

## Ejemplo de como hacer configurable un testplan de JMeter

► Área: [Verificación de Entrega Software](#)

► Carácter del recurso: [Recomendado](#)

Código: RECU-0392

Tipo de recurso: Ejemplo

### Introducción

El ejemplo es un script de test para la aplicación eCO. Aunque la aplicación es lo de menos, y para entender las modificaciones que aquí exponemos no es imprescindible ejecutar el script, sí que es deseable disponer de un entorno con la aplicación eCO instalada contra el que usted pueda probar el script.

El ejemplo parte de un script no configurable, de nombre eCO\_alta\_NO\_CONFIGURABLE.jmx. Básicamente, lo que hace el script (cada thread que crea el script) es hacer login en la aplicación eCO con ciertas credenciales de usuario, crear un expediente con un documento adjunto, tramitarlo y salir de la aplicación. El objetivo del ejemplo es explicar las modificaciones que lo convierten en el script eCO\_alta.jmx, que sí tiene un cierto grado de configuración.

#### !! IMPORTANTE !!

Los ficheros de este ejemplo corresponden a un caso real, por lo que ambos scripts deberían de funcionar (el segundo, eCO\_alta.jmx con la configuración adecuada en eCO\_alta.properties). Sin embargo, por seguridad, antes de publicar los ficheros hemos eliminado las credenciales reales que pudieran contener. Esto significa que el script eCO\_alta\_NO\_CONFIGURABLE.jmx no funciona tal cual. Para hacerlo ejecutable, teclee el nombre de usuario y password en su sitio adecuado: sampler de nombre /eco/validarLogin.do

Por ahora, el ejemplo sólo dispone de un script .bat que automatiza la ejecución del script .jmx desde línea de comando. Como para usuarios no Windows esto no será útil, he aquí la línea de comando de JMeter para ejecutar el script configurable:

```
{JMeter_HOME}/bin/jmeter -n -t eCO_alta.jmx -Jjmeter.save.saveservice.output_format=csv -l eCO_alta.csv -j eCO_al
```

Para información de que es cada uno de los ficheros incluidos en el ejemplo, ver el fichero LEAME.txt.

### Descripción

Los scripts de JMeter (ficheros .jmx), deberían tener siempre un cierto grado de configuración. Por ejemplo, datos como el hostname o dirección IP de la máquina en la que está desplegada la aplicación objetivo que testea el script, no deberían formar parte del fuente. De lo contrario, el script no es ejecutable en otros entornos.

Este documento explica, con un ejemplo, un conjunto de modificaciones básicas para dotar de un cierto grado de configuración a un script de JMeter.

Se asume que dependiendo del script al que se apliquen, las modificaciones explicadas en este ejemplo pueden ser suficientes o no para dotarlo del grado de configuración deseado. Lo que se pretende en el documento es exponer el *estado mínimo de configuración* que deberían tener todos los scripts de JMeter, NO una exposición exhaustiva de todas las posibilidades de configuración.

Este ejemplo está orientado principalmente a testers, o técnicos de otro tipo, involucrados en la implementación de scripts para JMeter.

Se asume que el lector conoce bien los conceptos de *Propiedad*, *Variable*, y *Función*, y como se usan estos tres artefactos en la implementación de scripts en JMeter. Si no es su caso, consultar la referencia [Construcción de planes de prueba con JMeter](#).

### Desarrollo

Cada uno de los puntos que siguen corresponde a una modificación al script eCO\_alta\_NO\_CONFIGURABLE.jmx para convertirlo en eCO\_alta.jmx:

- Eliminar todos los recursos embebidos en el HTML

En el script original se observa que hay peticiones [HTTP](#) para obtener recursos como hojas de estilo CSS, código JavaScript e imágenes. Probablemente porque las peticiones del script se capturaron con el [HTTP](#) Proxy Server de JMeter, sin

especificar que estos tipos de recursos se deben excluir de la captura. Esta no es realmente una modificación que haga el script más configurable, sino una simplificación en orden a facilitar las modificaciones que siguen.

Si lo que se desea es que el testplan realice estas peticiones para tenerlas en cuentas en la medición del rendimiento, veremos más adelante que no se pierde funcionalidad por el hecho de suprimirlas. Esto se discute más adelante.

- Agrupar peticiones

En el script original todas las peticiones cuelgan de un único controlador. Esto hace imposible saber que peticiones [HTTP](#) corresponden al proceso de login en la aplicación, cuales al alta y tramitación del expediente, y cuales al logout. Además el controlador no tiene ningún nombre descriptivo. En el script destino hemos agrupado las peticiones en tres controladores, un grupo de peticiones hace el login, otro el alta y la tramitación, y otro el logout. Esto además de ser más descriptivo, facilita el que podamos reutilizar las peticiones por ejemplo del login y del logout si tenemos que ampliar el script implementando nuevas interacciones.

- Crear componentes de configuración "[HTTP](#) Request Default" para contener la configuración común a varias peticiones [HTTP](#)

En el script original se observa que todos los componentes [HTTP](#) Request se ejecutan contra el mismo hostname (prueba.eco.i-administracion.junta-andalucia.es), con el mismo protocolo ([http](#)), contra el mismo puerto (por defecto el 80), ... En el script modificado, estos datos comunes a todas las peticiones [HTTP](#) se han colocado en el componente [HTTP](#) Request Defaults, dejando en blanco estos campos en todos los componentes [HTTP](#) Request. Además, en este caso, solo hemos necesitado un componente de este tipo, colocado en el testplan de forma que aplique a todos los samplers. Obsérvese que este componente tiene un check Retrieve All Embedded Resources from HTML Files. Activando este check hacemos que para cada respuesta de la aplicación, JMeter haga tantas peticiones como sea necesario para obtener cada uno de los recursos referenciados en la respuesta, igual que haría un navegador con la caché desactivada. El tiempo de la petición es el tiempo de la petición padre más las peticiones hijas correspondientes a cada uno de los recursos embebidos. Por esta razón, decíamos más arriba, que no se pierde funcionalidad por el hecho de eliminar del .jmx las peticiones que obtienen recursos como hojas de estilo, imágenes, javascript, ...

- Crear un fichero .properties para contener propiedades de configuración del script, que se pasarán a éste en el momento de la ejecución, mediante llamadas a la función `__P` de JMeter.

En nuestro caso, el fichero se llama igual que el script de JMeter, pero con la extensión .properties

En este fichero de configuración hemos creado una propiedad para cada uno de los datos que queremos que sean configurables en el script resultante. A saber: eco.user.name.create, eco.user.pass.create, eco.host, eco.adjunto.filename, eco.tg.count, eco.tg.rampup, eco.tg.duration, eco.tg.delay. Además, y esto es importante el fichero de propiedades es un fichero autodocumentado.

- Modificar el panel de control del Thread Group para que sus campos tomen los valores de las propiedades del fichero `eCO_alta.properties`, mediante la función de propiedades, mediante la función `__P`. Nótese que para que esto funcione, hay que pasar el fichero de propiedades a JMeter con el modificador `-q`

- Crear variables en el script modificado

En el script modificado hemos añadido un componente User Defined Variables dentro del único Thread Group. Si hubiese más de un Thread Group habría que plantearse si lo más conveniente es un componente User Defined Variables por cada Thread Group o bien uno que afecte a todos los Thread Group

Nótese que hemos definido cuatro variables (`V_USR_NAME_CREATE`, `V_USR_PASS_CREATE`, `V_HOST`, `V_ADJUNTO_FILENAME`), cada una de las cuales toma un valor devuelto por la función `__P` (esta función devuelve o bien el valor de una propiedad pasada a JMeter con el modificador `-J` ó `-q`, o bien el valor por defecto que se especifique). También hemos especificado que se usen los valores de estas variables en aquellas peticiones donde sea necesario, concretamente las credenciales de usuario en `/eco/validarLogin.do`, la variable `V_HOST` la hemos especificado en el componente [HTTP](#) Request Defaults, y en todos los componentes [HTTP](#) Header Manager donde aparece la cabecera Referer.

En el Thread Group no es necesario utilizar variables, basta con llamadas a la función `__P` para capturar directamente los valores de propiedades.

- En muchas peticiones, se utiliza un identificador de expediente (parámetro `PKEXP`). Como este número de expediente cambia de una ejecución a otra, lo capturamos en una variable `V_PKEXP` de la respuesta a la petición `/eco/altaExpedienteFin.do`, mediante un componente Regular Expression Extractor. En las peticiones siguientes sustituimos por una llamada a la variable el valor de `PK_EXP`.

## Ejemplos

Para obtener los ficheros del ejemplo, descargar el archivo .zip adjunto a la sección Documentos.

## Documentos

 [Ejemplo eCO alta](#) (145.96 KB)

Source URL: <http://madeja.i-administracion.junta-andalucia.es/servicios/madeja/contenido/recurso/392>