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

Documentation

Create and update companies

A Company resource in Cloud Talent Solution represents a single company. Any job belonging to that company refers to a job posting that is created for an open position in that company. It contains information such as the company name and address, as well as internal fields that tie the resource in the Cloud Talent Solution service back to your internal databases.

Create a company

To create a company, send a POST request to the `companies` endpoint, specifying at least the two required fields. Refer to the Quickstart: Create companies and jobs (https://cloud.google.com/talent-solution/job-search/docs/quickstart-jobs-and-companies) page for details on how to create a company.

GO JAVA PHP PYTHON RUBY

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

FEEDBACK (#)

```
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) (*talen
        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
}
```

Required fields

The following fields are required in create/update requests:

- `displayName` (https://cloud.google.com/talent-solution/job-search/docs/reference/rest/v4beta1/projects.companies#Company.FIELDS.display_name) : The name of the employer displayed with the job, for example, "Google LLC".
- `externalId` (https://cloud.google.com/talent-solution/job-search/docs/reference/rest/v4beta1/projects.companies#Company.FIELDS.external_id)

: Your internal ID for this company. This field allows you to map your internal identifiers to the company in Google's system. If the company doesn't have a separate internal identifier, set this field to the same value as `displayName`.

Commonly used fields

- `headquartersAddress` (https://cloud.google.com/talent-solution/job-search/docs/reference/rest/v4beta1/projects.companies#Company.FIELDS.headquarters_address)
  : The street address of the company's main headquarters, which may be different from the job location. Cloud Talent Solution only accepts a single HQ location per company. The service attempts to geolocate the address, and populates a more specific location when possible in `derivedInfo.headquartersLocation` (which is output only).
- `size`: A bucket value representing the size of the company in terms of number of employees, from `MINI` to `GIANT`. See the `size` (https://cloud.google.com/talent-solution/job-search/docs/reference/rest/v4beta1/projects.companies#CompanySize) reference for enums and their definitions.
- `eeoText`: A string containing the Equal Employment Opportunity legal disclaimer text to be associated with all jobs for this company.
- `keywordSearchableJobCustomAttributes` (https://cloud.google.com/talent-solution/job-search/docs/reference/rest/v4beta1/projects.companies#Company.FIELDS.keyword_searchable_job_custom_attributes)
  : Specifies which `customAttributes` on this company's jobs should be indexed and searchable by general keyword search.

Confidential companies

In cases where you want to post a confidential job, we recommend creating a separate company that mimics the company's fields, but obfuscates the `displayName`, `externalId`, and any other identifying fields.

In cases where the end employer should be anonymous (for example, the Staffing Agency use case), we recommend setting the `externalId` and `displayName` to random but unique values.

Retrieve a company

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).

CLOUDPLATFORM/GOLANG-SAMPLES/BLOB/MASTER/JOBS/V4/HOWTO/GET_COMPANY_SAMPLE.GO)

FEEDBACK (#)

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

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

// getCompany gets an existing company by its resource name.
func getCompany(w io.Writer, projectID, companyID string) (*talentpb.Company, erro
        ctx := context.Background()

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

        // Construct a getCompany request.
        companyName := fmt.Sprintf("projects/%s/companies/%s", projectID, companyI
        req := &talentpb.GetCompanyRequest{
                // The resource name of the company to be retrieved.
                // The format is "projects/{project_id}/companies/{company_id}".
                Name: companyName,
        }

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

```
        fmt.Fprintf(w, "Company: %q\n", resp.GetName())
        fmt.Fprintf(w, "Company display name: %q\n", resp.GetDisplayName())

        return resp, nil
}
```

List companies

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).

CLOUDPLATFORM/GOLANG-SAMPLES/BLOB/MASTER/JOBS/V4/HOWTO/LIST_COMPANY_SAMPLE.GO)

FEEDBACK (#)

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

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

// listCompanies lists all companies in the project.
func listCompanies(w io.Writer, projectID string) error {
        ctx := context.Background()

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

        // Construct a listCompanies request.
        req := &talentpb.ListCompaniesRequest{
                Parent: "projects/" + projectID,
        }
```

```
        it := c.ListCompanies(ctx, req)

        for {
                resp, err := it.Next()
                if err == iterator.Done {
                        return nil
                }
                if err != nil {
                        return fmt.Errorf("it.Next: %v", err)
                }
                fmt.Fprintf(w, "Listing company: %q\n", resp.GetName())
                fmt.Fprintf(w, "Display name: %v\n", resp.GetDisplayName())
                fmt.Fprintf(w, "External ID: %v\n", resp.GetExternalId())
        }
}
```

Delete a company

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).

UDPLATFORM/GOLANG-SAMPLES/BLOB/MASTER/JOBS/V4/HOWTO/DELETE_COMPANY_SAMPLE.GO)

FEEDBACK (#)

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

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

// deleteCompany deletes an existing company. Companies with
// existing jobs cannot be deleted until those jobs have been deleted.
func deleteCompany(w io.Writer, projectID, companyID string) error {
```

```
        ctx := context.Background()

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

        // Construct a deleteCompany request.
        companyName := fmt.Sprintf("projects/%s/companies/%s", projectID, companyI
        req := &talentpb.DeleteCompanyRequest{
                // The resource name of the company to be deleted.
                // The format is "projects/{project_id}/companies/{company_id}".
                Name: companyName,
        }

        if err := c.DeleteCompany(ctx, req); err != nil {
                return fmt.Errorf("DeleteCompany(%s): %v", companyName, err)
        }

        fmt.Fprintf(w, "Deleted company: %q\n", companyName)

        return nil
}

// [END job_search_delete_company
```


Last updated December 3, 2019.