* 名称: [输入框]

\* 描述: [输入框]

\* 应用ID: [输入框]

\* 时间类型: cron

\* cron表达式: 0 0 \*/\* ?

使用生成工具 验证Cron

高级配置

时区: 请选择

实例并发数: 1

确定 取消

也可以单击**高级设置**，设置**时区**和**实例并发数**。

4. 在**工作流详情**页面单击**创建 Job**或**导入 Job**，添加调度任务。

- **创建 Job:** 和创建调度任务的步骤一致，详情请参见[#unique\\_5/unique\\_5\\_Connect\\_42\\_section\\_ww9\\_0re\\_gta](#)。
- **导入 Job:** 将已创建的 Job 导入到工作流中。



#### 说明:

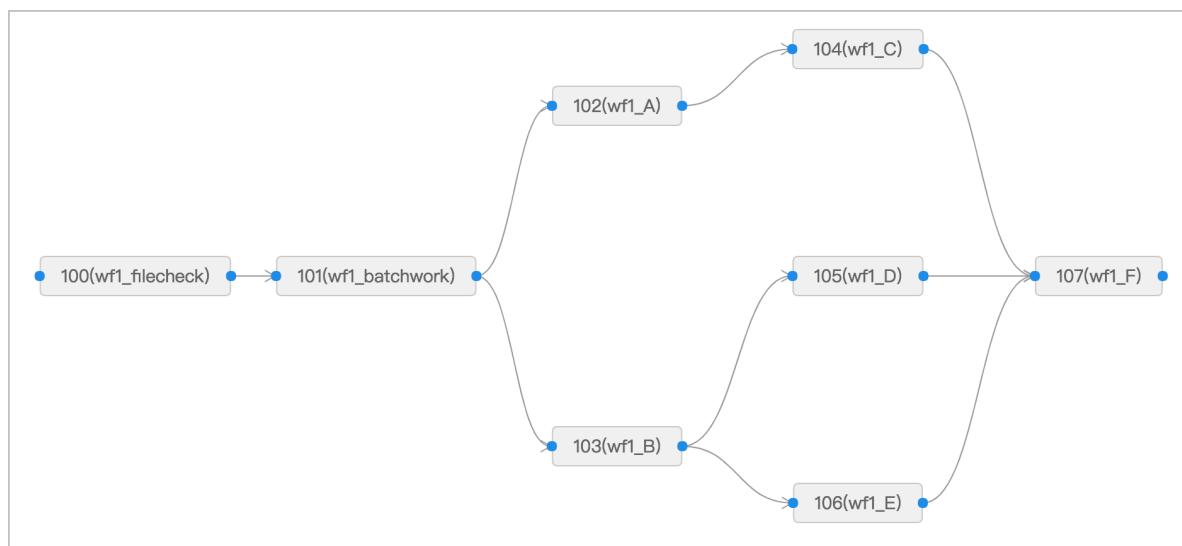
导入 Job 会弹出**导入Job将会移除该Job的定时配置**提示框，单击**确认**，则该 Job 不会独立调度，会跟随工作流的调度周期进行调度。

5. 调度任务添加完毕后，按住并拖动任务两侧的端点到其它任务的端点连接调度任务，形成依赖关系，组成完成的工作流。

在工作流中，上下游的任务还可以实现数据传递。详情请参见[如何通过工作流进行上下游数据传递](#)。

如果需要删除某两个任务间的依赖关系，选中它们之间的线条，单击 **Delete**；右键某个调度任务，在快捷菜单中单击**删除**，即可删除该任务。

一个工作流示意如下：



在该工作流中，101 执行完，102 和 103 会同时开始执行。104，105，106 都执行成功，107 才会开始执行。

6. 工作流配置完成后，单击发布。

工作流创建完成后，返回[流程管理](#)页面，可以查看是否已经包含创建的流程及相关信息。

## 相关操作

工作流发布之后，就会变成启用状态并自动开始调度。如果不马上启用，可以返回[流程管理](#)页面，在[操作列](#)单击[运行一次](#)测试下，没问题再手动启用。

您还可以对工作流进行编辑、查看操作记录和历史记录，以及删除。

## 3 API 调度

---

您可以使用 API 触发任务调度。

API 触发的使用方式请参见[#unique\\_11](#)。

# 4 重刷数据

通过重刷数据功能，您可以重新触发一段时间区间内的实例，来重刷业务数据。

## 重刷调度任务

如果您的业务发生变更，如数据库增加一个字段或者上一个月数据有错误，需要把过去一段时间的任务重新执行一遍，可以重刷调度任务数据。



### 说明：

任务和工作流都支持重刷数据（只支持天级别的调度周期）。

如果您之前执行的某个调度任务的数据出现偏差或遗漏，您可以通过重新设置执行参数并执行某个调度任务属性、获取数据。

1. 在**任务管理**页面任务列表的**操作**列单击图标，在弹出的菜单中单击**重刷任务**。



2. 在**重刷任务**页面设置**起止时间和数据时间**，单击**确定**。

- **起止日期**：指定重刷的时间区间，当前只支持 Cron 表达式任务。
- **数据时间**：指定重刷时间区间内的重刷时间。

### 示例

Cron 表达式为 0 0 2 \* \* ?，表示每天 2 点运行，这个时候重刷配置如下：

- 当前时间为 2019-01-01 10:00:00。
- 重刷任务的起止日期为 2018-10-01 ~ 2018-10-07，默认从 2018 年 10 月 1 日 00:00:00 起，到 2018 年 10 月 7 日 23:59:59 结束。
- 数据时间为 11:11:11。

则该任务会被重刷 7 次，生成 7 个实例。

序号	调度时间	数据时间
1	2019.1.1 10:00:00	2018.10.1 11:11:11
2	2019.1.1 10:00:00	2018.10.2 11:11:11
3	2019.1.1 10:00:00	2018.10.3 11:11:11
4	2019.1.1 10:00:00	2018.10.4 11:11:11
5	2019.1.1 10:00:00	2018.10.5 11:11:11
6	2019.1.1 10:00:00	2018.10.6 11:11:11

7	2019.1.1 10:00:00	2018.10.7 11:11:11
---	-------------------	--------------------