1. Asynchronous Apex
2. Monitor Asynchronous Apex

# Monitor Asynchronous Apex

## Learning Objectives

After completing this unit, you'll know:

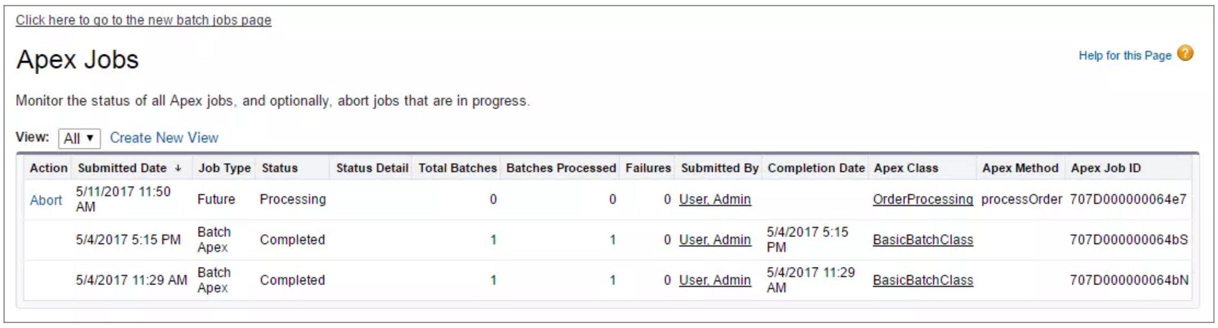
- How to monitor the different types of jobs.
- How to use the flex queue.

## Monitoring Asynchronous Jobs

The great thing about async jobs is that they work silently in the background. The tough thing about async jobs is that they work silently in the background. Luckily there are a few ways to monitor what is going on with your jobs under the covers.

You can monitor the status of all jobs in the Salesforce user interface. From **Setup**, enter `Jobs` in the Quick Find box, then select **Apex Jobs**.

The Apex Jobs page shows all asynchronous Apex jobs with information about each job's execution. The following screenshot shows one future method job and two completed batch jobs for the same Batch Apex class.


Click here to go to the new batch jobs page

Apex Jobs

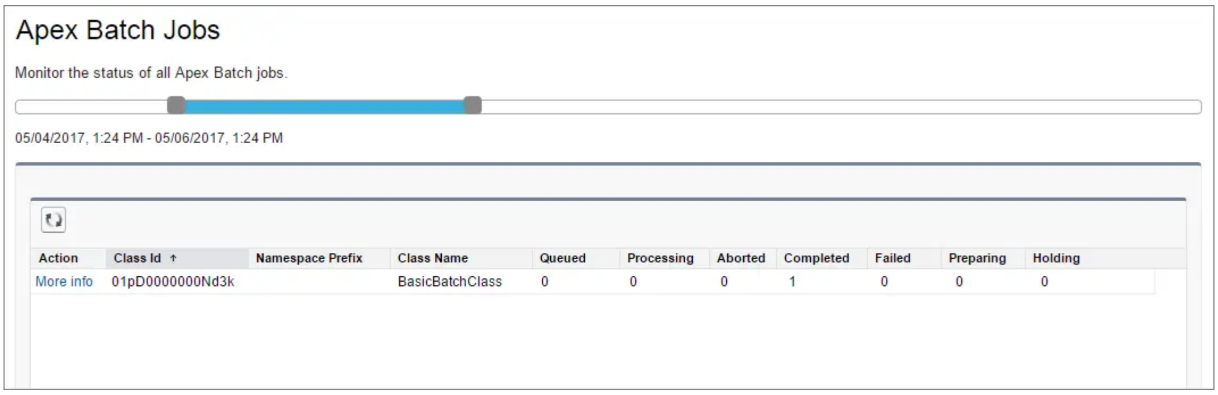
Help for this Page

Monitor the status of all Apex jobs, and optionally, abort jobs that are in progress.

View: All ▾ Create New View

| Action | Submitted Date ↓ | Job Type | Status | Status Detail | Total Batches | Batches Processed | Failures | Submitted By | Completion Date | Apex Class | Apex Method | Apex Job ID |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Abort | 5/11/2017 11:50 AM | Future | Processing | | 0 | 0 | 0 | User, Admin | | OrderProcessing | processOrder | 707D000000064e7 |
| | 5/4/2017 5:15 PM | Batch Apex | Completed | | 1 | 1 | 0 | User, Admin | 5/4/2017 5:15 PM | BasicBatchClass | | 707D000000064bS |
| | 5/4/2017 11:29 AM | Batch Apex | Completed | | 1 | 1 | 0 | User, Admin | 5/4/2017 11:29 AM | BasicBatchClass | | 707D000000064bN |

If you have many batch jobs, use the Batch Jobs page to view only batch jobs. To open the Apex Batch Jobs page, click the link at the top of the Apex Jobs page. Use the slider in the Apex Batch Jobs page to select a specific date range and narrow down the list of batch jobs displayed. You can view past jobs that haven't been deleted yet. The Batch Jobs page groups jobs by the batch class.

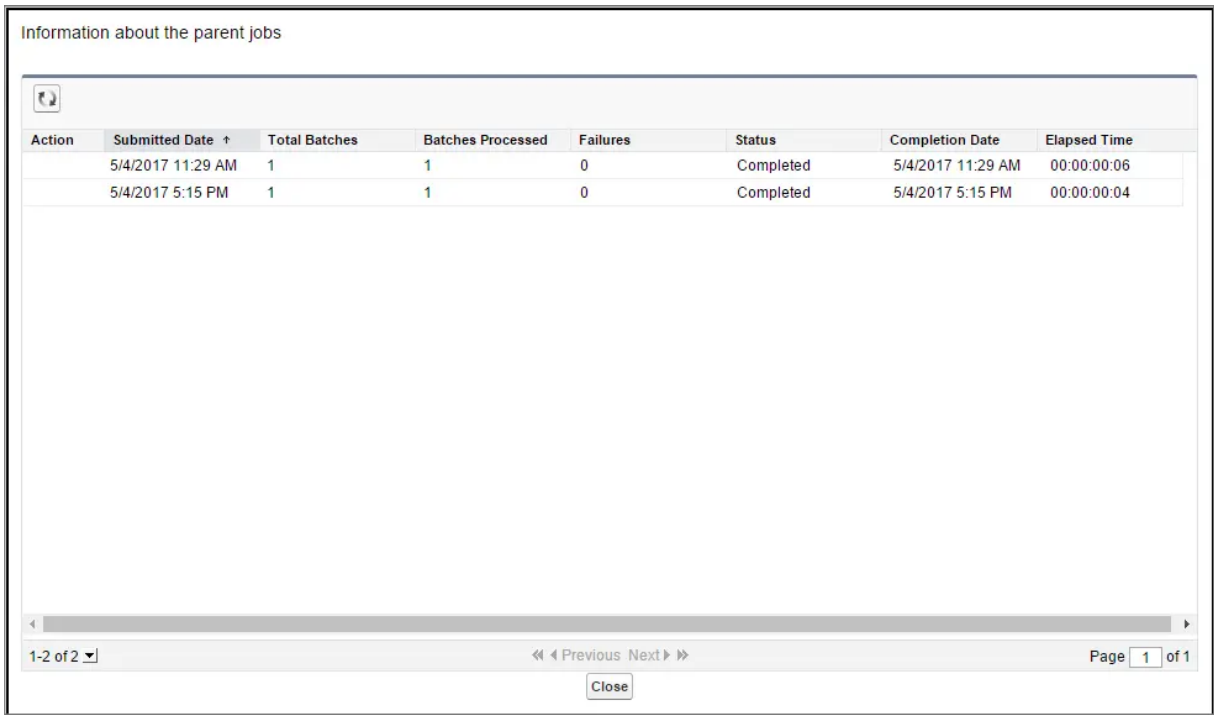

Apex Batch Jobs

Monitor the status of all Apex Batch jobs.

05/04/2017, 1:24 PM - 05/06/2017, 1:24 PM

| Action | Class Id ↑ | Namespace Prefix | Class Name | Queued | Processing | Aborted | Completed | Failed | Preparing | Holding |
|---|---|---|---|---|---|---|---|---|---|---|
| More info | 01pD0000000Nd3k | | BasicBatchClass | 0 | 0 | 0 | 1 | 0 | 0 | 0 |

Click More Info next to a class ID you're interested in to find out details about the jobs executed for that class. This image shows the popup that displays after clicking More Info. This batch class has two jobs that executed successfully.


Information about the parent jobs

| Action | Submitted Date ↑ | Total Batches | Batches Processed | Failures | Status | Completion Date | Elapsed Time |
|---|---|---|---|---|---|---|---|
| | 5/4/2017 11:29 AM | 1 | 1 | 0 | Completed | 5/4/2017 11:29 AM | 00:00:00:06 |
| | 5/4/2017 5:15 PM | 1 | 1 | 0 | Completed | 5/4/2017 5:15 PM | 00:00:00:04 |

You can also monitor the status of Apex jobs in the Apex Flex Queue, and reorder them to control which jobs are processed first. From Setup, enter `Jobs` in the Quick Find box, then select **Apex Flex Queue**.

## Monitoring Future Jobs

Future jobs show up on the Apex Jobs page like any other jobs. However, future jobs are not part of the flex queue currently.

You can query `AsyncApexJob` to find your future job, but there's a caveat. Since kicking off a future job does not return an ID, you have to filter on some other field such as `MethodName`, or `JobType`, to find your job. There are a few sample SOQL queries in this Stack Exchange post that might help.

## Monitoring Queued Jobs with SOQL

To query information about your submitted job, perform a SOQL query on AsyncApexJob by filtering on the job ID that the `System.enqueueJob` method returns.

```
AsyncApexJob jobInfo = [SELECT Status, NumberOfErrors
    FROM AsyncApexJob WHERE Id = :jobID];
```

Copy

## Monitoring Queue Jobs with the Flex Queue

The Apex Flex queue enables you to submit up to 100 batch jobs for execution. Any jobs that are submitted for execution are in holding status and are placed in the Apex Flex queue. Up to 100 batch jobs can be in the holding status.

Jobs are processed first-in first-out—in the order in which they're submitted. You can look at the current queue order and shuffle the queue, so that you could move an important job to the front, or less important ones to the back.

When system resources become available, the system picks up the next job from the top of the Apex Flex queue and moves it to the batch job queue. The system can process up to five queued or active jobs simultaneously for each organization. The status of these moved jobs changes from Holding to Queued. Queued jobs get executed when the system is ready to process new jobs. Like other jobs, you can monitor queued jobs in the Apex Jobs page.

## Monitoring Scheduled Jobs

After an Apex job has been scheduled, you can obtain more information about it by running a SOQL query on `CronTrigger`. The following sample queries the number of times the job has run, and the date and time when the job is scheduled to run again. It uses a jobID variable which is returned from the `System.schedule` method.

```
CronTrigger ct = [SELECT TimesTriggered, NextFireTime FROM CronTrigger WHERE Id = :jobID];
```

Copy

If you’re performing this query inside the execute method of your schedulable class, you can obtain the ID of the current job by calling `getTriggerId` on the `SchedulableContext` argument variable.

```
global class DoAwesomeStuff implements Schedulable {
    global void execute(SchedulableContext sc) {
         // some awesome code
         CronTrigger ct = [SELECT TimesTriggered, NextFireTime FROM CronTrigger WHERE Id = :sc.getTriggerId()];
    }

}
```

Copy

You can also get the job’s name and the job’s type from the `CronJobDetail` record associated with the `CronTrigger` record. To do so, use the `CronJobDetail` relationship when performing a query on `CronTrigger`. This example retrieves the most recent `CronTrigger` record with the job name and type from CronJobDetail.

```
CronTrigger job = [SELECT Id, CronJobDetail.Id, CronJobDetail.Name, CronJobDetail.JobType FROM CronTrigger ORDER BY CreatedDate DESC LIMIT 1];
```

Copy

Alternatively, you can query `CronJobDetail` directly to get the job’s name and type. The following example gets the job’s name and type for the `CronTrigger` record queried in the previous example. The corresponding `CronJobDetail` record ID is obtained by the `CronJobDetail.Id` expression on the `CronTrigger` record.

```
CronJobDetail ctd = [SELECT Id, Name, JobType FROM CronJobDetail WHERE Id = :job.CronJobDetail.Id];
```

Copy

And lastly, to obtain the total count of all Apex scheduled jobs, excluding all other scheduled job types, perform the following query. Note the value '7' is specified for the job type, which corresponds to the scheduled Apex job type. See CronJobDetail in the Resources section for a list of all types.

```
SELECT COUNT() FROM CronTrigger WHERE CronJobDetail.JobType = ‘7’
```

Copy

## Resources

- CronJobDetail

## Quiz Complete!

### +100 points

Asynchronous Apex
100%
Progress: 100%
View more modules