



# Job Search Index Design

Search is central to helping candidates discover relevant jobs. A dedicated search index (e.g., Elasticsearch) is used alongside the relational database to provide fast keyword search and filtering.

## Fields to Index

Field	Type	Purpose
<code>id</code>	integer (keyword)	Unique job identifier.
<code>title</code>	text (english analyzer)	Main job title; weighted heavily in search.
<code>description</code>	text (english analyzer)	Full job description.
<code>required_skills</code>	text or keyword	List of required skills. Use <code>text</code> with an analyzer to allow partial matching and synonyms.
<code>company_name</code>	text	Name of the company posting the job.
<code>country</code>	keyword	Primary country of the job; <code>null</code> if remote-only.
<code>eligible_countries</code>	keyword array	ISO country codes allowed for remote roles.
<code>is_remote</code>	boolean	Indicates remote work.
<code>visa_sponsorship</code>	boolean	Indicates visa sponsorship availability.
<code>relocation_assistance</code>	boolean	Indicates relocation support.
<code>category</code>	keyword	Job category/industry (e.g., Engineering, Marketing).
<code>posted_at</code>	date	Index for sorting by recency.

## Example Mapping (Elasticsearch)

```
PUT jobs-index
{
  "settings": {
    "analysis": {
      "analyzer": {
        "english": { "type": "standard", "stopwords": "_english_" }
      }
    }
  }
}
```

```

        },
    },
    "mappings": {
        "properties": {
            "id": { "type": "integer" },
            "title": { "type": "text", "analyzer": "english" },
            "description": { "type": "text", "analyzer": "english" },
            "required_skills": { "type": "text", "analyzer": "english" },
            "company_name": { "type": "text" },
            "country": { "type": "keyword" },
            "eligible_countries": { "type": "keyword" },
            "is_remote": { "type": "boolean" },
            "visa_sponsorship": { "type": "boolean" },
            "relocation_assistance": { "type": "boolean" },
            "category": { "type": "keyword" },
            "posted_at": { "type": "date" }
        }
    }
}

```

## Filterable Fields

- **Country:** filter by `country` for on-site roles or filter remote jobs whose `eligible_countries` contain the selected country.
- **Remote from Africa:** filter by `is_remote=true` and ensure that `eligible_countries` is empty or includes the user's country code.
- **Visa Sponsorship:** filter by `visa_sponsorship=true`.
- **Relocation Assistance:** filter by `relocation_assistance=true`.
- **Category/Industry:** filter by `category`.

## Query Examples

### Keyword Search with Filters

```

GET jobs-index/_search
{
    "query": {
        "bool": {
            "must": [
                { "multi_match": { "query": "software engineer", "fields": ["title^3", "description", "required_skills"] } }
            ],
            "filter": [
                { "term": { "country": "KEN" } },
                { "term": { "is_remote": true } },
                { "term": { "visa_sponsorship": true } },

```

```

        { "term": { "category": "Engineering" } }
    ]
}
},
"from": 0,
"size": 10,
"sort": [ { "posted_at": "desc" } ]
}

```

## Remote Jobs Open to African Countries

```

GET jobs-index/_search
{
  "query": {
    "bool": {
      "filter": [
        { "term": { "is_remote": true } },
        { "bool": {
          "should": [
            { "exists": { "field": "eligible_countries" } },
            { "match_all": {} }
          ]
        }
      }
    ]
  }
}

```

This query returns remote jobs regardless of specific countries. If we store allowed countries for remote jobs, additional filtering can be applied (e.g., `"term": { "eligible_countries": "NGA" }`).

## Sync Strategy

- 1. Indexing on Create/Update:** When a job is created or updated in the database, the backend invokes a `searchService.indexJob(job)` function. This function transforms the job record into the index format (flattening company name and skills) and issues an `index` or `update` request to the search engine. Jobs with `status != 'active'` can either be excluded or stored with a flag for limited visibility.
- 2. Deletion on Close/Reject:** When a job is closed or rejected (status changes to `closed`), the backend calls `searchService.deleteJob(jobId)` to remove it from the index. Alternatively, jobs can be retained with a `status` field for historical analytics but filtered out from candidate searches.

3. **Batch Reindexing:** A scheduled job (e.g., nightly) reindexes all active jobs from the database to handle drift or recovery scenarios.
4. **Monitoring & Quality Metrics:** Log search queries, result counts and click-through rates. Monitor the ratio of zero-result queries; adjust analyzers or synonyms accordingly. Track the distribution of filter usage (e.g., how often candidates filter for visa sponsorship). Periodically review sample search results to ensure relevance.

The search logic is fully explainable because it uses field weights and filters rather than black-box machine learning. All parameters are transparent and can be tuned manually.

---