

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

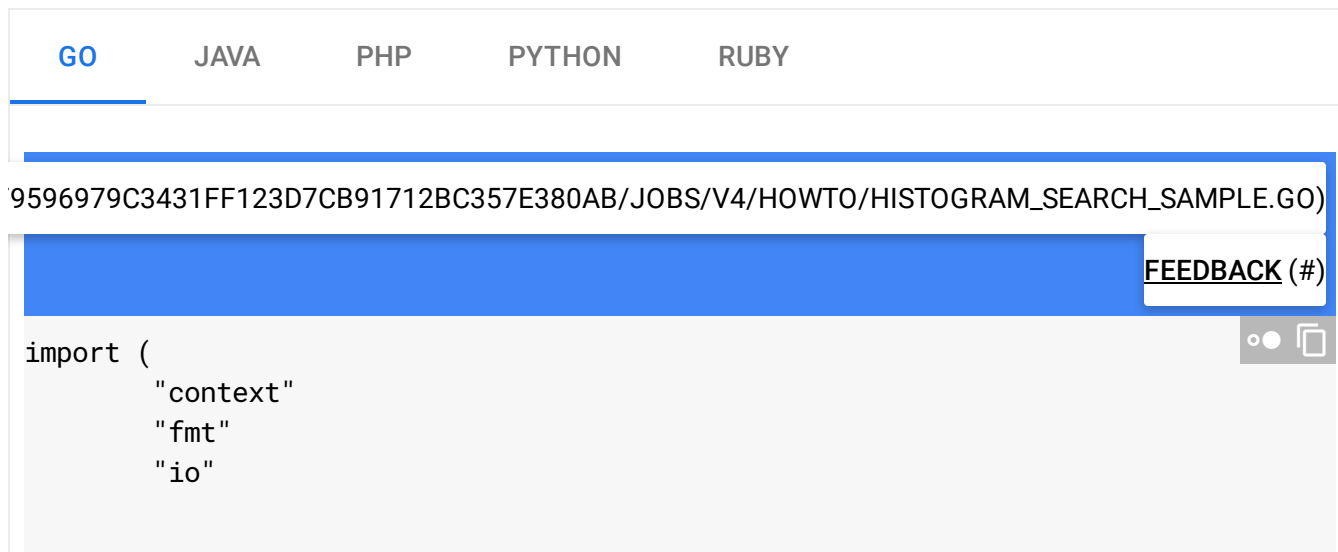
[Documentation](#)

Quickstart: Search

Cloud Talent Solution allows you to search for any currently active Jobs that you have posted to the Cloud Talent Solution system. Once you are able to make [search requests](https://cloud.google.com/talent-solution/job-search/docs/reference/rest/v4beta1/projects.jobs/search) (https://cloud.google.com/talent-solution/job-search/docs/reference/rest/v4beta1/projects.jobs/search), be sure to implement the Jobs analytics framework and implement [client events](https://cloud.google.com/talent-solution/job-search/docs/events) (https://cloud.google.com/talent-solution/job-search/docs/events). This is core to the value of Cloud Talent Solution and directly impacts improved search results.

Get a histogram of the number of Jobs by a certain field

The search API can return a histogram of all relevant Jobs indexed in Cloud Talent Solution that match the filters of the search query. Now that you've gotten the search results, you can get a count of the jobs by different facets. Leverage the histogram API to get the count of jobs by certain facets.



```

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

// histogramSearch searches for jobs with histogram queries.
func histogramSearch(w io.Writer, projectID, companyID string) error {
    ctx := context.Background()

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

    // Construct a searchJobs request.
    req := &talentpb.SearchJobsRequest{
        Parent: fmt.Sprintf("projects/%s", projectID),
        // Make sure to set the RequestMetadata the same as the associated
        // search request.
        RequestMetadata: &talentpb.RequestMetadata{
            // Make sure to hash your userId.
            UserId: "HashedUsrID",
            // Make sure to hash the sessionId.
            SessionId: "HashedSessionID",
            // Domain of the website where the search is conducted.
            Domain: "www.googlesample.com",
        },
        HistogramQueries: []*talentpb.HistogramQuery{
            {
                // More info on histogram facets, constants, and b
                // https://godoc.org/google.golang.org/genproto/go
                HistogramQuery: "count(base_compensation, [bucket(
            },
        },
    }
    if companyID != "" {
        req.JobQuery = &talentpb.JobQuery{
            CompanyNames: []string{fmt.Sprintf("projects/%s/companies/
        }
    }

    it := c.SearchJobs(ctx, req)

```

```
    for {
        resp, err := it.Next()
        if err == iterator.Done {
            return nil
        }
        if err != nil {
            fmt.Printf("it.Next: %v\n", err)
            return err
        }
        fmt.Fprintf(w, "Job: %q\n", resp.Job.GetName())
    }
}
```

In this case, we're getting the count of all jobs in the system, that are accepted by the filters specified, across the various facets - CATEGORY and CITY.

Use the autocomplete API to get suggested job titles

The complete API suggests job titles or company titles that the job seeker might be interested in based on what they've already typed. Use this to autocomplete potential results in the search bar of your UI.

[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>).

LOUDPLATFORM/GOLANG-SAMPLES/BLOB/MASTER/JOBS/V4/HOWTO/AUTO_COMPLETE_SAMPLE.GO)

[FEEDBACK \(#\)](#)

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

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

```
// jobTitleAutoComplete suggests the job titles of the given
// company identifier on query.
func jobTitleAutocomplete(w io.Writer, projectID, query string) (*talentpb.Complet
    ctx := context.Background()

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

    // Construct a completeQuery request.
    req := &talentpb.CompleteQueryRequest{
        Parent:      fmt.Sprintf("projects/%s", projectID),
        Query:        query,
        LanguageCodes: []string{"en-US"},
        PageSize:     5, // Number of completion results returned.
        Scope:        talentpb.CompleteQueryRequest_PUBLIC,
        Type:         talentpb.CompleteQueryRequest_JOB_TITLE,
    }

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

    fmt.Fprintf(w, "Auto complete results:")
    for _, c := range resp.GetCompletionResults() {
        fmt.Fprintf(w, "\t%v\n", c.Suggestion)
    }

    return resp, nil
}
```

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.