
quarksmart Documentation

Rafael Dantas

25 jan, 2019

Conteúdo

1	Instalação	3
2	Utilização	5
2.1	Mais detalhes	6

O **quarksmart** é uma biblioteca Python que permite a integração entre ferramentas de *Data Science* com os dados armazenados no sistema **QuarkSmart**.

Com apenas um comando é possível carregar seus dados para um formato que permite o tratamento de forma fácil e ágil. O quarksmart foi pensado para ser utilizado com outras bibliotecas de *Data Science*, como o pandas e o numpy, mas nada você pode utiliza-la como preferir.

CAPÍTULO 1

Instalação

A instalação da biblioteca é bem simples e pode ser realizada facilmente por meio do gerenciador de pacotes do python (pip).

Para instalar, abrir o terminal e rodar o comando abaixo do pip:

```
$ pip install quarksmart
```

Para verificar se a instalação aconteceu corretamente, basta importar o pacote e verificar a versão.

```
>>> import quarksmart
>>> quarksmart.__version__
... 1.0.0
```

Utilização

Nota: As informações que serem passadas nos parâmetros da função devem ser consultadas direto no sistema do QuarkSmart.

Carregando os dados com o quarksmart é muito fácil, basta utilizar a função `carregar_dados` e passar alguns parâmetros:

- `token` - é a chave de acesso que autentica o usuário no sistema. Esse Token pode ser encontrado no próprio sistema do QuarkSmart na página da [API REST](#).
- `entidade` - Identificador da Entidade, é um atalho para a entidade que permite encontra-la de forma mais fácil. Para encontra-lo, basta acessar a página de [listagem das entidades](#).
- `data` - quando se deseja consultar os dados da entidade em determinada data, é possível informa-la por meio desse parâmetro e o sistema retornará o conjunto de dados da entidade naquele momento. Quando esse parâmetro não é informado o sistema pega estado atual. Ex: `data='01/01/2019'`.

```
>>> import quarksmart as qs
>>> entidade = qs.carregar_dados("pessoa", \
...                             token="b40734876aeea537ec0dce2e371da4a1")
>>> type(entidade)
pandas.core.frame.DataFrame
```

O retorno da função é um **Dataframe** do *pandas*.

Todos os detalhes das funções da biblioteca podem ser encontrados na página de [funções](#).

2.1 Mais detalhes

2.1.1 carregar_dados

Paramêtros

Nota: As informações que serem passadas nos paramêtros da função devem ser consultadas direto no sistema do QuarkSmart.

Carregando os dados com o quarksmart é muito fácil, basta utilizar a funções `carregar_dados` e passar alguns paramêtros:

- `token` - é a chave de acesso que autentica o usuário no sistema. Esse Token pode ser encontrado no próprio sistema do QuarkSmart na página da [API REST](#).
- `entidade` - Identificador da Entidade, é um atalho para a entidade que permite encontra-la de forma mais fácil. Para encontra-lo, basta acessar a página de [listagem das entidades](#).
- `data` - quando se deseja consultar os dados da entidade em determinada data, é possível informa-la por meio desse parâmetro e o sistema retornará o conjunto de dados da entidade naquele momento. Quando esse parâmetro não é informado o sistema pega estado atual. Ex: `data='01/01/2019'`.

O retorno da função é um **Dataframe** do *pandas*.

```
>>> import quarksmart as qs
>>> entidade = qs.carregar_dados("colaboradores", \
...                             token="b40734876aeea537ec0dce2e371da4a1")
>>> type(entidade)
pandas.core.frame.DataFrame
```

Exemplo informando uma data:

```
>>> import quarksmart as qs
>>> entidade = qs.carregar_dados(entidade="movimentacoes-financeiro", \
...                             token="b40734876aeea537ec0dce2e371da4a1", data="01/12/2018")
>>> type(entidade)
pandas.core.frame.DataFrame
```

Erros

Quando o sistema não consegue carregar os dados, ele retorna uma mensagem no lugar dos dados, as mensagens podem ser:

- Falha de Autenticação. - quando o sistema não consegue autenticar o usuário com o Token informado.
- Falha ao carregar os dados - quando a entidade não é encontrada ou existe algum problema com os dados.

```
>>> entidade = qs.carregar_dados("colaboradores", \
...                             token="token_invalido")
>>> entidade
Falha de Autenticação.
```