

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

لغة البرمجة جافا

Java Programming Language

الدرس الرابع عشر

برمجة الويب

8-1. التطبيقات الصغيرة (Applets) .

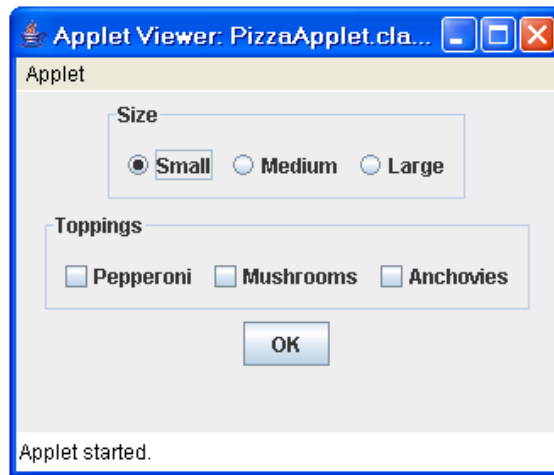
التطبيق الصغير (Applet) هو تطبيق بلغة الجافا صمم لتنفيذه في نافذة متصفح الانترنت. وعندما يقوم المستخدم بتصفح موقع ما أو صفحة ما على الانترنت متضمنة تطبيقاً صغيراً يتم تحميل الملف البايبي لذلك التطبيق الصغير (Applet) إلى حاسب المستخدم حيث يتم تنفيذه في مساحة محددة من نافذة المتصفح . وفي معظم الحالات تكون التطبيقات الصغيرة عبارة عن تطبيقات رسومية شبيهة بالتطبيقات ذات الواجهات الرسومية (Swing Application) . وفيما يلي توضيح لبعض النقاط المختلفة بينهما :

- كل تطبيق صغير (Applet) ممتد للصف (JApplet)، في حين أن كل تطبيق ذو واجهة رسومية (Swing Application) ممتد للصف (JFrame). كما أن المساحة المخصصة لكل تطبيق (Applet) صغير عند تصفح الانترنت تعالج من قبل متصفح الانترنت ، في حين أن المساحة المخصصة للتطبيق (Swing Application) تعالج من قبل نظام التشغيل وأدواته الرسومية .
- تنفذ التطبيقات (Swing Application) بشكل عام باستدعاء الدالة الساكنة انطلاقاً من كهيئة غرض يمثل التطبيق الممتد للصف (JFrame)، وبالمقابل فإن متصفح الانترنت يقوم بتهيئة غرض يمثل التطبيق الصغير (Applet) والممتد للصف (JApplet) ويقوم باستدعاء الدالة (init()) والتي تقوم بتشغيل التطبيق الصغير .
- يتم إغلاق التطبيقات الصغيرة بإغلاق صفحة الانترنت أو متصفح الانترنت ، وبالمقابل فإن تطبيقات الجافا بشكل عام يتم إغلاقها بزر الإغلاق الخاص بالإطار أو زر للخروج خاص بالتطبيق . وهذا ما لا نجد له لدى التطبيقات الصغيرة عند تحميلها وتشغيلها ضمن صفحات الانترنت فهي لا تتضمن زرراً للإغلاق ولا عنواناً أيضاً وهذه النافذة لا يمكن تغيير حجمها . ومن أهم الوابي والدوال الهامة المتعلقة بالصف (JApplet) موضحة بالجدول الآتي :

Constructor and methods	Description
JApplet()	الباني الافتراضي

void add(Component c)	دالة لإضافة كائن رسومي
void init()	دالة تحميل التطبيق الصغير
void setLayout(LayoutManager layout)	دالة إسناد مدير التصميم
void setLocation(int x, int y)	دالة تحديد موضع التطبيق الصغير
void setLocationRelativeTo(Component c)	دالة توسيط نافذة التطبيق الصغير وسط الشاشة
void setSize(int width,int height)	دالة تحديد أبعاد التطبيق الصغير
void start()	دالة تشغيل التطبيق الصغير
void stop()	دالة إيقاف عمل التطبيق الصغير

ولنحاول بناء تطبيق صغير اعتماداً على تطبيق سابق ، بحيث يأخذ الصورة أدناه عند طلب صفحة الانترنت المتضمنة له :



ولإنجاز ذلك نكتب صيغة التطبيق الصغير (Applet) كما هو موضح أدناه ، ومن ثم ترجمته وتحرير صفحة الانترنت التي يجب أن تتضمنه ولتكن (PizzaPage.html) ، وأخيراً يتم تصفح تلك الصفحة باستخدام الأمر (appletviewer) من محث الأوامر أي بالشكل :

```
C:\javac PizzaApplet.java
C:\appletviewer PizzaPage.html
```

```

import javax.swing.*;
import java.awt.event.*;
import javax.swing.border.*;
public class PizzaApplet extends JApplet
{
    private JButton buttonOK;
    private JRadioButton small, medium, large;
    private JCheckBox pepperoni, mushrooms, anchovies;
    public void init()
    {
        this.setSize(320,200);
        ButtonListener bl = new ButtonListener();
        JPanel mainPanel = new JPanel();
        JPanel sizePanel = new JPanel();
        Border b1 =BorderFactory.createTitledBorder("Size");
        sizePanel.setBorder(b1);
        ButtonGroup sizeGroup = new ButtonGroup();
        small = new JRadioButton("Small");
        small.setSelected(true);
        sizePanel.add(small);
        sizeGroup.add(small);
        medium = new JRadioButton("Medium");
        sizePanel.add(medium);
        sizeGroup.add(medium);
        large = new JRadioButton("Large");
        sizePanel.add(large);
        sizeGroup.add(large);
        mainPanel.add(sizePanel);
        JPanel topPanel = new JPanel();
        Border b2 =BorderFactory.createTitledBorder("Toppings");
        topPanel.setBorder(b2);
        pepperoni = new JCheckBox("Pepperoni");
        topPanel.add(pepperoni);
        mushrooms = new JCheckBox("Mushrooms");
        topPanel.add(mushrooms);
        anchovies = new JCheckBox("Anchovies");
        topPanel.add(anchovies);
        mainPanel.add(topPanel);
        buttonOK = new JButton("OK");
        buttonOK.addActionListener(bl);
    }
}

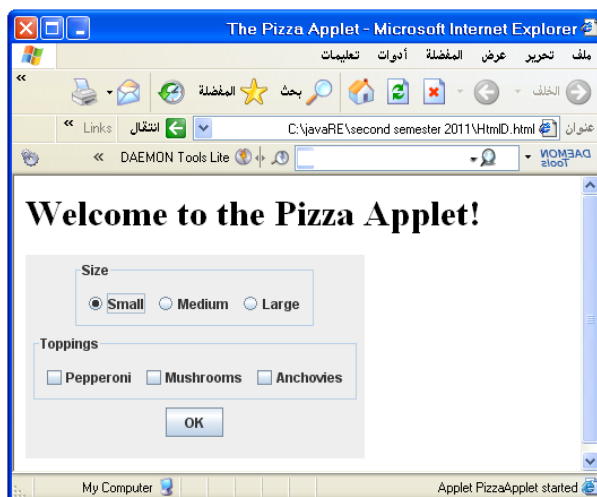
```

```
mainPanel.add(buttonOK);this.add(mainPanel);
this.setVisible(true);
}
private class ButtonListener implements ActionListener
{
public void actionPerformed(ActionEvent e)
{
if (e.getSource() == buttonOK)
{
String tops = "";
if (pepperoni.isSelected())
tops += "Pepperoni\n";
if (mushrooms.isSelected())
tops += "Mushrooms\n";
if (anchovies.isSelected())
tops += "Anchovies\n";
String msg = "You ordered a ";
if (small.isSelected())
msg += "small pizza with ";
if (medium.isSelected())
msg += "medium pizza with ";
if (large.isSelected())
msg += "large pizza with ";
if (tops.equals(""))
msg += "no toppings.";
else
msg += "the following toppings:\n" + tops;
JOptionPane.showMessageDialog(buttonOK, msg,
"Your Order",JOptionPane.INFORMATION_MESSAGE);
pepperoni.setSelected(false);
mushrooms.setSelected(false);
anchovies.setSelected(false);
small.setSelected(true);
}
}
}
}
```

كما يمكننا تحميل وتنفيذ هذا التطبيق الصغير بعد ترجمته وتحرير صفحة الانترنت المتضمنة له ، من خلال تصفح تلك الصفحة والتي تأخذ الصيغة التالية :

```
<html>
  <head>
    <title>The Pizza Applet</title>
  </head>
  <body>
    <H1>Welcome to the Pizza Applet!</H1>
    <APPLET code="PizzaApplet" width="300" height="180">
      Sorry, your browser isn't able to run Java applets.
    </APPLET>
  </body>
</html>
```

وعندها يأخذ التطبيق الصغير الشكل التالي كما هو موضح بالصورة أدناه :



8-2. تطبيقات المخدم الصغيرة (Servlets).

إن استخدام تطبيقات المخدم الصغيرة من أشهر الطرق لتطوير تطبيقات الويب . فالعديد من المواقع الشهيرة حالياً تعتمد على تطبيقات المخدم الصغيرة بحيث تمنحها القوة والفعالية. وقبل أن نستعرض ما هي تطبيقات المخدم الصغيرة ، لا بد من فهم آلية وعمل مخدّمات الويب .

إن مخدّمات الويب تستخدم البروتوكول (http) لإرسال الصفحات والوثائق لمتصفح الانترنت . فالمستخدم عادة يقوم بطلب الصفحة من خلال استخدام (URL) في برنامج متصفح الانترنت مع الاستعانة بالبروتوكول (http) ، حيث أن هذا البروتوكول (بروتوكول نقل النصوص التشعبية) يستخدم النموذج (طلب - إجابة) والذي يعني باختصار في أن المستخدم (الزبون) يرسل طلب للمخدم ، والذي بدوره يرد بالإجابة ويرسل المطلوب . ويمكن توضيح ذلك بالخطوات التالية :

١. تشغيل متصفح الانترنت وكتابة الرابط (URL) الذي يحدد مكان وجود الملف أو الصفحة التي نريد الوصول إليها .
٢. يرسل المتصفح طلب (رسالة طلب HTTP) إلى المخدم المطلوب والمعرف في الرابط .
٣. يستقبل المخدم الطلب ويرسل الملف المطلوب على شكل (رسالة رد HTTP).
٤. يستقبل المتصفح الملف على شكل صفحة انترنت يقوم المتصفح بتفسيرها وعرض الناتج.

وفي حقيقة الأمر يمكن أن نلاحظ بأن الاستخدام الأكثر لشبكة الانترنت يصب في التصفح، وبالتالي مما يضيفي صفة السكون أو الساكنة (Static) لمثل هذا الاستخدام (الصفحات ساكنة مهما تصفحتها فهي لا تتغير) . وبناء عليه فإن تطبيقات المخدم الصغيرة (servlets) قد أضفت شيء من الحركية لمتوى الصفحات والمواقع . وهي عبارة عن تطبيقات بلغة الجافا أي صفوف ممتدة للصف (javax.servlet.Servlet) ، وهذا الصف (Servlet) يدعم تنفيذ تطبيقات المخدم الصغيرة (servlets) على مخدم الويب بحيث يستجيب لطلب الزبون وإرسال ناتج تنفيذ تطبيق المخدم الصغير (servlet) إلى الزبون على شكل صفحة ويب . وتحصل هذه الاستجابة وفق الخطوات السابقة باستثناء الخطوة الثالثة التي تتلخص في الآتي :

٣. يستقبل المخدم الطلب والمتضمن تطبيق المخدم الصغير ليقوم لاحقاً بالبحث عن تطبيق الجافا (الملف البايتي) ويقوم بتنفيذه على المخدم ومن ثم يرسل ناتج التنفيذ على شكل (رسالة رد HTTP).

ويعني آخر بدلاً من إرسال محتوى الملف المطلوب يقوم المخدم بإرسال ناتج تنفيذ تطبيق المخدم الصغير (servlet) وبكل تأكيد هذا الناتج هو على شكل صفحة انترنت .
ولتنفيذ تطبيقات المخدم الصغيرة (servlets) على الحواسيب الشخصية لأهداف تدريبية وتعليمية لا بد من تنصيب البرمجيات اللازمة لجعل الحاسب مخدمًا قادراً على تنفيذها . ومن أشهر تلك الحزم والبرمجيات (tomcat) والتي يمكن أن تعمل كمخدم ويب. وفيما يلي خطوات تنصيب (tomcat 5.5) :

١. تحميل الملف المضغوط الذي يتضمن الحزمة (tomcat) من موقع (Apache web site).
٢. فك الضغط للحزمة السابقة ووضعها بمجلد اسم اختياري والأنسب (C:\tomcat).
٣. إضافة المتحول (JAVA_HOME) ضمن متحولات ومتغيرات بيئة نظام التشغيل والذي يشير إلى مكان وجود بيئة الجافا (JDK_5)، حيث يمكن الوصول لنافذة الإضافة بالشكل :
(Control Panel --> System --> Advanced Tab)
٤. نسخ الملف (C:\tomcat\common\lib\servlet-api.jar) إلى المجلد (C:\JDK_5\jre\lib\ext)
٥. تحرير ملف التهيئة (C:\tomcat\conf\context.xml) بإضافة (reloadable="true") إلى الوسم (<context>) بحيث يصبح بالشكل :
<Context reloadable="true">
٦. تعديل الملف (C:\tomcat\conf\web.xml) بإزالة التعليق في الأماكن التالية (أي إزالة الأسطر المتضمنة للإشارات (<!-- , -->) :

```
<!--
<servlet>
  <servlet-name>invoker</servlet-name>
  <servlet-class>
    org.apache.catalina.servlets.InvokerServlet
  </servlet-class>
  <init-param>
    <param-name>debug</param-name>
    <param-value>0</param-value>
  </init-param>
  <load-on-startup>2</load-on-startup>
</servlet>
-->
```

```
<!--
<servlet-mapping>
  <servlet-name>invoker</servlet-name>
  <url-pattern>/servlet/*</url-pattern>
</servlet-mapping>
-->
```

٧. إنشاء المجلد (C:\tomcat\webapps\ROOT\WEB-INF\classes)

ملاحظة هامة : بعد الانتهاء من تنصيب (tomcat) يمكن تشغيل وإغلاق خادم الويب خاصتك بإدخال الأوامر التالية من محث الأوامر :

```
C:\tomcat\bin>startup
C:\tomcat\bin>shutdown
```

مع التنويه إلى ضرورة تشغيل خادم الويب هذا عندما نريد تصفح صفحات تتضمن تطبيقات خادم صغيرة (servlets) .

إن معظم تطبيقات الخدم الصغيرة (servlets) تحتاج على الأقل استيراد الحزم البرمجية

التالية:

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

وكونها تطبيقات بلغة جافا فهي صفوف وممتدة للصف (HttpServlet) والذي يتضمن مجموعة من الدوال الموضحة بالجدول الآتي يمكن لتطبيق الخدم الصغير إعادة صياغتها :

methods
doDelete(HttpServletRequest request, HttpServletResponse response) throws IOException, ServletException
doGet(HttpServletRequest request, HttpServletResponse response) throws IOException, ServletException
doPost(HttpServletRequest request, HttpServletResponse response) throws IOException, ServletException
doPut(HttpServletRequest request, HttpServletResponse response) throws IOException, ServletException
void init() throws ServletException First time servlet is run
destroy() Servlet is destroyed

ولنحاول الآن إنشاء تطبيق مخدم صغير (servlet) ويتلخص في أنه عندما يقوم الزبون بطلب صفحة أو ملف يتضمن ذلك التطبيق الصغير (servlet) ، فإن تطبيق المخدم الصغير (servlet) يرد بالإجابة ويرسل صفحة انترنت تتمثل في ناتج تنفيذه . وللقيام بذلك نتبع ما يلي :

الخطوة الأولى : نقوم بكتابة نص تطبيق المخدم الصغير (servlet) كما هو موضح بالصيغة الآتية :

```
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>HelloWorld</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("<h1>Hello, World!</h1>");
        out.println("</body>");
        out.println("</html>");
    }
}
```

الخطوة الثانية : نقوم بترجمة تطبيق المخدم الصغير (servlet) . حيث نلاحظ بأن التطبيق السابق هو تطبيق بلغة الجافا قابل للترجمة غير قابل للتنفيذ لعدم احتوائه على الدالة الرئيسية (main()) ، لذا لابد من ترجمته ووضع الملف البايتي في الدليل :

C:\tomcat\webapps\ROOT\WEB-INF\classes

الخطوة الثالثة : نقوم بتشغيل مخدم الويب (tomcat) كما هو موضح بالصيغة الآتية :

C:\tomcat\bin>startup

الخطوة الرابعة : نقوم بتشغيل متصفح الانترنت وندخل العنوان الآتي :

http://localhost:8080/servlet/HelloWorld

ويأتباع الخطوات الأربع السابقة نحصل على النتيجة الموضحة بالصورة التالية :



مع التنويه إلى أن ناتج تنفيذ تطبيق المخدم الصغير على المخدم هو الصفحة الآتية :

```
<html>
  <head>
    <title>HelloWorld</title>
  </head>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>
```

ونلاحظ بأن تطبيق المخدم الصغير السابق (HelloWorld) يعيد صياغة الدالة (doGet()) ولنحاول الآن تحسين تطبيق المخدم الصغير البسيط السابق ولنسمه (HelloServlet) بإضافة بعض الديناميكية عليه بحيث يقوم بعرض رسالة ترحيبية بشكل عشوائي من أصل ست رسائل ، كما هو موضح بالصيغة الآتية:

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

```

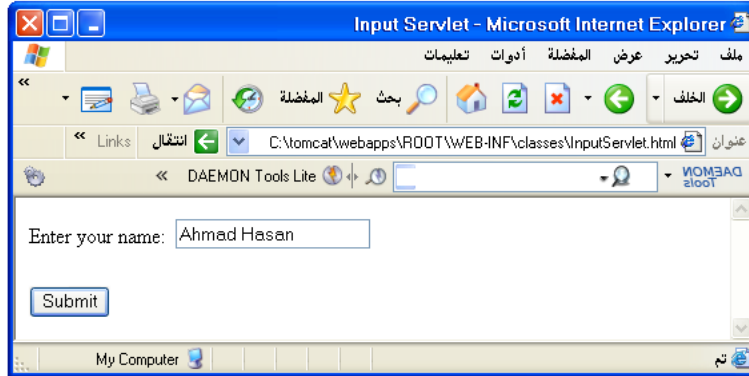
import javax.servlet.http.*;
import java.util.*;
public class HelloServlet extends HttpServlet
{
    public void doGet(HttpServletRequest request,HttpServletResponse
        response) throws IOException, ServletException
    {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        String msg = getGreeting();
        out.println("<html>");
        out.println("<head>");
        out.println("<title>HelloWorld Servlet</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("<h1>");
        out.println(msg);
        out.println("</h1>");
        out.println("</body>");
        out.println("</html>");
    }
    private String getGreeting()
    {
        String msg = "";
        int rand = (int)(Math.random() * (6)) + 1;
        switch (rand)
        {
            case 1:
                return "Hello!";
            case 2:
                return "Greetings!";
            case 3:
                return "Hi!";
            case 4:
                return "Good morning!";
            case 5:
                return "Good evening!";
            case 6:
                return "Good bye!";
        }
    }
}

```

```
        return null;
    }
}
```

ولنحاول مرة أخرى تحسين وتطوير تطبيق المخدم الصغير السابق ولنسمه بالاسم الجديد (InputServlet) بحيث يقوم بعرض رسالة (اسم ما ، جملة ما) تُرسل له من شاشة المستخدم على شكل نموذج ليعيدها المخدم من خلال تنفيذ تطبيقه الصغير . وفي هذه الحالة نقوم بإعادة صياغة الدالة (doPost) واستدعاء الدالة (getParameter()) من أجل عرض الطلب بحيث تعيد القيم المدخلة من قبل المستخدم عبر نمودجه لتصل المخدم وتطبيقه الصغير . ولإنجاز ذلك لا بد في البداية من تصميم وتحرير صفحة انترنيت تتضمن نموذجاً بالصيغة والصورة التاليتين :

```
<html>
  <head>
    <title>Input Servlet</title>
  </head>
  <body>
    <form action="/servlet/InputServlet" method="post">
      Enter your name:&nbsp;
      <input type="text" name="Name"> <br><br>
      <input type="submit" value="Submit">
    </form>
  </body>
</html>
```



وعندئذ بتصفح صفحة الانترنت السابقة يظهر النموذج المؤلف من مربع إدخال نصي يتيح للمستخدم إدخال اسمه على سبيل المثال . وبإنتهاء عملية الإدخال والضغط على الزر (Submit) يتم طلب تصفح تطبيق المخدم الصغير (InputServlet) فيقوم المخدم بتنفيذه على المخدم ويعيد صفحة انترنت بالصيغة والصورة التاليتين :

```
<html>
  <head>
    <title>Input Servlet</title>
  </head>
  <body>
    <h1>Hello Ahmad Hasan</h1>
  </body>
</html>
```



وتكون بذلك صيغة تطبيق المخدم الصغير كالتالي :

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class InputServlet extends HttpServlet
{
    public void doGet(HttpServletRequest request,HttpServletResponse
        response) throws IOException, ServletException
    {
        String name = request.getParameter("Name");
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Input Servlet</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("<h1>");
        out.println("Hello " + name);
        out.println("</h1>");
        out.println("</body>");
        out.println("</html>");
    }
    public void doPost(HttpServletRequest request,HttpServletResponse
        response) throws IOException, ServletException
    {
        doGet(request, response);
    }
}
```

تمرين : بفرض لدينا ملف نصي يحوي بعض البيانات المتعلقة بمجموعة من أقراص الأفلام

السينمائية (اسم الفيلم ، تاريخ العرض ، السعر) كالتالي :

It's a Wonderful Life	1946	14.95
The Great Race	1965	12.95

Young Frankenstein	1974	16.95	
The Return of the Pink Panther	1975	11.95	
Star Wars	1977	17.95	
The Princess Bride	1987	16.95	
Glory	1989	14.95	
Apollo13	1995	19.95	
The Game	1997	14.95	
The Lord of the Rings:The Fellowship of the Ring			2001 19.95

ولنحاول بناء تطبيق مخدم صغير يقوم باستخلاص بعض البيانات من هذا الملف . ومن أجل ذلك نقوم بإنشاء ثلاث صفوف وهي كالتالي :

-الصف (Movie) لتوصيف مجموعة الأفلام السابقة ويمكن أن يأخذ الصيغة الآتية :

```
public class Movie
{
    public String title;
    public int year;
    public double price;
    public Movie(String title, int year, double price)
    {
        this.title = title;
        this.year = year;
        this.price = price;
    }
}
```

-الصف (MovieIO) بالقراءة من الملف النصي وبناء قائمة لإضافة بيانات هذا الملف ، حيث أن تطبيق المخدم الصغير سيستعين بتلك القائمة للرد وإرسال المطلوب للمستخدم . ويمكن أن يأخذ الصيغة الآتية :

```
import java.io.*;
import java.util.*;
public class MovieIO
{
    public static ArrayList<Movie> getMovies()
    {
```

```
ArrayList<Movie> movies = new ArrayList<Movie>();
BufferedReader in = getReader("c:\\tomcat\\webapps\\root\\web-
                               inf\\classes\\MovieFile.txt");

Movie movie = readMovie(in);
while (movie != null)
{
    movies.add(movie);
    movie = readMovie(in);
}
return movies;
}
private static BufferedReader getReader(String name)
{
    BufferedReader in = null;
    try
    {
        File file = new File(name);
        in = new BufferedReader( new FileReader(file) );
    }
    catch (FileNotFoundException e)
    {
        System.out.println("The file doesn't exist.");
        System.exit(0);
    }
    catch (IOException e)
    {
        System.out.println("I/O Error");
        System.exit(0);
    }
    return in;
}
private static Movie readMovie(BufferedReader in)
{
    String title;
    int year;
    double price;
    String line = "";
    String[] data;
    try
    {
```

```

        line = in.readLine();
    }
    catch (IOException e)
    {
        System.out.println("I/O Error");
        System.exit(0);
    }
    if (line == null)
        return null;
    else
    {
        data = line.split("\t");
        title = data[0];
        year = Integer.parseInt(data[1]);
        price = Double.parseDouble(data[2]);
        return new Movie(title, year, price);
    }
}
}
}

```

-الصف (MovieServlet) ويمثل تطبيق المخدم الصغير والذي يأخذ الصيغة التالية:

```

import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.util.*;
public class ListMovies extends HttpServlet
{
    public void doGet(HttpServletRequest request,HttpServletResponse
        response) throws IOException, ServletException
    {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        String msg = getMovieList();
        out.println("<html>");
        out.println("<head>");
        out.println("<title>List Movies Servlet</title>");
    }
}

```

```
        out.println("</head>");
        out.println("<body>");
        out.println("<h1>Some of My Favorites</h1>");
        out.println("<h3>");
        out.println(msg);
        out.println("</h3>");
        out.println("</body>");
        out.println("</html>");
    }
    public void doPost(HttpServletRequest request, HttpServletResponse
        response) throws IOException, ServletException
    {
        doGet(request, response);
    }
    private String getMovieList()
    {
        String msg = "";
        ArrayList<Movie> movies = MovieIO.getMovies();
        for (Movie m : movies)
        {
            msg += m.year + ": ";
            msg += m.title + "<br>";
        }
        return msg;
    }
}
```

ولرؤية ما يقوم به تطبيق المخدم الصغير (MovieServlet) بإتباع الخطوات التي توضح تصفح تطبيقات المخدم الصغيرة نحصل على النتيجة الموضحة بالصورة التالية :



مع التنويه إلى أن ناتج تنفيذ تطبيق المستخدم الصغير على المستخدم هو الصفحة الآتية :

```
<html>
<head>
  <title>List Movies Servlet</title>
</head>
<body>
  <h1>Some of My Favorites</h1>
  <h3>
    1946: It?s a Wonderful Life<br>
    1965: The Great Race<br>
    1974: Young Frankenstein<br>
    1975: The Return of the Pink Panther<br>
    1977: Star Wars<br>
    1987: The Princess Bride<br>
    1989: Glory<br>
    1995: Apollo 13<br>
    1997: The Game<br>
    2001: The Lord of the Rings:The Fellowship of the Ring<br>
  </h3>
</body>
</html>
```

سأهم بنشر الكتاب ولك الأجر والثواب إن شاء الله

لا تنسوني من صالح دعائكم

تم بحمد الله