

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

[Documentation](#)

# Quickstart: Create companies and jobs

This tutorial is designed to get you started exploring and developing applications with Cloud Talent Solution. Familiarity with basic programming is assumed, though even without much programming knowledge, you should be able to follow along. After going through this tutorial, you should be able to use the [Reference](#)

(<https://cloud.google.com/talent-solution/job-search/docs/reference/rest/>) documentation to create your own basic applications.

This tutorial steps through a Cloud Talent Solution application using Java code. The purpose here is not to explain the Java client libraries, but to explain how to make calls to Cloud Talent Solution. Applications in Python and Node.js are essentially similar. Please [contact us](#) (<https://cloud.google.com/talent-solution/docs/support>) if you have any questions.

## Prerequisites

You have:

- followed the steps to create and set up your project, as listed in [Before You Begin](#) (<https://cloud.google.com/talent-solution/job-search/docs/before-you-begin>),
- set up your environment using [Application Default Credentials](#) (<https://developers.google.com/identity/protocols/application-default-credentials>),
- basic familiarity with Java or Python programming,
- installed the latest [Google API Java client](#) (<https://developers.google.com/api-client-library/java/google-api-java-client/setup>) or [Google API Python client](#) ([https://developers.google.com/api-client-library/python/start/get\\_started](https://developers.google.com/api-client-library/python/start/get_started)).

## Create a Company with Jobs and search for those Jobs

This tutorial walks you through a basic Cloud Talent Solution application, and guides you through creating a single Job resource that you associate to a Company resource. The next tutorial walks you through the steps to search for Jobs within the Company based on attributes of the Jobs and the search query. The `search` API attempts to return jobs most relevant to a job seeker's query based on available fields within a Job (company name, job title, job description, job categories, job location, and so on).

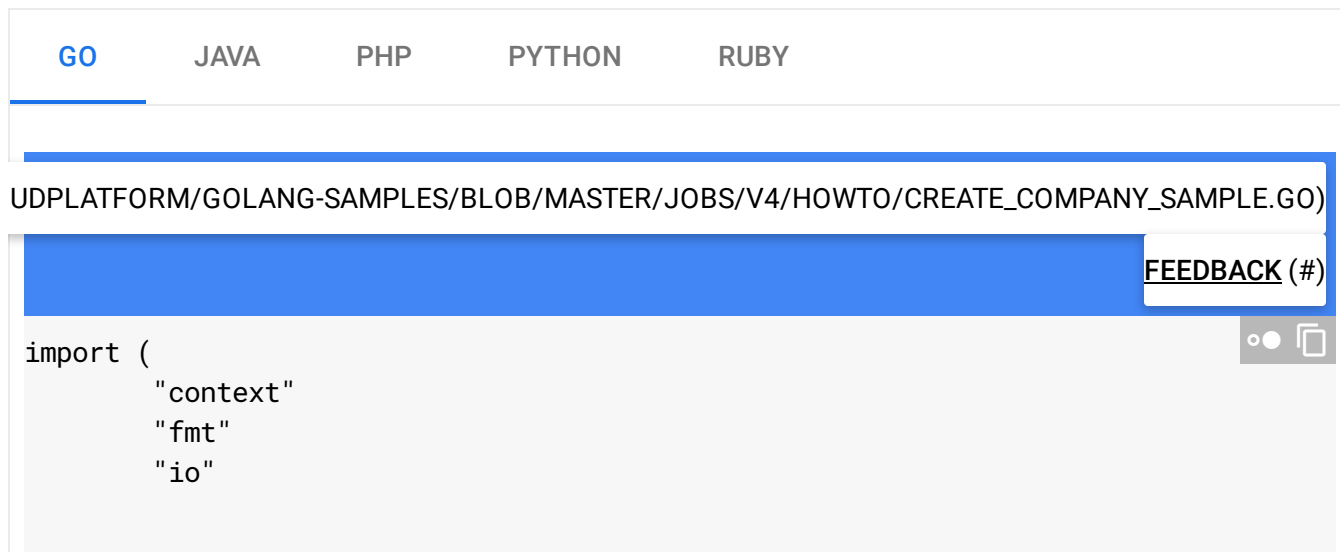
**Note:** Copy the code to your clipboard by mousing over the copy icon in the code snippet, and clicking. Copy this code into a file within your development directory.

### Create a Company

#### A Company

(<https://cloud.google.com/talent-solution/job-search/docs/reference/rest/v4beta1/projects.companies>) is an entity associated with a set of job listings. You must create the Company object before you can post Jobs on Cloud Talent Solution for that Company. You can send any freeform string as the `externalId` when creating the Company. This means you can use the primary key from your existing database (if you have one) when creating and referring to a Company.

**Note:** Cloud Talent Solution creation calls return a unique name for each Company/Job. Store these names as they are used to update/delete/reference Companies/Jobs.



```
UDPLATFORM/GOLANG-SAMPLES/BLOB/MASTER/JOBS/V4/HOWTO/CREATE_COMPANY_SAMPLE.GO
```

```
import (  
    "context"  
    "fmt"  
    "io"
```

```

    talent "cloud.google.com/go/talent/apiv4beta1"
    talentpb "google.golang.org/genproto/googleapis/cloud/talent/v4beta1"
)

// createCompany creates a company as given.
func createCompany(w io.Writer, projectID, externalID, displayName string) (*talentpb.CreateCompanyResponse, error) {
    ctx := context.Background()

    // Initializes a companyService client.
    c, err := talent.NewCompanyClient(ctx)
    if err != nil {
        return nil, fmt.Errorf("talent.NewCompanyClient: %v", err)
    }

    // Construct a createCompany request.
    req := &talentpb.CreateCompanyRequest{
        Parent: fmt.Sprintf("projects/%s", projectID),
        Company: &talentpb.Company{
            ExternalId: externalID,
            DisplayName: displayName,
        },
    }

    resp, err := c.CreateCompany(ctx, req)
    if err != nil {
        return nil, fmt.Errorf("CreateCompany: %v", err)
    }

    fmt.Fprintf(w, "Created company: %q\n", resp.GetName())

    return resp, nil
}

```

## Create a Job

To post a new Job resource, you must provide all of the required fields about the job along with the `companyName` of the Company you want to associate this Job with. You specified the `companyName` earlier when you created the Company resource.

The data object with the Job data filled in is sent to the Cloud Talent Solution endpoint using a POST request. The `name` field should not be set in the initial request since it is an "output only"

field of the `createJob` API: it's a part of the API response when a new job entity is created by the server. The API endpoint to interact with a Jobs resource is specified in the [Cloud Talent Solution Client Libraries document](https://cloud.google.com/talent-solution/job-search/docs/libraries) (<https://cloud.google.com/talent-solution/job-search/docs/libraries>)

The response to the request is a new Job object. It should include a Job name which uniquely represents the posting. The Job name is used when the posting needs to be updated or deleted. As a best practice, store this name and map it to your own unique ID for the Job.

The server returns an error if you attempt to create a new Job when another Job already exists in the system with the same `companyName`, `requisitionId` and `languageCode`.

Cloud Talent Solution also allows you to create Jobs that are specific to a location. See [Locations](https://cloud.google.com/talent-solution/job-search/docs/reference/rest/v4beta1/projects.jobs#Job.FIELDS.locations)

(<https://cloud.google.com/talent-solution/job-search/docs/reference/rest/v4beta1/projects.jobs#Job.FIELDS.locations>) for more information.

The following code creates a Job with just the required fields for the Company specified in the `companyName` field.

You can create a Job using the code sample below. See [Quickstart: Create companies and jobs](https://cloud.google.com/talent-solution/job-search/docs/quickstart-jobs-and-companies#create_a_job_in_your_company) ([https://cloud.google.com/talent-solution/job-search/docs/quickstart-jobs-and-companies#create\\_a\\_job\\_in\\_your\\_company](https://cloud.google.com/talent-solution/job-search/docs/quickstart-jobs-and-companies#create_a_job_in_your_company)) for more details.

[GO](#)[JAVA](#)[PHP](#)[PYTHON](#)[RUBY](#)

For more on installing and creating a Cloud Talent Solution client, see [Cloud Talent Solution Client Libraries](https://cloud.google.com/talent-solution/job-search/docs/libraries) (<https://cloud.google.com/talent-solution/job-search/docs/libraries>).

[LOB/5FF9596979C3431FF123D7CB91712BC357E380AB/JOBS/V4/HOWTO/CREATE\\_JOB\\_SAMPLE.GO](https://cloud.google.com/talent-solution/job-search/docs/quickstart-jobs-and-companies#create_a_job_in_your_company)

[FEEDBACK \(#\)](#)

```
import (  
    "context"  
    "fmt"  
    "io"  
  
    talent "cloud.google.com/go/talent/apiv4beta1"  
    talentpb "google.golang.org/genproto/googleapis/cloud/talent/v4beta1"
```

```
)

// createJob create a job as given.
func createJob(w io.Writer, projectID, companyID, requisitionID, title, URI, descr
    ctx := context.Background()

    // Initialize a jobService client.
    c, err := talent.NewJobClient(ctx)
    if err != nil {
        fmt.Printf("talent.NewJobClient: %v\n", err)
        return nil, err
    }

    jobToCreate := &talentpb.Job{
        CompanyName:  fmt.Sprintf("projects/%s/companies/%s", projectID,
        RequisitionId: requisitionID,
        Title:         title,
        ApplicationInfo: &talentpb.Job_ApplicationInfo{
            Uris: []string{URI},
        },
        Description:  description,
        Addresses:    []string{address1, address2},
        LanguageCode: languageCode,
    }

    // Construct a createJob request.
    req := &talentpb.CreateJobRequest{
        Parent: fmt.Sprintf("projects/%s", projectID),
        Job:     jobToCreate,
    }

    resp, err := c.CreateJob(ctx, req)
    if err != nil {
        fmt.Printf("Failed to create job: %v\n", err)
        return nil, err
    }

    fmt.Printf("Created job: %q\n", resp.GetName())

    return resp, nil
}
```

## Create a Job with custom fields

Cloud Talent Solution includes several job fields that are built in to the API schema. However, you may need additional fields that aren't present in the out-of-the-box options. While we recommended that you use the out-of-the box fields wherever possible, Cloud Talent Solution also provides some `customAttributes` fields for a Job. These can be filterable or non-filterable.

Refer to the [customAttributes](https://cloud.google.com/talent-solution/job-search/docs/reference/rest/v4beta1/projects.jobs#Job.FIELDS.custom_attributes)

([https://cloud.google.com/talent-solution/job-search/docs/reference/rest/v4beta1/projects.jobs#Job.FIELDS.custom\\_attributes](https://cloud.google.com/talent-solution/job-search/docs/reference/rest/v4beta1/projects.jobs#Job.FIELDS.custom_attributes)) documentation for more information.

The following code example shows how to create a Job with a `customAttribute`:

GOJAVAPHPPYTHONRUBY

For more on installing and creating a Cloud Talent Solution client, see [Cloud Talent Solution Client Libraries](https://cloud.google.com/talent-solution/job-search/docs/libraries) (<https://cloud.google.com/talent-solution/job-search/docs/libraries>).

DPLATFORM/GOLANG-SAMPLES/BLOB/MASTER/JOBS/V4/HOWTO/CUSTOM\_ATTRIBUTE\_SAMPLE.GOFEEDBACK (#)

```
import (  
    "context"  
    "fmt"  
    "io"  
  
    talent "cloud.google.com/go/talent/apiv4beta1"  
    "github.com/gofrs/uuid"  
    talentpb "google.golang.org/genproto/googleapis/cloud/talent/v4beta1"  
    money "google.golang.org/genproto/googleapis/type/money"  
)  
  
// createJobWithCustomAttributes creates a job with custom attributes.  
func createJobWithCustomAttributes(w io.Writer, projectID, companyID, jobTitle str  
    ctx := context.Background()  
  
    // Initialize a job service client.  
    c, err := talent.NewJobClient(ctx)  
    if err != nil {  
        return nil, fmt.Errorf("talent.NewJobClient: %v", err)  
    }  
}
```

```

// requisitionID should be the unique ID in your system
requisitionID := fmt.Sprintf("job-with-custom-attribute-%s", uuid.Must(uuid.NewRandom()))
jobToCreate := &talentpb.Job{
    Company:      fmt.Sprintf("projects/%s/companies/%s", projectID,
    RequisitionId: requisitionID,
    Title:        jobTitle,
    ApplicationInfo: &talentpb.Job_ApplicationInfo{
        Uris: []string{"https://googlesample.com/career"},
    },
    Description:   "Design, develop, test, deploy, maintain and improve",
    LanguageCode: "en-US",
    PromotionValue: 2,
    EmploymentTypes: []talentpb.EmploymentType{talentpb.EmploymentType_FULL_TIME},
    Addresses:     []string{"Mountain View, CA"},
    CustomAttributes: map[string]*talentpb.CustomAttribute{
        "someFieldString": {
            Filterable: true,
            StringValues: []string{"someStrVal"},
        },
        "someFieldLong": {
            Filterable: true,
            LongValues: []int64{900},
        },
    },
    CompensationInfo: &talentpb.CompensationInfo{
        Entries: []*talentpb.CompensationInfo_CompensationEntry{
            {
                Type: talentpb.CompensationInfo_BASE,
                Unit: talentpb.CompensationInfo_HOURLY,
                CompensationAmount: &talentpb.CompensationAmount{
                    Amount: &money.Money{
                        CurrencyCode: "USD",
                        Units: 1,
                    },
                },
            },
        },
    },
}

// Construct a createJob request.
req := &talentpb.CreateJobRequest{
    Parent: fmt.Sprintf("projects/%s", projectID),
    Job:    jobToCreate,
}

```

```
}

resp, err := c.CreateJob(ctx, req)
if err != nil {
    return nil, fmt.Errorf("CreateJob: %v", err)
}

fmt.Fprintf(w, "Created job with custom attributes: %q\n", resp.GetName())
fmt.Fprintf(w, "Custom long field has value: %v\n", resp.GetCustomAttribute())

return resp, nil
}
```

## Retrieve a Job

You can confirm that the Job was created by using the GET operation to GET the details of the Job. Note that it might take a few minutes for the Job to become available, based on the current volume of jobs being created in Cloud Talent Solution.

You can retrieve the details of a previously-inserted Job by sending a GET request to the Cloud Talent Solution API. The URI should include the previously-inserted Job name returned by the original CREATE request as a URL parameter.

The following example uses a GET operation to retrieve the details of a Job with a given name:

GOJAVAPHPPYTHONRUBY

For more on installing and creating a Cloud Talent Solution client, see [Cloud Talent Solution Client Libraries](https://cloud.google.com/talent-solution/job-search/docs/libraries) (<https://cloud.google.com/talent-solution/job-search/docs/libraries>).

S/BLOB/5FF9596979C3431FF123D7CB91712BC357E380AB/JOBS/V4/HOWTO/GET\_JOB\_SAMPLE.GO)FEEDBACK (#)

```
import (
    "context"
    "fmt"
    "io"

    talent "cloud.google.com/go/talent/apiv4beta1"
```



```
talentpb "google.golang.org/genproto/googleapis/cloud/talent/v4beta1"
)

// getJob gets an existing job by its resource name.
func getJob(w io.Writer, projectID, jobID string) (*talentpb.Job, error) {
    ctx := context.Background()

    // Initialize a jobService client.
    c, err := talent.NewJobClient(ctx)
    if err != nil {
        fmt.Printf("talent.NewJobClient: %v\n", err)
        return nil, err
    }

    // Construct a getJob request.
    jobName := fmt.Sprintf("projects/%s/jobs/%s", projectID, jobID)
    req := &talentpb.GetJobRequest{
        // The resource name of the job to retrieve.
        // The format is "projects/{project_id}/jobs/{job_id}".
        Name: jobName,
    }

    resp, err := c.GetJob(ctx, req)
    if err != nil {
        fmt.Printf("Failed to get job %s: %v\n", jobName, err)
        return nil, err
    }

    fmt.Fprintf(w, "Job: %q\n", resp.GetName())
    fmt.Fprintf(w, "Job title: %v\n", resp.GetTitle())

    return resp, err
}
```

## Search for Jobs

You've created your first Company and Job using the Cloud Talent Solution! You're now ready to conduct a [search](https://cloud.google.com/talent-solution/job-search/docs/quickstart-searching) across these Jobs.

## More APIs

For more information regarding the APIs and different configurations available, please see the latest [reference documentation](https://cloud.google.com/talent-solution/job-search/docs/apis) (https://cloud.google.com/talent-solution/job-search/docs/apis).

---

*Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/) (https://creativecommons.org/licenses/by/4.0/), and code samples are licensed under the [Apache 2.0 License](https://www.apache.org/licenses/LICENSE-2.0) (https://www.apache.org/licenses/LICENSE-2.0). For details, see our [Site Policies](https://developers.google.com/terms/site-policies) (https://developers.google.com/terms/site-policies). Java is a registered trademark of Oracle and/or its affiliates.*

*Last updated December 3, 2019.*