Batch Job Endpoints

Last Modified on 09/23/2021 9:20 am CDT

Overview of Batch Jobs

Definitions

Batch Job: We use a batch job to perform an operation on a set of records, where each record (corresponding to a "batch job item") is processed in its own background job. The batch job is identified by a BatchJobId.

> Batch Job Type: The operation being performed by a batch job is described by its BatchJobType. Be aware that specifying the BatchJobType is not enough to get the job picked up and processed; a listener for the BatchJobType must be configured first.
>
> Batch Job Parameters: Any configuration information that applies to all the batch job items is supplied via BatchJobParameters.
>
> Batch Job Status: A batch job can have a status of New, Processing, or Completed. The status of Completed does not mean that all items were processed successfully, but rather that all items were processed. To see if any items failed, check the metadata on the job details. To see why any failed, inspect the batch job items themselves.

Batch Job Item: A batch job item points to a record in the client's database that will be processed by a Background Job. The batch job item should have enough information on it that the process can identify the record to be acted on. A batch job item can have a status of New, Processing, Failed, or Completed.

Background job: When a process may take an excessive amount of time or may take an excessive amount of processing power, we schedule it or add it to a queue to be processed by our background job processing service. Processing typically begins within a matter of seconds unless the system is under heavy load. The creator of the background job does not wait for the job to be processed.

Why batch jobs?

The batch job system provides systems a unified approach for accessing information critical to processing and tracking background jobs. With batch jobs, we allow consumers of the API to monitor the status of work being performed in the background. Be aware that not all jobs performed in the background use the batch job system. Some systems (e.g. Background Checks, Text Messaging) have their own tables for tracking operations performed in the background and are not tracked using the batch jobs.

Batch Job Implementations

Bulk Resume Upload

In Bulk Resume Upload, we run multiple background jobs at once, for uploading resumes, deleting staged employees, and converting staged employees to permanent employees. For every group of resumes that are acted upon, a batch job is created and allows monitoring the status of each item within the group.

Get a list of staged employees and the statuses of their batch job items with the GET /Staging/Employees endpoint.

Upload Resume

1. Call the POST /batchjobs endpoint to create a batch job to track the parsing of your set of resumes. If you want to parse just one resume, you need not use a batch job and can just use POST /Employees/Resume. The following parameters should be supplied when creating a job to process multiple resumes:

```
{
"batchJobType": "Staged_Employee_Import_Resume_Data",
"batchJobParameters": {"skipInterestCodes":false}
}
```

where skipInterestCodes is false if interest codes should be parsed and true if interest codes should not be parsed from the resume.

Response:

```
{
"batchJobId": {int},
"batchJobItems": null
}
```

2. Call POST /Staging/Employees/Resumes with the batchJobId from above, once for each resume to be parsed. Each resume will result in a new batch job item. As soon as a resume has been added to the batch job, it is eligible to be parsed. Parsing is not deferred until all resumes have been uploaded. A Staged Employee record will be created for each resume that is parsed regardless of whether parsing succeeded, failed, or partially succeeded.

Delete Staged Employee

1. Call POST /Staging/Employees/delete with a list of staged employee records to be deleted. This operation cannot be undone. You will receive a response containing the id of the batch job to track the deletion of the records.

Convert Staged Employee to Permanent Employee

1. Call POST /batchjobs with the default parameters to be applied to all new employees created via the batch job:

```
{
"batchJobType": "Delete_Staged_Employees",
"batchJobParameters": {
    "serviceRepId":30256,
    "orderTypeId":3,
    "loggedMessage":"Manual Parse message logged 9/21/2021",
    "employmentCategoryId":"7f5f3059-c9ff-db11-8161-001143dc2454",
    "washedStatusId":127,
    "howHeardOfDetail":"ManualParse How Heard of Detail",
    "howHeardOfId":26,
    "messageActionId":254,
    "employeeStatusId":"AS",
    "branchId":1676,
    "isEmployeeActive":true
    }
}
```

where the batchJobParameters is the default settings for new employees.

Required:

serviceRepId is required and is the id of the service rep to be specified on the new employee records. A list of service reps can be found at GET /Datalists/ServiceReps.

branchId is required and indicates the branch the new employee records should belong to. A list of branches can be found at GET /Datalists/Branches.

isEmployeeActive is required and indicates if the new employees should be active or not upon creation.

Optional:

orderTypeId is the preferred type of job an employee is looking for. A list of order types can be found at GET /Datalists/OrderTypes.

messageActionId is the action to be used when creating the message specified in loggedMessage. If none is provided no message will be logged. A list of message actions can be found at GET Datalists/MessageActions. The message actions available to the caller are controlled via a security group.

loggedMessage, if provided, is the content of a message that will be logged on all new employee records. If a messageActionId is provided but loggedMessage is null, then the message creation will fail.

employmentCategoryId is the preferred type of employment the employee is looking for. A list of employment categories can be found at GET /Datalists/EmploymentCategories.

washedStatusId is the washed status new employees should be created with. A list of washed statuses can be found at GET /Datalists/WashedStatuses.

howHeardOfId indicates how the employees heard about the company. A list of how heard of options can be found at GET /Datalists/HowHeardOf. Please note that not all how heard of options are relevant to employees. To filter the list to just employees supply the value 1 to the originTypeId query parameter.

howHeardOfDetail allows for a free-form response as to how the employees heard about the company. If howHeardOfId (from GET /Datalists/HowHeardOf) has RequireDetails set to true, howHeardOfDetail is required.

employeeStatusId is the status that new records should be created at. If no status is provided, a system default will be used. A list of employee statuses can be found at GET /Datalists/EmployeeStatuses.

Response:

```
{
"batchJobId": {int},
"batchJobItems": null
}
```

2. Call POST /Staging/Employees/{id}/ConvertToEmployee with the batchJobId from above, once for each staged employee to be converted. Each staged employee will result in a new batch job item. As soon as a staged employee

has been added to the batch job, it is eligible to be converted. Converting is not deferred until all resumes have been uploaded.