

# Brazil + JDBC

Juin 2001, douin@cnam.fr

[http://jfod.cnam.fr/tp\\_cdi/douin/](http://jfod.cnam.fr/tp_cdi/douin/)

version du 26 Mai 2003 :

JDBC-SQL et Brazil

pré-requis : lecture de Tutorial JDBC de Sun

# Bibliographie

- Brazil
  - [Bra00] [www.sun.com/research/brazil](http://www.sun.com/research/brazil)
  - [Bra01] [www.ExperimentalStuff.com](http://www.ExperimentalStuff.com), <http://www.brazilhandlers.com:9090>
  
- JDBC
  - Le [tutorial JDBC](http://java.sun.com/docs/books/tutorial/jdbc/index.html) de Sun  
<http://java.sun.com/docs/books/tutorial/jdbc/index.html>
  
- la base de données hypersonic
  - <http://sourceforge.net/projects/hsqldb/>
  
- Un tutorial SQL
  - [http://jfod.cnam.fr/tp\\_cdi/Tp12\\_1/sqltutorial.htm](http://jfod.cnam.fr/tp_cdi/Tp12_1/sqltutorial.htm)

# Sommaire

- JDBC
  - introduction succincte
- hypersonic
  - Une base de données entièrement écrite en Java
  - Sa mise en œuvre
- Brazil+JDBC
  - HTTP <-> SQL
  - SqlHandler
  - SqlTemplate

# Java/JDBC : principes

- Le [tutorial JDBC](#) de Sun, un pré-requis

# Java/JDBC

package java.sql

- Chargement du "driver" associé à la base
- classe **Connection**
  - 1) établir une connexion avec la base
- classe **Statement**
  - 2) engendre une instruction : syntaxe SQL
- classe **ResultSet**
  - 3) exécution
    - `ResultSet rs = stmt.executeQuery("select * from ..... ")`
      - puis `rs.next()`
    - `int res = stmt.executeUpdate("insert into .... ")`

# Hypersonic version 1.6

- <http://www.hsqldb.org>

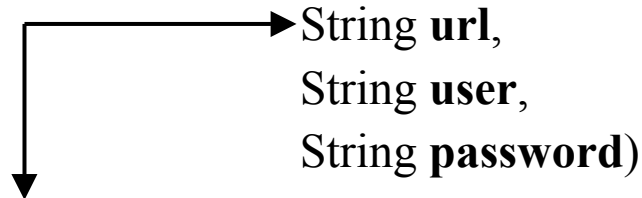
- Chargement du "driver"

- `Class.forName("org.hsqldb.jdbcDriver");`

- établir une connexion :

- `Connection con = DriverManager.getConnection(`

- `String url,`
    - `String user,`
    - `String password)`



- quelles sont les **url** ?, où est la base ?

- Mémoire

- `url = jdbc:hsqldb:.`

- Fichier local

- `url = jdbc:hsqldb:test`

- Socket

- `url = jdbc:hsqldb://localhost`

- Web / HTTP

- `url = jdbc:hsqldb:http://localhost/`

# Hypersonic,java/JDBC, suite

- Un exemple :

```
Class.forName("org.hsqldb.jdbcDriver");           // driver jdbc

con = DriverManager.getConnection(
    "jdbc:hsqldb:http://localhost:8181/LA_BASE",    // url
    "ou "jdbc:hsqldb:LA_BASE",
    "sa",                                           // user
    "" );                                          // password

stmt = con.createStatement();
int res=stmt.executeUpdate("create table lundi ..... ");

ResultSet rs =stmt.executeQuery("select * from lundi");

int res=stmt.executeUpdate("insert into lundi ..... ");
```

# Commandes SQL : syntaxe

- **CREATE TABLE** name ( columnDefinition [, ...] )  
columnDefinition: column DataType [ [NOT] NULL] [PRIMARY KEY]  
DataType: { INTEGER | DOUBLE | VARCHAR | DATE | TIME |... }
- **SELECT** [DISTINCT] { selectExpression | table.\* | \* } [, ... ]  
[INTO newTable] **FROM** tableList  
[WHERE Expression]  
    [ORDER BY selectExpression [{ASC | DESC}] [, ...] ]  
    [GROUP BY Expression [, ...] ]  
    [UNION [ALL] selectStatement]
- **INSERT INTO** table [ (column [,...]) ] { VALUES(Expression [,...]) | SelectStatement }

# Hypersonic 1.6

- **Serveur Web : intranet/internet**

```
java -cp hsqldb.jar org.hsqldb.WebServer -port 8181 -silent false -database LA_BASE
```

- **Serveur socket : intranet**

```
java -cp hsqldb.jar org.hsqldb.Server -port 899 -silent false -database LA_BASE
```

# Sources Java : 3 applications

- **create** : création d'une table : nommée lundi
- **insert** : Insertion dans cette table
- **select** : lecture de cette table
- **en commun** : le chargement du Driver :

```
String url ="jdbc:hsqldb:LA_BASE";  
//ou String url ="jdbc:hsqldb:http://localhost:8181/LA_BASE";  
Connection con;  
Statement stmt;  
try {  
    Class.forName("org.hsqldb.jdbcDriver");  
} catch(java.lang.ClassNotFoundException e) {  
    System.err.print("ClassNotFoundException: " + e.getMessage());  
}  
con = DriverManager.getConnection(url,"sa","");  
stmt = con.createStatement();
```

# Exemple : create table

```
try {
```

```
    String sql = "CREATE TABLE lundi ( " +  
                "nom          VARCHAR, " +  
                "prenom     VARCHAR, " +  
                "e_mail     VARCHAR, " +  
                "sujet      INTEGER)  ";
```

```
    stmt.executeUpdate(sql);
```

```
    stmt.close();
```

```
    con.close();
```

```
} catch(SQLException ex) {...
```

# CreateLundi.java, au complet

```
import java.sql.*;
public class CreateLundi {

    public static void main(String[] args) {
        try {
            Class.forName("org.hsqldb.jdbcDriver");

            Connection con = DriverManager.getConnection("jdbc:hsqldb:LA_BASE","sa","");
                // ou "jdbc:hsqldb:http://localhost:8181/LA_BASE"

            Statement stmt = con.createStatement();
            String sql = "CREATE TABLE Lundi ( " +
                "nom        VARCHAR, " +
                "prenom     VARCHAR, " +
                "e_mail     VARCHAR, " +
                "sujet      integer ) ";

            stmt.executeUpdate(sql);
            stmt.close();
            con.close();
        } catch (Exception e){
            System.err.println("Exception in createTables()");
            System.err.println(e);
        }
    }
}
```

# À faire chez soi

- Copier/coller/compiler/exécuter le texte du transparent précédent : CreateLundi.java
- Démarrer le serveur web/hsq1 sur une autre machine ou dans un autre répertoire
- Modifier/compiler CreateLundi.java en conséquence
- Exécuter CreateLundi

# Exemple de source Java : insert

```
String sql = "insert into LUNDI " +  
            "values('Lambert', 'michel' ,      " +  
            " '<A HREF=\"mailto:mlambert@noos.fr\">mlambert@noos.fr</A>', 1)");  
  
int res = stmt.executeUpdate(sql);  
  
stmt.close();  
con.close();  
  
} catch(SQLException ex) {...
```

# InsertLundi.java

```
import java.sql.*;
public class InsertLundi{
    public static void main(String[] args) {
        try {Class.forName("org.hsqldb.jdbcDriver");
            Connection con = DriverManager.getConnection("jdbc:hsqldb:LA_BASE","sa","");
                // ou "jdbc:hsqldb:http://localhost:8181/LA_BASE"
            Statement stmt = con.createStatement();
            String sql= "insert into lundi values( " + "'Lambert','michel', " +
                " 'mlambert@noos.fr</a>',1\)";
            stmt.executeUpdate\(sql\);

            sql = "insert into Vendredi values\( " + "'Robleda','bruno', " +
                " 'brunodb@club-internet.fr</a>',1\\)";
            stmt.executeUpdate\\(sql\\);

            stmt.close\\(\\);
            con.close\\(\\);
        } catch \\(Exception e\\){
            System.err.println\\("Exception in creatTables\\(\\)"\\);
            System.err.println\\(e\\);
        }
    }
}
```

# À faire chez soi

- Copier/coller/compiler/exécuter le texte du transparent précédent : InsertLundi.java
- Démarrer le serveur web/hsqldb sur une autre machine ou dans un autre répertoire
- Modifier/compiler InsertLundi.java en conséquence
- Exécuter InsertLundi

# Exemple de source Java : select

```
String sql = "select * from lundi";
ResultSet rs = stmt.executeQuery(sql);
ResultSetMetaData meta = rs.getMetaData();
int columns = meta.getColumnCount();

while (rs.next()) {
    for(int i=1;i<=columns;i++){
        System.out.println(rs.getString(meta.getColumnLabel(i)));
    } // for
} // while

stmt.close();
con.close();
} catch(SQLException ex) {...
```

# SelectLundi.java

```
import java.sql.*;
public class SelectLundi {
    public static void main(String[] args) {
        try {
            Class.forName("org.hsqldb.jdbcDriver");

            Connection con = DriverManager.getConnection("jdbc:hsqldb:LA_BASE","sa","");
            // ou "jdbc:hsqldb:http://localhost:8181/LA_BASE"

            Statement stmt = con.createStatement();
            String sql = "select * from lundi";
            ResultSet rs = stmt.executeQuery(sql);
            ResultSetMetaData meta = rs.getMetaData();
            int columns = meta.getColumnCount();

            while (rs.next()) {
                for(int i=1;i<=columns;i++){
                    System.out.print(rs.getString(i) + " ");
                }
                System.out.println();
            } // while
            stmt.close();
            con.close();
        } catch (Exception e){ System.err.println(e);}
    }
}
```

# À faire chez soi

- Copier/coller/compiler/exécuter le texte du transparent précédent : `SelectLundi.java`
- Démarrer le serveur web/hsq1 sur une autre machine ou dans un autre répertoire
- Modifier/compiler `SelectLundi.java` en conséquence
- Exécuter `SelectLundi`

# Brazil

- Interface HTTP <-> SQL-JDBC

[http://lmi80.cnam.fr:9090/hypersonic/?query=select+\\*+from+Lundi](http://lmi80.cnam.fr:9090/hypersonic/?query=select+*+from+Lundi)

<http://lmi80.cnam.fr:9090/hypersonic/?update=insert+into+Lundi> .....

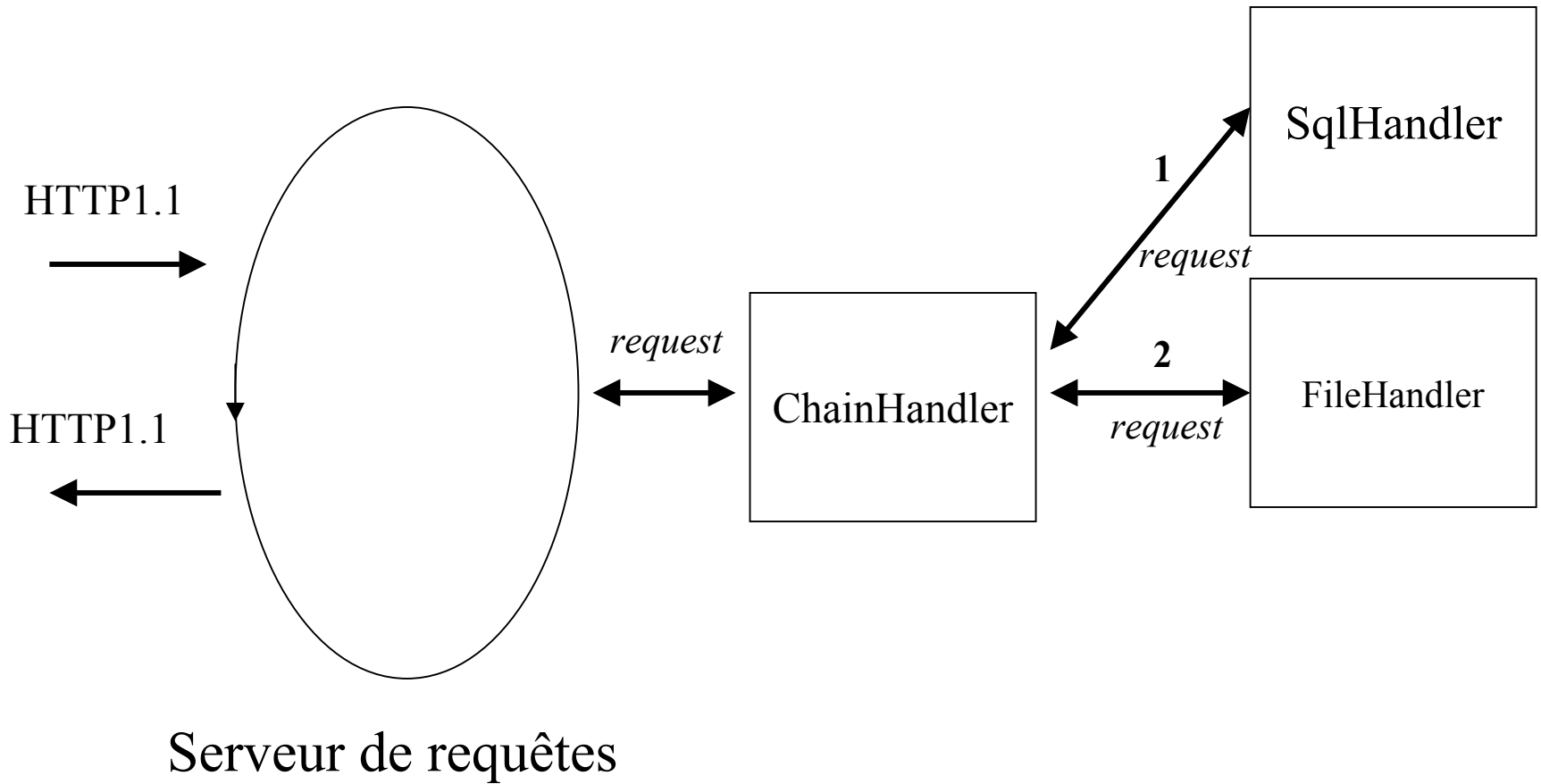


**select \* from lundi**

**résultat**

<b>NOM</b>	<b>PRENOM</b>	<b>E_MAIL</b>	<b>SUJET</b>
Lambert	michel	<a href="mailto:mlambert@noos.fr">mlambert@noos.fr</a>	1
Roble da	bruno	<a href="mailto:brunodb@club-internet.fr">brunodb@club-internet.fr</a>	1

# Serveur Brazil + SqlHandler



# Le fichier de configuration

```
main.class=sunlabs.brazil.server.ChainHandler
main.handlers=sql file
#
sql.class=cnam.sql.SqlHandler
sql.name=/hypersonic/
# les paramètres jdbc du handler
sql.driver=org.hsqldb.jdbcDriver
sql.url=jdbc:hsqldb:LA_BASE
# ou sql.url=jdbc:hsqldb:http://localhost:8181/LA_BASE
sql.user=sa
sql.password=
```

# Structure de SqlHandler

- méthode init
  - lecture des paramètres du fichier de configuration
- méthode respond
  - paramètres query et update

# Structure de SqlHandler.java,init

```
public boolean init(Server server, String prefix){
    this.server = server; this.propsPrefix = prefix;
    try {
        name      = server.props.getProperty(propsPrefix + "name");
        driver    = server.props.getProperty(propsPrefix + "driver");
        url       = server.props.getProperty(propsPrefix + "url");
        user      = server.props.getProperty(propsPrefix + "user");
        password  = server.props.getProperty(propsPrefix + "password");
    } catch (Exception e1) {...return false;}

    try {
        Class.forName(driver);
    } catch(Exception e) {...return false;}

    return true;
}
```

# SqlHandler respond

```
public boolean respond(Request request) throws IOException{
    String urlPrefix = request.props.getProperty(propsPrefix, name);
    if (!request.url.startsWith(urlPrefix)) {
        server.log(Server.LOG_INFORMATIONAL, null, "Not " + name + "
                prefix: " + request.url);
        return false;
    }
    Hashtable t = request.getQueryData();

    String result; String
    query=null; Connection con=null;
    Statement stmt=null;
    try{
        con = DriverManager.getConnection(url,user,password);
        stmt = con.createStatement();

// lecture des paramètres HTTP
```

# SqlHandler respond suite

```
// lecture des paramètres HTTP
query = (String)t.get("query");
    if (query != null){
        ResultSet rs = stmt.executeQuery(query);
        result = query + formatTable(rs);
    } else {
        String update = (String)t.get("update");
        if( update != null){
            int res = stmt.executeUpdate(update);
            result = update + res;
        } else result = "unknown, only (query or update)";
    }
    stmt.close();
    con.close(); ...
```

# SqlHandler : formatTable

```
public static String formatTable(ResultSet rs) throws SQLException{
    ResultSetMetaData meta = rs.getMetaData();
    StringBuffer html = new StringBuffer();
    html.append("<table border=2 BGCOLOR=\"#fffb0d8\">\r\n");
    html.append("<caption><big><b>r&eacute;sultat</b></caption>\r\n");
    int columns = meta.getColumnCount();
    html.append("<tr BGCOLOR=\"#00fd00\">\r\n");
    for(int i=1;i<=columns;i++){
        html.append("<td><big><b>"+meta.getColumnLabel(i)+
            "</b></big></td>\r\n");
    } // for
    html.append("</tr>\r\n");
}
```

**résultat**

**NOM**

**PRENOM**

**E\_MAIL**

**SUJET**


# SqlHandler : formatTable suite

...

```
while (rs.next()) {
    html.append("<tr>\r\n");
    for(int i=1;i<=columns;i++){
        html.append("<td>\r\n");
        html.append(" " + rs.getString(meta.getColumnLabel(i)) + "\r\n");
        html.append("</td>\r\n");
    } // for
    html.append("</tr>\r\n");
} // while
html.append("</table>\r\n");
return html.toString();
}
```

Lambert	michel	<a href="mailto:mlambert@noos.fr">mlambert@noos.fr</a>	1
Robleda	bruno	<a href="mailto:brunodb@club-internet.fr">brunodb@club-internet.fr</a>	1
Nguyen	thuan	<a href="mailto:the_thuan@yahoo.fr">the_thuan@yahoo.fr</a>	1

# SqlHandler : le résultat

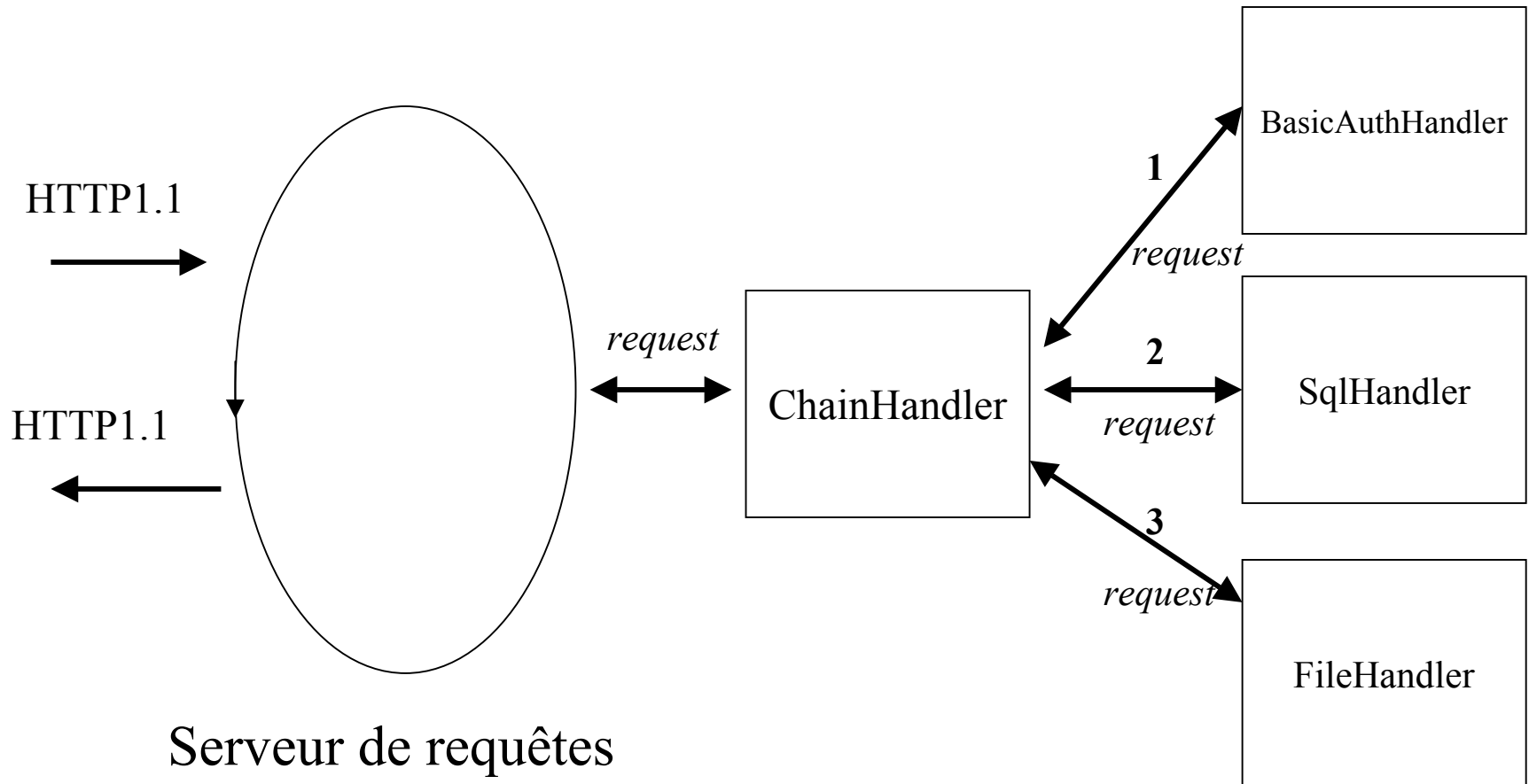
Adresse  [http://lmi80.cnam.fr:9090/hypersonic/?query=select+\\*+from+lundi](http://lmi80.cnam.fr:9090/hypersonic/?query=select+*+from+lundi)

**select \* from lundi**

**résultat**

<b>NOM</b>	<b>PRENOM</b>	<b>E_MAIL</b>	<b>SUJET</b>
Lambert	michel	<a href="mailto:mlambert@noos.fr">mlambert@noos.fr</a>	1
Robleda	bruno	<a href="mailto:brunodb@club-internet.fr">brunodb@club-internet.fr</a>	1
Nguyen	thuan	<a href="mailto:the_thuan@yahoo.fr">the_thuan@yahoo.fr</a>	1
Minchella	michel	<a href="mailto:p_minchella@hotmail.com">p_minchella@hotmail.com</a>	1
Derrien	alain	<a href="mailto:alain_derrien@yahoo.fr">alain_derrien@yahoo.fr</a>	2
Ciret	evariste	<a href="mailto:eciret@yahoo.fr">eciret@yahoo.fr</a>	2
Lebougre	herve	<a href="mailto:herve.lebougre@libertysurf.fr">herve.lebougre@libertysurf.fr</a>	2
Rodriguese	jose	<a href="mailto:jhrodrigues@wanadoo.fr">jhrodrigues@wanadoo.fr</a>	2
Routaboul	joel	<a href="mailto:jroutab@club-internet.fr">jroutab@club-internet.fr</a>	2
Abo	robert	<a href="mailto:Robert.ABO@sema.fr">Robert.ABO@sema.fr</a>	3
Martial	eric	<a href="mailto:emartial@club-internet.fr">emartial@club-internet.fr</a>	4

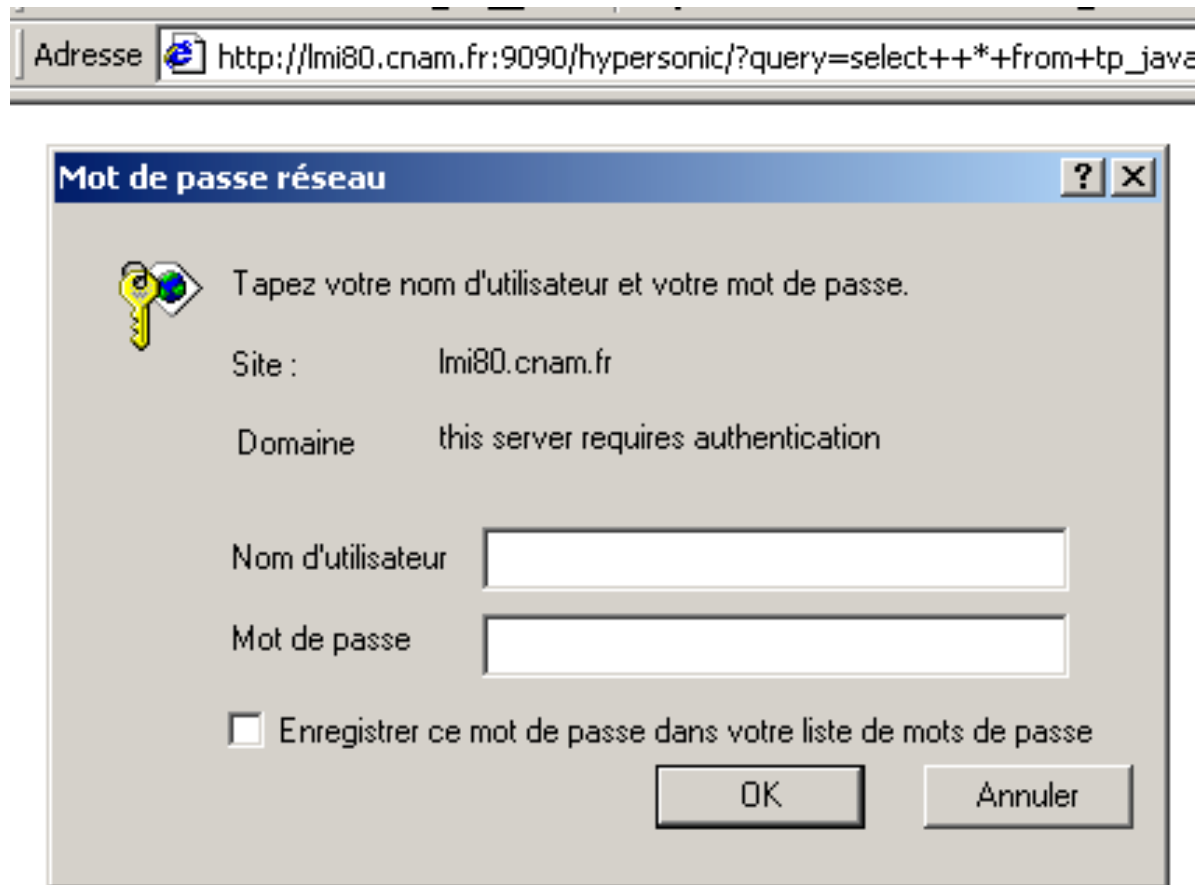
# Serveur Brazil avec authentication



# Le fichier de configuration

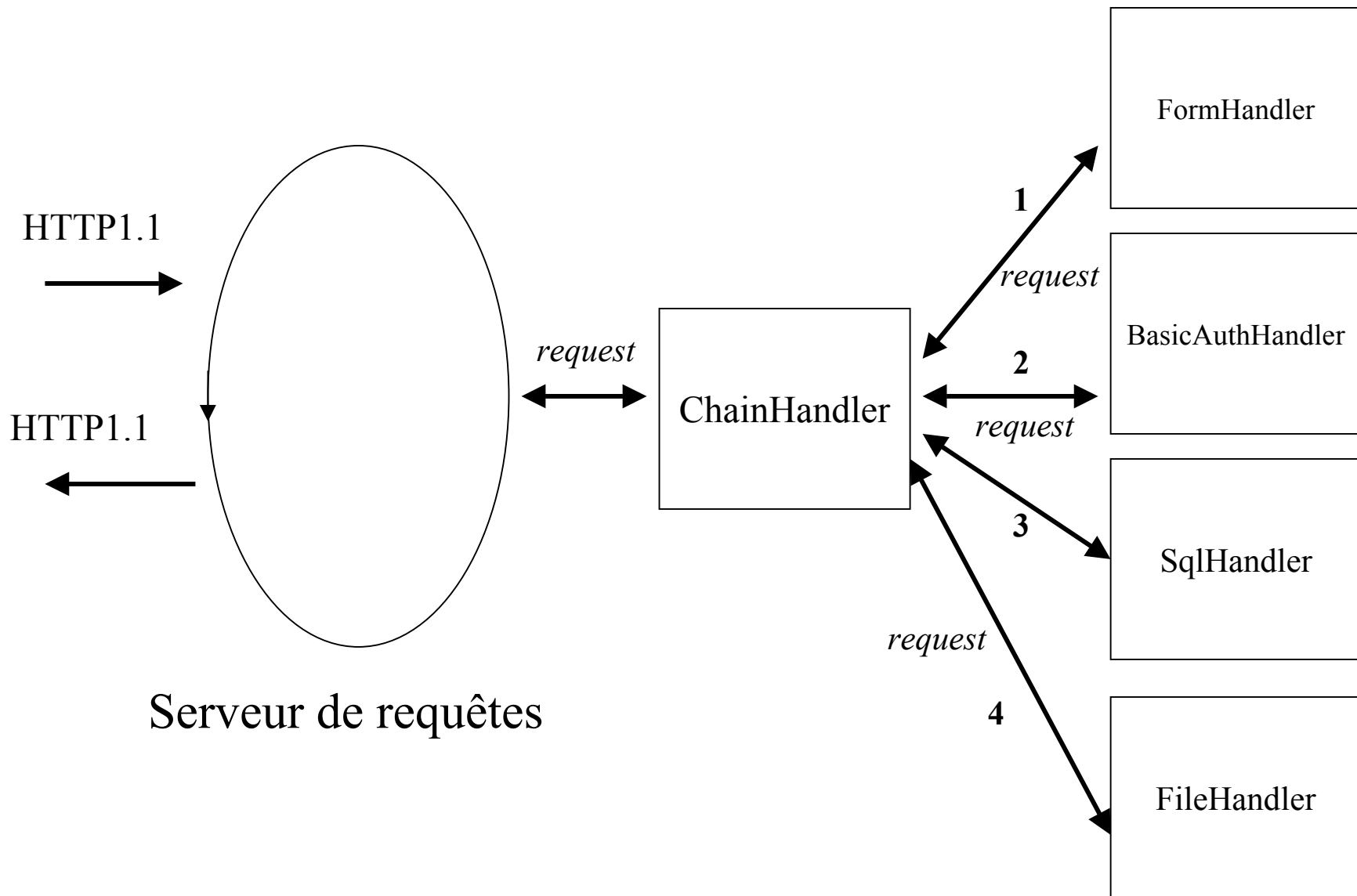
```
main.class=sunlabs.brazil.server.ChainHandler
main.handlers=auth sql file
#
auth.class=sunlabs.brazil.handler.BasicAuthHandler
auth.prefix=/hypersonic/
auth.mapFile=credentials
auth.session=account
auth.realm=this server requires authentication
auth.message=pas de SQL possible pour vous !!!
#
sql.class=sql.SqlHandler
sql.name=/hypersonic/
sql.driver=org.hsql.jdbcDriver
sql.url=jdbc:hsqldb:LA_BASE
sql.user=sa
sql.password=
```

# BasicAuthHandler : le résultat



# Architecture possible

- FormHandler
  - insertion en base de données
- SqlHandler
  - interrogation, modification



# Formulaire : FormHandler

nom	<input type="text"/>	prénom	<input type="text"/>	Quel groupe ?	<input type="text" value="groupe1"/>
votre mél	<input type="text" value="mail@mailhost.com"/>				
Ajouter cette URL à notre table			<input type="button" value="ajouter mon nom"/>		

```
<FORM ACTION="http://lmi80.cnam.fr:9090/insert/">  
<INPUT TYPE=hidden name="table" value="Lundi">  
<TABLE BORDER CELLPADDING="2">
```

.....

# Structure de FormHandler

- méthode init : chargement du jdbcdriver
- méthode respond :

```
Hashtable t = request.getQueryData();
String sql = "insert into " + (String)t.get("table") +
            " values(" +
            "\"" + (String)t.get("nom") + "\", " +
            "\"" + (String)t.get("prenom") + "\", " +
            "\"" + (String)t.get("e_mail") + ")";
int res = stmt.executeUpdate(sql);
```

# le fichier de configuration

```
form.class=cnam.sql.FormHandler  
form.name=/insert/  
form.driver=org.hsqldb.jdbcDriver  
form.url=jdbc:hsqldb:LA_BASE  
#ou form.url=jdbc:hsqldb:http://localhost:8181/LA_BASE  
form.user=sa  
form.password=
```

# Brazil + BSL

- Brazil et les Templates
- balises HTML, interprétées avant d'être transmises
  - `<sql`
    - SqlTemplate
  - `<foreach` et `<if`
    - BSLTemplate
  - `<property`
    - PropsTemplate

# Rappel BSL

- `<! Le fichier TestSql.html`
- `<table border>`
- `<foreach name=a glob=*> <tr>`
- `<td><property a.name> </td>`
- `<td><property a.value> </td>`
- `<tr>`
- `</foreach>`
- `</table>`

Une table HTML : visualisation de `server.props`

- BSL :
- `<if> ...<elseif> ... <else> ... </if>`
- `<foreach> ... </foreach>`
- `<property name=xxx>` est remplacé par `xxx`

# Exemple

- Accès aux propriétés du serveur (server.props)
- `<html>`
- la valeur log : `<property log>`
- la racine root : `<property root>`
- `<tag>a href=<property name=test.url></tag> un_lien</a>`
- `<if name=liste.CNAM_URL.${field}.${index} match=http:++>`

# Usage de BSL : le résultat

Adresse  <http://lmi93.cnam.fr:9999/TestSql.html>

Test SqlTemplate, serveur Brazil, le 22 Mai 2001

main.handlers	sql file
mime.html	text/html
handler	main
sql.url	jdbc:HypersonicSQL:http://163.173.228.80:8181
mime.css	text/x-css-stylesheet
sql.class	sunlabs.brazil.template.TemplateHandler
mime.txt	text/plain
mime.gif	image/gif
mime.jpg	image/jpeg

# SqlHandler / SqlTemplate

- `<sql prefix=inscrit> select * from lundi</sql>`

*#le fichier de configuration*

```
sql.class=sunlabs.brazil.template.TemplateHandler
sql.templates= sunlabs.brazil.template.DebugTemplate \
               sunlabs.brazil.template.PropsTemplate \
               sunlabs.brazil.sql.SqlTemplate \
               sunlabs.brazil.template.BSLTemplate
```

```
sql.driver=org.hsqldb.jdbcDriver
```

```
sql.url=jdbc:hsqldb:LA_BASE
```

```
sql.sqlPrefix=param
```

```
param.user=sa
```

```
param.password=
```

# SqlTemplate + BSLTemplate

```
<sql prefix=inscrit> SELECT * FROM lundi</sql>
```

```
<TABLE border=2>
```

```
<foreach name=index property=inscrit.rows >
```

```
<tr>
```

```
<foreach name=field list="NOM PRENOM SUJET E_MAIL">
```

```
<td>
```

```
<property name=inscrit.LUNDI.${field}.${index}>
```

```
</td>
```

```
</foreach>
```

```
</tr>
```

```
</foreach>
```

```
</TABLE>
```

# SqlTemplate : le résultat



Test SqlTemplate, serveur Brazil, le 22 Mai 2001

Lambert	michel	1	<a href="mailto:mlambert@noos.fr">mlambert@noos.fr</a>
Robleda	bruno	1	<a href="mailto:brunodb@club-internet.fr">brunodb@club-internet.fr</a>
Nguyen	thuan	1	<a href="mailto:the_thuan@yahoo.fr">the_thuan@yahoo.fr</a>
Minchella	michel	1	<a href="mailto:p_minchella@hotmail.com">p_minchella@hotmail.com</a>
Derrien	alain	2	<a href="mailto:alain_derrien@yahoo.fr">alain_derrien@yahoo.fr</a>
Ciret	evariste	2	<a href="mailto:eciret@yahoo.fr">eciret@yahoo.fr</a>
Lebougre	herve	2	<a href="mailto:herve.lebougre@libertysurf.fr">herve.lebougre@libertysurf.fr</a>
Rodriguese	jose	2	<a href="mailto:jhrodrigues@wanadoo.fr">jhrodrigues@wanadoo.fr</a>
Routaboul	joel	2	<a href="mailto:jroutab@club-internet.fr">jroutab@club-internet.fr</a>
Abo	robert	3	<a href="mailto:Robert.ABO@sema.fr">Robert.ABO@sema.fr</a>
Martial	eric	4	<a href="mailto:emartial@club-internet.fr">emartial@club-internet.fr</a>

```
<HTML><HEAD>  <TITLE>SqlTemplate</TITLE></HEAD>
<BODY><P>Test SqlTemplate, serveur Brazil, le 26 Mai 2003<P>

<sql prefix=inscrit> SELECT * FROM lundi</sql>
<debug apres la requete SQL: >
<TABLE border=2>
<tr>
<foreach name=field list=" NOM PRENOM SUJET E_MAIL">
<td>
<b><property name=field></b>
</td>
</foreach>
</tr>
  <foreach name=index property=inscrit.rows >
    <tr>
      <foreach name=field list="NOM PRENOM SUJET E_MAIL">
        <td><property name=inscrit.LUNDI.${field}.${index}></td>
      </foreach>
    </tr>
  </foreach>
</TABLE>
<p>
</BODY></HTML>
```

# Conclusion