

# Documentação do ITSQUERY Tickets

## Introdução

ITSQUERY Tickets é um software desenvolvido em Python com PySide6 para consulta de tickets na API do Movidesk. Ele permite selecionar um período através de um calendário e exportar os dados em formato JSON. O software utiliza um token de autenticação e possui uma interface gráfica intuitiva com barra de progresso.

Esta documentação explica brevemente os trechos mais relevantes do código fonte do software.

## Estrutura do Código

### Importação de Bibliotecas

```
Python
import sys

import os

import json

import requests

from datetime import timedelta

from PySide6.QtWidgets import QApplication, QWidget, QLabel, QPushButton,
QVBoxLayout, QDateEdit, QLineEdit, QFileDialog, QProgressBar

from PySide6.QtCore import QDate, Qt, QThread, Signal, QTimer

from PySide6.QtGui import QIcon
```

As bibliotecas são utilizadas para:

- `sys, os`: manipulação de sistema e caminhos de arquivos.
- `json`: manipulação de dados JSON.
- `requests`: realiza requisições HTTP.
- `datetime`: manipulação de datas.
- `PySide6`: cria a interface gráfica do usuário (GUI).

## Dicionários de Conversão

Python

```
ORIGIN_DICT = {...}
```

```
TYPE_DICT = {...}
```

Estes dicionários convertem códigos retornados pela API do Movidesk em descrições legíveis.

## Função Utilitária: `get_resource_path`

Python

```
def get_resource_path(relative_path):  
  
    if getattr(sys, 'frozen', False):  
  
        base_path = sys._MEIPASS  
  
    else:  
  
        base_path = os.path.abspath(".")  
  
    return os.path.join(base_path, relative_path)
```

Esta função permite localizar arquivos (como ícones) tanto em modo de desenvolvimento quanto quando o programa está compilado.

## Classe `TokenInputDialog`

Python

```
class TokenInputDialog(QWidget):
```

Esta classe cria uma janela para solicitar o token da API ao usuário. Após a entrada do token, a interface principal é carregada.

## Função `ensure_list_length`

Python

```
def ensure_list_length(lst, length, default="NA"):
```

Garante que uma lista tenha um tamanho fixo, preenchendo valores padrões caso seja menor.

## Classe **ConsultaThread**

Python

```
class ConsultaThread(QThread):
```

Esta classe executa a consulta à API do Movidesk em uma thread separada, evitando que a interface fique travada. Ela:

- Constrói a URL da requisição.
- Faz a requisição HTTP.
- Processa os dados retornados.
- Emite sinais para a interface indicar sucesso ou erro.

## Classe **HelpdeskQueryApp**

Python

```
class HelpdeskQueryApp(QWidget):
```

Esta classe gerencia a interface principal. Ela:

- Solicita ao usuário o período da consulta.
- Exibe uma barra de progresso.
- Inicia a **ConsultaThread**.
- Processa e exibe os resultados.
- Permite salvar os dados em um arquivo JSON.
- Permite iniciar uma nova consulta.

## Função **submit**

Python

```
def submit(self):
```

Executa a consulta, inicializa a barra de progresso e oculta o botão de consulta até a finalização.

## Função **update\_fake\_progress**

Python

```
def update_fake_progress(self):
```

Simula progresso enquanto os dados estão sendo carregados.

## Funções **on\_consulta\_concluida** e **on\_erro\_consulta**

Python

```
def on_consulta_concluida(self, data):
```

```
def on_erro_consulta(self, erro_msg):
```

Manipulam a resposta da consulta, exibindo os resultados ou mensagens de erro.

## Função **save\_as**

Python

```
def save_as(self):
```

Permite ao usuário salvar os resultados da consulta como um arquivo JSON.

## Função `new_query`

Python

```
def new_query(self):
```

Fecha a janela atual e reinicia o processo de entrada do token.

## Execução do Programa

Python

```
if __name__ == "__main__":
```

Cria e inicia a aplicação, exibindo a tela de entrada do token.

## Conclusão

O ITSQUERY Tickets é uma ferramenta eficiente para consulta de tickets no Movidesk, proporcionando uma interface intuitiva e um processo simplificado de obtenção de dados.