بسم الله الرحمن الرحيم،،

الجافا من اللغات جيدة التعامل مع الويب و التطبيقات الموزعه (Distributed Applications) لما بها من إمكانيات مبنيه داخل اللغه و دوالها الأساسيه تمكنها من عمل ذلك.

مكونات الويب (Web Components) في الجافا:
المكونين الأساسين للويب في الجافا هما ال Servlets و ال JSP، و لقد ظلا لفتره ليست بقصيره هما المكونان الوحيدان المعتمدان كمعيار لمكونات الويب في الجافا قبل ظهور ال JSF.
ولعل أهم هذه المكونات الثلاثه هو ال Servlets و التي هي أساس باقي المكونات.

و بالتالي لايمكنك كتابة أي تطبيق ويب في الجافا إلا بعد معرفه ال Servlets معرفه أساسيه على الأقل.
تأتي ال JSP في المرتبه الثانيه من حيث الأهميه، و إن كنت لا تسطيع كتابة تطبيق ويب (كبير) إلا بإتقانك لهذين المكونين، لذا أردت أن أشرع في كتابة هذا الكتيب لأهمية هذين المكونيين لكل مطوري تطبيقات الويب في الجافا عسى الله أن ينفعنا به.

في البداية لم يكن هناك JSP، فقط كان هناك Servlets ثم ظهرت بعد ذلك ال JSP- و سنعرف لاحقا كيف كان تطوير التطبيقات يحدث في تلك الأثناء- فلما ظهرت ال JSP إتجه المطورون لإستخدامها بدلا عن ال Servlets و إن كان ذلك خاطئا، لأن كلاهما يكمل الأخر، ولاتسطيع أن تكتب برنامج إحترافي إلا إذا كان يحتوي على الإثنين معا مكونان ما يسمي بنمط النموذج-العرض-التحكم (Model-View-Controller)، و هو من أهم الأنماط المستخدم في عالم الويب عموما وليس مع الجافا فقط.

دعنا الأن ننتقل لمثال عملي لأن معظم المطوريين يكرهون هذا الكلام النظري، فأنتم الأن تريدون أن تروا مثال عملي لهذه التقنيه المسماه ب Servlets &JSP (لاحظ أني إعتبرتهم تقنيه واحده، لأنهما في البداية كما ذكرت متكاملان معا، كما أن ال JSP -كما ستري لاحقا- هي في الأول و الأخر عباره عن Servlet!)

تهيئة بيئة التطوير:
كما تعودنا مع الجافا، فإنك لاتحتاج بيئة تطوير متكامله (IDE) لكتابة برمجك، وإن كانت تقوم بتسريع عملة التطوير ، لذا سنبدأ الأمثلة بإستخدام محرر نصوص بسيط مثل VIM أو Gedit أو Notepad و إن كنا سنتستخدم Eclipse لاحقا. (يكمنك إستخدام بيئة التطوير المفضله لديك مثل NetBeans فلن تجد فرق كبير، فأنت مبرمج جافا التي يملكها مجتمع المصادر المفتوحه، ولست مبرمج ميكروسوفت (أقصد ميكروسوف حرفيا و لا أقصد سي شارب مثلا).

المهم، ال Notepad لكتابة برامجك، فماذا عن الباقي؟
طبعا كما تعودت فسوف تحتاج لتحميل ال JDK من موقع sun ، كما سوف تحتاج خادم ويب (Web Server) لتنفيذ برامجك، من أشهر أمثلة خوادم الويب هو Apache tomcat و يمكنك تحميله من على موقع Apache.

قم بتثبيت ال JDK التي قمت بتحميلها (من المفترض أنك تستطيع أن تقوم بهذه العمليه)، بعد ذلك تعال سويا نقوم بتثبت خادم الويب apache tomcat.

لنفترض أن نسخة التومكات التي قمت بتحميلها تحمل هذا الإسم apache-tomcat-6.0.18.zip وتقع في المجلد /home/mhewedy/
ملاحظه، لأني أستخدم نظام التشغيل لينوكس سوف أستخدم المسارات و الأوامر الخاصه به مفترضا أنك تستخدمه أيضا، فإن لم تكن تستخدمه فهذه هي فرصتك للتعرف على هذا النظام التشغيل العملاق.
ملاحظة، إن كنت تستخدم أي من أنظمة اليونكس الأخري فالكلام التالي ينطبق عليك أيضا.

قم بعمل مجلد في مجلد المنزل لديك و سميه programs

```
cd ~
mkdir programs
```

بعد ذلك قم بنسخ التوكات التي قد حملته من موقع apache إلي المجلد programs:

```
cp apache-tomcat-6.0.18.zip ~/programs
```

(قد تختلف نسخة تومكات لديك، ولكن لا يهم)
بعد ذلك قم بفك الضغط:

```
cd programs
unzip apache-tomcat-6.0.18.zip
```

قم بعمل رابط للتومكات في مجلد المنزل لديك:

```
cd
ln -sf programs/apache-tomcat-6.0.18 tomcat
```

وبالتالي فللوصول للتومكات كل ما عليك هو :

```
cd
cd tomcat
```

قم بتنفيذ التالي:

```
cd ~/tomcat/bin
chmod +x *.sh
```

بقيت خطوه قبل بدء الخادم:
قم بتحرير .bashrc (قد يسمي .bash_profile في بعض أنظمة يونكس الأخري) الموجود في مجلد المنزل لديك و قم بإضافة التالي:

```
export JAVA_HOME=/path/to/jdk
export CATALINA_HOME=$HOME/tomcat
```

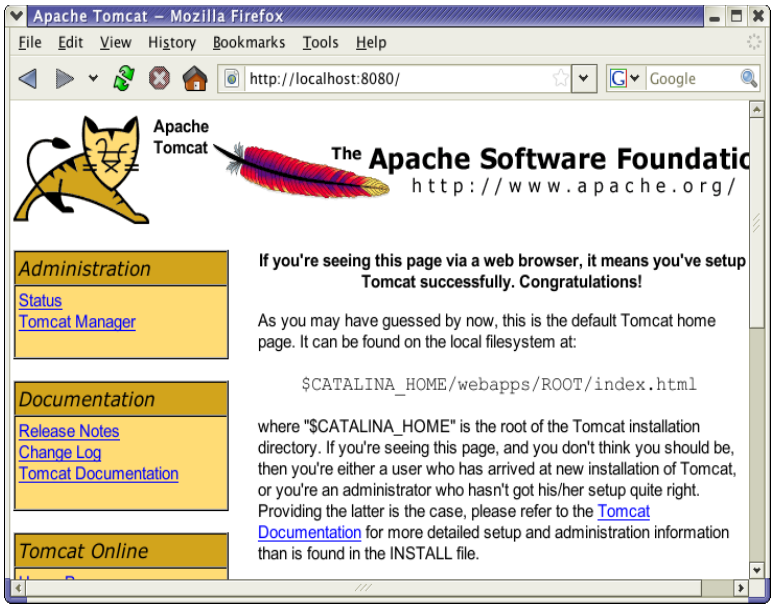
قم بفتح shell جديد و إكتب :

```
cd $HOME/tomcat/bin
./startup.sh
```

إفتح متصفحك المفضل و قم بالتوجه للعنوان الأتي :

http://localhost:8080/

ستلاحظ ظهور الصفحة التاليه:



و هذا يعني أنه تم التثبت و التشغيل بنجاح.
لغلق الخادم بقم بعمل التالي:

```
cd $HOME/tomcat/bin
./shutdown.sh
```

دعنا نبدأ أول مثال لنا:
قم بعمل ملجد جديد و ليكن إسمه Chapter1، قم بعمل مجلد بداخله و سميه WEB-INF
ملاحظه: إلتزم بشكل الأحرف في WEB-INF
داخل WEB-INF قم بعمل ملف و سمه web.xml و أكتب فيه التالي:

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
	xmlns="http://java.sun.com/xml/ns/javaee"
xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
	xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-
app_2_5.xsd"
	id="WebApp_ID" version="2.5">
	<display-name>Chapter1</display-name>
</web-app>
```

لا تقلق، سنقم بالشرح لاحقا.

بجانب المسار WEB-INFقم بعمل ملف و سمه index.jsp وقم بكتابة التالي به:

```
<%@page language="java" pageEncoding="UTF-8" %>
<html>
<head><title>Chapter1<title></head>
<body dir=”rtl”>
تحية الإسلام هي :
<br />

<b>
	<%
		out.println ("السلام عليكم و رحمة الله وبركاته");
	%>
</b>
</body>
</html>
```

بالنسبه للسطر :

```
<%@page language="java" pageEncoding="UTF-8" %>
```

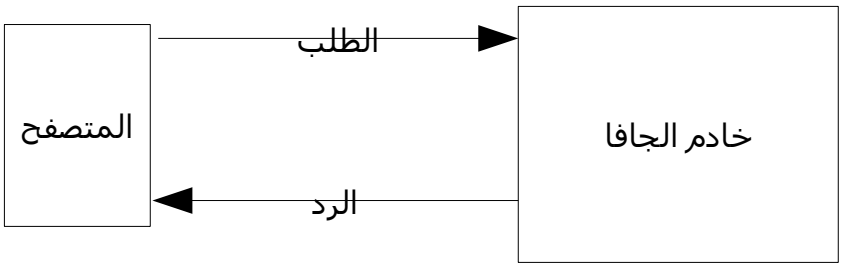
يسمي JSP Directive (أي موجه ال JSP) و سوف نتحدث عن موجهات ال jsp لاحقا.
لاحظ الأسطر من ٩-١١:

```
<%
	out.println ("السلام عليكم و رحمة الله وبركاته");
%>
```

فنحن هنا نطلب من المستعرض أن يجعل الخادم هو من يقوم بتفسير هذا الجزء من الكود.
الداله println الموجوده في المتغير out تقوم بطباعة مخلاتها إلى response stream

تماما كما كنت تتعامل مع الجداول (Streams) في ال IO .
فالمتغير out هي يمثل الجدول الخاص بالرد.
دعني أفسر أكثر، كل طلب ترسله للخادم يصحبه تكوين متغيرين ؛ متغير الطلب و متغير الرد (request variable and response variable) ، فمتغير الطلب هنا هو عباره عن كائن من النوع HttpServletRequest و متغير الرد هنا هو عباره عن كائن من نوع HttpServletResponse وكلاهما موجود تحت الحقيبه ((package javax.servlet.http

لاحظ الشكل التالي:



و المتغير out الذي تراه في المثال، ماهو إلا ال output Stream (جدول المخرجات) الخاص بالمتغير response، فمن البديهي أن يكون **للطلب** جدول مدخلات يحتوي على البيانات التي يقوم بإرسالها مستعرض الويب إلي الخادم، و **للرد** جدول للمخرجات يقوم الخادم بإرسالها إلي مستعرض الويب ليعرضها للمستخدم.

لاحظ أننا لم نقوم بالحصول علي كائن الرد ثم الحصول على المتغير out منه، فمن المفترض أن الكود كان يكون بالشكل التالي:

```
HttpServletResponse response =  .....
OutputStream out = response.getOutputStream();
out.println ("السلام عليكم و رحمة الله وبركاته");
```

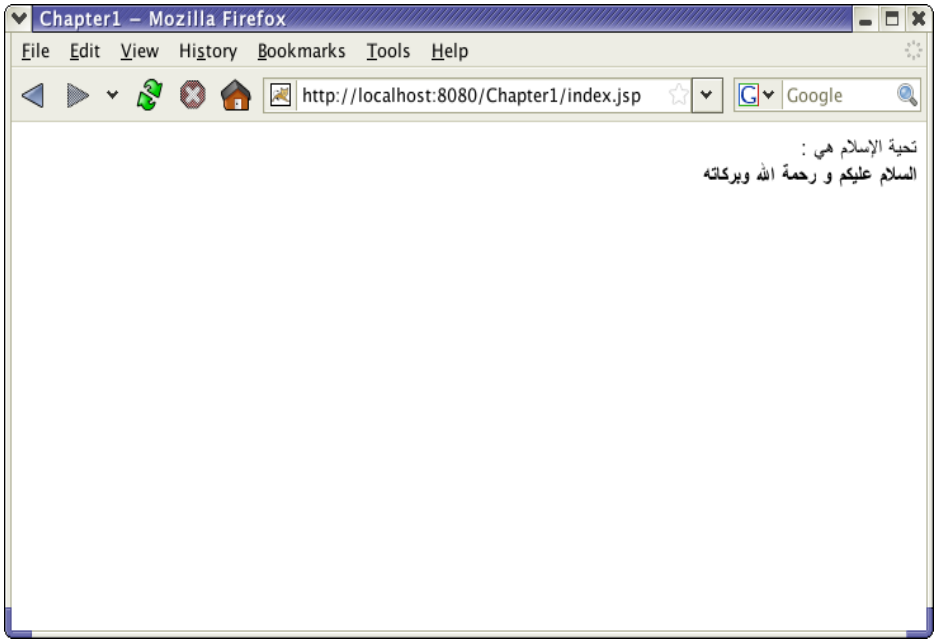
فماذا حدث، ولم تم إختزال هذا الكود في السطر الموضح بالمثال أعلاه؟؟
هذا لأن أي صفحة JSP بها عدد من ما يسمي ب ال implicit objects أي الكائنات الضمنيه، وهي كائنات موجوده بين الضفتين <% و %>، وماعليك إلا إستخدامها مباشره، لأن المحرك (الخادم) سوف يقوم بإنتاجها لك مباشرة.

دعنا الأن نقوم بحزم التطبيق لكي نقوم بعمل نشر (deployment) له داخل الخادم.
إذهب إلي المجلد Chapter1 و قم بتنفيذ الأوامر التاليه:

```
jar cf Chapter1.war WEB-INF/ index.jsp
```

لاحظ تكون الملف Chapter1.war لك في نفس المسار.
قم بنسخه إلي المجلد webapps داخل تومكات ثم قم بتشغيل تومكات ووجه متصفحك للعنوان
http://localhost:8080/Chapter1/index.jsp

```
mv Chapter1.war ~/tomcat/webapps
cd ~/tomcat/bin
./startup.sh
```



قم بفتح بعرض الكود الناتج عن الصفحه ستجد الأتي:

```
<html>
<head><title>Chapter1<title></head>
<body dir="rtl">
تحية الإسلام هي :
<br />

<b>
        السلام عليكم و رحمة الله وبركاته

</b>
</body>
</html>
```

لاحظ يا أخي، فإن الكود الناتج من الصفحة ليس بها أي كود خاص بال JSP

هينئأ لك يا أخي، لقد قمت بكتابة و نشر تطبيق ويب بالجافا بعدة خطوات سهله.
أرأيت، الموضوع ليس بالمعقد، ولكن قد يبدوا لك غريبا بعض الشئ و لكنه موضوع شيق يستحق المتابعه.