```
import requests
import os
from dotenv import load_dotenv

load_dotenv()

SEAFILE_SERVER = os.getenv("SEAFILE_SERVER", "http://127.0.0.1")
SEAFILE_USERNAME = os.getenv("SEAFILE_USERNAME", "me@example.com")
SEAFILE_PASSWORD = os.getenv("SEAFILE_PASSWORD", "asecret")
SEAFILE_LIBRARY_ID = os.getenv("SEAFILE_LIBRARY_ID", "45a9db89-ab6b-44f5-b927-21e68ed5e248")

print("▯ SEAFILE_USERNAME =", SEAFILE_USERNAME)
print("▯ SEAFILE_PASSWORD =", SEAFILE_PASSWORD)

def get_seafile_token():
  url = f"{SEAFILE_SERVER}/api2/auth-token/"
  headers = {
      "Content-Type": "application/x-www-form-urlencoded"
  }
  data = {
      "username": SEAFILE_USERNAME,
      "password": SEAFILE_PASSWORD
  }
```

```
    print(" [AUTH] URL:", url)

    print(" [AUTH] Data:", data)

    resp = requests.post(url, headers=headers, data=data)

    print(" [AUTH] Response:", resp.text)

    resp.raise_for_status()

    return resp.json()["token"]


def ensure_seafile_directory_exists(token, folder_name):

    dir_api = f"{SEAFILE_SERVER}/api2/repos/{SEAFILE_LIBRARY_ID}/dir/"

    headers = {

        "Authorization": f"Token {token}",

        "Content-Type": "application/x-www-form-urlencoded"

    }


    # Klasör var mı kontrol et

    params = {

        'p': f'/{folder_name}'

    }

    resp = requests.get(dir_api, headers=headers, params=params)

    print(f" [CHECK DIR] status: {resp.status_code}, response: {resp.text}")


    if resp.status_code == 404:

        data = f"operation=mkdir&parent_dir=/&name={folder_name}"

        mkdir_resp = requests.post(dir_api, headers=headers, data=data)

        print(f" [MKDIR] status: {mkdir_resp.status_code}, response: {mkdir_resp.text}")

        mkdir_resp.raise_for_status()

    elif resp.status_code == 200:

        print(f" [MKDIR] Folder '{folder_name}' already exists")
    else:
        resp.raise_for_status()


def upload_file_to_seafile(local_file_path, folder_name='/', remote_file_name=None):
    token = get_seafile_token()

    # ⃣ Klasör var mı, yoksa yarat!
    if folder_name and folder_name != '/':
        ensure_seafile_directory_exists(token, folder_name)
        upload_path = f'/{folder_name}'
    else:
        upload_path = '/'

    # 1⃣ Doğru dizine özel upload link al!
    upload_link_url = f"{SEAFILE_SERVER}/api2/repos/{SEAFILE_LIBRARY_ID}/upload-link/?p={upload_path}"
    headers = {
        "Authorization": f"Token {token}"
    }

    resp = requests.get(upload_link_url, headers=headers)
    resp.raise_for_status()
    upload_link = resp.text.strip('"')

    # 2⃣ Dosyayı yükle
    with open(local_file_path, 'rb') as f:
        files = {
            'file': (remote_file_name or os.path.basename(local_file_path), f)
        }
        data = {
            'parent_dir': upload_path
        }
        upload_resp = requests.post(upload_link, headers=headers, files=files, data=data)

        print(f"□ [UPLOAD] Response: {upload_resp.status_code}, {upload_resp.text}")

        upload_resp.raise_for_status()
        return upload_resp.text

    token = get_seafile_token()
    if folder_name and folder_name != '/':
        ensure_seafile_directory_exists(token, folder_name)
    upload_link_url = f"{SEAFILE_SERVER}/api2/repos/{SEAFILE_LIBRARY_ID}/upload-link/"
    headers = {
        "Authorization": f"Token {token}"
    }
    resp = requests.get(upload_link_url, headers=headers)
    resp.raise_for_status()
    upload_link = resp.text.strip('"')
    with open(local_file_path, 'rb') as f:
        files = {
            'file': (remote_file_name or os.path.basename(local_file_path), f)
        }
        data = {
            'parent_dir': f'/{folder_name}' if folder_name else '/'
        }
        upload_resp = requests.post(upload_link, headers=headers, files=files, data=data)
```

```
print("□ [UPLOAD] Response:", upload_resp.text)

upload_resp.raise_for_status()

return upload_resp.text
```