Job Search documentation (https://cloud.google.com/talent-solution/job-search/)

Documentation

# Batch operations

Cloud Talent Solution supports asynchronous batch operations, which allows you to put multiple API calls into a single HTTP request. The returned response type is `google.longrunning.operation`. The detailed status of the batch operation can be retrieved by calling the `GetOperation` method using the `name` (https://cloud.google.com/talent-solution/matching-engine/docs/reference/rpc/google.longrunning#getoperationrequest) field in `google.longrunning.operation`.

Using batching yields decreased latency in the API response time and higher throughput. Note that successfully receiving a batch API response indicates only that the batch request has been created. Actual data processing is done asynchronously. Batch operations created more than 30 days ago will be no longer accessible by the `GetOperation` method.

**Each batch is limited to 200 requests**.

## Batch operation progress indicator

The `google.longrunning.operation` created by calling the batch create or update method falls under one of the following states:

- STATE_UNSPECIFIED (the default value)
- INITIALIZING
- PROCESSING
- SUCCEEDED
- FAILED

In state INITIALIZING, `success_count`, `failure_count`, and `total_count` fields are set to 0. If the operation is in state PROCESSING the `success_count` and `failure_count` may increase. In state SUCCEEDED, `success_count` + `failure_count` is always equal to `total_count`, and `success_count` > 0. In state FAILED, `success_count` is always 0.

**Note:** Only the `GetOperation` method is currently only supported on the `google.longrunning.operation` resource.

## Batch create jobs

The following code sample demonstrates how to batch create jobs:

JAVA PHP NODE.JS MORE

FEEDBACK (#)

```
public List<Job> batchCreateJobs(List<Job> jobs)
    throws ApiException, ExecutionException, InterruptedException {
  try {
    BatchCreateJobsRequest batchCreateJobsRequest =
        BatchCreateJobsRequest.newBuilder().setParent(FORMATTED_PARENT).addAllJobs

    JobOperationResult jobOperationResult = jobServiceClient
        .batchCreateJobsAsync(batchCreateJobsRequest).get();
    System.out.println("======jobOperationResult======\n" + jobOperationResult);
    List<Job> jobsCreated = new ArrayList<>();
    for (JobResult jobResult : jobOperationResult.getJobResultsList()) {
      jobsCreated.add(jobResult.getJob());
    }
    System.out.println("======Jobs created======\n" + jobsCreated);
    return jobsCreated;
  } catch (ApiException e) {
    System.out.println("Got exception while batch creating jobs");
    throw e;
  }
}
```

# Batch update jobs

The following code sample demonstrates how to batch update jobs:

JAVA PHP NODE.JS MORE

FEEDBACK (#)

```
public List<Job> batchUpdateJobs(List<Job> jobs)
    throws ApiException, ExecutionException, InterruptedException {
  try {
    BatchUpdateJobsRequest batchUpdateJobsRequest =
        BatchUpdateJobsRequest.newBuilder().setParent(FORMATTED_PARENT).addAllJobs

    JobOperationResult jobOperationResult = jobServiceClient
        .batchUpdateJobsAsync(batchUpdateJobsRequest).get();
    List<Job> jobsUpdated = new ArrayList<>();
    for (JobResult jobResult : jobOperationResult.getJobResultsList()) {
      jobsUpdated.add(jobResult.getJob());
    }
    System.out.println("======Jobs updated======\n" + jobsUpdated);
    return jobsUpdated;
  } catch (ApiException e) {
    System.out.println("Got exception while batch updating jobs");
    throw e;
  }
}
```



*Last updated November 1, 2019.*