LABORATORIO 9:

- JAVA.IO - ENTRADA/SALIDA
- JAVA.NET - URL

Versión:2013-05-20

Conocimientos previos

- Temario visto en las clases de teoría.
- Se recomienda haber leído en casa esta guía antes de asistir a la sesión de laboratorio.

Objetivos

1. Practicar con las clases del paquete java.io dedicadas a manejar las tareas de entrada/salida y uso de ficheros.
2. Practicar en el uso de la clase java.net.URL para acceder a páginas web y descargar su contenido.

Documentación para el alumno

- API de java. (http://download.oracle.com/javase/6/docs/api/)
- Guía de alumno – Laboratorio 9.
- Recursos disponibles en moodle para este día.

Trabajo previo

Esta tarea se deberá realizar en horario libre, **antes de la sesión de laboratorio 9 de su grupo**:

Conteste a las siguientes preguntas:

- En que package de java se encuentra la clase URL.

- ¿Una persona puede leer y entender el código HTML de una página de Internet sin usar un navegador? ¿Porqué?
- Cual es el tipo del fichero necesario para descargar en mi ordenador una foto de Internet.
- Quiero crear el fichero /tmp/ejemplo/datos/hola.txt, pero en mi disco no existen los directorios especificados en esa ruta. ¿Cómo se pueden crear esos directorios?
- ¿Qué diferencia hay entre un InputStream y un Reader?

ACTIVIDADES

Descripción de las actividades

El objetivo final de esta sesión de laboratorio es crear un programa que descargue todas las fotografías usadas en una página Web escrita en HTML.

Dado el URL de una pagina web, el programa se conectará a dicha página, descargará su contenido, y lo analizará buscando referencias a fotografías. Para cada fotografía encontrada, se conectará al sitio web donde está alojada la fotografía y copiará su contenido en un fichero local.

Así, una vez terminada la ejecución del programa, tendremos en nuestro disco duro una copia de todas las fotografías usadas en la página web analizada.

Esta sesión de laboratorio está dividida en cuatro actividades. En cada actividad irán desarrollándose pequeñas tareas de entrada/salida y manejo de URLs que terminarán con la realización de la aplicación descrita. Las actividades a realizar son:

- **Actividad 1**: Conectarse a una página Web con contenido HTML y mostrar el código HTML por la pantalla.
- **Actividad 2**: Cambiar la actividad 1 para que el contenido de la página HTML se escriba en un fichero.
- **Actividad 3**: Ampliar la actividad 2 para analizar el contenido de la página HTML buscando referencias a fotografías. Se escribirá por pantalla el URL de las fotografías usadas en la página HTML.
- **Actividad 4**: Modificar la actividad 3 para que se descarguen en ficheros las fotografías referenciadas.

Se ha creado en moodle una clase llamada `DescargaFotos` que puede usarse para realizar todas las actividades propuestas. Para cada actividad se ha creado un método diferente que el alumno deberá completar. Modifique el método ***main*** de esta clase para probar los métodos que desarrolle.

Para realizar estas actividades debe crear un proyecto java en Eclipse llamado ***Laboratorio9*** y un paquete llamado ***es.upm.dit.prog.l9.*** Copie el fichero ***DescargaFotos.java*** en este paquete.

ACTIVIDAD 1

Mostrar por pantalla una página HTML

1.1 Descripción

En esta actividad se completará el método:

```
void leeHTML(URL url) throws Exception;
```

Este método debe mostrar por pantalla el contenido de la página web apuntada por el parámetro **url**. Supondremos que la página Web está escrita en HTML. Si no es una página HTML, el método podría escribir cualquier tipo de basura por pantalla, pero no nos preocuparemos de este problema en esta actividad.

El método lanzará excepciones si no puede realizar su tarea.

1.2 Prueba

Para probar esta actividad modifique el método ***main*** para ejecutar las siguientes sentencias:

```
URL url = new URL("http://www.dit.upm.es");
leeHTML(url);
```

En pantalla debe aparecer el código HTML de la página:

```
<?xml version="1.0" encoding="iso-8859-1"?><!DOCTYPE html PUBLIC "-//W3C//DTD
XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="es" xml:lang="es">
<head>
<title>DIT-UPM, Dpto. de IngenierÌa de Sistemas Telem·ticos - Inicio</title>
<meta name="description" content="Departamento de IngenierÌa de Sistemas
Telem·ticos, ETSI TelecomunicaciÛn, Universidad PolitÈcnica de Madrid, EspaÒa."
/>
<meta name="keywords" content="telem·tica, telecomunicaciÛn" />
<meta name="Generator" content="Joomla! - Copyright (C) 2005 - 2006 Open Source
Matters. All rights reserved." />
<meta name="robots" content="index, follow" />
<base href="http://www.dit.upm.es/" />
                    <link rel="alternate"

continúa . . . .
```

ACTIVIDAD 2

Descargar a fichero una página HTML

2.1 Descripción

En esta actividad se completará el método:

```
void copiaHTML(URL url,String destino) throws Exception;
```

Este método debe copiar el contenido de la página web apuntada por el parámetro ***url*** en un fichero. Supondremos que la página web a copiar está escrita en HTML.

El parámetro destino contiene el path y el nombre del fichero donde se copiará el contenido de la página HTML.

El método lanzará excepciones si no puede realizar su tarea.

2.2 Prueba

Para probar esta actividad modifique el método ***main*** para ejecutar las siguientes sentencias:

```
URL url = new URL("http://www.dit.upm.es");
copiaHTML(url, "/tmp/pagina.html");
```

En el directorio ***/tmp*** debería aparecer un fichero llamado ***pagina.html*** con el contenido de la pagina web analizada.

Utilice un navegador web para ver el fichero creado. Debería ver la página principal del DIT.

ACTIVIDAD 3

Buscar fotografías usadas en una página HTML

3.1 Descripción

En esta actividad se completará el método:

```
void buscaFotos(URL url) throws Exception;
```

Este método escribe por pantalla los URLs de las fotografías usadas en la página HTML apuntada por el parámetro ***url***.

Este método lanzará excepciones si no puede realizar su tarea.

En necesario que el contenido de la página sea HTML para que este método funcione. HTML se usa la etiqueta IMG para incluir una imagen en una página, donde el atributo SRC es la ruta o el URL a la imagen a incluir. Así la etiqueta:

```
<IMG SRC=”planeta.gif”>
```

inserta en una página web la fotografía ***planeta.gif***. Las palabras ***IMG*** y ***SRC*** pueden usarse tanto en mayúsculas como en minúsculas.

El método a realizar en esta actividad debe analizar línea a línea el contenido de la página HTML buscando cadenas de texto con el formato:

```
<IMG SRC="url_a_una_imagen">
```

Si encontramos una cadena de texto con el formato anterior, extraeremos el texto entre comillas (`url_a_una_imagen`), que será una referencia a la fotografía utilizada en la página. Escribiremos la URL de la fotografía por la pantalla.

El método `buscaFotos` debe escribir siempre un URL absoluto por pantalla. Nótese que el texto `url_a_una_imagen` puede ser un path relativo, un path absoluto, o una URL. Esto debe tenerse en cuenta para calcular cual es el URL absoluto de la imagen. Para facilitar esta tarea se proporciona un método ya resuelto, llamado ***extraeFotoUrlDeImg***.

```
URL extraeFotoUrlDeImg(URL url_base, String html)
                                        throws Exception
```

El método `extraeFotoUrlDeImg` explora un substring HTML en busca de una etiqueta ***IMG***, devolviendo el URL absoluto de la imagen apuntada, o `null` si no

encuentra una etiqueta ***IMG***. Si el substring HTML analizado contiene varias etiquetas ***IMG***, sólo se considerará la primera de ellas, ignorando las demás. El parámetro ***url_base*** es el URL que apunta a la página HTML que estamos analizando. El parámetro ***html*** es un String con una línea de la página HTML que estamos analizando.

3.2 Prueba

Para probar esta actividad modifique el método ***main*** para ejecutar las siguientes sentencias:

```
URL url = new URL("http://www.dit.upm.es");
buscaFotos(url);
```

Por pantalla debería aparecer los siguientes URLs (pueden variar si la página inicial del DIT ha sido modificada desde que se editó este documento):

```
http://www.dit.upm.es/figures/logos/dit08.gif
http://www.dit.upm.es/templates/spanish_red/images/css_larger.png
http://www.dit.upm.es/figures/logos/etsitandupm.gif
http://www.dit.upm.es/figures/logos/www2009madrid_text.gif
http://www.advanticsys.com/images/products/cm3000.jpg
```

ACTIVIDAD 4

Descargar fotografías usadas en página HTML

4.1 Descripción

En esta actividad se completará el método:

```
void copiaFotos(URL url, String directorio)
                                    throws Exception;
```

Este método descarga las fotografías usadas en la página HTML apuntada por el parámetro ***url***. Es una modificación de la actividad 3 en la que en vez de escribir por pantalla el URL de las fotografías, se crearán ficheros con las propias fotografías.

El parámetro ***url*** apunta a la página web HTML a explorar en busca de fotografías. El parámetro ***directorio*** es el nombre del directorio donde deben descargarse las fotografías, es decir, el directorio donde se crearán los ficheros con las copias de las fotografías.

4.2 Prueba

Para probar esta actividad modifique el método ***main*** para ejecutar las siguientes sentencias:

```
URL url = new URL("http://www.dit.upm.es");
copiaFotos(url, "/tmp");
```

En el directorio ***/tmp*** deberían aparecer los ficheros *dit08.gif*, *css_larger.png*, *etsitandupm.gif*, etc… Compruebe que su contenido son imágenes.

Hoja de Respuestas Laboratorio 9

A entregar al inicio de la sesión del Laboratorio 9 de su grupo

Pregunta 1: En que package de java se encuentra la clase URL.

Pregunta 2: ¿Una persona puede leer y entender el código HTML de una página de Internet sin usar un navegador? ¿Porqué?

Pregunta 3: ¿De que tipo será el fichero donde se copia una página htlm con el método copiaHTML.

Pregunta 4: Ponga dos ejemplos de URLs que referencien a fotos.

Pregunta 5: Cual es el tipo del fichero necesario para descargar en mi ordenador una foto de Internet..

Pregunta 6: Quiero crear el fichero /tmp/ejemplo/datos/hola.txt, pero en mi disco no existen los directorios especificados en esa ruta. ¿Cómo se pueden crear esos directorios?

Pregunta 7: ¿Qué diferencia hay entre un InputStream y un Reader?

FAQ

1 - Quiero crear el fichero /tmp/ejemplo/datos/hola.txt, pero en mi disco no existen los directorios especificados en esa ruta. ¿Cómo se pueden crear esos directorios?

```
String ruta = "/tmp/ejemplo/datos/hola.txt";
File f = new File(ruta);
f.getParentFile().mkdirs();
```

2 - ¿Qué diferencia hay entre un InputStream y un Reader?

El InputStream es para leer bytes, y el Reader para leer chars.