

Компоненты приложения

Планировщик заданий Quartz

Версия 8.0



Эта документация предоставляется с ограничениями на использование и защищена законами об интеллектуальной собственности. За исключением случаев, прямо разрешенных в вашем лицензионном соглашении или разрешенных законом, вы не можете использовать, копировать, воспроизводить, переводить, транслировать, изменять, лицензировать, передавать, распространять, демонстрировать, выполнять, публиковать или отображать любую часть в любой форме или посредством любые значения. Обратный инжиниринг, дизассемблирование или декомпиляция этой документации, если это не требуется по закону для взаимодействия, запрещены.

Информация, содержащаяся в данном документе, может быть изменена без предварительного уведомления и не может гарантировать отсутствие ошибок. Если вы обнаружите какие-либо ошибки, сообщите нам о них в письменной форме.

Содержание

Планировщик заданий Quartz	4
Рекомендации по настройке планировщика заданий	4
Политики Quartz для обработки не отработавших вовремя заданий	8

Планировщик заданий Quartz



Рекомендации по настройке планировщика заданий

Все политики Quartz по обработке не отработавших вовремя заданий можно разделить условно на три группы: `Ignore misfire policy` (Игнорировать), `Run immediately and continue` (Выполнить немедленно и продолжать) и `Discard and wait for next` (Отменить и ожидать следующего задания). Далее приведены рекомендации, в каких случаях использовать ту или иную политику.

Ignore misfire policy

Эта политика представлена одноименной константой `MisfireInstruction.IgnoreMisfirePolicy = -1`. Ее рекомендуется применять в том случае, когда необходимо гарантировать, что все запуски триггера будут обязательно выполнены, даже если имеется несколько просроченных запусков триггера. Например, есть задание А с периодичностью в 2 минуты. Из-за нехватки рабочих потоков Quartz или выключения планировщика время следующего старта задания А (`NEXT_FIRE_TIME`) отстает на 10 минут от текущего времени. Если необходимо, чтобы все 5 просроченных запусков задания обязательно выполнились, то следует использовать политику `IgnoreMisfirePolicy`.

Эту политику рекомендуется применять для триггеров, которые при каждом запуске используют некоторые уникальные данные, и важно, чтобы все запуски триггера были рано или поздно выполнены. Например, есть задание В, которое выполняется с интервалом в 1 час и генерирует отчет за интервал времени от `PREV_FIRE_TIME` до `PREV_FIRE_TIME + 1` час. При этом планировщик был выключен на 8 часов. В таком случае необходимо, чтобы после включения планировщика все 8 просроченных запусков задания В были в конечном итоге запущены, и таким образом сгенерированы отчеты за каждый из часов простоя.

Следует также отметить, что применение этой политики к триггерам, которые не привязаны к каким-либо уникальным данным, может привести к ненужному загромождению очереди планировщика и ухудшению общей производительности приложения. Например, для каждого из пользователей Creatio настроена синхронизация почты Exchange с интервалом в 1 минуту. При этом 1.5 часа выполнялось обновление. После обновления Quartz, прежде чем приступить к заданиям, запланированным на текущее время, будет выполнять для каждого из пользователей синхронизацию почты 90 раз. Хотя достаточно выполнить просроченную синхронизацию почты однократно, а затем вернуться к выполнению задания согласно расписанию.

Run immediately and continue

К этой группе политик относятся:

- `SimpleTrigger.FireNow` ;
- `SimpleTrigger.RescheduleNowWithExistingRepeatCount` ;
- `SimpleTrigger.RescheduleNowWithRemainingRepeatCount` ;
- `CronTrigger.FireOnceNow` ;

- `CalendarIntervalTrigger.FireOnceNow` .

Эти политики следует применять, если необходимо просроченное задание выполнить как можно скорее, но один раз, а далее выполнять задание согласно расписанию. В такую категорию входят, например, задания по синхронизации почты (такие, как `<user>@<server>_LoadExchangeEmailsProcess_<userId>` , `SyncImap_<user>@<server>_<userId>`), а также задачи `RemindingCountersJob` , `SyncWithLDAPProcess` .

Например, для каждого из пользователей настроена синхронизация почты с интервалом 5 минут, запускаемая триггерами `<user>@<server>_LoadExchangeEmailProcess_<userId>Trigger` . При этом сайт обновлялся с 1:30 до 2:43. Когда сайт был запущен (в 2:43), время следующего запуска для триггеров `<user>@<server>_LoadExchangeEmailsProcess_<userId>Trigger` будет обновлено на текущее (т. е., на 2:43). Соответственно, просроченные задания будут запущены один раз в 2:43 и далее будут выполняться согласно расписанию (т. е., в 2:48, 2:53, 2:58 и т. д.).

Discard and wait for next

К этой группе политик относятся:

- `SimpleTrigger.RescheduleNextWithRemainingCount` ;
- `SimpleTrigger.RescheduleNextWithExistingCount` ;
- `CronTrigger.DoNothing` ;
- `CalendarIntervalTrigger.DoNothing` .

Эти политики следует применять для заданий, которые необходимо выполнять строго в определенное время. Например, есть задание по сбору статистики, запускаемое каждый день в 3 часа ночи, когда нет активных пользователей на сайте (используется `CronTrigger`). Это задание достаточно ресурсоемкое и длительное, и в рабочее время его запускать нельзя, потому что это может замедлить работу пользователей. В таком случае нужно применить политику `CronTrigger.DoNothing` . В итоге, если по какой-то причине задание не было выполнено вовремя, следующий запуск будет запланирован на 3 часа ночи следующих суток.

Конфигурирование Quartz

thread count

Если планировщик не успевает своевременно обрабатывать задания или некоторые задания не выполнились ни разу, то, возможно, поможет увеличение числа потоков Quartz. Для этого необходимо установить необходимое количество потоков в файле `Web.config` загрузчика приложения.

Установка необходимого количества потоков в файле `Web.config` загрузчика приложения

```
<add key="quartz.threadPool.threadCount" value="5" />
```

Важно. Файл `Web.config` загрузчика приложения расположен в корневом каталоге установленного приложения Creatio.

misfireThreshold

Если увеличение числа потоков Quartz нежелательно (например, из-за ограниченных ресурсов), то оптимизировать выполнение заданий может изменение настройки `misfireThreshold` в файле `Web.config` загрузчика приложения.

Например, есть приложение с числом заданий, намного большим, чем число потоков. При этом большинство заданий выполняются с достаточно малым интервалом (1 минута). Значение настройки `misfireThreshold` равно одной минуте, число потоков равно 3.

Изменение настройки `misfireThreshold` в файле `Web.config` загрузчика приложения

```
<add key="quartz.jobStore.misfireThreshold" value="60000" />
<add key="quartz.threadPool.threadCount" value="3" />
```

Также для большинства заданий используются политики из группы `Run immediately and continue`. Это означает следующее. Quartz для большинства заданий, у которых `MISFIRE_INSTR` не равно -1 и `NEXT_FIRE_TIME` меньше текущего времени на 1 минуту (60000 мс), будет регулярно устанавливать время следующего запуска на текущее время. Это означает, что будет теряться изначальный порядок запланированных заданий, потому что всем заданиям будет установлено время запуска на текущее время. Как следствие, увеличивается вероятность того, что Quartz чаще будет брать в работу одни и те же задания, при этом игнорируя другие.

На рисунке отображена очередь планировщика после 15 минут работы. Задания, у которых значение `PREV_FIRE_TIME` равно `NULL`, ни разу не выполнялись. Очевидно, что таких заданий достаточно много.

Затем следует увеличить значение `misfireThreshold` до 10 минут (очистив при этом `PREV_FIRE_TIME` в `QRTZ_TRIGGERS`).

TRIGGER_NAME	TRIGGER_GROUP	NEXT_FIRE_TIME	PREV_FIRE_TIME	Last repeat interval, min	TRIGGER_STATE	TRIGGER_TYPE	START_TIME	MISFIRE_INSTR
MinutelyJob_15Trigger	TestGroup	2017-11-21 19:17:20.860	NULL	NULL	WAITING	SIMPLE	2017-11-21 19:17:20.860	3
MinutelyJob_16Trigger	TestGroup	2017-11-21 19:17:20.863	NULL	NULL	WAITING	SIMPLE	2017-11-21 19:17:20.863	3
MinutelyJob_17Trigger	TestGroup	2017-11-21 19:17:20.867	NULL	NULL	WAITING	SIMPLE	2017-11-21 19:17:20.867	3
MinutelyJob_18Trigger	TestGroup	2017-11-21 19:17:20.867	NULL	NULL	WAITING	SIMPLE	2017-11-21 19:17:20.867	3
MinutelyJob_19Trigger	TestGroup	2017-11-21 19:17:20.870	NULL	NULL	WAITING	SIMPLE	2017-11-21 19:17:20.870	3
SyncImap_postulga.alexey@gmail.com_73b869f-34f3...	IMAP	2017-11-21 19:17:20.870	NULL	NULL	WAITING	SIMPLE	2017-11-21 19:17:20.870	3
MinutelyJob_5Trigger	TestGroup	2017-11-21 19:17:20.873	NULL	NULL	WAITING	SIMPLE	2017-11-21 19:17:20.873	3
Terrasoft.Configuration.CampaignFailoverHandlerTrigger	CampaignJobGroup	2017-11-21 19:17:20.873	NULL	NULL	WAITING	SIMPLE	2017-11-21 19:17:20.873	3
Terrasoft.Configuration.TriggerEmailFailoverHandlerTri...	Mailing	2017-11-21 19:17:20.877	NULL	NULL	WAITING	SIMPLE	2017-11-21 19:17:20.877	3
Terrasoft.Configuration.BulkEmailFailoverHandlerTrigger	Mailing	2017-11-21 19:17:20.880	NULL	NULL	WAITING	SIMPLE	2017-11-21 19:17:20.880	3
MinutelyJob_6Trigger	TestGroup	2017-11-21 19:17:20.880	NULL	NULL	WAITING	SIMPLE	2017-11-21 19:17:20.880	3
MinutelyJob_7Trigger	TestGroup	2017-11-21 19:17:20.883	NULL	NULL	WAITING	SIMPLE	2017-11-21 19:17:20.883	3
MinutelyJob_0Trigger	TestGroup	2017-11-21 19:17:20.883	2017-11-21 19:04:37.643	12.72065943000	WAITING	SIMPLE	2017-11-21 19:17:20.883	3
MinutelyJob_11Trigger	TestGroup	2017-11-21 19:17:20.887	2017-11-21 19:04:37.663	12.72035926333	WAITING	SIMPLE	2017-11-21 19:17:20.887	3
MinutelyJob_2Trigger	TestGroup	2017-11-21 19:17:20.887	2017-11-21 19:04:37.667	12.72034262833	WAITING	SIMPLE	2017-11-21 19:17:20.887	3
CESWebHooksSyncTrigger	Mailing	2017-11-21 19:17:20.890	2017-11-21 19:08:20.733	9.00257776333	WAITING	CAL_INT	2017-11-21 18:39:38.140	1
MandrillScheduledMailingTrigger	Mailing	2017-11-21 19:17:20.890	2017-11-21 19:08:20.740	9.00252771833	WAITING	CAL_INT	2017-11-21 18:39:38.197	1
MT_287466734163338083	DEFAULT	2017-11-21 19:18:20.857	NULL	NULL	WAITING	SIMPLE	2017-11-21 19:08:42.410	0
Terrasoft.Configuration.ML.MLModelTrainerJob, Terra...	DEFAULT	2017-11-21 19:18:20.860	NULL	NULL	WAITING	SIMPLE	2017-11-21 19:08:42.433	0
Terrasoft.Configuration.EmailMining.EmailMiningJob, T...	DEFAULT	2017-11-21 19:18:20.860	NULL	NULL	WAITING	SIMPLE	2017-11-21 19:08:42.450	0
Terrasoft.Configuration.NotificationsJobTrigger	NotificationsCountersGroup	2017-11-21 19:18:20.863	2017-11-21 19:08:20.727	10.00226696166	WAITING	SIMPLE	2017-11-21 19:18:20.863	3
A.Postulga@terrasoft.ru_LoadExchangeEmailsProces...	Exchange	2017-11-21 19:18:20.863	2017-11-21 19:08:20.743	10.00205025333	WAITING	SIMPLE	2017-11-21 19:18:20.863	3
MinutelyJob_8Trigger	TestGroup	2017-11-21 19:18:20.867	2017-11-21 19:08:20.743	10.00205029000	WAITING	SIMPLE	2017-11-21 19:18:20.867	3
MinutelyJob_9Trigger	TestGroup	2017-11-21 19:18:20.870	NULL	NULL	WAITING	SIMPLE	2017-11-21 19:18:20.870	3
MinutelyJob_10Trigger	TestGroup	2017-11-21 19:18:20.870	NULL	NULL	WAITING	SIMPLE	2017-11-21 19:18:20.870	3
MinutelyJob_11Trigger	TestGroup	2017-11-21 19:18:20.873	NULL	NULL	WAITING	SIMPLE	2017-11-21 19:18:20.873	3
MinutelyJob_12Trigger	TestGroup	2017-11-21 19:18:20.873	NULL	NULL	WAITING	SIMPLE	2017-11-21 19:18:20.873	3
c999bb9e-070e-411f-8d19-181e6393b6e8	DEFAULT	2017-11-21 19:18:25.517	NULL	NULL	WAITING	SIMPLE	2017-11-21 19:08:25.517	0
RemindingCountersJob_73b869f-34f3-4#20-ab4d-748...	RemindingCountersGroup	2017-11-21 19:18:48.613	2017-11-21 19:17:48.613	1.00000000000	BLOCKED	CAL_INT	2017-11-21 19:08:48.613	0
MinutelyJob_3Trigger	TestGroup	2017-11-21 19:19:20.723	2017-11-21 19:18:20.723	1.00000000000	BLOCKED	SIMPLE	2017-11-21 19:08:20.723	3
MinutelyJob_4Trigger	TestGroup	2017-11-21 19:19:20.723	2017-11-21 19:18:20.723	1.00000000000	BLOCKED	SIMPLE	2017-11-21 19:08:20.723	3
Terrasoft.Configuration.DelayedNotifyingTrigger	DelayedNotificationGroup	2017-11-21 19:23:20.697	2017-11-21 19:18:20.697	5.00000000000	WAITING	SIMPLE	2017-11-21 19:08:20.697	3
SyncWithLDAPProcessTrigger	LDAP	2017-11-21 19:43:35.327	NULL	NULL	WAITING	CAL_INT	2017-11-21 18:43:35.327	0
NotificationCleanerTrigger	NotificationCleanerGroup	2017-11-22 02:00:00.000	NULL	NULL	WAITING	CRON	2017-11-21 18:39:37.000	0
ActualizeActiveContactsTrigger	Mailing	2017-11-22 02:00:00.000	NULL	NULL	WAITING	CRON	2017-11-21 18:39:37.997	1
GenerateAnniversaryRemindersTrigger	GenerateAnniversaryRem...	2017-11-22 03:00:00.000	NULL	NULL	WAITING	CRON	2017-11-21 19:08:25.000	0
Terrasoft.Configuration.DelayedNotificationCleaningTr...	DelayedNotificationGroup	2017-11-22 18:39:37.843	NULL	NULL	WAITING	SIMPLE	2017-11-21 18:39:37.843	3

Увеличение значения misfireThreshold

```
<add key="quartz.jobStore.misfireThreshold" value="600000" />
```

После 15 минут работы планировщика очередь будет выглядеть следующим образом:

TRIGGER_NAME	TRIGGER_GROUP	NEXT_FIRE_TIME	PREV_FIRE_TIME	Last repeat interval, min	TRIGGER_STATE	TRIGGER_TYPE	START_TIME	MISFIRE_INSTR
MinutelyJob_13Trigger	TestGroup	2017-11-21 19:23:20.960	2017-11-21 19:22:20.960	1.000000000000	WAITING	SIMPLE	2017-11-21 19:20:20.960	3
Terrasoft.Configuration.TriggerEmailFailoverHandlerTri...	Mailing	2017-11-21 19:25:20.980	2017-11-21 19:20:20.980	5.000000000000	WAITING	SIMPLE	2017-11-21 19:20:20.980	3
Terrasoft.Configuration.BulkEmailFailoverHandlerTrigger	Mailing	2017-11-21 19:25:20.980	2017-11-21 19:20:20.980	5.000000000000	WAITING	SIMPLE	2017-11-21 19:20:20.980	3
RemindingCountersJob_73b869f-34f3-420-ab4d-748...	RemindingCountersGroup	2017-11-21 19:25:50.787	2017-11-21 19:24:50.787	1.000000000000	BLOCKED	CAL_INT	2017-11-21 19:22:50.787	0
MinutelyJob_3Trigger	TestGroup	2017-11-21 19:26:20.723	2017-11-21 19:25:20.723	1.000000000000	BLOCKED	SIMPLE	2017-11-21 19:08:20.723	3
MinutelyJob_4Trigger	TestGroup	2017-11-21 19:26:20.723	2017-11-21 19:25:20.723	1.000000000000	BLOCKED	SIMPLE	2017-11-21 19:08:20.723	3
Terrasoft.Configuration.DelayedNotifyingTrigger	DelayedNotificationGroup	2017-11-21 19:28:20.697	2017-11-21 19:23:20.697	5.000000000000	WAITING	SIMPLE	2017-11-21 19:08:20.697	3
SyncInmap_postulga_alexey@gmail.com_73b869f-34f3...	IMAP	2017-11-21 19:32:39.110	2017-11-21 19:20:20.973	12.30228216333	WAITING	SIMPLE	2017-11-21 19:32:39.110	3
MinutelyJob_5Trigger	TestGroup	2017-11-21 19:32:39.117	2017-11-21 19:20:20.977	12.30234882000	WAITING	SIMPLE	2017-11-21 19:32:39.117	3
MinutelyJob_6Trigger	TestGroup	2017-11-21 19:32:39.117	2017-11-21 19:20:20.983	12.30224896000	WAITING	SIMPLE	2017-11-21 19:32:39.117	3
MinutelyJob_7Trigger	TestGroup	2017-11-21 19:32:39.120	2017-11-21 19:20:20.983	12.30226555000	WAITING	SIMPLE	2017-11-21 19:32:39.120	3
MinutelyJob_0Trigger	TestGroup	2017-11-21 19:32:39.120	2017-11-21 19:20:20.987	12.30224884166	WAITING	SIMPLE	2017-11-21 19:32:39.120	3
MinutelyJob_1Trigger	TestGroup	2017-11-21 19:32:39.123	2017-11-21 19:20:20.987	12.30224826833	WAITING	SIMPLE	2017-11-21 19:32:39.123	3
MinutelyJob_2Trigger	TestGroup	2017-11-21 19:32:39.123	2017-11-21 19:20:20.990	12.30224890666	WAITING	SIMPLE	2017-11-21 19:32:39.123	3
Terrasoft.Configuration.Notifications.JobTrigger	NotificationsCountersGroup	2017-11-21 19:32:39.127	2017-11-21 19:20:21.057	12.30116543500	WAITING	SIMPLE	2017-11-21 19:32:39.127	3
A.Postulga@terrasoft.ru_LoadExchangeEmailsProces...	Exchange	2017-11-21 19:32:39.130	2017-11-21 19:20:21.060	12.30116547833	WAITING	SIMPLE	2017-11-21 19:32:39.130	3
MinutelyJob_8Trigger	TestGroup	2017-11-21 19:32:39.130	2017-11-21 19:20:21.060	12.30116555333	WAITING	SIMPLE	2017-11-21 19:32:39.130	3
MinutelyJob_9Trigger	TestGroup	2017-11-21 19:32:39.133	2017-11-21 19:20:21.063	12.30116544666	WAITING	SIMPLE	2017-11-21 19:32:39.133	3
MinutelyJob_10Trigger	TestGroup	2017-11-21 19:32:39.133	2017-11-21 19:20:21.067	12.30116546166	WAITING	SIMPLE	2017-11-21 19:32:39.133	3
MinutelyJob_11Trigger	TestGroup	2017-11-21 19:32:39.137	2017-11-21 19:20:21.067	12.30116550166	WAITING	SIMPLE	2017-11-21 19:32:39.137	3
MinutelyJob_12Trigger	TestGroup	2017-11-21 19:32:39.140	2017-11-21 19:20:21.070	12.30118216666	WAITING	SIMPLE	2017-11-21 19:32:39.140	3
CESWebHooksSyncTrigger	Mailing	2017-11-21 19:32:39.143	2017-11-21 19:20:38.140	12.01673093833	WAITING	CAL_INT	2017-11-21 18:39:38.140	1
MandrilScheduledMailingTrigger	Mailing	2017-11-21 19:32:39.143	2017-11-21 19:20:38.197	12.01579751166	WAITING	CAL_INT	2017-11-21 18:39:38.197	1
MinutelyJob_14Trigger	TestGroup	2017-11-21 19:32:39.147	2017-11-21 19:21:20.963	11.30308236666	WAITING	SIMPLE	2017-11-21 19:32:39.147	3
MinutelyJob_15Trigger	TestGroup	2017-11-21 19:32:39.150	2017-11-21 19:21:20.963	11.30308227500	WAITING	SIMPLE	2017-11-21 19:32:39.150	3
MinutelyJob_16Trigger	TestGroup	2017-11-21 19:32:39.150	2017-11-21 19:21:20.967	11.30308224833	WAITING	SIMPLE	2017-11-21 19:32:39.150	3
MinutelyJob_17Trigger	TestGroup	2017-11-21 19:32:39.153	2017-11-21 19:21:20.967	11.30308234666	WAITING	SIMPLE	2017-11-21 19:32:39.153	3
MinutelyJob_18Trigger	TestGroup	2017-11-21 19:32:39.210	2017-11-21 19:21:20.970	11.30399938833	WAITING	SIMPLE	2017-11-21 19:32:39.210	3
MinutelyJob_19Trigger	TestGroup	2017-11-21 19:32:39.213	2017-11-21 19:21:20.970	11.30401564500	WAITING	SIMPLE	2017-11-21 19:32:39.213	3
MT_520821574169140644	DEFAULT	2017-11-21 19:32:39.217	NULL	NULL	WAITING	SIMPLE	2017-11-21 19:22:30.010	0
Terrasoft.Configuration.ML.ModelTrainerJob, Terra...	DEFAULT	2017-11-21 19:32:39.227	NULL	NULL	WAITING	SIMPLE	2017-11-21 19:22:30.037	0
Terrasoft.Configuration.EmailMining.EmailMiningJob, T...	DEFAULT	2017-11-21 19:32:39.227	NULL	NULL	WAITING	SIMPLE	2017-11-21 19:22:30.047	0
Terrasoft.Configuration.CampaignFailoverHandlerTrigger	CampaignJobGroup	2017-11-21 19:35:20.977	2017-11-21 19:20:20.977	15.000000000000	WAITING	SIMPLE	2017-11-21 19:20:20.977	3
0778a81c-e91-402a-826a-77ab0e3afb25	DEFAULT	2017-11-21 19:37:25.223	NULL	NULL	WAITING	SIMPLE	2017-11-21 19:22:25.223	0
SyncWithLDAPProcessTrigger	LDAP	2017-11-21 19:43:35.327	NULL	NULL	WAITING	CAL_INT	2017-11-21 18:43:35.327	0
NotificationCleanerTrigger	NotificationCleanerGroup	2017-11-22 02:00:00.000	NULL	NULL	WAITING	CRON	2017-11-21 18:39:37.000	0
ActualizeActiveContactsTrigger	Mailing	2017-11-22 02:00:00.000	NULL	NULL	WAITING	CRON	2017-11-21 18:39:37.997	1
GenerateAnniversaryRemindersTrigger	GenerateAnniversaryRem...	2017-11-22 03:00:00.000	NULL	NULL	WAITING	CRON	2017-11-21 19:22:25.000	0
Terrasoft.Configuration.DelayedNotificationCleaningTri...	DelayedNotificationGroup	2017-11-22 18:39:37.843	NULL	NULL	WAITING	SIMPLE	2017-11-21 18:39:37.843	3

Очевидно, что ни разу не запущенных заданий стало намного меньше.

Увеличение `misfireThreshold` приводит к тому, что планировщик более равномерно выполняет задания. Другими словами, выполняются практически все задания из очереди. Очевидно, что из-за нехватки потоков планировщик не успевает каждое из заданий выполнять через минуту. Это видно по колонке `[Last repeat interval]`, значение в которой равно `NEXT_FIRE_TIME - PREV_FIRE_TIME`, мин. Однако, при этом планировщик выполняет каждое из заданий.

batchTriggerAcquisitionMaxCount

Увеличение `batchTriggerAcquisitionMaxCount` может оптимизировать работу планировщика, если не используется кластеризованная конфигурация Quartz (используется один узел планировщика).

Политики Quartz для обработки не отработавших вовремя заданий

В Quartz существуют как политики, общие для всех типов триггеров, так и политики, специфичные для конкретного типа триггера. В таблице перечислены все политики, используемые для триггеров `SimpleTrigger`, `CronTrigger` и `CalendarIntervalTrigger`.

Политики триггеров

Политика Quartz	Значение	Значение	Тип триггера
	MISFIRE_INSTR	Terrasoft.Core.Scheduler.AppSchedulerMisfireInstruction	

IgnoreMisfirePolicy	-1	IgnoreMisfirePolicy	для всех типов
---------------------	----	---------------------	----------------

Описание поведения IgnoreMisfirePolicy

Триггеры с IgnoreMisfirePolicy будут обязательно выполнены. Соответственно, для таких триггеров Quartz не будет обновлять время следующего запуска (NEXT_FIRE_TIME).

Все просроченные задания Quartz попытается выполнить как можно скорее, после чего вернется к изначальному расписанию триггера. Например, запланировано задание с триггером SimpleTrigger на 10 повторений. Исходные условия:

- START_TIME = 9:00;
- REPEAT_COUNT = 9 (первое выполнение + 9 повторений);
- REPEAT_INTERVAL = 0:15.

Если планировщик был выключен с 8:50 до 9:20, то при включении Quartz попытается как можно скорее выполнить 2 просроченных задания (в 9:00 и 9:15). Далее он будет выполнять 8 оставшихся заданий по расписанию — в 9:30; 9:45 и т.д.

SmartPolicy	0	SmartPolicy	для всех типов
-------------	---	-------------	----------------

Описание поведения SmartPolicy

Используется Quartz по умолчанию для всех типов триггеров. В зависимости от типа и конфигурации триггера Quartz выберет соответствующую политику. В псевдокоде ниже приведен алгоритм выбора для версии Quartz 2.3.2.

```
if (TRIGGER_TYPE == 'SIMPLE') // Триггер Simple.
    if (REPEAT_COUNT == 0) // Без повторов.
        MISFIRE_INSTR = 1 // SimpleTrigger.FireNow
    else if (REPEAT_COUNT == -1) // Повторять бесконечно.
        MISFIRE_INSTR = 4 // SimpleTrigger.RescheduleNextWithRemainingCount
    else // Указано количество повторов.
        MISFIRE_INSTR = 2 // SimpleTrigger.RescheduleNowWithExistingRepeatCount
else if (TRIGGER_TYPE == 'CAL_INT') // Триггер CalendarInterval.
    MISFIRE_INSTR = 1 // CalendarIntervalTrigger.FireOnceNow
else if (TRIGGER_TYPE == 'CRON') // Триггер Cron.
    MISFIRE_INSTR = 1 // CronTrigger.FireOnceNow
```

SimpleTrigger.FireNow	1	FireNow	SimpleTrigger
-----------------------	---	---------	---------------

Описание поведения SimpleTrigger.FireNow

Применяется для триггеров SimpleTrigger у которых значение REPEAT_COUNT равно 0 (триггеры, рассчитанные на 1 запуск). Если значение REPEAT_COUNT не равно 0, то будет применена политика

`SimpleTrigger.RescheduleNowWithRemainingRepeatCount` .

Например, запланировано задание с триггером `SimpleTrigger` . Исходные условия:

- `START_TIME = 9:00 ;`
- `REPEAT_COUNT = 0 .`

Если планировщик был выключен с 8:50 до 9:20, то при включении Quartz попытается как можно скорее выполнить задание. В результате, в 9:20 или чуть позже задание будет выполнено.

<code>SimpleTrigger.RescheduleNowWithExistingRepeatCount</code>	2	<code>RescheduleNowWithExistingRepeatCount</code>	<code>SimpleTrigger</code>
---	---	---	----------------------------

Описание поведения `SimpleTrigger.RescheduleNowWithExistingRepeatCount`

Планировщик пытается как можно скорее выполнить первое просроченное задание. Все остальные запуски триггера будут выполнены с интервалом `REPEAT_INTERVAL` .

Например, запланировано задание с триггером `SimpleTrigger` на 10 повторений. Исходные условия:

- `START_TIME = 9:00 ;`
- `REPEAT_COUNT = 9 ;`
- `REPEAT_INTERVAL = 0:15 .`

Если планировщик был выключен с 8:50 до 9:20, то при включении Quartz выполнит первое просроченное задание, назначенное на 9:00, (из заданий в 9:00 и 9:15) в 9:20. Оставшиеся 9 запусков выполнит в 9:35, 9:50 и т.д. .

На заметку. Для методов `AppScheduler.ScheduleMinutelyJob` поведение

`RescheduleNowWithExistingRepeatCount` ИДЕНТИЧНО `RescheduleNowWithRemainingRepeatCount` , а поведение `RescheduleNextWithRemainingCount` ИДЕНТИЧНО `RescheduleNextWithExistingCount` , поскольку используются триггеры с `REPEAT_COUNT = -1` .

<code>SimpleTrigger.RescheduleNowWithRemainingRepeatCount</code>	3	<code>RescheduleNowWithRemainingRepeatCount</code>	<code>SimpleTrigger</code>
--	---	--	----------------------------

Описание поведения `SimpleTrigger.RescheduleNowWithRemainingRepeatCount`

Планировщик пытается как можно скорее выполнить первое просроченное задание. Остальные просроченные задания игнорируются. Планировщик выполняет оставшиеся задания, которые не были просрочены, с интервалом `REPEAT_INTERVAL` .

Например, запланировано задание с триггером `SimpleTrigger` на 10 повторений. Исходные условия:

- `START_TIME = 9:00 ;`
- `REPEAT_COUNT = 9 ;`
- `REPEAT_INTERVAL = 0:15 .`

Если планировщик был выключен с 8:50 до 9:20, то при включении Quartz выполнит первое просроченное задание (из заданий в 9:00 и 9:15) в 9:20. Второе просроченное задание будет проигнорировано, и оставшиеся 8 запусков будут выполнены в 9:35, 9:50 и т.д.

<code>SimpleTrigger.RescheduleNextWithRemainingCount</code>	4	<code>RescheduleNextWithRemainingCount</code>	<code>SimpleTrigger</code>
---	---	---	----------------------------

Описание поведения `SimpleTrigger.RescheduleNextWithRemainingCount`

Планировщик игнорирует просроченные задания и ждет следующего планового запуска задания. При наступлении времени следующего запуска будут выполнены оставшиеся не просроченные задания с интервалом `REPEAT_INTERVAL`.

Например, запланировано задание с триггером `SimpleTrigger` на 10 повторений. Исходные условия:

- `START_TIME = 9:00;`
- `REPEAT_COUNT = 9;`
- `REPEAT_INTERVAL = 0:15.`

Если планировщик был выключен с 8:50 до 9:20, то при включении Quartz выполнит оставшиеся 8 непросроченных заданий в 9:30; 9:45 и т.д.

<code>SimpleTrigger.RescheduleNextWithExistingCount</code>	5	<code>RescheduleNextWithExistingCount</code>	<code>SimpleTrigger</code>
--	---	--	----------------------------

Описание поведения `SimpleTrigger.RescheduleNextWithExistingCount`

Планировщик будет ждать следующего времени запуска и выполнит все оставшиеся задания с интервалом `REPEAT_INTERVAL`.

Например, запланировано задание с триггером `SimpleTrigger` на 10 повторений. Исходные условия:

- `START_TIME = 9:00;`
- `REPEAT_COUNT = 9;`
- `REPEAT_INTERVAL = 0:15.`

Если планировщик был выключен с 8:50 до 9:20, то при включении Quartz выполнит все 10 заданий в 9:30; 9:45 и т.д.

<code>CronTrigger.FireOnceNow</code>	1	-	<code>CronTrigger</code>
--------------------------------------	---	---	--------------------------

Описание поведения `CronTrigger.FireOnceNow`

Планировщик пытается как можно скорее выполнить первое просроченное задание. Остальные просроченные задания игнорируются. Оставшиеся непросроченные задания планировщик выполняет согласно с расписанием.

Например, запланировано задание с триггером `CronTrigger`: `CRON_EXPRESSION = '0 0 9-17 ? * MON-FRI'` (с понедельника по пятницу с 9:00 до 17:00). Если планировщик был выключен с 8:50 до 10:20, то при включении в 10:20 Quartz выполнит первое просроченное задание из двух (в 9:00 и 10:00).

Далее задания будут выполняться в 11:00, 12:00 и т.д.

<code>CronTrigger.DoNothing</code>	2	-	CronTrigger
------------------------------------	---	---	-------------

Описание поведения `CronTrigger.DoNothing`

Планировщик игнорирует все просроченные задания. Оставшиеся непросроченные задания выполняются согласно с расписанием.

Например, запланировано задание с триггером `CronTrigger`: `CRON_EXPRESSION = '0 0 9-17 ? * MON-FRI'` (с понедельника по пятницу с 9:00 до 17:00). Если планировщик был выключен с 8:50 до 10:20, то при включении Quartz начнет выполнять задания с 11:00 (в 11:00, 12:00 и т.д.).

<code>CalendarIntervalTrigger. FireOnceNow</code>	1	-	Calendar Interval Trigger
---	---	---	---------------------------------

Описание поведения `CalendarIntervalTrigger.FireOnceNow`

Поведение аналогично `CronTrigger.FireOnceNow`.

<code>CalendarIntervalTrigger. DoNothing</code>	2	-	Calendar Interval Trigger
---	---	---	---------------------------------

Описание поведения `CalendarIntervalTrigger.DoNothing`

Поведение аналогично `CronTrigger.DoNothing`.

Особенности работы с планировщиком при использовании горизонтального масштабирования описаны в статье [Настроить горизонтальное масштабирование](#).