

# Les cookies en HTTP (Hyper Text Transfert Protocol)

# Rappel : Format générale des messages HTTP entre un client et un serveur

La syntaxe d'**une requête HTTP** est la suivante (<crLf> signifie retour chariot ou saut de ligne):

**METHODE URL VERSION<crLf>**

**EN-TETE : Valeur<crLf>**

**EN-TETE : Valeur<crLf>**

**Ligne vide<crLf>**

**CORPS DE LA REQUETE**

La syntaxe d'**une réponse HTTP** est la suivante :

**VERSION-HTTP CODE EXPLICATION<crLf>**

**EN-TETE : Valeur<crLf>**

**EN-TETE : Valeur<crLf>**

**Ligne vide<crLf>**

**CORPS DE LA REPONSE**

# Exemple de requête HTTP

- Format ASCII (texte lisible)

***GET /~berthet/ HTTP1.1***

***Host: etudiant.univ-mlv.fr***

***User-Agent: Mozilla/5.0(WindowsNT6.1; WOW64; rv:12.0)Firefox/12.0***

***Accept: text/html,application/xhtml+xml,***

***Accept-Language: fr,fr-fr,en-us,en***

***Accept-Encoding: gzip, deflate***

- La requête contient sur la première ligne :
  - la **méthode** à utiliser pour récupérer le document (**GET**) ;
  - Le **chemin** de l'objet demandé (**/~berthet**) ;
  - la **version** du protocole HTTP à invoquer (**HTTP1.1**).

Puis viennent les **en tête de la requête** sous la forme

***Variable : Valeur***

# Exemple de Réponse HTTP

- Le serveur répond à la requête en envoyant une **réponse HTTP** composée de trois parties.

**HTTP/1.1 200 OK**

**Date: Wed, 26 Sep 2012 13:40:13 GMT**

**Server: Apache**

**Last-Modified: Tue, 08 Sep 2009 12:07:55 GMT**

**Content-Length: 460**

**Content-Type: text/html**

...

**<HTML><HEAD>....**

- L'état de la réponse (**HTTP/1.1 200 OK**) est une ligne de texte avec la version HTTP, un code d'état décrivant le résultat (200 pour un acquittement...) et un texte.
- L'en-tête de réponse (**Date:..**) contient des informations relatives au type du serveur et une ligne vide annonçant les données.
- Les données (**<HTML>....**) HTTP généralement au format html.

# Les Méthodes de la requête :

Commande	Description
GET	Demande au serveur de renvoyer le contenu de l'information pointée par l'URL spécifiée dans la ligne de commande. Il peut s'agir d'un simple fichier HTML ou multimédia (image, son, ...), voire d'un programme CGI.
HEAD	Cette commande est similaire à la précédente mais ne renvoie que l'en-tête associé à la ressource demandée (par exemple, la date de dernière modification d'un fichier, ...).
POST	Permet au client d'envoyer des données au serveur, comme par exemple le contenu d'un formulaire renseigné par l'utilisateur.
DELETE	Suppression de la ressource située à l'URL spécifié (pas implémenté dans la plupart des serveurs)
PUT	Remplace la ressource visée par le contenu de la requête
TRACE	réalise un message de test aller/retour en suivant le chemin de la ressource visée.

# Les Cookies

- Le dialogue **Client-Serveur HTTP** a été prévu *sans état* ou *state-less*. Cela signifie que **le serveur ne stocke aucune information relative à une transaction (couple requête/réponse) avec un client**. On va donc stocker côté client des informations.
- Un cookie est **une information envoyée par le serveur**, stockée côté client et permettant d'établir des transactions avec état. Utilisable à partir d'HTTP1.1
- Le serveur place dans sa réponse, un entête appelé *Set-Cookie*, dont la syntaxe est la suivante :  
*Set-Cookie: Nom=Valeur; expires=Date; path=Chemin; domain=Domaine; secure ; http-only ;*
- En Javascript , on peut aussi stocker des données dans le navigateur des visiteurs avec l'une des deux APIs **Web Storage** ou **IndexedDB** (voir cours M Gambette)

# Champ Nom et valeur du cookie

- Permet d'envoyer au client **une valeur associée à un identifiant** afin que ces informations soient stockées en local, sous forme d'**une chaîne de caractères**.
- Cette chaîne peut contenir des caractères spéciaux, comme des espaces ou virgules, qui peuvent être reproduits tels quels s'il n'y a pas de risque d'ambiguïté ou codés suivant le système d'encodage des URL (%Valeur).
- Seul ce champ n'est pas facultatif dans une directive **Set-Cookie**.

*Exemple :*

*set-cookie : sessionId=259-7379481-4791100*

# Champ Expires

- Permet d'indiquer **la date d'expiration du cookie**, c'est-à-dire la date à partir de laquelle le client peut l'effacer.
- Si ce champ n'est pas précisé, le cookie sera effacé lorsque l'utilisateur quittera le navigateur.
- Un cookie avec un champs expire ou max-age, devient un cookie permanent, car il est conservé à la fermeture du navigateur.
- *! Les navigateurs web peuvent utiliser la restauration de session, ce qui fait de la plupart des cookies des cookies permanents, comme si le navigateur n'avait jamais été fermé.*



# Restrictions d'utilisation : Domain et Path

- Le champs « **Domain** », spécifie le nom de domaine d'application du cookie.
- **La valeur du cookie ne sera envoyée qu'aux serveurs appartenant au domaine précisé.**

*Exemple* : si le domaine est égal à `ibm.fr`, alors les machines `research.ibm.fr` et `sales.ibm.fr` (fictives) pourront accéder au cookie considéré.

- Si le domaine n'est pas spécifié, la valeur par défaut est l'adresse DNS de la machine ayant généré le cookie.

# Restrictions d'utilisation : Domain et Path

- Le champs « **Path** » est utilisé pour désigner les chemins de l'URL auxquelles le cookie est accessible.
- Si le champ *Domain* est renseigné, on concatène la valeur de ce champ à celle du *path*.

*Exemple* : si *path=/doc*, alors les ressources */doc/index.html*, */doc/toc.html*, etc. recevrons le cookie, à condition bien sûr que la valeur éventuelle du champ *Domain* corresponde à la machine considérée.

*Remarques* : la valeur *path=/* permet de spécifier tous les documents d'un serveur. Si *path* est omis, on suppose que sa valeur est celle du chemin d'accès

# Secure et HTTPOnly

- Si le champ « **Secure** » est mentionné le cookie ne sera envoyé que si la transmission s'effectue via une version sécurisée du protocole HTTP (HTTPS).
- Pour empêcher les attaques de cross-site scripting (XSS), on peut utiliser les cookies **HttpOnly**. Ceux ci sont alors inaccessibles à l'**API JavaScript** *Document.cookie*. Ils sont uniquement envoyés au serveur.
- Par exemple, les cookies qui persistent la session côté serveur n'ont pas besoin d'être accessibles via JavaScript, et l'option HttpOnly doit être définie.

# Exemple : première requête vers un serveur

En-têtes HTTP en direct

En-têtes   Générateur   Configuration   A propos

En-têtes HTTP

```

http://src-intranet.univ-mlv.fr/login/index.php

GET /login/index.php HTTP/1.1
Host: src-intranet.univ-mlv.fr
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; fr; rv:1.9.2.8) Gecko/20100722 Firefox/3.6.8 (.NET ...
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: fr,fr-fr;q=0.8,en-us;q=0.5,en;q=0.3
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 115
Connection: keep-alive

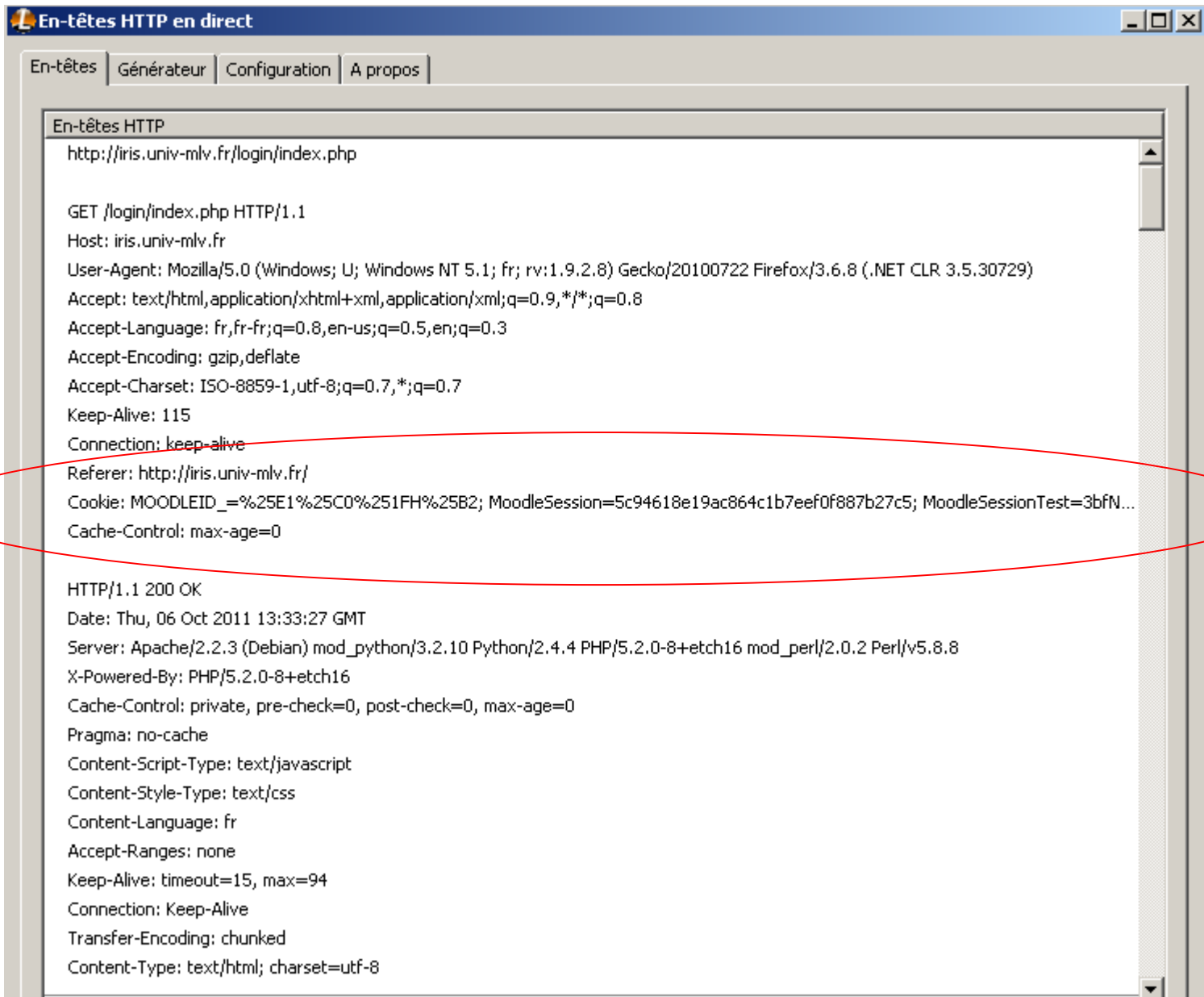
HTTP/1.1 200 OK
Date: Thu, 06 Oct 2011 10:37:46 GMT
Server: Apache/2.2.3 (Debian) mod_python/3.2.10 Python/2.4.4 PHP/5.2.0-8+etch16 mod_perl/2.0.2 Perl/v...
X-Powered-By: PHP/5.2.0-8+etch16
Set-Cookie: MoodleSession=afae515d501d117cbc3fbacc53f68299; path=/
Set-Cookie: MoodleSessionTest=VqLNGYJxQ0; path=/
Set-Cookie: MOODLEID_=deleted; expires=Wed, 06-Oct-2010 10:37:45 GMT; path=/
Set-Cookie: MOODLEID_=%25ED%25C3%251CC%25B7d; expires=Mon, 05-Dec-2011 10:37:46 GMT; path=/
Cache-Control: private, pre-check=0, post-check=0, max-age=0
Pragma: no-cache
Content-Script-Type: text/javascript
Content-Style-Type: text/css
Content-Language: fr
Accept-Ranges: none
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Transfer-Encoding: chunked
Content-Type: text/html; charset=utf-8
    
```

# Retour du cookie

- Avant d'envoyer une requête, le navigateur parcourt la liste des cookies qu'il possède.
- S'il y a occurrence entre l'URL de la ressource contenue dans la requête et les différents champs définissant le domaine d'application d'un cookie, la valeur de celui-ci est insérée dans la requête sous la forme d'une en tête.
- Si plusieurs cookies sont applicables, le client les renvoie sur une ligne suivant la syntaxe :

*Cookie: Nom1=valeur1; Nom2=valeur2; ...*

# Exemple : retour du cookie



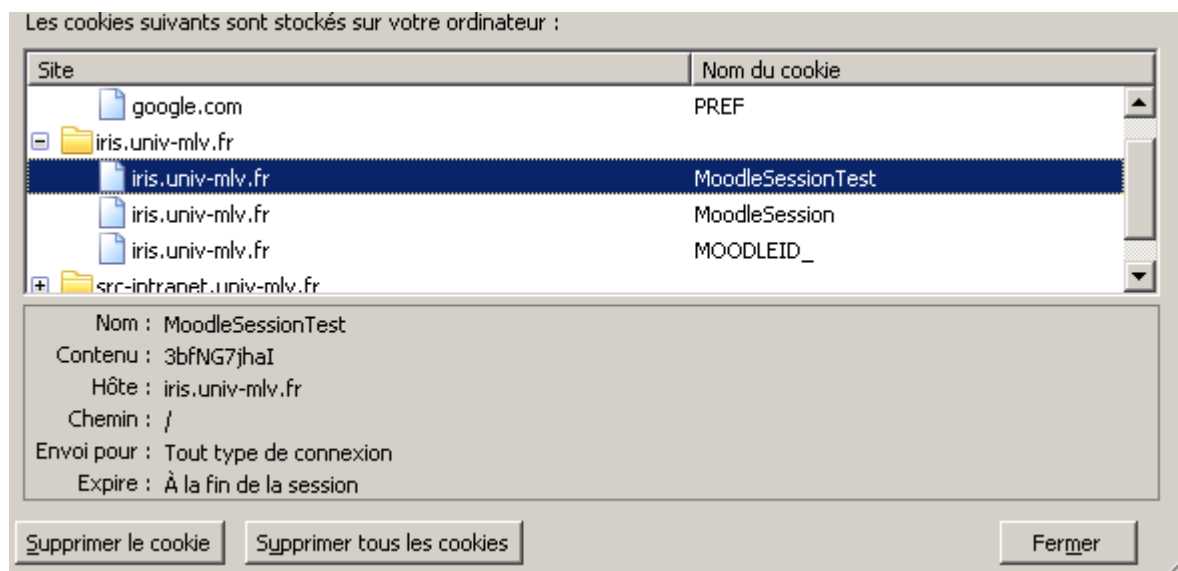
```
En-têtes HTTP
http://iris.univ-mlv.fr/login/index.php

GET /login/index.php HTTP/1.1
Host: iris.univ-mlv.fr
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; fr; rv:1.9.2.8) Gecko/20100722 Firefox/3.6.8 (.NET CLR 3.5.30729)
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: fr,fr-fr;q=0.8,en-us;q=0.5,en;q=0.3
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 115
Connection: keep-alive
Referer: http://iris.univ-mlv.fr/
Cookie: MOODLEID_=%25E1%25C0%251FH%25B2; MoodleSession=5c94618e19ac864c1b7eef0f887b27c5; MoodleSessionTest=3bfN...
Cache-Control: max-age=0

HTTP/1.1 200 OK
Date: Thu, 06 Oct 2011 13:33:27 GMT
Server: Apache/2.2.3 (Debian) mod_python/3.2.10 Python/2.4.4 PHP/5.2.0-8+etch16 mod_perl/2.0.2 Perl/v5.8.8
X-Powered-By: PHP/5.2.0-8+etch16
Cache-Control: private, pre-check=0, post-check=0, max-age=0
Pragma: no-cache
Content-Script-Type: text/javascript
Content-Style-Type: text/css
Content-Language: fr
Accept-Ranges: none
Keep-Alive: timeout=15, max=94
Connection: Keep-Alive
Transfer-Encoding: chunked
Content-Type: text/html; charset=utf-8
```

# Limitations et configuration

- Un navigateur ne peut recevoir plus de 300 cookies, de 4Ko maximum chacun avec une limite de 20 cookies pour un même domaine.
- Si le client a atteint cette limite de 300 cookies, il pourra effacer le cookie le plus ancien afin d'en stocker un nouveau.
- Il est possible d'effacer à la main les cookies stockés sur un client.
- Il est possible d'ignorer l'envoi des cookies par un serveur ou de les refuser systématiquement et automatiquement.



# Tracking-cookie

- Technique pour suivre un utilisateur sur **plusieurs sites partenaires**.
  - Les régies publicitaires peuvent ainsi vous afficher des publicités différentes sur chaque site partenaire
  - Elles peuvent aussi vous proposer des offres spéciales très ciblées par rapport aux articles que vous avez vu sur les sites partenaires
- Chaque site partenaire dépose ses propres cookies sur votre machine (**cookies tiers**).
- Les espaces publicitaires inclus dans les pages du partenaire déposent également des cookies
- En **recoupant leurs bases de données** la régie pub vous suit de site partenaire en site partenaire et connaît vos pages vues et donc vos habitudes de navigation.



# Exercice 1 : site Elearning

- Dans les paramètres de votre navigateur favori :
  - 1) Comment faites vous pour identifier les cookies du site `elearning.univ-eiffel.fr` ?
  - 2) Sous Firefox, ajoutez l'extension Cookie Quick Manager.
  - 3) Combien de cookies ce serveur a t'il déposé dans votre navigateur ?
  - 4) Pouvez vous identifier celui qui permet de suivre votre session ?
  - 5) Est il persistant ?
  - 6) D'autres serveurs peuvent ils le récupérer ?
  - 7) Quels sont les chemins sur le serveur qui peuvent y accéder ?

## Exercice 2 : Site Abritel

- Commencez par lire la déclaration relative aux cookies <https://www.abritel.fr/legal/cookies>
  - 1) Le terme Cookie utilisé dans la déclaration correspond il au Cookie HTTP que nous venons d'étudier ?
  - 2) Relever la liste des objectifs de l'utilisation de ces cookies
  - 3) Relever la liste des informations collectées qui permettent d'atteindre ces objectifs
  - 4) Quelles sont les 4 catégories de cookies gérées par Abritel (groupe expedia)
  - 5) Est on obligé d'accepter tous les cookies ?
  - 6) Lesquelles peuvent bloquer l'accès au site si on les refuse ?

# Exercice 3 : site Abritel

- Allez sur la page d'accueil : <https://abritel.fr>
  - 1) N'acceptez pas tous les cookies mais choisir de personnaliser. Quels sont les cookies « essentiels » ?
  - 2) Proviennent ils tous du site Expedia ?
  - 3) Quels autres sites (entreprises) vont alors laissés des cookies sur votre navigateur ?
  - 4) Qui sont les entreprises impliquées dans les cookies marketing ?
  - 5) Critéo ajoute beaucoup de cookies, en recherchant sur le site de cette entreprise, quel service apporte elle à Abritel ?
  - 6) Qui sont les entreprises qui déposent des cookies pour l'analyse du fonctionnement du site ?
  - 7) Quel est le but du cookie de Yandex.Metrica ? Vérifiez en allant voir leur solution sur leur site.

# Exercice 4 : Abritel

- Supprimez les cookies correspondant au site abritel.fr si vous en avez. Lancez l'analyseur réseau, désactivez le cache et connectez vous au site [www.abritel.fr](http://www.abritel.fr)
  - 1) Identifiez la première requête. Les cookies sont ils dans l'entête de la requête ou dans celui de la réponse ? Justifiez
  - 2) Vous n'avez pas encore accepté les cookies, vérifiez qu'ils ne sont pas encore enregistrés dans les paramètres de votre navigateur.
  - 3) Acceptez maintenant tous les cookies, parmi les pages html demandées, quels sites vous renvoient des cookies ? Dans quel but d'après vous ?
  - 4) Activez maintenant le cache, et rechargez la page, vous devez trouver des cookies dans les entête de requête.
  - 5) Retrouvez ces cookies dans le cache du navigateur