Spécification des méthodes APIs

Projet YourList : Florian Coquillat, Damien Fontes

Partie Connexion :

fonction : cookieConnected()

- Objectif : Définir les cookies pour un compte non entreprise lors de la connexion.
    - ‘connected’ = ‘true’
    - compteEntreprise = ‘false’
- Sortie : ‘ok’ lorsque que les cookies sont définis.

fonction : cookieConnectedEntreprise()

- Objectif : Définir les cookies pour un compte entreprise lors de la connexion.
    - ‘connected’=’true’
    - compteEntreprise=’true’
- Sortie : ‘ok’ lorsque que les cookies sont définis.

fonction : testConnexion()

- Objectif : Récupérer les identifiants, hash le mot de passe (sha256), appel login(id,mdp) pour vérifier si l’utilisateur non compte entreprise existe en base de données.
- Entrée : id, mdp
    - id : login de l’utilisateur
    - mdp : mot de passe de l’utilisateur
- Sortie : True, False
    - ‘True’, si l’utilisateur existe en base de données
    - ‘False’, si l’id et le mdp ne correspondent pas à un utilisateur non compte entreprise.

fonction : testConnexionEntreprise()

- Objectif : Récupérer les identifiants, hash le mot de passe (sha256), appel loginEntreprise(login,mdp) pour vérifier si l’utilisateur compte entreprise existe en base de données.
- Entrée : login, mdp
    - login : login de l’utilisateur
    - mdp : mot de passe de l’utilisateur
- Sortie : True, False
    - ‘True’, si l’utilisateur existe en base de données
    - ‘False’, si l’id et le mdp ne correspondent pas à un utilisateur compte entreprise.

fonction : isConnected()

- Objectif : Regarde les cookies. Vérifie que l’utilisateur soit connecté et que ce ne soit pas un compte entreprise.
- Sortie : True, False, entreprise
    - ‘True’, si l’utilisateur est connecté et n’est pas un compte entreprise.
    - ‘False’, si l’utilisateur n’est pas connecté.
    - ‘entreprise’, si l’utilisateur est connecté mais sur un compte entreprise. L’utilisateur est alors déconnecté.

fonction : isConnectedEntreprise()

- Objectif : Regarde les cookies. Vérifie que l'utilisateur soit connecté et que ce soit un compte entreprise.
- Sortie : True, False, entreprise
    - 'True', si l'utilisateur est connecté sur un compte entreprise.
    - 'False', si l'utilisateur n'est pas connecté ou est connecté sur un compte non entreprise. L'utilisateur est alors déconnecté.

fonction : getDataRechercher()

- Objectif : Effectuer une recherche sur l'API serpapi qui renvoie les vidéos youtube en lien avec la recherche
- Donnée : apiKey, url
    - apiKey : clé de l'API serpapi
    - url : Url de l'API
- Entrée : input : mot clé à rechercher
- Sortie : JSON comprenant toutes les vidéos récupérées par la requête à l'API

Partie Utilisateur :

fonction : createUser()

- Objectif : Créer un utilisateur non entreprise
- Entrée : login, mdp, mdpConfirm, nom, prenom, email
    - login : login de l'utilisateur à créer
    - mdp : mot de passe
    - mdpConfirm : confirmation du mot de passe
    - nom : nom de l'utilisateur
    - prenom : prénom de l'utilisateur
    - email : email de l'utilisateur
- Sortie : ok, Login déjà existant., Mot de passe différent.
    - 'ok' : Le compte a été créé.
    - 'Login déjà existant.' : Message d'erreur affiché dans le html.
    - 'Mot de passe différent.' : Message d'erreur affiché dans le html.

fonction : createUserEntreprise()

- Objectif : Créer un utilisateur entreprise
- Entrée : login, mdp, mdpConfirm, entreprise, email
    - login : login du compte entreprise à créer
    - mdp : mot de passe
    - mdpConfirm : confirmation du mot de passe
    - entreprise: nom de l'entreprise
    - email : email de l'entreprise
- Sortie : ok, Login déjà existant., Mot de passe différent.
    - 'ok' : Le compte a été créé.
    - 'Login déjà existant.' : Message d'erreur affiché dans le html.
    - 'Mot de passe différent.' : Message d'erreur affiché dans le html.

fonction : getUserByLogin()

- Objectif : Récupérer un utilisateur dans la base de données grâce au login.
- Entrée : login : login de l'utilisateur cherché
- Sortie : JSON, None
    - JSON contenant les données de l'utilisateur
    - None : Si l'utilisateur n'existe pas

fonction : getUserById()

- Objectif : Récupérer un utilisateur dans la base de données grâce à l'identifiant.
- Entrée : id : identifiant de l'utilisateur cherché
- Sortie : Tableau, 'Erreur'
    - Tableau contenant les donnée de l'utilisateur
    - 'Erreur' : Si l'utilisateur n'existe pas

fonction : updateUser()

- Objectif : Modifier un utilisateur avec les nouvelles valeurs.
- Entrée : id, login, nom, prenom, email
    - id : identifiant de l'utilisateur à modifier,
    - login : nouveau login de l'utilisateur.
    - nom : nouveau nom de l'utilisateur.
    - prénom: nouveau prénom de l'utilisateur.
    - email : nouvel email de l'utilisateur.
- Sortie : 'ok' : Utilisateur modifier

Partie Playlist :

fonction : getPlaylist()

- Objectif : Récupérer les playlists d'un utilisateur grâce à son identifiant.
- Entrée : id : identifiant de l'utilisateur
- Sortie : Tableau de playlist, None
    - Tableau de playlists contenant les données de chaque playlist
    - None : Si aucune playlist n'a été trouvé

fonction : creationPlaylist()

- Objectif : Créer une nouvelle playlist dans la table 'playlist' et une nouvelle ligne dans la table 'possede' (table liant les playlists aux utilisateurs).
- Entrée : id, nomPlaylist
    - id : identifiant de l'utilisateur
    - nomPlaylist : nom de la playlist
- Sortie : 'ok' : playlist et possede créés
- Données : nomCreateur, idPlaylist
    - nomCreateur : nom du créateur de la playlist, récupéré en recherchant le nom de l'utilisateur par son id.
    - idPlaylist : id de la playlist créée, renvoyé par la fonction effectuant la requête SQL.

fonction : addVideoPlaylist()

- Objectif : Ajouter une vidéo dans une playlist d'un utilisateur. Si la vidéo n'existe pas dans la table 'video', ajoute la vidéo à la table.
- Entrée : idPlaylist, titre, lien, duree, site, thumbnail, vues
    - idPlaylist : identifiant de la playlist à laquelle on ajoute une vidéo,
    - titre : titre de la vidéo,
    - duree : durée de la vidéo,
    - site : site (plateforme) d'où provient la vidéo,
    - thumbnail : miniature de la vidéo,
    - vues : nombre de vues de la vidéo.
- Sortie : 'ok', 'alreadyExist'
    - 'ok' : vidéo ajoutée à la playlist
    - 'alreadyExist' : vidéo déjà existante dans la playlist.

fonction : verifPossede()

- Objectif : Vérifier si un utilisateur possède bien une playlist.
- Entrée : id, idPlaylist
    - id : identifiant de l'utilisateur,
    - idPlaylist : identifiant de la playlist.
- Sortie : 'ok', 'Playlist n'appartenant pas à l'utilisateur'
    - 'ok' : La playlist appartient à l'utilisateur,
    - 'Playlist n'appartenant pas à l'utilisateur' : message d'erreur affiché en html.

fonction : playlistById()

- Objectif : Renvoyer une playlist grâce à son identifiant.
- Entrée : id : identifiant de la playlist à renvoyer.
- Sortie : Tableau, None
    - Tableau contenant toutes les informations de la playlist,
    - None : Pas de playlist trouvée avec cet identifiant.

fonction : getVideoPlaylist()

- Objectif : Renvoyer toutes les vidéos d'une playlist.
- Entrée : id : identifiant de la playlist.
- Sortie : Tableau, 'Playlist vide'
    - Tableau de vidéos se trouvant de la playlist,
    - 'Playlist vide' : La playlist n'a pas de vidéo.

fonction : supprimerVideoPlaylist()

- Objectif : Supprimer une vidéo d'une playlist. Il faut donc supprimer une ligne de la table contient (table liant les playlists aux vidéos)
- Entrée : idVideo, idPlaylist
    - idVideo : identifiant de la vidéo à supprimer;
    - idPlaylist : identifiant de la playlist où se trouve la vidéo.
- Sortie : 'ok', 'Vidéo déjà supprimée'
    - 'ok' : Ligne de la table contient supprimée,
    - 'Vidéo déjà supprimée' : La ligne contenant l'idVideo et l'idPlaylist de la table contient n'existe pas ou a déjà été supprimée.

fonction : modifierTitrePlaylist()

- Objectif : Modifier le nom d'une playlist grâce à son identifiant.
- Entrée : idPlaylist, titre
    - idPlaylist : identifiant de la playlist à modifier,
    - titre : nouveau nom de la playlist.
- Sortie : 'ok' : Le nom de la playlist a été changé.

fonction : supprimerPlaylist()

- Objectif : Supprimer une playlist. Il faut aussi supprimer toutes les lignes où son identifiant apparaît dans les tables 'contient' et 'possede'.
- Entrée : idPlaylist, idUtilisateur
    - idPlaylist : identifiant de la playlist à supprimer,
    - idUtilisateur : identifiant de l'utilisateur qui possède la playlist.
- Sortie : 'ok', 'Playlist Vide'
    - 'ok' : Toutes les lignes ont été supprimées,
    - 'Playlist Vide' : Pas de ligne dans la table 'contient' à supprimer, on sort plus tôt de la fonction.

Partie Pub :

fonction : uploaderEntreprise()

- Objectif : Créer une nouvelle publicité dans la table 'pub'.
- Entrée : titrePub, lienPub, idUser, file, format
    - titrePub : nom de la nouvelle publicité,
    - lienPub : URL de redirection de la publicité,
    - idUser : utilisateur entreprise à qui appartient la publicité,
    - file : image de la publicité,
    - format : format de l'image (728x90 ou 320x50) en px.
- Sortie : render_template("uploaderEntreprise.html) : renvoie la page html uploaderEntreprise.

fonction : getPubByIdUtilisateur()

- Objectif : Renvoyer toutes les publicités liées à un compte entreprise.
- Entrée : id : identifiant de l'utilisateur.
- Sortie : Tableau de pub, None
    - Tableau de toutes les publicités ainsi que leurs informations,
    - None : Pas de publicités liées au compte

fonction : getPubByIdUIdP()

- Objectif : Renvoyer une publicité en fonction de son identifiant et de l'identifiant de l'utilisateur.
- Entrée : id, idPub
    - id : identifiant de l'utilisateur,
    - idPub : identifiant de la publicité
- Sortie : Tableau, None
    - Tableau de toutes les informations liées à la publicités,
    - None : Pas de publicités liées au compte.

fonction : getRandomPubByFormat()

- Objectif : Renvoyer un nombre de publicités choisies aléatoirement en fonction de leur format.
- Entrée : pubNmb, format
    - pubNmb : Nombre de publicités à renvoyer,
    - format : format de l'image (728x90 ou 320x50) en px.
- Sortie : Tableau de pub, None
    - Tableau de toutes les publicités ainsi que leurs informations,
    - None : Pas de publicités dans l'application.

fonction : clickPub()

- Objectif : Ajouter 1 au nombre de fois qu'un utilisateur à cliquer sur la publicité.
- Entrée : idPub : identifiant de la pub à modifier.
- Sortie : Identifiant de la pub modifiée.

Partie Redirection :

fonction : connexion()

- Objectif : Permet de se rendre sur la page connexion.html
- Sortie : render_template("connexion.html) : renvoie la page html connexion.

fonction : inscription()

- Objectif : Permet de se rendre sur la page inscription.html
- Sortie : render_template("inscription.html) : renvoie la page html inscription.

fonction : video()

- Objectif : Permet de se rendre sur la page video.html
- Sortie : render_template("video.html) : renvoie la page html video.

fonction : playlist()

- Objectif : Permet de se rendre sur la page playlist.html
- Sortie : render_template("playlist.html) : renvoie la page html playlist.

fonction : compte()

- Objectif : Permet de se rendre sur la page compte.html
- Sortie : render_template("compte.html) : renvoie la page html compte.

fonction : modifierPlaylist()

- Objectif : Permet de se rendre sur la page modifierPlaylist.html
- Sortie : render_template(“modifierPlaylist.html) : renvoie la page html modifierPlaylist.

fonction : inscriptionEntreprise()

- Objectif : Permet de se rendre sur la page inscriptionEntreprise.html
- Sortie : render_template(“inscriptionEntreprise.html) : renvoie la page html inscriptionEntreprise.

fonction : connexionEntreprise()

- Objectif : Permet de se rendre sur la page connexionEntreprise.html
- Sortie : render_template(“connexionEntreprise.html) : renvoie la page html connexionEntreprise.

fonction : compteEntreprise()

- Objectif : Permet de se rendre sur la page compteEntreprise.html
- Sortie : render_template(“compteEntreprise.html) : renvoie la page html compteEntreprise.

fonction : pubEntreprise()

- Objectif : Permet de se rendre sur la page pubEntreprise.html
- Sortie : render_template(“pubEntreprise.html) : renvoie la page html pubEntreprise.