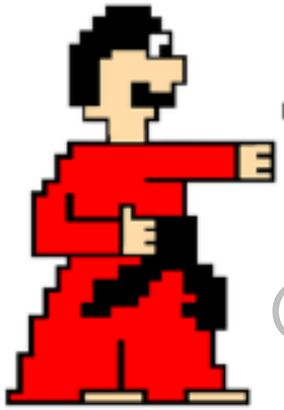


Fuzzfinder Revisited:

Capturing Query Behaviour Using Elasticsearch

CARLY RICHMOND

@CarlyLRichmond

elastic

@CarlyLRichmond

elastic

2000

@CarlyLRichmond

elastic


CAPTCHA
Type the word above
Verify

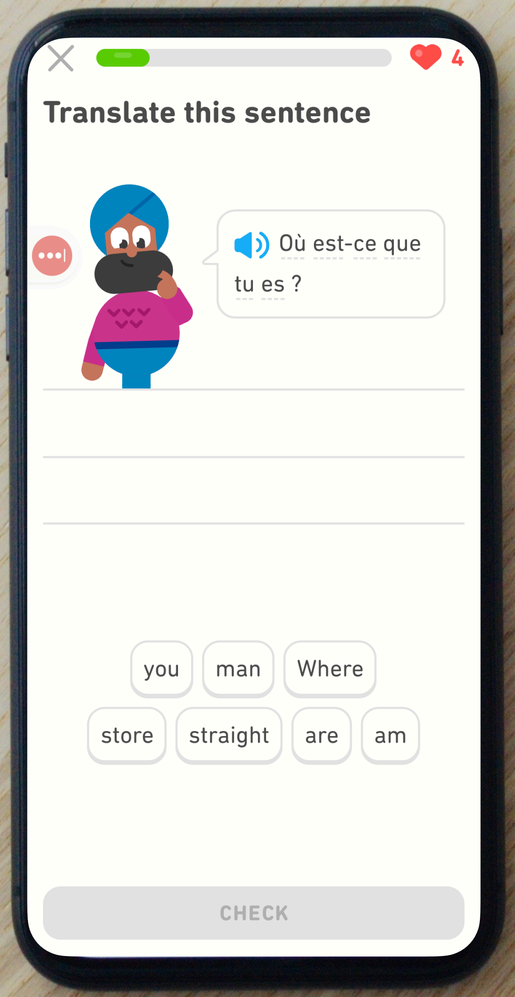
Massive-scale online collaboration | Luis von Ahn | TEDxCMU 2012

GWAP


The ESP Game - Netscape 6
0:11
Time Left
The ESP Game
2100
score
Taboo Words
MAN
BEARD
Your Guesses
HAT
Type your next guess:
Pass
© 2002-2003 Carnegie Mellon University, all rights reserved. Patent Pending.

Game With A Purpose

Game With A Purpose



2011

@CarlyLRichmond

elastic

University of Glasgow | Department of Computing Science

VIA VERITAS VITA

Fu-Finder: Using Human Computation Games to Understand User Querying Behaviour and Quality

Masters Thesis

Carly Louise O'Neil

22nd April 2011

Fu-Finder: A Game for Studying Querying Behaviours

Test your Search-Fu

Carly O'Neil
School of Computing Science
University of Glasgow
United Kingdom
oneilcl@dcs.gla.ac.uk

James Purvis
School of Computing Science
University of Glasgow
United Kingdom
0801303P@student.gla.ac.uk

Leif Azzopardi
School of Computing Science
University of Glasgow
United Kingdom
leif@dcs.gla.ac.uk

ABSTRACT

Usually the focus of evaluation within Information Retrieval has been placed largely upon the system. However, the individual user and their submitted queries are typically the greatest source of variation in the search process. This demonstration paper presents Fu-Finder, a fun and enjoyable game that measures the user's querying abilities (or search-fu). This game provides useful data for the study of user querying behaviour and assesses how well users can find specific web pages using different search engines.

Categories and Subject Descriptors: H.3.3 Information Storage and Retrieval: Information Search and Retrieval

General Terms: Performance, Experimentation

Keywords: Search-Fu, Querying Behaviour, Findability

1. INTRODUCTION

Searching and finding particular webpages can sometimes feel like a game of skill and chance. Choosing the "right" query terms to illicit the desired response is largely dependant upon the user. The user needs to be able to understand their information need, the system that they are using, how it works, what it is in the document (imagined, ideal, actual and/or known) and what terms would help to distinguish it from other documents. Not a particularly easy task. This has led to the term, *search-fu* being coined which denotes the skill of user's search abilities i.e., someone with weak search-fu will be unlikely to find relevant documents, while someone with strong search-fu will be able to find almost anything. So the question is, how much search-fu does a user possess? Or stated more formally, how well can a user satisfy their information needs when using a search engine?

The prototype system we have developed to help answer this question is a human computation game (HCG). HCGs have been developed to obtain data for various purposes (such as image annotation, character recognition, etc) [8]. Recently, the game Page Hunt was developed to help optimise the Bing search engine as a way to help annotate pages that are difficult to find [3]. However, rather than trying to optimise a particular search engine, we have developed Fu-Finder to evaluate the user's ability at querying for pages across a number of search engines. The game is currently hosted and available to play[1], while the code is open source and freely available to download as part of the PuppyIR project housed on SourceForge[2]. In this paper, we describe the game developed and some of the initial results.

2. EVALUATIONS USING GAMES

Human computation games provide a novel tool for solving computationally challenging problems (which are easy for humans, but hard for computers). The success of such games has been their ability to generate large quantities of reliable and useful data by providing users with enjoyment and points as payment rather than money or course credit. The added advantage is that these games can be undertaken in a controlled environment (i.e. within the remit of the game rules/setup) much like a standard or traditional IR experiment, but without the problems of using paid-workers through crowd sourcing. While HCGs are often developed to solve a difficult problem, such as von Ahn's ESP game, where players independently annotate an images [7], in IR human computation games have been used for evaluation purposes.

Two notable games developed in IR are: Book Explorer [2] and PageHunt [3]. In [2], they developed a human computation game to assess relevancy of electronic books to topics. The main motivation for incorporating this task in a game was to provide a means of obtaining relevancy judgements for large document collections without needing to provide fiscal payment to study participants. This has been outlined as a prime motivation for including a computationally-hard task within a game with a purpose [7]. The Book Explorer game, whose model is a variant of the inversion-problem model [8], has two player roles, which can be adopted by individuals or teams. The *explorer* locates and highlights relevant sections of a document; while the *reviewer* provides relevancy feedback on documents [2]. Reviewers can only enter relevancy judgements for documents that have been evaluated by at least one explorer player. Furthermore, reviewers also serve as a cheating-detection mechanism as they examine explorer input, thus creating a multi-level validation system.

While the results from the study appear to show some

CIKM'11, October 24–28, 2011, Glasgow, Scotland, UK.

[1] http://www.dcs.gla.ac.uk/access/fufinder
[2] http://sourceforge.net/projects/puppyir/

elastic

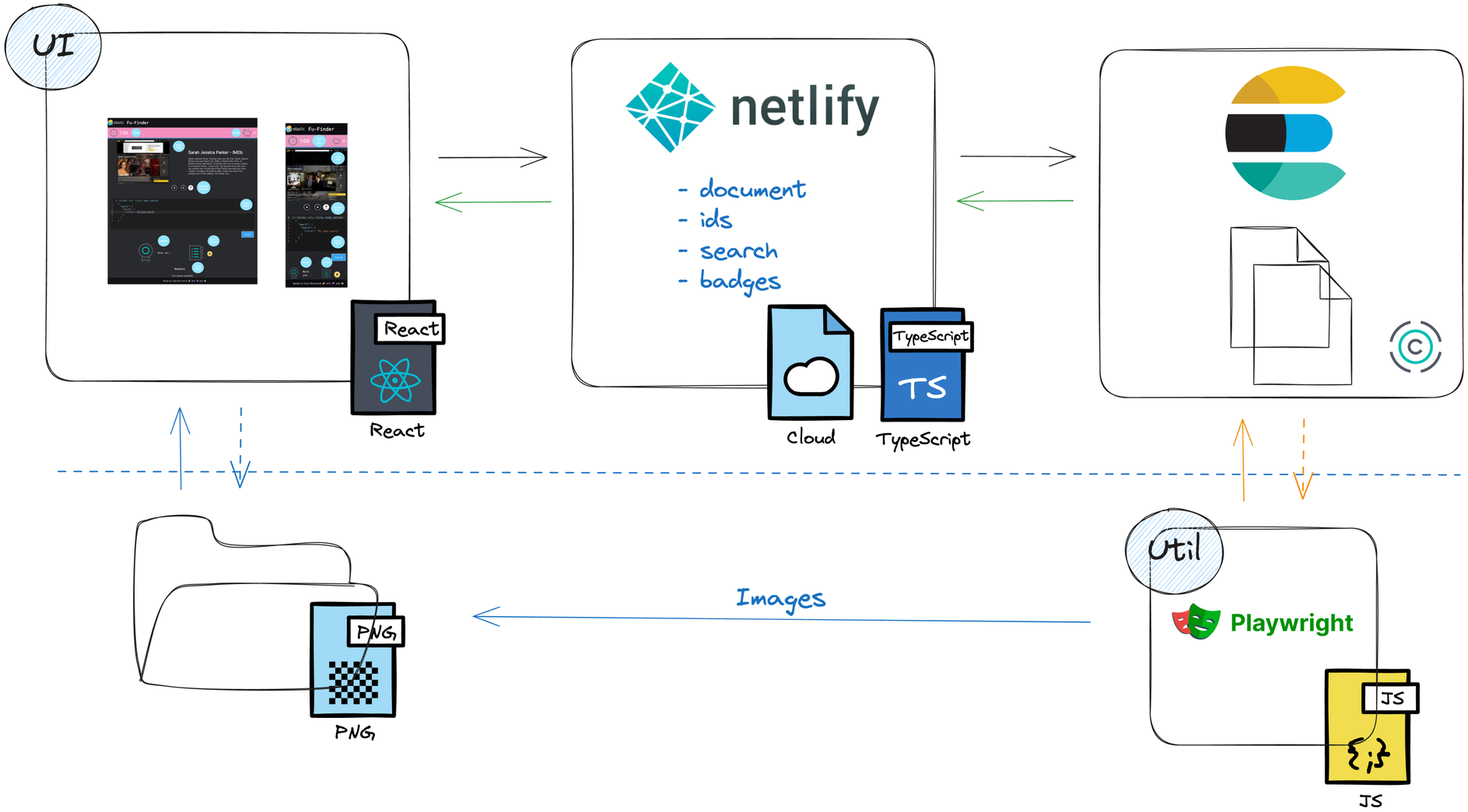
2023

@CarlyLRichmond

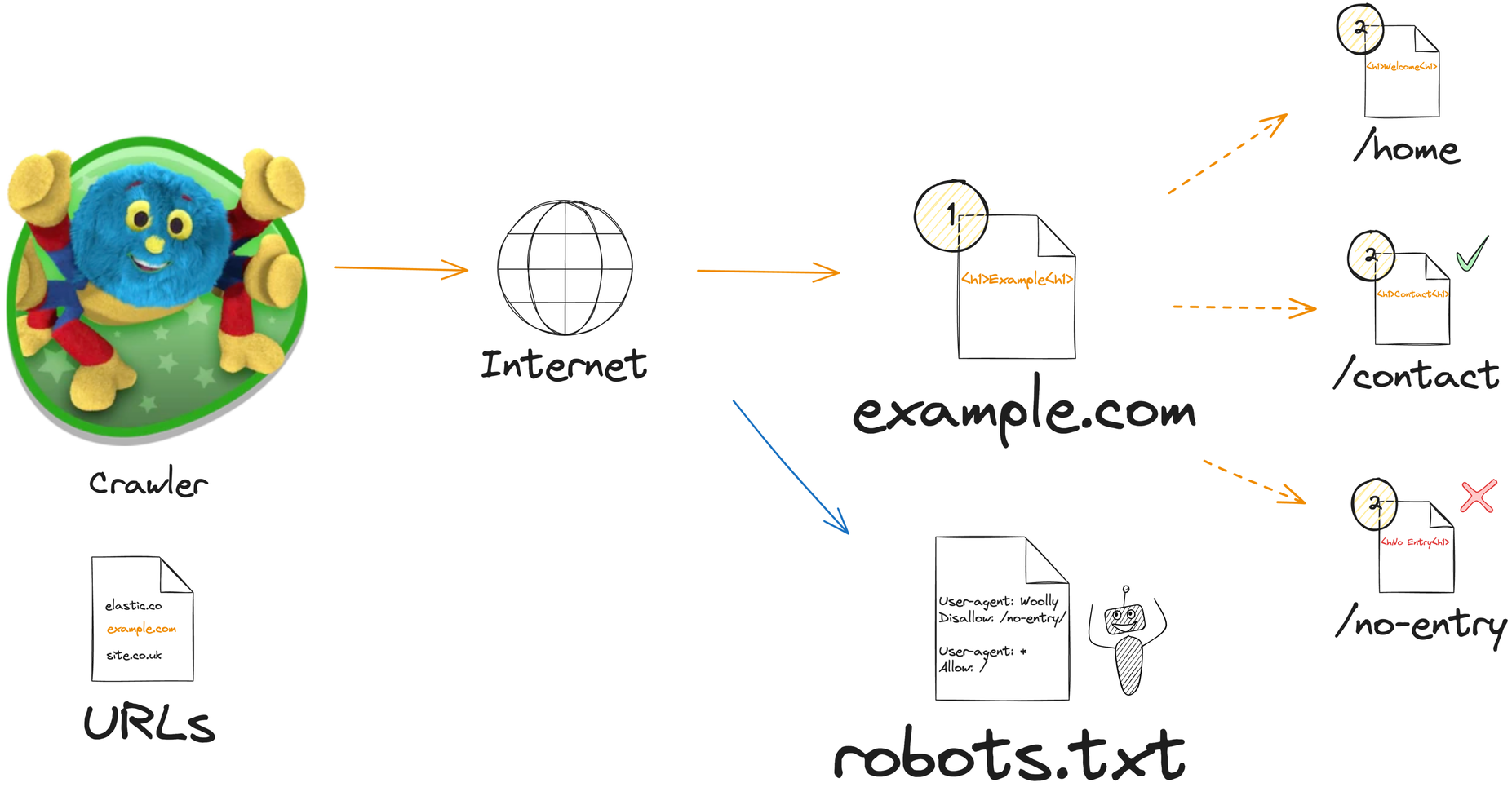
elastic


elastic

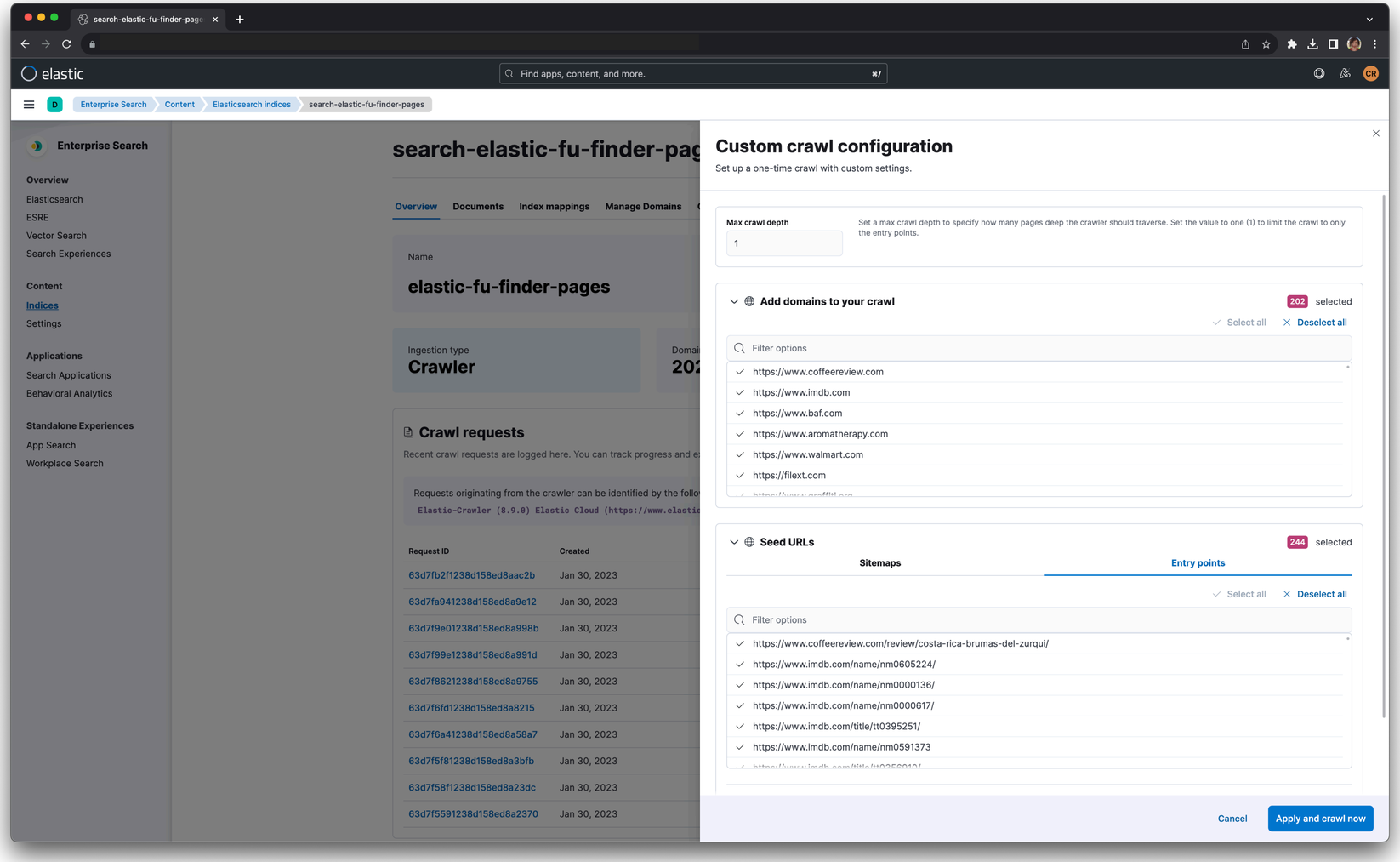
Architecture



Web Crawler


Crawler
elastic.co
example.com
site.co.uk
URLs
Internet
1
<h1>Example</h1>
example.com
2
<h1>Welcome</h1>
/home
2
<h1>Contact</h1>
/contact
2
<h1>No Entry</h1>
/no-entry
User-agent: Woolly
Disallow: /no-entry/
User-agent: *
Allow: /
robots.txt

Elastic Web Crawler



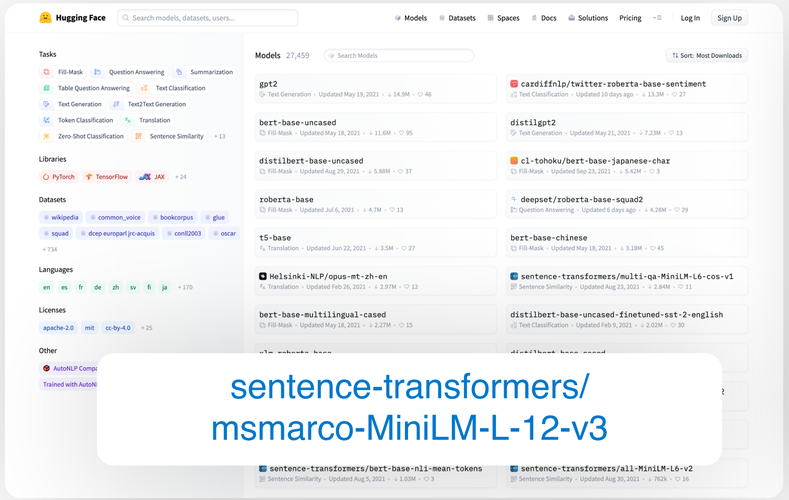
Alt: Storm Crawler | http://stormcrawler.net/

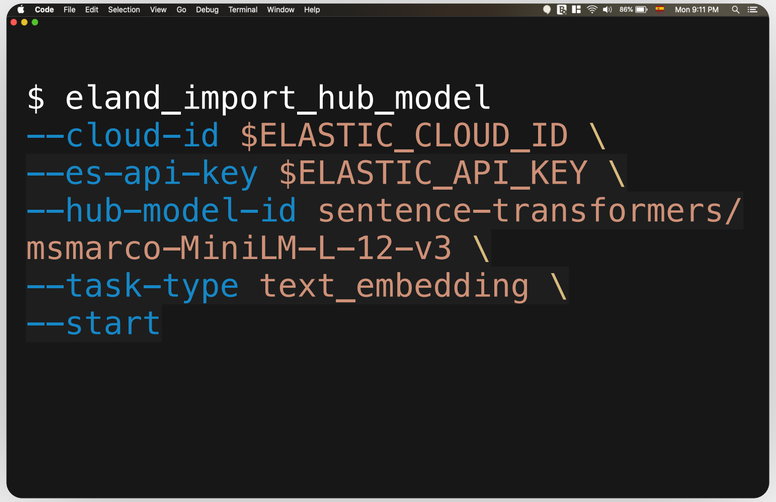
Keyword Document

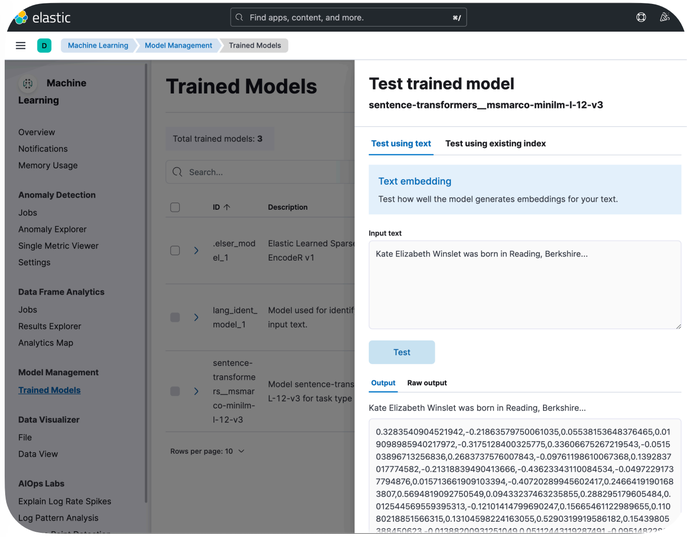
keyword-kate-winslet-imdb.json


```
{
        "_index": "search-elastic-fu-finder-pages",
        "_id": "63d3a0d21238d1dc003b9c79",
        "_score": 4.4962697,
        "_source": {
          "last_crawled_at": "2023-01-27T14:46:47Z",
          "additional_urls": [
            "https://www.imdb.com/name/nm0000701/"
          ],
          "body_content": "Kate Elizabeth Winslet was born in Reading, Berkshire...",
          "domains": [
            "https://www.imdb.com"
          ],
          "title": "Kate Winslet - IMDb",
          "meta_keywords": "Biography, Photos, Awards, News",
          "url": "https://www.imdb.com/name/nm0000701/",
          "url_scheme": "https",
          "url_port": 443,
          "url_host": "www.imdb.com",
          "url_path_dir2": "nm0000701",
          "url_path": "/name/nm0000701/",
          "url_path_dir1": "name"
        }
      }
```

Setting up the Embedding Model







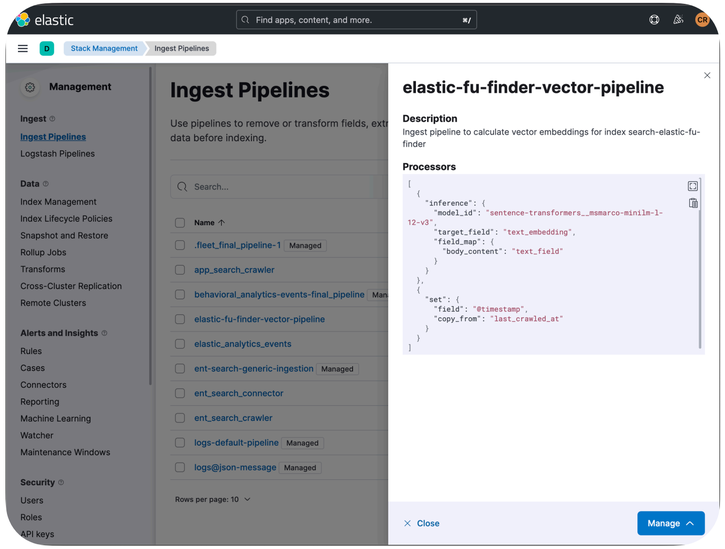
PyTorch → eland → Elastic

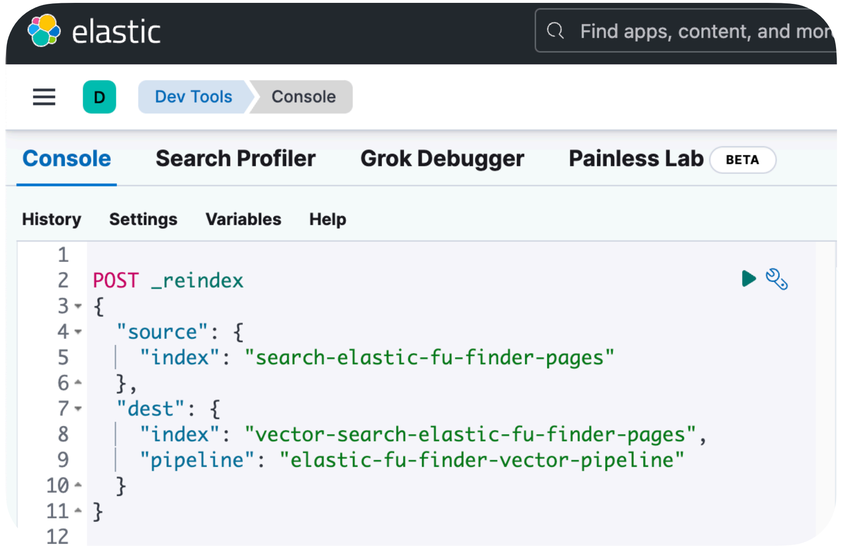
Select the appropriate model

Load the model to the cluster

Manage models

Generate Embeddings





PyTorch

Enrich data using inference processor

Reindex data using ingest pipeline

Generate embeddings

elastic

Vector Document

vector-kate-winslet-imdb.json


```
{
        "_index": "vector-search-elastic-fu-finder-pages",
        "_id": "63d3a0d21238d1dc003b9c79",
        "_score": 4.4962697,
        "_source": {
          "last_crawled_at": "2023-01-27T14:46:47Z",
          "text_embedding": {
            "predicted_value": [
              0.23861193656921387,
              -0.3623441457748413,
            ],
            "is_truncated": true,
            "model_id": "sentence-transformers__msmarco-minilm-l-12-v3"
          },
          "additional_urls": [
            "https://www.imdb.com/name/nm0000701/"
          ],
          "body_content": "Kate Elizabeth Winslet was born in Reading, Berkshire...",
          "domains": [
            "https://www.imdb.com"
          ],
          "title": "Kate Winslet - IMDb",
          "meta_keywords": "Biography, Photos, Awards, News",
          "url": "https://www.imdb.com/name/nm0000701/",
          "url_scheme": "https",
          "url_port": 443,
          "url_host": "www.imdb.com",
          "url_path_dir2": "nm0000701",
          "url_path": "/name/nm0000701/",
          "url_path_dir1": "name"
        }
      }
```

Elasticsearch Client

elasticsearch.ts

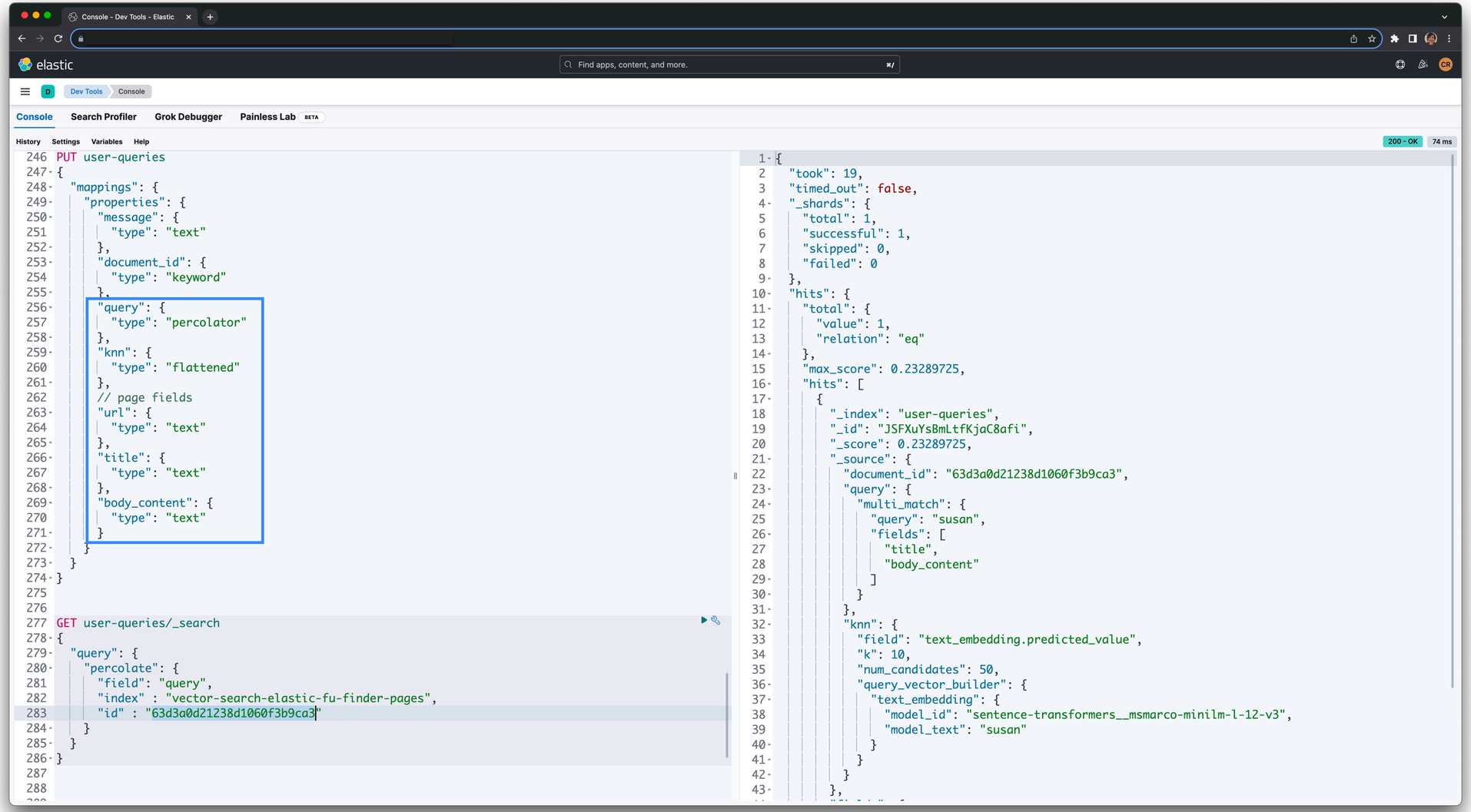
```ts
const vectorSearchIndex = 'vector-search-elastic-fu-finder-pages';

const cloudID = process.env.ELASTIC_CLOUD_ID;
const apiKey = process.env.ELASTIC_API_KEY;

const client = new Client({
    cloud: { id: cloudID },
    auth: { apiKey: apiKey },
});

export async function getSearchResults(query: any) {
  const keyword = query.query;
  const vector = query.knn;

  return client.search({
    index: vectorSearchIndex,
    profile: true,
    query: keyword,
    knn: vector
  });
}
```

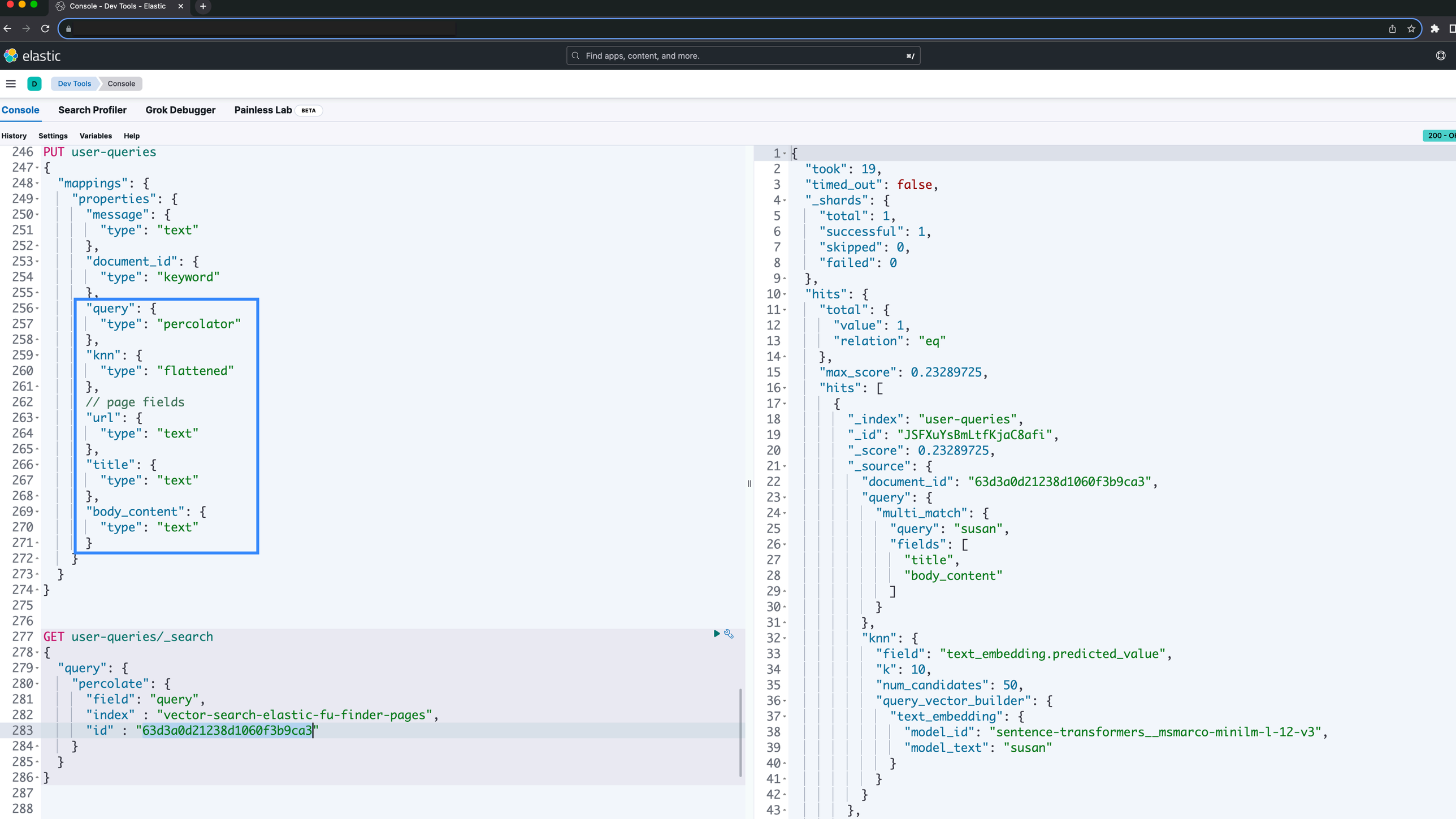
Hits Processing

get-results.ts

```
async function getResults(newQuery: string) {
  setShowSpinner(true);

    try {
      const response = await axios.post('.netlify/functions/search', { queryString: newQuery });
      const results = response.data.hits?.hits;
      const queryTypes = response.data.profile?.shards[0].searches.flatMap((search: { query: any[]; }) => {
          return search.query.map((element: { type: any; }) => {
            return element.type;
          })
        });

      // Check for match and calculate score here

       }
    }
    catch(error) {
      console.log('Unable to get results');
    }
    finally {
        setShowSpinner(false);
      }
  }
```

Percolators





Console - Dev Tools - Elastic

elastic

Find apps, content, and more. ⌘/

D Dev Tools Console

Console Search Profiler Grok Debugger Painless Lab BETA

History Settings Variables Help

200 - OK

```
PUT user-queries
{
  "mappings": {
    "properties": {
      "message": {
        "type": "text"
      },
      "document_id": {
        "type": "keyword"
      },
      "query": {
        "type": "percolator"
      },
      "knn": {
        "type": "flattened"
      },
      // page fields
      "url": {
        "type": "text"
      },
      "title": {
        "type": "text"
      },
      "body_content": {
        "type": "text"
      }
    }
  }
}


GET user-queries/_search
{
  "query": {
    "percolate": {
      "field": "query",
      "index" : "vector-search-elastic-fu-finder-pages",
      "id" : "63d3a0d21238d1060f3b9ca3"
    }
  }
}
```

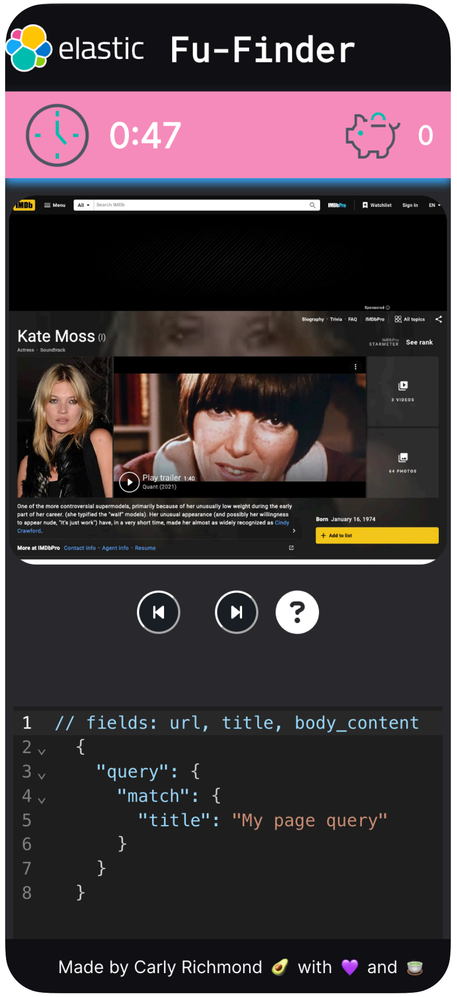
```
{
  "took": 19,
  "timed_out": false,
  "_shards": {
    "total": 1,
    "successful": 1,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": {
      "value": 1,
      "relation": "eq"
    },
    "max_score": 0.23289725,
    "hits": [
      {
        "_index": "user-queries",
        "_id": "JSFXuYsBmLtfKjaC8afi",
        "_score": 0.23289725,
        "_source": {
          "document_id": "63d3a0d21238d1060f3b9ca3",
          "query": {
            "multi_match": {
              "query": "susan",
              "fields": [
                "title",
                "body_content"
              ]
            }
          },
          "knn": {
            "field": "text_embedding.predicted_value",
            "k": 10,
            "num_candidates": 50,
            "query_vector_builder": {
              "text_embedding": {
                "model_id": "sentence-transformers__msmarco-minilm-l-12-v3",
                "model_text": "susan"
              }
            }
          }
        },
      },
```

Elasticsearch Client

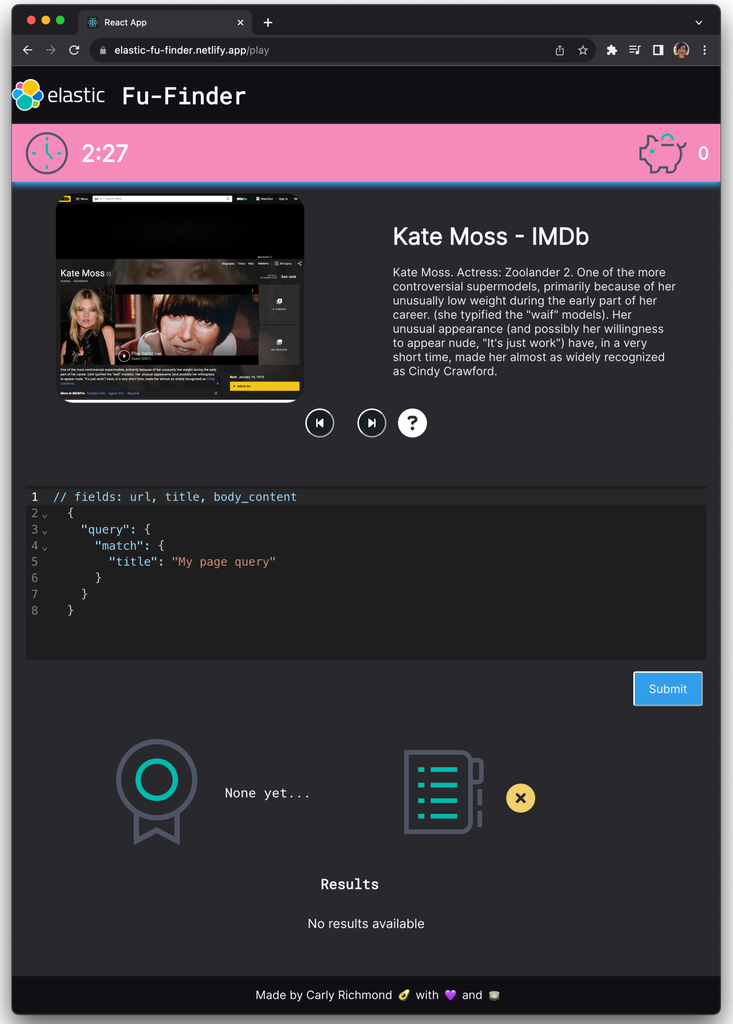
index-query-elasticsearch.ts

```
const userQueryIndex = 'user-queries';

const cloudID = process.env.ELASTIC_CLOUD_ID;
const apiKey = process.env.ELASTIC_API_KEY;

const client = new Client({
    cloud: { id: cloudID },
    auth: { apiKey: apiKey },
});

export async function indexGameSearch(documentID: string, searchTerms: any) {
  let document = {
    document_id: documentID,
    query: searchTerms.query,
    knn: searchTerms.knn
  };

  return client.index({
    index: userQueryIndex,
    document: document
  });
}
```

Let's Play!



SCAN ME



elastic

DANKE!
THANK YOU!
MERCI!
GRAZIE!
GRACIAS!
DANK JE WEL!

• • • • • • • • • •