



Les protocoles du Web

Serveurs Web, HTTP et autres protocoles - Formulaires HTML5

Dans ce TD, vous allez rafraîchir et compléter vos connaissances sur les protocoles importants de l'internet à savoir : SMTP, POP3 et surtout HTTP, observer la communication entre un client et un serveur Web et refaire des formulaires Web complets avec toutes les vérifications nécessaires et un beau CSS !

Exercice 1. Le protocole SMTP

Utilisation du protocole Simple Mail Transfert Protocol : Envoi d'un message.

- Consultez le RFC correspondant à SMTP. Les RFC se trouvent sur https://www.rfc-editor.org/search/rfc_search.php et leur traduction en français sur <http://abcdrfc.free.fr/>
- Envoyer "manuellement" un mail en utilisant la commande *telnet* ou *netcat* pour se connecter au port 25 du serveur `smtp.univ-orleans.fr`
- Ecrire un petit programme en java ou bash vous permettant d'envoyer un mail.

Exercice 2. Utilisation de POP3

Le protocole POP3 (PostOffice Protocol) vous permet de récupérer des mails. On pourra lire les spécifications de ce protocole sur les RFC (Request For Comments) : <http://abcdrfc.free.fr/rfc-vf/rfc1939.html> On va maintenant récupérer un mail sur `mailetu.univ-orleans.fr` ou sur toute autre messagerie POP3 accessible.

- Assurez-vous d'avoir au moins un message en attente sur cette messagerie.
- Quel serait l'échange si cette connexion n'était pas sécurisée ?
- En utilisant POP3 SSL dans la librairie `poplib` tapez le code Python ci-dessus qui vous permet de récupérer vos messages.
- Mettez comme login la lettre o suivie de votre numéro d'étudiant.

En utilisant par exemple le langage Python :

```
1 import getpass, poplib
2
3 Mailbox = poplib.POP3_SSL('mailetu.univ-orleans.fr', '995')
4 Mailbox.user('login')
5 passwd = getpass.getpass('Password:')
6 Mailbox.pass_(passwd)
```

Developpement Web Lic Pro Web et Mobile (TD n°1) 2018–2019

```
7 numMessages = len(Mailbox.list()[1])
8 for i in range(numMessages):
9     for msg in Mailbox.retr(i+1)[1]:
10         print(msg)
11 Mailbox.quit()
```

Exercice 3. Apache

Sur chaque machine de l'IUT tourne un serveur Web Apache

- Connectez-vous sur `http://localhost` pour vous en assurer
- Créez un répertoire **www** à la racine de votre HOME contenant une page Web simple de test.
- Procédez à sa “publication” en vérifiant que tous les contenus de ce dossier son accessibles à tout le monde en lecture.
- Attention, les répertoires doivent aussi être traversables ce qui inclus votre HOME. Les commandes à faire sont donc généralement : `chmod -R a+rX ~/www` et `chmod a+x ~`.
- Ces commandes sont à faire une seule fois et dépendent de la configuration spécifique du serveur. Si les modules suexec et suphp d'Apache sont activés, les droits sont plus simples à configurer !
- Attention : Les fichiers n'ont pas à être exécutables ! C'est une faille de sécurité que de rendre tout exécutable ou de tout mettre en écriture pour tout le monde. Ne le faites pas !
- Visualisez votre site dans un navigateur à l'adresse `http://localhost/~vous` où *vous* est votre nom de login.
- Consultez les fichiers de configuration d'Apache dans `/etc/apache2/`.
- Testez également le serveur Web intégré de PHP :
`php -S localhost:8000` ou celui de Python :
`python3 -m http.server`.

Exercice 4. Le protocole HTTP

Utilisation du protocole HyperText Transfer Protocol. On va maintenant se connecter directement sur un serveur Web.

- Faire un netcat sur le port 80 d'une machine sur laquelle tourne un serveur web et essayez de communiquer avec le serveur par le protocole HTTP.
- Essayez essentiellement GET /
- Que constatez-vous ?
- Que se passe-t-il si vous demandez une page qui n'existe pas (par exemple GET /toto) ?
- Observez les échanges entre le navigateur et le serveur Web en utilisant l'onglet "réseau" des outils de developpement.

Developpement Web Lic Pro Web et Mobile (TD n°1) 2018–2019

Exercice 5. HTTP 2

Découverte du protocole HTTP version 2

1. Commencez par lire l'introduction au protocole HTTP 2 par Daniel Haxx : <https://daniel.haxx.se/http2/http2-fr.pdf>
2. On peut aussi lire la RFC <https://tools.ietf.org/html/rfc7540> ou les explications en français <http://www.bortzmeyer.org/7540.html>
3. Trouvez certains sites qui utilisent le protocole HTTP 2
4. Utilisez les outils de débogage du navigateur et notamment l'onglet Réseaux pour observer les échanges avec un site prévu pour HTTP 2.
5. Faut-il minifier et compacter tous les fichiers d'un site Web avec le protocole HTTP 2 ?

Exercice 6. Utilisation d'un script CGI

On va maintenant (ré)utiliser les CGI avec Apache. Dans votre HOME, placez-vous dans le répertoire `www` et créez-y un dossier `cgi-bin` ayant le droit dans lequel vous aurez placé `hello.cgi` ci-dessous. Le fichier `hello.cgi` doit être rendu exécutable. Puis pointez votre navigateur sur `http://localhost/~login/cgi-bin/hello.cgi`

```
#!/bin/bash
# -*- shell-script -*-

cat <<EOF
Content-type: text/plain

hello world!
EOF
```

Exercice 7. QUERY_STRING

1. Ecrivez un formulaire simple avec saisie d'un nom et d'un prénom dans des champs texte
2. Ecrivez un script bash qui récupérera les valeurs de ces champs et qui affichera les variables d'environnement concernant le client. Le script cgi doit se trouver dans `www/cgi-bin` et le formulaire dans `www`.
3. Testez la validation du formulaire par un POST ou un GET. Comment le tester dans le CGI, afficher le cas échéant la `QUERY_STRING`

Exercice 8. bonus

Concevez un formulaire permettant d'uploader une photo et un CGI qui la transforme en niveau de gris avant de l'afficher ? On pourra utiliser l'utilitaire `convert` d'Imagemagick.

Developpement Web Lic Pro Web et Mobile (TD n°1) 2018–2019

Exercice 9. Formulaires HTML5 et JS

Mettre en place un petit formulaire interrogeant une personne sur son identité, son adresse mail, son téléphone, les langues maitrisées, etc. avec des zones de texte, des listes déroulantes, radio et cases à cocher, et si possible un slider, une date. Utiliser un tableau et des checkboxes pour saisir la liste des langues parlées. On utilisera la méthode GET de soumission du formulaire. Expérimentez-là avec un bouton Submit. Utilisez les principaux nouveaux champs HTML5 comme *email*, *tel*, et des attributs comme *required* ou *pattern* pour effectuer des vérifications simples de validité des entrées. On effectuera les vérifications plus complexes (comme vérifier qu'une date est valide ou ne tombe pas un dimanche) en JavaScript.

Exercice 10. La réponse en PHP

Programmez à présent la réponse à ce formulaire en PHP. Faites d'abord un simple récapitulatif des données saisies par l'utilisateur. Dans un second temps procédez côté serveur à toutes vérifications déjà faites côté client. Utilisez pour cela des expressions régulières ou des filtres PHP. Le formulaire avec toutes ses vérifications côté client, un CSS agréable, si possible responsive et sa réponse en PHP avec toutes ses vérifications côté serveur est à rendre pour le 18 Septembre à votre enseignant de TD/TP via le dépôt Celene prévu à cet effet.