

Java /internet

URL

- ❑ <http://www.bib.org:80/javafaq/book/index.html?isbn=56888#toc>
- ❑ `ftp://anonymous:anonymous@ftp.bib.org/c%3a/stuff`
 - ❖ protocol
 - ❖ port
 - ❖ authority
 - ❖ chemin
 - ❖ référence
 - ❖ requête
 - ❖ user

Construire une URL

- `public URL(String url) throws MalformedURLException`
- `public URL(String protocol, String hostname, String file) throws MalformedURLException`
- `public URL(String protocol, String host, int port, String file) throws MalformedURLException`
- `public URL(URL base, String relative) throws MalformedURLException`

Remarques

- ❑ tous les protocoles ne sont pas compris par la machine virtuelle...
- ❑ exception `MalformedURLException`

Exemple: Quels protocoles?

```
import java.net.*;
public class ProtocolTester {
    public static void main(String[] args) {
        testProtocol("http://www.adc.org");
        testProtocol("https://www.amazon.com/exec/obidos/order2/");
        testProtocol("ftp://metalab.unc.edu/pub/languages/java/javafaq/");
        testProtocol("mailto:elharo@metalab.unc.edu");
        testProtocol("telnet://dibner.poly.edu/");
        testProtocol("file:///etc/passwd");
        testProtocol("gopher://gopher.anc.org.za/");
        testProtocol("ldap://ldap.itd.umich.edu/o=University%20of%20Michigan,c=US?postalAddress");
        testProtocol("jar:http://cafeaulait.org/books/javaio/ioexamples/javaio.jar!"
            +"/com/macfaq/io/StreamCopier.class");
        testProtocol("nfs://utopia.poly.edu/usr/tmp/");
        testProtocol("jdbc:mysql://luna.metalab.unc.edu:3306/NEWS");
        testProtocol("rmi://metalab.unc.edu/RenderEngine");
        testProtocol("doc:/UsersGuide/release.html");
        testProtocol("netdoc:/UsersGuide/release.html");
        testProtocol("systemresource://www.adc.org/+/index.html");
        testProtocol("verbatim:http://www.adc.org/");
    }
}
```

Quels protocoles (suite)

```
private static void testProtocol(String url) {  
  
    try {  
        URL u = new URL(url);  
        System.out.println(u.getProtocol() +  
            " is supported");  
    }  
    catch (MalformedURLException ex) {  
        String protocol = url.substring(0,url.indexOf(':'));  
        System.out.println(protocol + " is not supported");  
    }  
  
}
```

```
http is supported  
https is supported  
ftp is supported  
mailto is supported  
telnet is not supported  
file is supported  
gopher is supported  
ldap is not supported  
jar is supported  
nfs is not supported  
jdbc is not supported  
rmi is not supported  
doc is supported  
netdoc is supported  
systemresource is supported  
verbatim is supported
```

Décomposer l'url

```
public static void decomposeURL(String st) {
    try{
        URL u = new URL(st);
        System.out.println("URL :" + u);
        System.out.println("Protocole : " + u.getProtocol( ));
        System.out.println("UserInfo :" + u.getUserInfo( ));
        String host = u.getHost( );
        if (host != null) {
            int atSign = host.indexOf('@');
            if (atSign != -1) host = host.substring(atSign+1);
            System.out.println("hôte " + host);
        }
        else {
            System.out.println("hôte null.");
        }
    }
}
```

suite

```
        System.out.println("Port :" + u.getPort( ));
        System.out.println("Path : " + u.getPath( ));
        System.out.println("Ref : " + u.getRef( ));
        System.out.println("Query: " + u.getQuery( ));
    }
    catch (MalformedURLException ex)
        {System.err.println(st + " URL malformée"); }
}
```


Résultat

[http://www.bib.org:80/javafaq/book/index.html?
isbn=56888#toc](http://www.bib.org:80/javafaq/book/index.html?isbn=56888#toc)

URL : [http://www.bib.org:80/javafaq/book/
index.html?isbn=56888#toc](http://www.bib.org:80/javafaq/book/index.html?isbn=56888#toc)

Protocole : http

UserInfo : null

hôte www.bib.org

Port : 80

Path : /javafaq/book/index.html

Ref : toc

Query: isbn=56888

Obtenir des données

- ❑ `public InputStream openStream() throws IOException`
- ❑ `public URLConnection openConnection() throws IOException`
- ❑ `public URLConnection openConnection(Proxy proxy) throws IOException`
- ❑ `public Object getContent() throws IOException`
- ❑ `public Object getContent(Class[] classes) throws IOException`

Charger le contenu d'une URL

```
public static void chargerFichier(String url) {
    try {
        URL u = new URL(url);
        InputStream in = u.openStream();
        in = new BufferedInputStream(in);
        Reader r = new InputStreamReader(in);
        int c;
        while ((c = r.read()) != -1) {
            System.out.print((char) c);
        }
    } catch (MalformedURLException ex) {
        System.err.println(url + " mauvaise URL");
    } catch (IOException e) {
        System.err.println(e + " problème de lecture url "
            +url);
    }
}
```

Type du contenu

```
public static void typeObjetURL(String st) {  
    try {  
        URL u = new URL(st);  
        try {  
            Object o = u.getContent();  
            System.out.println("L'objet est un " +  
                               o.getClass().getName());  
        } catch (IOException ex) {System.err.println(ex);}  
    } catch (MalformedURLException ex) {  
        System.err.println(st + " URL malformée");  
    }  
}
```

`http://www.liafa.jussieu.fr`

```
L'objet est un sun.net.www.protocol.http.HttpURLConnection  
$HttpInputStream
```

Et encore

- ❑ URLEncoder
- ❑ URLDecoder

URLEncode: Example

```
public class QueryString {
    private StringBuffer query = new StringBuffer();
    public QueryString(String name, String value) {
        encode(name, value);
    }
    public synchronized void add(String name, String value) {
        query.append('&');
        encode(name, value);
    }
    private synchronized void encode(String name, String value){
        try {
            query.append(URLEncoder.encode(name, "UTF-8"));
            query.append('=');
            query.append(URLEncoder.encode(value, "UTF-8"));
        }
        catch (UnsupportedEncodingException ex) {
            throw new RuntimeException("??");
        }
    }
    //...
}
```

Exemple (suite)

```
 /...  
 public String getQuery() {  
     return query.toString();  
 }  
 public String toString() {  
     return getQuery();  
 }  
 }
```

Authentication

- Classe (abstraite) Authenticator
 - ❖ PasswordAuthentication représente un couple password + user
 - ❖ Méthode getPasswordAuthentication() à redéfinir pour obtenir un PasswordAuthenitcation
 - ❖ Méthode setDefault(Authenticator) définit l'Authenticator pour le système
 - C'est à lui que s'adressera le système à chaque fois que nécessaire

Pour apache:

- Fichier `.htaccess`: (il faut aussi un `AllowOverride AuthConfig` dans `httpd.conf`)

```
AuthType Basic
```

```
AuthName "restreint"
```

```
AuthUserFile /Users/hf/sites/.psswd
```

```
Require user hf
```

- ❖ `.psswd` est le fichier des mots de passe
- ❖ `htpasswd -c /Users/hf/sites/.psswd hf`

Exemple

html et swing

- plusieurs packages permettent de visualiser et travailler avec html
 - ❖ swing peut utiliser html

« compiler » html

- Des classes internes de [HTMLEditorKit](#) permettent de « parser » du html:
 - ❖ [HTMLEditorKit.ParserCallback](#) pour définir les actions (callback) au cours du « parsing » du texte
 - ❖ [HTMLEditorKit.parser](#) définit la méthode `parse(Reader r, HTMLEditorKit.ParserCallback cb, boolean ignoreCharSet)`
 - ❖ [HTMLEditorKit.LinkController](#) souris sur les hyperliens.

ParserCallback

```
class UrdansPage extends HTMLToolkit.ParserCallback {
    private Writer out;
    boolean silence = true;
    public UrdansPage(Writer out) { this.out = out; }
    public void handleText(char[] text, int position) {
        try {
            if (!silence) out.write(text);
        } catch (IOException e) {}
    }
    public void handleStartTag(HTML.Tag tag, MutableAttributeSet at, int p){
        try { if (tag == HTML.Tag.A) {
            silence = false;
            out.write("url:" + at.getAttribute(HTML.Attribute.HREF) + " \"); }
        } catch (IOException e) {}
    }
}
```

Suite

```
public void handleEndTag(HTML.Tag tag, int position) {
    try {
        if (tag == HTML.Tag.A) {
            silence = true;
            out.write('\n' + "\n");
        }
    } catch (IOException e) {
        System.err.println(e);
    }
}

public void handleSimpleTag(HTML.Tag tag, MutableAttributeSet
    attributes, int position) {}
```

fin

```
public static void main(String[] args) {
    ParserGetter kit = new ParserGetter();
    HTMLEditorKit.Parser parser = kit.getParser();
    HTMLEditorKit.ParserCallback callback = new UrlDansPage(new
        OutputStreamWriter(System.out));

    try {
        URL u = new URL("http://www.liafa.jussieu.fr/");
        InputStream in = u.openStream();
        InputStreamReader r = new InputStreamReader(in);
        parser.parse(r, callback, true);
    } catch (IOException e) {
        System.err.println(e);
    }
}
```

Swing

- le texte de la plupart des composants utilisant du texte de Swing (labels, buttons, menu items, ...) peut être du HTML

Exemple:

```
import javax.swing.*;
public class HTMLLabelApplet extends JApplet {
    public void init() {
        JLabel theText = new JLabel(
            "<html>Voilà un exemple de HTML dans label <b>bold</b> "
            + "en <i>italic</i> ici. <P> "
            + "avec des paragraphes, des nouvelles lignes,<hr> "
            + "<font color=red>couleurs</font> "
            + "et plus généralement les constructions htmln</html>");
        this.getContentPane().add(theText);
    }
}
```


JEditorPane

- ❑ JEditorPane contient une implémentation de HTML :
- ❑ constructeurs:
 - ❖ [JEditorPane](#) ()
 - ❖ [JEditorPane](#) ([String](#) url)
 - ❖ [JEditorPane](#) ([String](#) type, [String](#) text)
 - ❖ [JEditorPane](#) ([URL](#) initialPage)
- ❑ méthodes
 - ❖ public void setPage([URL](#) page) throws [IOException](#)
 - ❖ public void setPage([String](#) url) throws [IOException](#)
 - ❖ public void setText([String](#) html)

Exemple

```
import java.io.IOException;
import javax.swing.JEditorPane;
import javax.swing.JFrame;
import javax.swing.JScrollPane;
import javax.swing.WindowConstants;
public class BrowserMinimal {
    public BrowserMinimal(String st) {
        JEditorPane jep = new JEditorPane();
        jep.setEditable(false);
        try {
            jep.setPage(st);
        }
        catch (IOException ex) {
            jep.setContentType("text/html");
            jep.setText("<html>impossible de charger "+st+" </html>");
        }
        JScrollPane scrollPane = new JScrollPane(jep);
        JFrame f = new JFrame("exemple");
        f.setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);
        f.setContentPane(scrollPane);
        f.setSize(512, 342);
        f.setVisible(true);}
}
```

Hyperliens

- Pour manipuler les hyperliens:
 - ❖ HyperlinkEvent
 - ❖ HyperlinkListener
 - avec une seule méthode
 - `public hyperlinkUpdate(HyperlinkEvent e);`

Exemple

```
import javax.swing.JEditorPane;
import javax.swing.event.HyperlinkEvent;
import javax.swing.event.HyperlinkListener;
public class SuivreLien implements HyperlinkListener{
    private JEditorPane pane;
    public SuivreLien(JEditorPane pane) {
        this.pane = pane;
    }
    public void hyperlinkUpdate(HyperlinkEvent evt) {
        if (evt.getEventType() ==
HyperlinkEvent.EventType.ACTIVATED) {
            try {
                pane.setPage(evt.getURL());
            }
            catch (Exception ex) {
                pane.setText("<html>impossible de trouver " +
evt.getURL() + "</html>");
            }
        }
    }
}
```

Exemple SimpleBrowser

```
import java.awt.EventQueue;
import java.awt.Frame;
import java.io.IOException;
import javax.swing.JEditorPane;
import javax.swing.JFrame;
import javax.swing.JScrollPane;
import javax.swing.WindowConstants;
public class SimpleBrowser {
    public SimpleBrowser(String initialPage) {
        JEditorPane jep = new JEditorPane();
        jep.setEditable(false);
        jep.addHyperlinkListener(new SuivreLien(jep));
        try {
            jep.setPage(initialPage);
        } catch (IOException ex) {
            jep.setContentType("text/html");
            jep.setText("<html>impossible de charger"
+initialPage+" </html>");
            ex.printStackTrace();
        }
    }
}
```

Exemple (suite)

```
JScrollPane scrollPane = new JScrollPane(jep);
JFrame f = new JFrame("Un Browser Simple");
f.setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);
f.setContentPane(scrollPane);
f.setSize(512, 342);
EventQueue.invokeLater(new FrameShower(f));
}
private static class FrameShower implements Runnable {
    private final Frame frame;
    FrameShower(Frame frame) {
        this.frame = frame;
    }
    public void run() {
        frame.setVisible(true);
    }
}
}
```

URLConnection

- ❑ URLConnection est une classe abstraite qui représente une connexion active spécifiée par une URL
- ❑ Principe:
 - ❖ construire un objet URL
 - ❖ invoquer `openConnection()` de cet objet URL: retourne un objet `URLConnection`
 - ❖ configurer la connexion
 - ❖ lire les "headers"
 - ❖ construire une input stream et lire les données
 - ❖ construire une output stream et écrire les données
 - ❖ fermer la connexion

Méthodes

- `setAllowUserInteraction` autorise/interdit l'interaction avec un utilisateur
- `setDoInput` autorise/interdit l'entrée
- `setDoOutput` autorise/interdit la sortie
- `setIfModifiedSince`
- `setUseCaches`
- `getContent`
- `getHeaderField`
- `getInputStream`
- `getOutputStream`
- `getContentEncoding`
- `getContentLength`
- `getContentType`
- `getDate`
- `getExpiration`
- `getLastModified`

Ouverture d'une page:

```
import java.net.*;
import java.io.*;
public class ConnexionURL {
    public static void main(String[] args) {
        BufferedReader Entree = new BufferedReader(new InputStreamReader(System.in));
        String url;
        try {
            while ((url = Entree.readLine()) != null) {
                URL u = new URL(url);
                URLConnection uc = u.openConnection();
                InputStream raw = uc.getInputStream();
                InputStream buffer = new BufferedInputStream(raw);
                Reader r = new InputStreamReader(buffer);
                int c;
                while ((c = r.read()) != -1) {System.out.print((char) c);}
            }
        } catch (MalformedURLException ex) { System.err.println(url + " URL
        Malformée");
        } catch (IOException ex) {System.err.println(ex);
        }
    }
}
```

Ouverture d'une page (avec codage correct)

```
public class SourceViewer3 {
    public static void main (String[] args) {
        for (int i = 0; i < args.length; i++) {
            try {
                // set default encoding
                String encoding = "ISO-8859-1";
                URL u = new URL(args[i]);
                URLConnection uc = u.openConnection();
                String contentType = uc.getContentType();
                int encodingStart = contentType.indexOf("charset=");
                if (encodingStart != -1) {
                    encoding = contentType.substring(encodingStart+8);
                }
                InputStream in = new BufferedInputStream(uc.getInputStream());
                Reader r = new InputStreamReader(in, encoding);
                /...
            }
        }
    }
}
```

```

//...
    int c;
    while ((c = r.read()) != -1) {
        System.out.print((char) c);
    }
}
catch (MalformedURLException ex) {
    System.err.println(args[0] + " URL?");
}
catch (IOException ex) {
    System.err.println(ex);
}
} // end if
} // end main
} // end SourceViewer3

```

Date dernière modification

```
public class DerniereModif {
    public static void main(String args[]) {
        for (int i=0; i < args.length; i++) {
            try {
                URL u = new URL(args[i]);
                HttpURLConnection http=(HttpURLConnection)u.openConnection();
                http.setRequestMethod("HEAD");
                System.out.println(u + "a été modifiée "
                    + new Date(http.getLastModified()));
            } // end try
            catch (MalformedURLException ex) {
                System.err.println(args[i] + " URL??");
            }
            catch (IOException ex) {
                System.err.println(ex);
            }
        } // end for
    } // end main
} // end DernierModif
```

Lire les headers

```
public class HeaderViewer {
    public static void main(String args[]) {
        for (int i=0; i < args.length; i++) {
            try {
                URL u = new URL(args[i]);
                URLConnection uc = u.openConnection();
                System.out.println("Content-type: " + uc.getContentType());
                System.out.println("Content-encoding: "
                    + uc.getContentEncoding());
                System.out.println("Date: " + new Date(uc.getDate()));
                System.out.println("Last modified: "
                    + new Date(uc.getLastModified()));
                System.out.println("Expiration date: "
                    + new Date(uc.getExpiration()));
                System.out.println("Content-length: " + uc.getContentLength());
            } // end try
        }
    }
}
```

(Suite)

```
//...
catch (MalformedURLException ex) {
    System.err.println(args[i] + "URL?? ");
}
catch (IOException ex) {
    System.err.println(ex);
}
System.out.println();
} // end for
} // end main
} // end HeaderViewer
```

Afficher le header

```
public class afficheHeaders {
    public static void main(String args[]) {
        for (int i=0; i < args.length; i++) {
            try {
                URL u = new URL(args[i]);
                URLConnection uc = u.openConnection();
                for (int j = 1; ; j++) {
                    String header = uc.getHeaderField(j);
                    if (header == null) break;
                    System.out.println(uc.getHeaderFieldKey(j) + ": " + header);
                } // end for
            } // end try
            catch (MalformedURLException ex) {
                System.err.println(args[i] + "URL ???");
            }
            catch (IOException ex) {
                System.err.println(ex);
            }
            System.out.println();
        } // end for
    } // end main
} // end afficheHeaders
```

Protocoles

- ❑ À partir d'une url, `openConnection()` permet d'obtenir une `URLConnection`
 - ❖ La classe concrète correspondante est déterminée par la partie protocole (exemple `http`) de l'url.
 - ❖ Protocole: interactions avec le client, génération des requêtes, interprétation des headers etc.
 - ❖ Content: conversion des données dans un format java

Protocoles

□ Classes

- ❖ URL
- ❖ URLConnection (abstraite)
- ❖ URLStreamHandler (abstraite)
- ❖ URLStreamHandlerFactory (interface)
- ❖ Pour un nouveau protocole:
 - Écrire une sous classe concrète URLConnection, une sous classe concrète URLStreamHandler
 - Pour utiliser ce protocole: implémenter URLStreamHnadlerFactory

Protocoles...

- À partir d'une url extraction de la partie protocole (exemple mailto)
 - ❖ transmettre à URLStreamHandlerFactory
 - ❖ qui détermine l'URLStreamHandler correspondant au protocole

Protocole...

- ❑ Construction d'un objet URL
- ❑ Extraction partie protocole
- ❑ Le constructeur de l'URL détermine l'URLStreamHandler:
 - ❖ Dans le cache
 - ❖ Si un URLStreamHandlerFactory est installé lui demander
 - ❖ Essayer plusieurs instanciations (...)
 - ❖ Si échec MalformedURLException
- ❑ Appel openConnection()
- ❑ Demander à l'URLStreamHandler de retourner une URLConnection adaptée (IOException)
- ❑ Interagir avec la ressource distante par cette URLConnection

Exemple (finger)

```
% telnet rama.poly.edu 79
```

```
Trying 128.238.10.212...
```

```
Connected to rama.poly.edu.
```

```
Escape character is '^]'.
```

Login	Name	TTY	Idle	When	Where
jacola	Jane Colaginae	*pts/7		Tue 08:01	208.34.37.104
marcus	Marcus Tullius	pts/15	13d	Tue 17:33	farm-dialup11.poly.e
matewan	Sepin Matewan	*pts/17	17:	Thu 15:32	128.238.10.177
hengpi	Heng Pin	*pts/10		Tue 10:36	128.238.18.119
nadats	Nabeel Datsun	pts/12	56	Mon 10:38	128.238.213.227

FingerURLConnection

```
public class FingerURLConnection extends URLConnection {
    private Socket connection = null;
    public final static int DEFAULT_PORT = 79;
    public FingerURLConnection(URL u) {
        super(u);
    }
    public synchronized InputStream getInputStream( ) throws IOException {
        if (!connected) this.connect( );
        InputStream in = this.connection.getInputStream( );
        return in;
    }
    public String getContentType( ) {
        return "text/plain";
    }
    //...
```

(Suite)

```
public synchronized void connect( ) throws IOException {
    if (!connected) { int port = DEFAULT_PORT; }
    this.connection = new Socket(url.getHost( ), port);
    OutputStream out = this.connection.getOutputStream( );
    String names = url.getFile( );
    if (names != null && !names.equals("")) {
        names = names.substring(1);
        names = URLDecoder.decode(names);
        byte[] result;
        try {
            result = names.getBytes("ASCII");
        }
        catch (UnsupportedEncodingException ex) {
            result = names.getBytes( );
        }
        out.write(result);
    }
    out.write('\r'); out.write('\n');out.flush( ); this.connected = true;
}
}
```

Handler

```
import java.net.*;
import java.io.*;
public class Handler extends URLStreamHandler {
    public int getDefaultPort( ) {
        return 79;
    }
    protected URLConnection openConnection(URL u) throws
        IOException {
        return new FingerURLConnection(u);
    }
}
```

URLStreamHandlerFactory

```
import java.net.*;
public class MaFactory implements URLStreamHandlerFactory {
    public URLStreamHandler createURLStreamHandler(String
        protocol) {
        if (protocol.equalsIgnoreCase("finger")) {
            return new Handler( );
        }
        else {
            //...
            return null;
        }
    }
}
```


Et ensuite

```
public class SourceViewer {
    public static void main (String[] args) {
        URL.setURLStreamHandlerFactory(new MaFactory( ));
        try {
            URL u = new URL(args[0]);
            InputStream in = new BufferedInputStream(u.openStream( ));
            Reader r = new InputStreamReader(in);
            int c;
            while ((c = r.read( )) != -1) {System.out.print((char) c); }
        }
        catch (MalformedURLException ex) {System.err.println(args[0]+" mauvaise URL"); }
        catch (IOException ex) { System.err.println(ex); }
    }
}
```

Cookies (RFC 2965)

```
set-cookie = "Set-Cookie2:" cookies
cookies = 1#cookie
cookie = NAME "=" VALUE *("; " set-cookie-av)
NAME = attr
VALUE = value
set-cookie-av = "Comment" "=" value
| "CommentURL" "=" <"> http_URL <">
| "Discard"
| "Domain" "=" value
| "Max-Age" "=" value
| "Path" "=" value
| "Port" [ "=" <"> portlist <"> ]
| "Secure"
| "Version" "=" 1*DIGIT
portlist = 1#portnum
portnum = 1*DIGIT
```

Cookies en java

- Gestion par un `CookieHandler` de [`java.net.CookieHandler`](#) (abstraite)
 - ❖ Définit un "callback" permettant de gérer les états http.
 - Une url de protocole http gèrera les cookies par un `CookieHandler` (les cookies seront traités et mis en cache et insérés dans les requêtes)

CookieHandler

□ Définir le "CookieHandler"

- `setDefault(cookieHandler)`
- `getDefault()`

□ Deux méthodes

- ❖ `put(uri, responseHeaders)`
 - Met les cookies du cache dans la réponse
- ❖ `get(uri, requestHeaders)`
 - Retourne les cookies applicables pour l'uri de la requête (un `Map<String,List<String>>`)
- ❖ (`requestHeader` est un `Map<String>,List<String>>`)

Une implémentation

- CookieManager est une implémentation de CookieHandler:
 - ❖ Une politique de gestion CookiePolicy
 - ❖ Un stockage des cookies CookieStore

 - ❖ Mise en place:
 - ❖ `java.net.CookieManager cm = new java.net.CookieManager();`
`java.net.CookieHandler.setDefault(cm);`

Example:

```
public class FetchCookie {
    public static void monCookie(String urlString) throws Exception {
        CookieManager manager = new CookieManager();
        manager.setCookiePolicy(CookiePolicy.ACCEPT_ALL);
        CookieHandler.setDefault(manager);
        URL url = new URL(urlString);
        URLConnection connection = url.openConnection();
        Object obj = connection.getContent();
        CookieStore cStore = manager.getCookieStore();
        List<URI> luri = cStore.getURIs();
        for (URI uri : luri) {
            System.out.println("url:" + uri + " cookie:" + cStore.get(uri));
        }
    }
}
```

Cookies

- ❑ Par défaut les cookies ne sont pas persistants
 - ❖ On peut redéfinir la politique (`CookiePolicy`) et le stockage (`CookieStore`) et définir un nouveau `CookieManager`.

Autre politique

```
public class BlacklistPolicy implements CookiePolicy{
    String[] blacklist;
    public BlacklistPolicy(String[] list) {blacklist = list; }
    public boolean shouldAccept(URI uri, HttpCookie cookie) {
        String host;
        try {
            host = InetAddress.getByName(uri.getHost()).getCanonicalHostName();
        } catch (UnknownHostException e) { host = uri.getHost(); }
        for (int i=0; i<blacklist.length; i++) {
            if (HttpCookie.domainMatches(blacklist[i], host)) {
                return false;
            }
        }
        return CookiePolicy.ACCEPT_ORIGINAL_SERVER.shouldAccept(uri, cookie);
    }
}
```


Et...

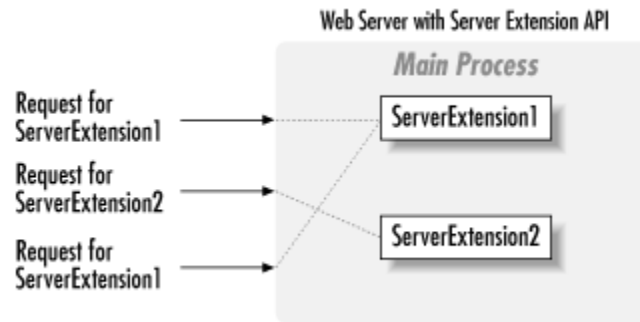
```
String[] list = new String[]{ ".bar.com" };  
CookieManager cm = new CookieManager(null,  
    new BlacklistPolicy(list));  
CookieHandler.setDefault(cm);
```

Autre stockage

Servlets

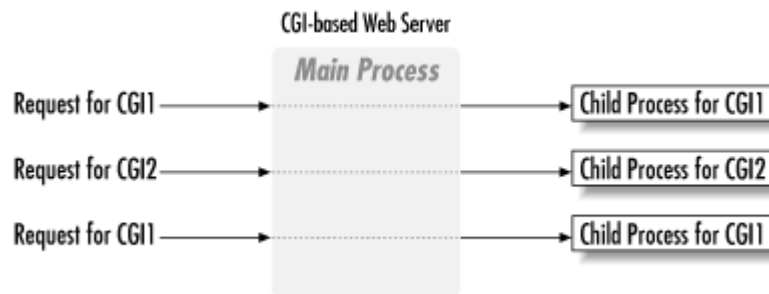
Contenu dynamique

- ❑ Perl/cgi indépendant de la plateforme
- ❑ Extensions du serveur exemple ASP



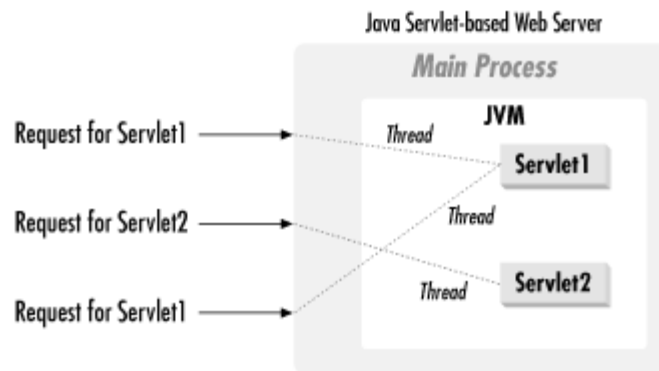
Contenu dynamique...

- ❑ Un serveur transmet des pages html...
- ❑ Comment le rendre dynamique?
 - ❖ CGI et variantes
 - ❖ Traiter des requêtes des clients: méthodes post et get: un requête un processus



Contenu dynamique servlet

- Servlet: utiliser la machine virtuelle java du serveur



Avec quoi utiliser les servlets?

- ❑ Serveur tomcat (java) d'apache utilise les packages javax.servlet et javax.servlet.http
peut s'utiliser avec un serveur apache
- ❑ De nombreux autres serveurs
- ❑ Plug-in sur des serveurs
- ❑ <http://www.servlets.com>

Servlets

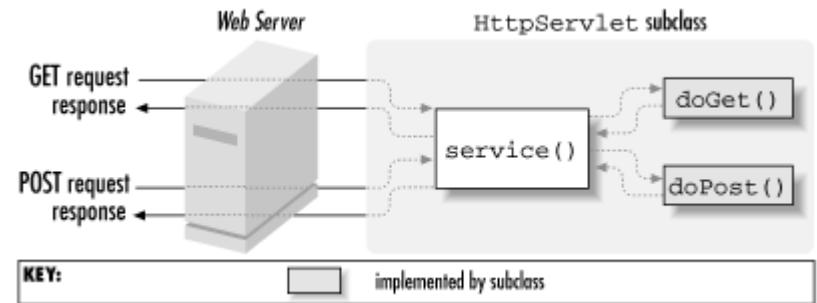
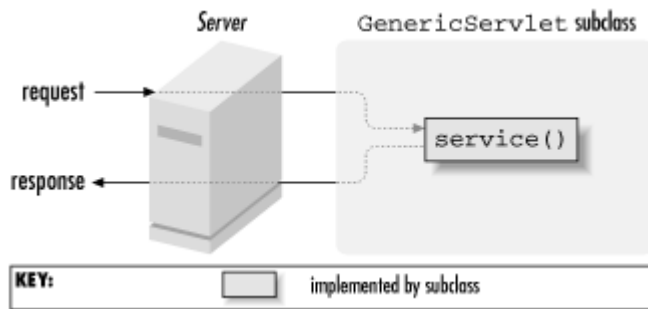
- ❑ Portabilité
- ❑ Puissance (java)
- ❑ Efficacité
- ❑ Sûreté (par l'intermédiaire de la jvm)
- ❑ Extensibilité et flexibilité:
 - Produire du html directement ou transformer du xml
 - Utilisation des JavaServer Pages (script qui génère des servlets)

Servlet

- ❑ *Rappel: les requêtes des clients (essentiellement) par post ou get*
- ❑ Package javax.servlet classes et interfaces pour les servlets (indépendant du protocole)
- ❑ Package javax.servlet.http

Servlet et java

- ❑ Pas de main() mais une méthode service()
- ❑ Version générique: Version http



Exemple basique

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

public class HelloWorld extends HttpServlet {
    public void doGet(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException {
        res.setContentType("text/html");
        PrintWriter out = res.getWriter();
        out.println("<HTML>");
        out.println("<HEAD><TITLE>Bonjour</TITLE></HEAD>");
        out.println("<BODY>");
        out.println("<BIG>Bonjour à tous</BIG>");
        out.println("</BODY></HTML>");
    }
}
```

Et maintenant comment faire?

□ Avec Apache tomcat:

- ❖ Le code source de l'applet dans
- ❖ *server_root/webapps/ROOT/WEB-INF/classes*
- ❖ Compiler avec `javax.servlet` et `javax.servlet.http` dans le « classpath » (présents dans *server_root/lib/servlet.jar*)
- ❖ Lancer le serveur tomcat (`startup.sh` dans *server_root/bin*)
- ❖ (le serveur écoute sur le port 8080 par défaut)
- ❖ Le client demande l'url:
 - **`http://localhost: 8080/servlet/HelloWorld`**

Et pour traiter des données

□ Une page:

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Introductions</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
<FORM METHOD=GET ACTION="/servlet/Hello">
```

```
If you don't mind me asking, what is your name?
```

```
<INPUT TYPE=TEXT NAME="name"><P>
```

```
<INPUT TYPE=SUBMIT>
```

```
</FORM>
```

```
</BODY>
```

```
</HTML>
```



- ❑ *server_root/webapps/ROOT.*
- ❑ url: `http://server:8080/form.html`
- ❑ `http://server:8080/servlet/Hello?name=Jacques+Dupont`

Traiter la réponse

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class Hello extends HttpServlet {
    public void doGet(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException {
        res.setContentType("text/html");
        PrintWriter out = res.getWriter();
        String name = req.getParameter("name");
        out.println("<HTML>");
        out.println("<HEAD><TITLE>Hello, " + name + "</TITLE></HEAD>");
        out.println("<BODY>");
        out.println("Hello, " + name);
        out.println("</BODY></HTML>");
    }
    public String getServletInfo() {
        return "Une servlet qui sait dire bonjour à quelqu'un" ;
    }
}
```

Pour les autres requêtes

□ Pour POST

```
public void doPost(HttpServletRequest req,  
    HttpServletResponse res) throws ServletException,  
    IOException {  
    doGet(req, res);  
}
```

□ Et

```
<FORM METHOD=POST ACTION="/servlet/Hello">
```


Pour head

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class Hello extends HttpServlet {
    public void doGet(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException {

        res.setContentType("text/html");
        if (req.getMethod().equals("HEAD")) return;
        PrintWriter out = res.getWriter();
        String name = req.getParameter("name");
        out.println("<HTML>");
        out.println("<HEAD><TITLE>Hello, " + name + "</TITLE></HEAD>");
        out.println("<BODY>");
        out.println("Hello, " + name);
        out.println("</BODY></HTML>");
    }
}
```

Web app

- ❑ Collection de servlets, de JSP, de document html, images etc...
- ❑ Les web apps sont dans des répertoires de *server_root/webapps/ROOT*
- ❑ Exemple dans le répertoire essai:
 - index.html
 - feedback.jsp
 - images/banner.gif
 - images/jumping.gif
 - WEB-INF/web.xml
 - WEB-INF/lib/bhawk4j.jar
 - WEB-INF/classes/MyServlet.class
 - WEB-INF/classes/com/mycorp/frontend/CorpServlet.class
 - WEB-INF/classes/com/mycorp/frontend/SupportClass.class

WEB-INF

- ❑ Contient les informations de configuration de la webapp
- ❑ *WEB-INF/classes* contient les classes
- ❑ *WEB-INF/lib* les bibliothèques
- ❑ *web.xml* est le descripteur de déploiement

Web.xml

□ Exemple:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE web-app
  PUBLIC "-//Sun Microsystems, Inc.//DTD web Application 2.2//EN"
  "http://java.sun.com/j2ee/dtds/web-app_2_2.dtd">
<web-app>
  <servlet>
    <servlet-name>
      hi
    </servlet-name>
    <servlet-class>
      HelloWorld
    </servlet-class>
  </servlet>
</web-app>
```

suite

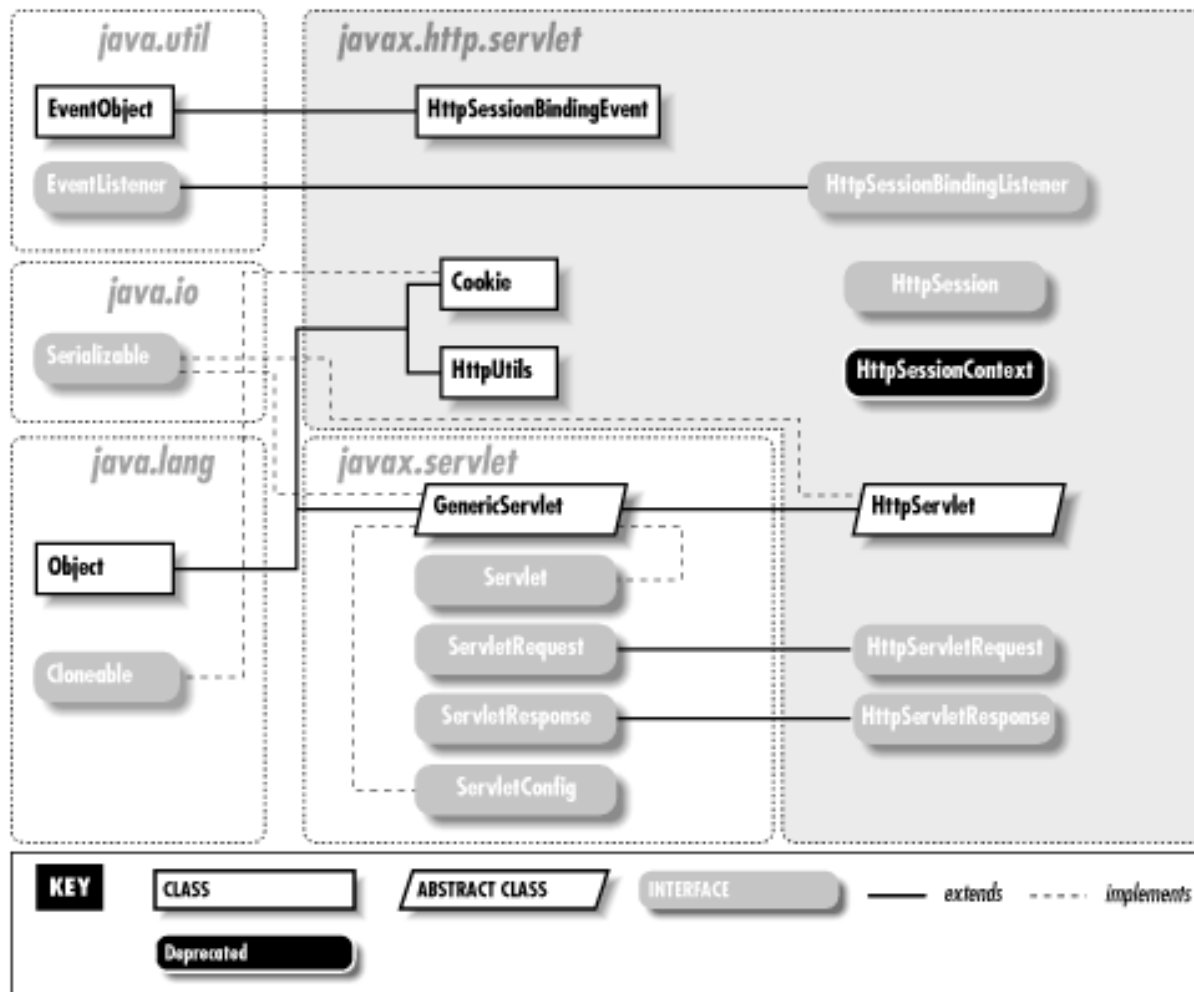
```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE web-app
  PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.2//EN"
  "http://java.sun.com/j2ee/dtds/web-app_2_2.dtd">
<web-app>
  <servlet>
    <servlet-name>
      hi
    </servlet-name>
    <servlet-class>
      HelloWorld
    </servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>
      hi
    </servlet-name>
    <url-pattern>
      /hello.html
    </url-pattern>
  </servlet-mapping>
</web-app>
```

Plus précisément...

généralités

- ❑ Une servlet reçoit une requête soit d'une autre servlet soit d'un navigateur et fournit une réponse vers le navigateur, soit effectue un `forward()` vers une autre servlet.
- ❑ Session: activité mise en œuvre par le même utilisateur
- ❑ Application : toute l'activité.
 - ❖ Nom + attributs

Classes



Servlet

□ Package javax.servlet.http

- HttpServlet gestion des servlet pour recevoir des requêtes et envoyer des réponses
- HttpServletRequest interface des requêtes
- HttpServletResponse interface des réponses
- HttpSession gestion de la session
- ServletContext gestion du contexte de l'application
- RequestDispatcher lancement de servlet (forward())

HttpServlet

- ❑ protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException
- ❑ protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException
- ❑ void init()
- ❑ void destroy()
- ❑ ServletContext getServletContext()

HttpServletRequest

- ❑ `String getParameter(nom)` valeur d'un paramètre de formulaire
- ❑ `String[] getParameterValues(nom)` idem pour plusieurs valeurs
- ❑ `Enumeration getParameterNames()` nom de tous les paramètres du formulaire
- ❑ `void setAttribute(nom, objet)`
- ❑ `Object getAttribute(nom)`
- ❑ `Enumeration getAttributeNames()`
- ❑ `void removeAttribute()`

HttpServletRequest suite

- ❑ `Cookie[] getCookies()`
- ❑ `HttpSession getSession()`
- ❑ `RequestDispatcher getRequestDispatcher(path)`

HttpServletResponse

- ❑ PrintWriter `getWriter()` pour obtenir où écrire le texte
- ❑ `void setContentType("text/html")`
- ❑ `Void addCookie(cookie)`

RequestDispatcher

- Permet de transmettre le contrôle à une autre servlet
 - ❖ `forward()`: le flux de sortie de la servlet est annulé, seul le flux de sortie de la servlet destination du forward est pris en compte
 - `getRequestDispatcher("index.html").forward()`
 - ❖ `Include()` : inclusion dynamique du flux de la servlet destination

Quelques exemples de servlet

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class HelloWorld extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse
        response) throws IOException, ServletException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Hello World!</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("<h1>Hello World!</h1>");
        out.println("</body>");
        out.println("</html>");
    }
}
```

requete

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class RequestInfo extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse
response)
    throws IOException, ServletException
    {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<html>");
        out.println("<body>");
        out.println("<head>");
        out.println("<title>Request Information Example</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("<h3>Request Information Example</h3>");
    }
    //...
```


suite

```
out.println("Method: " + request.getMethod());
out.println("Request URI: " + request.getRequestURI());
out.println("Protocol: " + request.getProtocol());
out.println("PathInfo: " + request.getPathInfo());
out.println("Remote Address: " + request.getRemoteAddr());
out.println("</body>");
out.println("</html>");
```

```
}
```

```
public void doPost(HttpServletRequest request, HttpServletResponse
response)
```

```
throws IOException, ServletException
```

```
{
```

```
    doGet(request, response);
```

```
}
```

```
}
```

formulaire

```
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class RequestParamExample extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse
response)
    throws IOException, ServletException
    {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("GET Request. No Form Data Posted");
    }
//...
```

Suite

```
public void doPost(HttpServletRequest request,
    HttpServletResponse res)
    throws IOException, ServletException
{
    Enumeration e = request.getParameterNames();
    PrintWriter out = res.getWriter ();
    while (e.hasMoreElements()) {
        String name = (String)e.nextElement();
        String value = request.getParameter(name);
        out.println(name + " = " + value);
    }
}
```

header

```
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class RequestHeaderExample extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws IOException, ServletException
    {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        Enumeration e = request.getHeaderNames();
        while (e.hasMoreElements()) {
            String name = (String)e.nextElement();
            String value = request.getHeader(name);
            out.println(name + " = " + value);
        }
    }
}
```

Cookies

```
public class CookieExample extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws IOException, ServletException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        Cookie[] cookies = request.getCookies();
        for (int i = 0; i < cookies.length; i++) {
            Cookie c = cookies[i];
            String name = c.getName();
            String value = c.getValue();
            out.println(name + " = " + value);
        }
        String name = request.getParameter("cookieName");
        if (name != null && name.length() > 0) {
            String value = request.getParameter("cookieValue");
            Cookie c = new Cookie(name, value);
            response.addCookie(c);
        }
    }
}
```

Session

```
public class SessionExample extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws IOException, ServletException{
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        HttpSession session = request.getSession(true);
        Date created = new Date(session.getCreationTime());
        Date accessed = new Date(session.getLastAccessedTime());
        out.println("ID " + session.getId());
        out.println("Created: " + created);
        out.println("Last Accessed: " + accessed);
        String dataName = request.getParameter("dataName");
        if (dataName != null && dataName.length() > 0) {
            String dataValue = request.getParameter("dataValue");
            session.setAttribute(dataName, dataValue);
        } //...
```

Suite

```
// print session contents

    Enumeration e = session.getAttributeNames();
    while (e.hasMoreElements()) {
        String name = (String)e.nextElement();
        String value =

session.getAttribute(name).toString();
        out.println(name + " = " + value);
    }
}
}
```

JSP

□ Script pour générer des servlets

- ❖ Des données statiques (HTML)

- ❖ Des directives JSP

 - exemple:

 - `<jsp:directive.include file="unAutreFichier" />`

- ❖ Eléments de script

 - exemple:

 - `<%! int variableDeClasse = 0; %>`

- ❖ Actions JSP

 - Exemple:

 - `<jsp:useBean id="nomDeBean" class="package.Bean" scope="request" />`

Un exemple (netbeans)

- ❑ index.jsp un formulaire simple
- ❑ reponse.jsp la réponse
- ❑ La javabean pour le traitement:
TraiteNom.java

Remarque...

- Netbeans ou eclipse permettent de gérer les servlets et les jsp aisément...