

Teste técnico: Backend Software Developer

Objetivo: desenvolver uma aplicação Java para navegar por um website em busca de um termo fornecido pelo usuário e listar as URLs onde o termo foi encontrado.

Requisitos

1. A interação do usuário com a aplicação deve acontecer por meio de uma API HTTP, a ser disponibilizada na porta 4567.

Duas operações devem ser suportadas:

- a. POST: inicia uma nova busca por um termo (*keyword*).

Requisição:

```
POST /crawl HTTP/1.1
Host: localhost:4567
Content-Type: application/json
Body: {"keyword": "security"}
```

Resposta:

```
200 OK
Content-Type: application/json
Body: {"id": "30vblllyb"}
```

- b. GET: consulta resultados de busca.

Requisição:

```
GET /crawl/30vblllyb HTTP/1.1
Host: localhost:4567
```

Resposta:

```
200 OK
Content-Type: application/json
{
  "id": "30vblllyb",
  "status": "active",
  "urls": [
    "http://hiring.axreng.com/index2.html",
    "http://hiring.axreng.com/htmlman1/chcon.1.html"
  ]
}
```

2. O termo buscado deve ter no mínimo 4 e no máximo 32 caracteres. A busca deve ser *case insensitive*, em qualquer parte do conteúdo HTML (incluindo *tags* e comentários)..
3. O *id* da busca deve ser um código alfanumérico de 8 caracteres gerado automaticamente.
4. A URL base do website em que as análises são realizadas é determinada por uma variável de ambiente. As buscas devem seguir links (absolutos e relativos) em elementos *anchor* das páginas visitadas **se e somente se** eles possuírem a mesma URL base.

5. A aplicação deve suportar a execução de múltiplas buscas simultâneas. Informações sobre buscas em andamento (status *active*) ou já concluídas (status *done*) devem ser mantidas indefinidamente enquanto a aplicação estiver em execução.
6. Enquanto uma busca está em andamento, seus resultados parciais já encontrados devem ser retornados pela operação GET.
7. O projeto deve seguir a estrutura base fornecida. Os arquivos `Dockerfile` e `pom.xml` **não podem ser modificados**. Qualquer outro arquivo fornecido pode ser modificado.
8. A partir do diretório raiz do projeto, os seguintes dois comandos, executados em sequência, devem fazer a compilação e inicialização da aplicação:

```
docker build . -t axreng/backend
docker run
-e BASE_URL=http://hiring.axreng.com/
-p 4567:4567 --rm axreng/backend
```

9. O código fonte da solução deve ser entregue em um arquivo tar (.tar ou .tar.gz). Caso o código seja disponibilizado publicamente, solicitamos gentilmente que quaisquer referências à Axur (incluindo em nomes de pacotes e *hosts*) sejam removidas de todos os arquivos antes da publicação.

Avaliação

Testes automatizados serão aplicados pela equipe de avaliação da Axur, de forma que **soluções que não seguirem à risca os requisitos serão desclassificadas**.

Além do atendimento dos requisitos, a qualidade interna da solução será avaliada rigorosamente, de acordo com padrões e práticas amplamente disseminados na literatura. O desempenho da solução em relação a tempo de processamento e uso de memória também será objeto de avaliação.

Auxílio ao desenvolvimento

Disponibilizamos como recursos que podem auxiliar o desenvolvimento da solução um website simples e a API HTTP de uma aplicação funcional. Ela está configurada para fazer buscas no website fornecido (com limite de 100 resultados).

Website: <http://hiring.axreng.com/>

Aplicação: <http://testapp.axreng.com:4567/>