

Instruções para teste das funções de busca

Pedro Aurélio Coelho de Almeida

14/0158103

Linguagens de Programação

27 de Maio de 2017

1 Funcionalidades presentes

As funcionalidades implementadas podem ser resumidas em:

- Visualizar a lista com o nome das empresas cadastradas na base de dados.
- Visualizar a lista com o nome das funções a serem desempenhadas nas empresas cadastradas na base de dados.
- Visualizar a lista com os cursos presentes na base de dados.
- Visualizar a lista de nome das empresas com os cursos exigidos para estagiar que são compatíveis com o do aluno que está logado.
- Saber se uma dada empresa está cadastrada na base de dados, a partir de uma busca por seu nome.
- Obter a lista de funções nas empresas cujos cursos necessários para estagiar são compatíveis com o curso do aluno logado.
- Obter as funções a serem desempenhadas nas empresas de acordo com a algum curso presente na base de dados.
- Obter as funções a serem desempenhadas na empresa de acordo com o nome da empresa presente na base de dados.
- Obter as funções a serem desempenhadas na empresa de acordo com a data de criação delas (do mês inicial fornecido até a data atual).
- Obter as funções a serem desempenhadas na empresa de acordo com a data de criação delas (do mês final fornecido para datas anteriores).
- Obter as funções a serem desempenhadas na empresa de acordo com a data de criação delas (entre duas datas especificadas).
- Obter as funções a serem desempenhadas na empresa de acordo com a combinação dos 5 filtros acima.

2 Formas de uso

Essas funções de configuração também foram testadas com python3 (ambiente do django foi previamente configurado para permitir utilizar python3).

2.1 Configurações iniciais

Observação: se quiser testar com todos os dados presentes em *comandos_sql.txt* (presente na pasta api/Busca), uma maneira fácil de repopular o banco de dados é apagar o db.sqlite3 e executar os comandos que se seguem.

Para que a listagem de empresas, cursos e vagas não de erro quando for rodar a aplicação no servidor local, execute os seguintes passos:

- `python manage.py makemigrations api`
- `python manage.py sqlmigrate api 0001`
- `python manage.py migrate`

Para popular o banco de dados, abra um editor bd e execute os comandos presentes no arquivo *comandos_sql.txt* (na pasta api/Busca) num editor de BD e salve as inserções feitas.

Para rodar entre na pasta e inicialize com o comando:

python manage.py runserver

2.2 URLs para executar os itens

As urls estão na mesma ordem em relação ao que foi descrito na seção Funcionalidades presentes.

- `127.0.0.1:8000/api/busca/SearchCompany/`
- `127.0.0.1:8000/api/busca/SearchVacancy/`
- `127.0.0.1:8000/api/busca/course/`
- `127.0.0.1:8000/api/busca/SearchCompany/?id=user_id`
- `127.0.0.1:8000/api/busca/SearchCompany/?name=comapny_name`
- `127.0.0.1:8000/api/busca/SearchOportunity/?id=user_id`
- `127.0.0.1:8000/api/busca/SearchVacancy/?course=course_name`
- `127.0.0.1:8000/api/busca/SearchVacancy/?company=comapny_name`
- `127.0.0.1:8000/api/busca/SearchVacancy/?icreated=YYYYMM`
- `127.0.0.1:8000/api/busca/SearchVacancy/?fcreated=YYYYMM`
- `127.0.0.1:8000/api/busca/SearchVacancy/?fcreated=YYYYMM&icreated=YYYYMM`

Para realizar a combinacao entre entre os filtros, use:
`127.0.0.1:8000/api/busca/SearchVacancy/?course=&company=&icreated=&fcreated=`

A correspondência das variáveis é dada a seguir.

- *user_id* é o número correspondente ao id do usuário logado como estudante.
- *company_name* é o nome da companhia que se deseja conhecer dados sobre.
- *course_name* é o nome do curso que se deseja conhecer dados sobre.
- *YYYYMM* são os números no formato ano mes (*AAAAMM*).

Para realizar a combinação de filtros, colocar após o '=' uma das variáveis listadas acima que é correspondente ao campo.

127.0.0.1:8000 é o endereço retornado ao rodar o comando

python manage.py runserver