

Systemes d'Information

Méthodes Avancées (2)

Université Lyon 1
UFR informatique

L. Médini, mars 2008

Plan des cours

- ❑ Langages de description de l'information
- ❑ Protocole HTTP et programmation côté serveur
 - Rappels HTTP
 - Infrastructures des applications Web
 - ❑ Passage des arguments et des données (CGI, servlets...)
 - Programmation côté serveur en Java
- ❑ Architectures réparties (CORBA, RMI)
- ❑ Objets distribués (Javabeans, EJB)
- ❑ Web services (SOA, WSDL, SOAP, UDDI)
- ❑ Projet

Le protocole HTTP

▣ Place de http dans le modèle OSI

Application	HTTP
Présentation	...
Session	...
Transport	TCP
Réseau	IP
Liaison	...
Physique	...

Le protocole HTTP

- ❑ Orienté connexion
 - Sur TCP, maintien d'une connexion pour chaque envoi (requête, réponse)
 - Contrôle de la transmission
 - Connexion fermée une fois la transmission réussie
- ❑ Sans état
 - chaque page Web est transmise dans une connexion séparée (sauf pour les connexions persistantes)
 - Le protocole ne conserve pas l'historique des requêtes précédentes
- ❑ RFC
 - <http://www.w3.org/Protocols/rfc2616/rfc2616.html>
- ❑ Requêtes/réponses

Le protocole HTTP

□ Format d'une requête

<i>Méthode URL Version</i> <crLf>	GET http://www.monsite.com HTTP/1.0
<i>En-tête : Valeur</i> <crLf>	Accept : text/html
.	If-Modified-Since : Saturday, 15-January-2000 14:37:11 GMT
.	User-Agent : Mozilla/4.0 (compatible; MSIE 5.0; Windows 95)
<i>En-tête : Valeur</i> <crLf>	
<i>Ligne vide</i> <crLf>	
<i>Corps de la requête</i>	

Le protocole HTTP

□ Méthodes de requête HTTP

GET : permet d'obtenir des informations (document, graphique, le résultat d'une requête dans une base de données, ...) peut envoyer des informations dans l'URL limités en taille à 240 caractères sur certains serveurs (données d'un formulaire simple)

POST : permet de poster des informations (une données graphique, des données pour une base de données...) en quantité illimitée

HEAD : demande les en-têtes de la réponse ; seulement pour récupérer la taille des documents, l'heure de modification...

PUT et **DELETE** : agissent sur les documents du serveur (peu utilisées à cause des problèmes de sécurité)

TRACE : retourne le contenu exact de la demande (utilisée pour le débogage)

OPTIONS : pour demander au serveur les méthodes et les options qu'il supporte

Le protocole HTTP

□ Champs d'en-tête d'une requête HTTP

Accept : type MIME qu'un client sait gérer (application/pdf, image/jpeg...)

Accept-Charset : jeu de caractères (ISO-8859-1...)

Accept-Encoding : type de compression qui peut être décodée (gzip, compress...)

Accept-Language : fr par exemple

Authorization : accéder à des pages Web protégés par mot de pass

Connection : gestion des connexions http persistantes pour récupérer un fichier HTML et plusieurs pages associées en une seule connexion de socket (option par défaut en HTTP 1.1)

Content-Length : taille des données du corps d'une requête POST en octets

Content-type : utilisée par un client quand un document est attaché à POST

Cookie : pour renvoyer au serveur un cookie qu'il a envoyé

Le protocole HTTP

□ Champs d'en-tête d'une requête HTTP

Expect : demande d'acquittement au serveur d'un document attaché

From : adresse e-mail de la personne responsable de la requête (envoyée par les robots de recherche mais pas par les navigateurs)

Host : nom d'hôte en numéro de port du client

If-Match et **If-None-Match** : requêtes conditionnelles

If-Modified-Since : la page sera transmise au client uniquement si elle a été modifiée après une date spécifiée ; elle pourra ainsi être mise en mémoire cache par le navigateur (voir **getLastModified**)

If-Unmodified-Since : ...

Pragma : permet de spécifier les comportements des intermédiaires (demander à un serveur proxy de transmettre la requête même s'il dispose d'une copie locale, ou de ne pas mettre la réponse en cache)

Proxy-Authorization : authentifie un client auprès d'un serveur proxy

Le protocole HTTP

□ Champs d'en-tête d'une requête HTTP

Referer : indique l'URL de la page Web à partir de laquelle la requête a été émise (ex. : page contenant un lien)

Upgrade : spécifie un protocole autre que HTTP 1.1

User-Agent : identifie le navigateur ou le client

Via : défini par les passerelles et les serveurs proxy par lesquels la requête passe

...

Le protocole HTTP

▣ Format d'une réponse

Version-HTTP Code Message

En-Tête : Valeur

HTTP/1.0 200 OK

.

Date : Sat, 15 Jan 2000 14:37:12 GMT

En-Tête : Valeur

Content-Type : text/plain

Ligne vide

Last-Modified : Fri, 14 Jan 2000 08:25:13 GMT

Corps de la réponse

<html>

<head>

<title>La page qui dit bonjour</title>

</head>

<body>

<p>Coucou</p>

</body>

</html>

Le protocole HTTP

▣ Codes d'état d'une réponse HTTP

Les codes d'état sont classés en 5 catégories

- ▣ 100 à 199 : le client doit répondre avec une autre action
- ▣ 200 à 299 : la requête à réussi
- ▣ 300 à 399 : fichiers déplacés ayant une en-tête Location précisant leurs nouvelles adresses
- ▣ 400 à 499 : erreur au niveau du client
- ▣ 500 à 599 : erreur au niveau du serveur

Le protocole HTTP

□ Codes d'état d'une réponse HTTP

■ Exemples

200 (OK) : code envoyé par défaut

204 (No Content) : document non modifié ; demande au navigateur de continuer d'afficher le document précédent car aucun nouveau document n'est disponible

302 (Found) : le document demandé se trouve ailleurs ; possibilité de rediriger automatiquement le client

400 (Bad Request) : erreur de syntaxe dans la requête

403 (Forbidden) : l'accès à la ressource est interdit

404 (Not Found) : aucune ressource n'est présente à l'adresse demandée par le client

500 (Internal Server Error) : souvent dues à des erreurs de programmation côté serveur...

Le protocole HTTP

□ En-têtes d'une réponse HTTP

Cache-Control : définit une politique de cache pour la ressource
(no-cache, public...)

Content-Encoding : Type de codage du corps de la réponse

Content-Language : Type de langage du corps de la réponse

Content-Length : Longueur du corps de la réponse
(uniquement pour les connexions persistantes)

Content-Type : Type de contenu du corps de la réponse (type MIME) ; exemples :

application/msword ; application/octet-stream (données binaires ou inconnues) ; application/pdf ; application/x-gz

application/x-java-serialized-object

multipart/form-data

text/html ; text/plain (texte brut)

image/jpeg ; video/mpeg

Le protocole HTTP

□ En-têtes d'une réponse HTTP

Date : Date de début de transfert des données

Expires : Date limite de consommation des données

Forwarded : Utilisé par les machines intermédiaires entre le browser et le serveur

Location : Redirection vers une nouvelle URL associée au document

Server : Caractéristiques du serveur ayant envoyé la réponse

Programmation côté serveur (généralités)

□ Applications web

■ Définition

- [Ensembles de] programmes fonctionnant côté serveur
 - Peuvent être réparties sur plusieurs machines
 - Fonctionnent selon un choix d'infrastructure

■ Principe

- Récupération et traitement de requêtes HTTP
 - Récupération des paramètres éventuels
- Génération dynamique de contenus
 - Pages web,
 - Fichiers texte,
 - Images...
- Renvoi des contenus à travers le serveur web

■ Exemples d'applications courantes

- Moteur de recherche,
- Webmail,
- Commerce électronique...

Programmation côté serveur (généralités)

□ Applications web

■ Exemples de technologies sous-jacentes

- CGI (Common Gateway Interface) : interface de programmation entre un serveur web et un module logiciel

- Passage des données de la requête aux programmes,
- Renvoi des contenus générés (et du type MIME) au serveur.

- Infrastructures middleware (CORBA, RMI, J2EE, .Net...) : protocoles de communication entre modules logiciels

- Traitements réalisés dans plusieurs modules logiciels,
- Répartition de ces modules sur plusieurs machines,
- Utilisation de langages éventuellement hétérogènes.

- Services web : agrégation d'applications web hétérogènes

- protocoles de découverte et de description des fonctionnalités des applications web réparties,
- Protocoles de communication entre ces applications,
- Permettent la composition dynamique de services en fonction des besoins du client

Programmation côté serveur (généralités)

□ Applications web

■ Common Gateway Interface (CGI)

□ Principes

- Permet d'exécuter des applications sur un serveur pour le compte d'un client (« Passerelle » entre le Web et une machine)
- Indépendante du langage de programmation (Perl, C/C++ + Java...)
- Utilise des variables d'environnement pour les échanges avec le serveur Web (CONTENT_TYPE, QUERY_STRING...)

□ Problèmes

- Surcharge de la machine (création d'un process pour chaque requête)
- Sécurité (protection des données, restriction d'accès...)

Programmation côté serveur (généralités)

□ Applications web

■ Common Gateway Interface (CGI)

□ Technique

- Interface CGI accessible depuis un répertoire spécifique (habituellement : /cgi-bin pour les serveurs Apache)
- Récupération des paramètres d'une requête HTTP
 - Méthode GET : variable QUERY_STRING
 - Méthode POST : entrée standard (STDIN)
- Envoi du résultat au serveur sur la sortie standard (STDOUT)
 - précédés des en-têtes HTTP (par exemple, le type de contenu : « content-type : text/html »)
 - suivis de deux sauts de ligne (une ligne blanche les sépare du contenu de la réponse)
- La sortie standard doit être fermée à la fin de la réponse.

Programmation côté serveur (généralités)

- Applications web
 - Common Gateway Interface (CGI)
 - Exemple de script CGI en Perl

```
#!/usr/local/bin/perl
```

```
print "content-type: text/html\n\n";
```

```
print "<html>\n";
```

```
print "<head><title>Variables d'environnement</title></head>\n";
```

```
print "<body>\n";
```

```
print "Voici l'adresse du client : $ENV{$REMOTE_ADDR} <br />\n";
```

```
print "Voici la requete recue : $ENV{$QUERY_STRING} <br />\n";
```

```
print "</body>";
```

```
print "</html>";
```

Programmation côté serveur (généralités)

□ Applications web

■ Outils et plateformes de développement

□ Java

- serveur : Tomcat (container de servlets)
- programmes : servlets, JSP...

□ Microsoft™ .Net Framework

- Serveur : IIS
- Langages VB, C#...

□ Python

- Serveur d'applications : Zope, Plone
- Programmes : scripts python, scriptlets, DTML...

□ Php

- Langage de scripts
- S'installe sur la quasi-totalité des serveurs

Programmation côté serveur (généralités)

- ❑ Récupération des données côté serveur
 - Quelle que soit l'infrastructure choisie, il existe des méthodes permettant à l'application Web d'accéder
 - ❑ À la requête, pour récupérer les paramètres
 - ❑ À la réponse, pour la construire dynamiquement
 - D'autres types d'accès sont fonction du type d'application
 - ❑ Contexte (applications Web : Java, PHP...)
 - ❑ Primitives systèmes (shell Unix)
 - ❑ Bases de données...
 - Exemples (Java)
 - ❑ `String getParameter(String name)`
 - ❑ `String [] getParameterValues(String name)`
 - ❑ `Enumeration getParameterNames()`

Programmation côté serveur (généralités)

❑ Le suivi de session HTTP

- HTTP est un protocole sans état : déconnexion TCP à chaque requête/réponse
 - ❑ Le serveur peut avoir besoin de garder la trace des interactions précédentes avec le client
 - ➔ Les sessions sont simulées par le serveur en envoyant un identifiant qui est renvoyé par le client à chaque requête
- 3 possibilités
 - ❑ Champ input HIDDEN qui contient l'identifiant de la session (dans un formulaire HTML)
 - ❑ Ré-écriture d'URL : l'identifiant est rajouté à la fin de chaque URL (méthode GET)
 - ❑ Cookies : information positionnée par le serveur sur le client puis renvoyée par le client à chaque requête et dont la durée de vie dépasse la session
- Stockage au niveau du serveur
 - ❑ Objets HttpSession (Java) ou Session (ASP)

Programmation côté serveur (Java)

- Les outils de programmation Java serveur
 - Tomcat
 - Contenu
 - Serveur web (Apache)
 - Moteur de servlets/JSP (Catalina)
 - Avantages
 - Installation/configuration simple
 - « XML compliant »
 - Standard (fourni avec l'EDI NetBeans)
 - Extensible (sert de base à J2EE)
 - Gratuit...
 - Packages Java
 - javax.servlet
 - javax.servlet.http
 - javax.servlet.jsp
 - javax.servlet.jsp.tagext

Programmation côté serveur (Java)

□ Les outils de programmation Java serveur

■ Les servlets

□ Définition

(<http://java.sun.com/products/servlet/whitepaper.html>)

- Servlets are protocol- and platform-independent server side components, written in Java, which dynamically extend Java enabled servers.
- They provide a general framework for services built using the request-response paradigm.
- Their initial use is to provide secure web-based access to data which is presented using HTML web pages, interactively viewing or modifying that data using dynamic web page generation techniques.
- Since servlets run inside servers, they do not need a graphical user interface.

Programmation côté serveur (Java)

□ Les outils de programmation Java serveur

■ Les servlets

□ Concrètement

- Objets (classes) Java
 - Composants d'application (« petits » programmes)
 - Derrière un serveur (Web, mais pas seulement)
- Dans un « Container »
 - Pas d'accès direct au serveur
 - Accès protégé aux autres objets métier de l'application
- Encapsulation standardisée des données
 - Requêtes
 - Réponses
 - Session
 - Application web (« contexte »)

Programmation côté serveur (Java)

- Les outils de programmation Java serveur

- Les servlets

- Exemple de code

```
import javax.servlet.*;
```

```
import javax.servlet.http.*;
```

```
public class NewServlet extends HttpServlet {
```

```
    public void init(ServletConfig config) throws ServletException {  
        super.init(config); ... }  
    public void destroy() { ... }
```

```
    protected void doGet(HttpServletRequest request, HttpServletResponse response)  
        throws ServletException, IOException {  
        response.setContentType("text/html");  
        PrintWriter out = response.getWriter();  
        out.println("<html><head><title>Hello World!</title></head>");  
        out.println("<body><h1>Hello World!</h1></body></html>");  
    }
```

```
    protected void doPost(HttpServletRequest request, HttpServletResponse  
        response) throws ServletException, IOException { ... }  
}
```

Programmation côté serveur (Java)

□ Les outils de programmation Java serveur

■ Les Java Server Pages (JSP)

□ Les JSP permettent de

- « Mélanger » du code HTML statique et du code dynamiquement généré par le serveur
- Écrire une page HTML comme si elle était statique, et de n mettre du code que quand nécessaire (scripts)
- Accéder aux même données/objets qu'une servlet
- Inclure une autre JSP/servlet
- Définir des balises personnalisées

- Remarque : une JSP est compilée en servlet à la première utilisation

Programmation côté serveur (Java)

- Les outils de programmation Java serveur
 - Les Java Server Pages (JSP)
 - Syntaxe des balises de scripts : `<% code %>`
 - Trois types de scripts
 - Des expressions, qui sont évaluées et insérées
`<%= new java.util.date() %>`
<jsp:expression>
new java.util.date()
</jsp:expression>

Programmation côté serveur (Java)

- Les outils de programmation Java serveur
 - Les Java Server Pages (JSP)
 - Syntaxe des balises de scripts : `<% code %>`
 - Trois types de scripts
 - Scriptlets : morceaux de code Java dans la page
`<% response.setContentType("text/plain"); %>`
<jsp:scriptlet>
`response.setContentType("text/plain");`
</jsp:scriptlet>

Programmation côté serveur (Java)

▣ Les outils de programmation Java serveur

■ Les Java Server Pages (JSP)

▣ Syntaxe des balises de scripts : `<% code %>`

▪ Trois types de scripts

- Déclarations : permettent de définir des méthodes ou des champs qui seront insérés dans le corps de la servlet

```
<%! private int VariableGlobale = 0; %>
```

```
<jsp:declaration>
```

```
private int VariableGlobale = 0;
```

```
</jsp:declaration>
```

Programmation côté serveur (Java)

- Les outils de programmation Java serveur
 - Les Java Server Pages (JSP)
 - Variables prédéfinies dans les scripts
 - **request**
 - **response**
 - **out**
 - **session**
 - **application**
 - **page**
 - ...

Programmation côté serveur (Java)

□ Les outils de programmation Java serveur

■ Les Java Server Pages (JSP)

□ Syntaxe des balises de directives : `<%@ code %>`

▪ Trois types de directives

- **page** : permet de modifier les données de la page (import de packages, spécification d'un type de contenu, gestion des sessions)

```
<%@ page import="java.util.*" %>
```

- **include** : permet d'inclure des fichiers ou autres servlets/JSP,

```
<%@ include page="/monJSP.jsp" flush="true" %>
```

- **taglib** : permet d'utiliser des bibliothèques de balises personnalisées

```
<%@ taglib uri="..." prefix="..." %>
```

Programmation côté serveur (Java)

▣ Les outils de programmation Java serveur

■ Les Java Server Pages (JSP)

- ▣ Exemple de code simple : une JSP qui compte le nombre de fois où elle a été appelée

```
<html>
```

```
  <head><title>Déclarations et expressions</title></head>
```

```
  <body>
```

```
    <h1>Déclarations JSP</h1>
```

```
    <%! private int accessCount = 0; %>
```

```
    <p>Cette page a été accédée <%= ++accessCount %>  
    fois depuis le démarrage du serveur</p>
```

```
  </body>
```

```
</html>
```