Apex Reference Guide / System Namespace / FlexQueue Class

FlexQueue Class

Contains methods that reorder batch jobs in the Apex flex queue.

Namespace

System

Usage

You can place up to 100 batch jobs in a holding status for future execution. When system resources become available, the jobs are taken from the top of the Apex flex queue and moved to the batch job queue. Up to five queued or active jobs can be processed simultaneously for each org. When a job is moved out of the flex queue for processing, its status changes from Holding to Queued. Queued jobs are executed when the system is ready to process new jobs.

Use this class's methods to reorder your `Holding` jobs in the flex queue. As best practice and for safe usage, a `FlexQueue` reorder method must be the final statement in a transaction.

Example

This example moves a job to the front of the flex queue so that it's executed immediately. The job is moved by calling the `System.FlexQueue.moveJobToFront()` method with the high priority job ID as the parameter.

```
ID highPriorityJobId = Database.executeBatch(new HighPriorityBatchClass(), 200);
boolean jobMovedToFrontOfQueue = FlexQueue.moveJobToFront(highPriorityJobId);
```

- FlexQueue Methods

See Also

- Monitoring the Apex Flex Queue
- *Apex Developer Guide*: Using Batch Apex

FlexQueue Methods

The following are methods for `FlexQueue`.

- **moveAfterJob(jobToMoveId, jobInQueueId)**
  Moves the job with the ID *jobToMoveId* immediately after the job with the ID *jobInQueueId* in the flex queue. You can move *jobToMoveId* forward or backward in the queue. If either job isn't in the queue, it throws an element-not-found exception. Returns `true` if the job is moved, or `false` if *jobToMoveId* is already immediately after *jobInQueueId*, so no change is made.
- **moveBeforeJob(jobToMoveId, jobInQueueId)**
  Moves the job with the ID *jobToMoveId* immediately before the job with the ID *jobInQueueId* in the flex queue. You can move *jobToMoveId* forward or backward in the

Moves the specified job the end of the flex queue, to index position `(size - 1)`. All jobs after the job's starting position move one spot forward. If the job isn't in the queue, it throws an element-not-found exception. Returns `true` if the job is moved, or `false` if the job is already at the end of the queue, so no change is made.

- **moveJobToFront(jobId)**
  Moves the specified job to the front of the flex queue, to index position `0`. All other jobs move back one spot. If the job isn't in the queue, it throws an element-not-found exception. Returns `true` if the job is moved, or `false` if the job is already at the front of the queue, so no change is made.

moveAfterJob(jobToMoveId, jobInQueueId)

Moves the job with the ID *jobToMoveId* immediately after the job with the ID *jobInQueueId* in the flex queue. You can move *jobToMoveId* forward or backward in the queue. If either job isn't in the queue, it throws an element-not-found exception. Returns `true` if the job is moved, or `false` if *jobToMoveId* is already immediately after *jobInQueueId*, so no change is made.

Signature

```
public static Boolean moveAfterJob(Id jobToMoveId, Id jobInQueueId)
```

Parameters

jobToMoveId

Type: Id

The ID of the job to move.

jobInQueueId

Type: Id

The ID of the job to move after.

Return Value

Type: Boolean

moveBeforeJob(jobToMoveId, jobInQueueId)

Moves the job with the ID *jobToMoveId* immediately before the job with the ID *jobInQueueId* in the flex queue. You can move *jobToMoveId* forward or backward in the queue. If either job isn't in the queue, it throws an element-not-found exception. Returns `true` if the job is moved, or `false` if *jobToMoveId* is already immediately before *jobInQueueId*, so no change is made.

Signature

```
public static Boolean moveBeforeJob(Id jobToMoveId, Id jobInQueueId)
```

Parameters

jobToMoveId

Type: Id

The ID of the job to move.

jobInQueueId

Type: Id

The ID of the job to use as a reference point.

Return Value

Moves the specified job the end of the flex queue, to index position `(size - 1)`. All jobs after the job's starting position move one spot forward. If the job isn't in the queue, it throws an element-not-found exception. Returns `true` if the job is moved, or `false` if the job is already at the end of the queue, so no change is made.

Signature

```
public static Boolean moveJobToEnd(Id jobId)
```

Parameters

jobId

Type: Id

The ID of the job to move.

Return Value

Type: Boolean

moveJobToFront(jobId)

Moves the specified job to the front of the flex queue, to index position `0`. All other jobs move back one spot. If the job isn't in the queue, it throws an element-not-found exception. Returns `true` if the job is moved, or `false` if the job is already at the front of the queue, so no change is made.

Signature

```
public static Boolean moveJobToFront(Id jobId)
```

Parameters

jobId

Type: Id

The ID of the job to move.

Return Value

Type: Boolean

DID THIS ARTICLE SOLVE YOUR ISSUE?
Let us know so we can improve!

Share your feedback

DEVELOPER CENTERS	POPULAR RESOURCES	COMMUNITY
Heroku	Documentation	Trailblazer Community
MuleSoft	Component Library	Events and Calendar
Tableau	APIs	Partner Community
Commerce Cloud	Trailhead	Blog

Privacy Information Terms of Service Legal Use of Cookies Trust Cookie Preferences

✓✕ Your Privacy Choices Responsible Disclosure Contact