

Tests et structuration du projet :

<http://webtp.fil.univ-lille1.fr/~texierl/Rezozio/index.php>

Cette url correspond à la page d'accueil et permet d'aller sur toutes les autres pages (certaines nécessitant une authentification).

Page d'un membre (exemple) :

<http://webtp.fil.univ-lille1.fr/~texierl/Rezozio/pageMembre.php?login=Flash>

Page de mise à jour (que si une personne est authentifiée) :

<http://webtp.fil.univ-lille1.fr/~texierl/Rezozio/MiseAJour.php>

Crédits : <http://webtp.fil.univ-lille1.fr/~texierl/Rezozio/credits.php>

Voici deux comptes vous permettant de tester les différentes fonctionnalités.

Login	Mot de passe
Iris	Zozio
Willy	voyage

PAGE D'ACCUEIL

Sur la page d'accueil, l'utilisateur peut se créer un compte grâce au formulaire d'inscription qui fait appel au web service `createUser`. Ce formulaire ne peut pas être modifié pendant l'envoi des données et la création de ce compte. Un message indiquant que le chargement est en cours est affiché pendant le traitement de ces données. Une fois que cela est terminé, un message s'affiche afin de prévenir l'utilisateur que son compte a été créé avec succès.

Il peut également, sur cette page, se connecter (une fois son compte créé) grâce au formulaire de connexion qui fait appel au web service `login`. Lors du chargement et de la vérification de la validité des informations, le formulaire est désactivé et un message s'affiche pour indiquer que le chargement est en cours. Si l'utilisateur s'est trompé d'identifiants, un message est alors affiché afin de lui indiquer que son login et/ou mot de passe est erroné. Dans le cas contraire, cela authentifie l'utilisateur.

Si la personne n'est pas connectée, elle peut voir les messages de tous les utilisateurs. Dans le cas contraire, elle ne voit que les messages des personnes auxquelles elle est abonnée. Les messages sont triés du plus récent au plus ancien. Ces messages sont récupérés grâce au web services `findMessages` avec des paramètres différents suivant le cas. L'avatar ainsi que le login et le nom de l'auteur du message sont affichés. Ces informations sont récupérées grâce aux web services `getUser` et `getAvatar`. Un clic sur le nom ou le login de l'auteur d'un message ouvre la page de ce membre.

Lorsqu'une personne est citée, (i.e. la personne ayant écrit ce message a cité cette personne en ajoutant un @ devant son login. Seules les personnes ayant un compte au moment où le message est publié sera traité comme une citation) son login est affiché en orange dans le message. Un passage de la souris sur ce nom affiche le nom et la photo de la personne correspondante. Un clic sur son login ouvre la page de ce membre.

L'utilisateur peut aussi chercher un autre utilisateur suivant son nom ou son login (ou une partie de son nom ou de son login) grâce au web service `findUsers`. Pour cela, il rentre ces informations dans le champ approprié et envoie le formulaire. Une liste est alors affichée. L'utilisateur peut ainsi choisir une personne de cette liste en cliquant dessus. Lors de la sélection d'une personne le fil des messages se réduit à ceux de cette seule personne. Ces messages sont également récupérés grâce au web services `findMessages`. Ces informations sont également récupérées grâce aux web services `getUser` et `getAvatar`.

Un bouton est disponible à la fin des messages affichés si tous les messages ne sont pas affichés. En effet, les messages sont affichés par quinzaine. Un clic sur ce bouton affichera de nouveaux messages (soit 15 de plus soit moins s'il reste moins de 15 messages à afficher).

Un lien vers la page de crédits est également présent sur cette page.

Le web service `isLogged` est utile sur cette page afin de savoir si la personne est connectée ou non et donc affiché ce qu'il faut sur la page en fonction. En effet, si la personne est connectée, les formulaires de connexion et d'inscription ne sont alors plus visibles. Cependant, de nouvelles fonctionnalités sont présentes. En effet, l'utilisateur dispose d'un bouton permettant de se déconnecter grâce au web service `logout`. Un lien vers son compte est également disponible.

De plus, il a la possibilité de publier un message grâce au formulaire présent. Lors de l'envoi de son message, le formulaire est désactivé et un message s'affiche pour indiquer que le traitement est en cours. Ce formulaire fait appel au webservice `postMessage`. Une fois le message traité et ajouté à la base de données, un message indique que le message a été ajouté avec succès.

PAGE D'UN MEMBRE

Le web service `isLogged` est également utile sur cette page afin de savoir si la personne est connectée ou non et donc affiché ce qu'il faut sur la page en fonction.

Si la personne n'est pas connectée, la page d'un membre est composée des informations (login, nom, description) du membre récupérées grâce au web service `getUser` ainsi que de son avatar récupéré avec le web service `getAvatar`. On y trouve aussi un lien vers la page des crédits et un lien vers la page d'accueil.

Si la personne est connectée, elle a accès à des fonctionnalités supplémentaires. En effet, un bouton pour se déconnecter est présent. Ce bouton fait appel au web service `logout`. Il peut également consulter la liste de ses abonnements grâce au web service `abonnements` et la liste de ses abonnées grâce au web service `abonnees`. Un bouton permet d'afficher ses listes. Une fois affichées, elles peuvent être cachées grâce au bouton correspondant. De plus, il peut voir les messages écrits par ce membre et les messages où ce membre est cité. Pour les afficher, il doit appuyer sur le bouton correspond. Il peut après les avoir affichés, les cacher grâce au bouton correspondant. Le fonctionnement des messages est le même que la page d'accueil.

Si la page visitée ne correspond pas à celle du membre connecté, un bouton concernant l'abonnement est présent. Celui-ci affiche «S'abonner» ou bien «Se désabonner» suivant le cas déterminé grâce au web service `aboOuNon`. Si la personne connectée est déjà abonnée à ce membre alors le bouton affiche «S'abonner». Dans le cas contraire, il affiche «Se désabonner». Un clic sur le bouton permet de s'abonner ou de se désabonner suivant le cas grâce au web service `AboDesabo`.

Si la page visitée correspond à celle du membre connecté, un lien vers la page de mise à jour du profil est présent.

PAGE DE MISE A JOUR

Si la personne n'est pas connectée, une redirection vers la page d'accueil est effectuée.

Si elle est connectée, une mise à jour des informations (nom, mot de passe, description) est disponible grâce à un formulaire faisant appel au web service `setProfile`. Elle peut également grâce à l'aide d'un second formulaire et du web service `uploadAvatar` mettre la photo de son avatar à jour. Le formulaire envoyé est indisponible pendant l'envoi des données et un message indiquant que le traitement est en cours est affiché. Une fois fini, un message s'affiche afin que l'utilisateur sache que ses informations (ou son avatar suivant le formulaire envoyé) ont bien été mises à jour.

Un lien vers la page des crédits est disponible ainsi qu'un lien vers son compte et la page d'accueil.

Toutes les fonctionnalités demandées sont opérationnelles.

Les fichiers :

Javascript (dossier js)

requetes.js

Ce fichier comporte toutes les requêtes utiles pour la page d'accueil (page : index.php). Il comporte également les fonctions utiles aux traitements de l'affichage des messages sur cette page.

requetesMAJ.js

Ce fichier comporte toutes les requêtes utiles pour la mise à jour du profil (i.e. mise à jour de ses informations et de son avatar) d'une personne connectée. On y trouve également une fonction permettant la désactivation ou l'activation d'un formulaire suivant le cas.

requetesMembre.js

Ce fichier comporte toutes les requêtes utiles pour la page d'un membre. Il affiche, sur la page, différentes choses suivant si l'utilisateur est connecté ou non ainsi que suivant si la page visitée correspond à la page de l'utilisateur connecté.

Fonctions et classes (dossier lib)

biblio.php

Ce fichier comporte toutes les fonctions utiles aux web services. Chaque fonction porte le même nom que son web service associé. De plus, quelques fonctions utiles dans plusieurs autres fonctions sont définies afin de les répéter plusieurs fois.

connexion.php

Ce fichier permet la connexion à la base de données (postgresql).

Identite.class.php

Ce fichier définit la classe « identité » et ses fonctions (construction et getters) associées.

MessageMention.class.php

Ce fichier définit la classe « MessageMention » et les fonctions permettant le traitement des messages afin de les enregistrer dans la base de données.

Webservices (dossier services)

AboDesabo.php

Ce web service sert à abonner ou désabonner une personne à une autre suivant le cas.

abonnements.php

Ce web service sert à nous donner la liste des abonnements d'une personne.

abonnes.php

Ce web service sert à nous donner la liste des abonnés d'une personne.

aboOuNon.php

Ce web service sert à savoir si une personne est abonnée à une autre.

createUser.php

Ce web service sert à créer un nouvel utilisateur. Sa description est par défaut vide et une image par défaut lui est attribuée.

findMessages.php

Ce web service sert à trouver des messages dans la base de données suivant divers critères.

getAvatar.php

Ce web service sert à afficher l'avatar d'un utilisateur.

getUser.php

Ce web service sert à nous donner les informations d'un utilisateur.

isLogged.php

Ce web service sert à nous donner les informations sur la personne connectée (s'il y en a une de connectée).

login.php

Ce web service sert à connecter quelqu'un au site.

logout.php

Ce web service sert à déconnecter quelqu'un du site si cette personne est connectée.

postMessage.php

Ce web service sert à poster un message sur le site si la personne est connectée.

setProfile.php

Ce web service sert à mettre à jour les informations d'un utilisateur (mot de passe, nom, description) si cette personne est connectée.

uploadAvatar.php

Ce web service sert à mettre à jour l'avatar d'un utilisateur si cette personne est connectée.

CSS

rezozioCSS.css

Ce fichier définit tout le CSS relatif au site.

Lien de pages

credits.php

Ce fichier contient les crédits pour les photos utilisées sur le site.

index.php

Ce fichier (avec l'aide du fichier requetes.js) permet l'affichage de la page d'accueil. Le contenu de cette page est adapté suivant si une personne est connectée ou non.

MiseAJour.php

Ce fichier (avec l'aide du fichier requetesMAJ.js) permet l'affichage de la page permettant la mise à jour du profil si la personne est connectée. Sinon, cela renvoie vers la page d'accueil.

pageMembre.php

Ce fichier (avec l'aide du fichier requetesMembre.js) permet l'affichage de la page d'un membre suivant le login donné en mode get.