Client Serveur-Image-Fichier

Ce code est une application développée en **Lazarus (Free Pascal)** qui permet d’envoyer et de récupérer des fichiers depuis un serveur via des requêtes **HTTP (GET & POST)** en utilisant la bibliothèque `fphttpclient`.

Analyse du Code

L'application contient une **interface graphique** avec des boutons pour :

1. **Télécharger un fichier** depuis un serveur via une requête GET.
2. **Envoyer un fichier** vers un serveur via une requête POST.
3. **Afficher les réponses** du serveur dans un `TMemo`.

Déclaration et Interface (Unit1.pas)

```
unit Unit1;

{$mode objfpc}{$H+}

interface
```

- `{$mode objfpc}{$H+}` : Active le mode objet Free Pascal et la gestion avancée de la mémoire.
- `interface` : Déclare les classes et les fonctions accessibles depuis d'autres unités.

Bibliothèques utilisées

```
uses
  Classes, SysUtils, Forms, Controls, Graphics, Dialogs, StdCtrls, fphttpclient,
  fpjson;
```

- **`Classes, SysUtils`** : Gestion de fichiers et de chaînes de caractères.
- **`Forms, Controls, Graphics, Dialogs`** : Composants de l'interface utilisateur.
- **`StdCtrls`** : Éléments comme `TButton`, `TLabel`, `TMemo`, etc.
- **`fphttpclient`** : Permet d'envoyer des requêtes HTTP.
- **`fpjson`** : Manipulation de JSON (pas utilisé directement ici).

Déclaration de `TForm1` (Formulaire principal)

```
type
  { TForm1 }
  TForm1 = class(TForm)
    BtnGetFile: TButton;  // Bouton pour récupérer un fichier (GET)
    BtnUploadFile: TButton;  // Bouton pour envoyer un fichier (POST)
    MemoResponse: TMemo;  // Zone de texte pour afficher les réponses du serveur
    OpenDialog: TOpenDialog;  // Boîte de dialogue pour sélectionner un fichier
    BtnSendRequest: TButton;  // Bouton supplémentaire (non utilisé ici)
    EdtUrl: TEdit;  // Champ pour saisir l'URL du serveur
    EdtAutorisation: TEdit;  // Champ pour saisir le token d'autorisation
    Label1: TLabel;  // Libellé pour l'URL
    Label2: TLabel;  // Libellé pour le token d'autorisation
```

- L'application possède des champs pour **saisir l’URL** du serveur et **le token d’authentification**.
- Un `TButton` permet d’envoyer et de récupérer des fichiers.

Récupération d'un fichier via HTTP GET

Fonction `SendGetRequest` :

```
function TForm1.SendGetRequest(FileName,_url,Auth: string): string;
```

- Récupère un fichier depuis un serveur HTTP.
- `FileName` : Nom du fichier à récupérer.
- `_url` : Adresse du serveur (exemple : `192.168.1.123:9001/stream/`).
- `Auth` : Token d'autorisation.

Étapes :

1. Construction de l'URL en encodant le nom du fichier.
2. Création du client HTTP (`TFPHTTPClient`).
3. Ajout du token d’authentification (`Authorization`).
4. Envoi de la requête `GET` et récupération du fichier.
5. Sauvegarde du fichier téléchargé dans `./image/`.

```
URL := 'http://'+_url+ URLEncode(ExtractFileName(FileName));
AUTH_TOKEN := 'Bearer '+Auth;
HttpClient := TFPHTTPClient.Create(nil);

HttpClient.AddHeader('Authorization', AUTH_TOKEN);
HttpClient.AddHeader('Content-Type', 'application/octet-stream');
Stream := TMemoryStream.Create;

SavePath := GetCurrentDir+'/image/'+ExtractFileName(FileName);
ShowMessage(SavePath);

try
  HttpClient.Get(URL, Stream);  // Exécute la requête GET
  Stream.SaveToFile(SavePath);  // Sauvegarde le fichier téléchargé
  Result := SavePath;
except
  on E: Exception do
    ShowMessage('Error: '+E.Message);
end;
```

Envoi d'un fichier via HTTP POST

Fonction `SendPostRequest` :

```
function TForm1.SendPostRequest(FileName,_url,Auth: string): string;
```

- Permet d'envoyer un fichier vers un serveur.

Étapes :

1. Construction de l’URL en encodant le nom du fichier.
2. Création d’un client HTTP (`TFPHTTPClient`).
3. Ajout du token d’authentification (`Authorization`).
4. Lecture du fichier et envoi via une requête `POST`.
5. Gestion des réponses du serveur.

```
URL := 'http://'+_url+ URLEncode(ExtractFileName(FileName));
AUTH_TOKEN := 'Bearer '+Auth;
Client := TFPHTTPClient.Create(nil);
FileStream := TFileStream.Create(FileName, fmOpenRead);
Response := TStringStream.Create('');

try
  Client.AddHeader('Authorization', AUTH_TOKEN);
  Client.AddHeader('Content-Type', 'application/octet-stream');

  try
    Client.RequestBody := FileStream;
    Client.Post(URL , Response);
    ResponseCode := Client.ResponseStatusCode;

    if ResponseCode = 201 then
      Result := 'File uploaded successfully : '+FileName
    else
      Result := 'Upload failed. HTTP Code: ' + IntToStr(ResponseCode) + ' - ' +
Response.DataString;
  except
    on E: Exception do
      Result := 'Error: ' + E.Message;
  end;

finally
  FileStream.Free;
  Response.Free;
  Client.Free;
end;
```

Actions des Boutons

Bouton `BtnGetFileClick` (Téléchargement via GET)

```
procedure TForm1.BtnGetFileClick(Sender: TObject);
var
  FileName: string;
begin
  FileName := InputBox('Nom Fichier', 'Enter Nom Fichier:', '');
  if FileName <> '' then
    MemoResponse.Lines.Text := SendGetRequest(FileName, EdtUrl.Text,
EdtAutorisation.Text);
end;
```

- Affiche une boîte de dialogue pour demander le **nom du fichier**.
- Exécute `SendGetRequest()` pour récupérer le fichier.
- Affiche la réponse dans `MemoResponse`.

Bouton `BtnUploadFileClick` (Envoi via POST)

```
procedure TForm1.BtnUploadFileClick(Sender: TObject);
begin
  if OpenDialog.Execute then
  begin
    MemoResponse.Lines.Add('Chargement: ' + OpenDialog.FileName);
    MemoResponse.Lines.Text := SendPostRequest(OpenDialog.FileName, EdtUrl.Text,
EdtAutorisation.Text);
  end;
end;
```

- Ouvre une boîte de dialogue (`OpenDialog`) pour sélectionner un fichier.
- Exécute `SendPostRequest()` pour envoyer le fichier au serveur.
- Affiche la réponse dans `MemoResponse`.

Résumé

Cette application Lazarus permet **d'envoyer et de télécharger des fichiers** vers un serveur via des requêtes HTTP.

Elle utilise `fphttpclient` pour gérer les requêtes **GET** (téléchargement) et **POST** (envoi).

L'utilisateur saisit l'URL et le **token d'authentification** avant d'effectuer une action.

Les fichiers téléchargés sont **enregistrés dans le dossier `./image/`**.

Besoin d'une amélioration ou d'une fonctionnalité supplémentaire ?