

Servei Característiques dispositius mòbils (WURFL) amb Canigó3

A qui va dirigit

Aquest document va dirigit a tots aquells que hagin d'emprar el servei de característiques de dispositius mòbils amb Canigó3.

Versió de Canigó

Els passos descrits en aquest document són d'aplicació a les versions de Canigó 3.

Introducció

El WURFL és un repositori de descripció de terminals (<http://wurfl.sourceforge.net/>) que conté informació sobre milers de terminals mòbils. Consisteix en un fitxer XML que actua com una base de dades amb informació dels diferents terminals: resolució de pantalla, si suporta javascript, fabricant...

El principal objectiu d'aquest repositori és crear una base de coneixement centralitzada i independent sobre terminals mòbils. A més, es tracta d'un projecte de programari lliure.

Per poder consultar la informació referent a un terminal, existeix una API Java que facilita la consulta d'aquesta informació des d'una aplicació J2ee. Aquesta API pot obtenir "tota" la informació del terminal a partir de l'user-agent, que és una capçalera que s'envia en totes les peticions HTTP i que normalment serveix per identificar un navegador.

Per obtenir més informació d'aquest repositori podeu consultar les següents adreces:

Documentació de l'API: <http://www.scientiamobile.com/api-doc/java/>

Definició de les característiques: http://wurfl.sourceforge.net/help_doc.php

Aquest howto pretén ser una guia per als desenvolupadors que hagin de fer servir el servei WURFL per obtenir les característiques del dispositiu mòbil amb Canigó3. S'emprarà API de tercers (<http://wurfl.sourceforge.net/njava/>) i s'indicarà com s'ha de configurar per integrar-la en la aplicació.

Es parteix de la base que es té una aplicació web mòbil desenvolupada o en desenvolupament, i que es volen conèixer algunes de les característiques del dispositiu mòbil amb què s'està visualitzant la web.

Configuració

Per poder utilitzar aquest servei caldrà configurar les dependències i els paràmetres de l'API dins l'aplicació Canigó.

1. Afegir dependències

El primer pas és afegir la dependència de l'API al **pom.xml**.

```
<dependency>
  <groupId>net.sourceforge.wurfl</groupId>
  <artifactId>wurfl</artifactId>
  <version>1.3.1.1</version>
</dependency>
```

2. Configuració de l'API

Es realitzarà mitjançant la configuració d'un únic Bean. Aquesta configuració es realitza mitjançant el fitxer `wurfl_ctx.xml` ubicat a `"scr/main/resources/spring/"` per tal que el framework instanciï automàticament el Bean.

Servei Característiques dispositius mòbils (WURFL) amb Canigó3

Configuració mínima

```
<bean class="net.sourceforge.wurfl.spring.SpringWurflManager" />
```

A partir de la configuració mínima es poden personalitzar els següents paràmetres.

Ubicació de la base de dades wurfl.zip

La ubicació per defecte d'aquest fitxer és `"/WEB-INF/wurfl.zip"`. Per poder modificar-la cal indicar la nova ubicació mitjançant la propietat `"wurfl"`

```
<bean class="net.sourceforge.wurfl.spring.SpringWurflManager">
  <property name="wurfl" value="/nova_ubicacio/mycustomwurfl.zip"/>
</bean>
```

Aplicació de Pegats al fitxer de configuració

En el cas d'aplicar un fitxer pegat s'haurà d'indicar la ubicació en la propietat `"wurflPatch"`. En el cas d'haver d'aplicar més d'un fitxer patch cal definir en la propietat `wurflPatches` la llista de fitxers patch a aplicar.

```
<bean class="net.sourceforge.wurfl.spring.SpringWurflManager">
  <!--Ubicació per defecte -->
  <!--<property name="wurfl" value="/WEB-INF/wurfl.zip"/>-->

  <!--Si només s'aplica un patchfile-->
  <!--<property name="wurflPatch" value="/WEB-INF/web_browsers_patch.xml"/>-->

  <!-- Si s'apliquen 2 o més fitxers patchfiles -->
  <property name="wurflPatches">
    <list>
      <value>/WEB-INF/web_browsers_patch.xml</value>
      <value>/WEB-INF/web_browsers_patch2.xml</value>
    </list>
  </property>
</bean>
```

Servei Característiques dispositius mòbils (WURFL) amb Canigó3

Fitxer de configuració wurft_ctx.xml d'exemple.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--

    Copyright (c) 2011 ScientiaMobile Inc.

    This program is free software: you can redistribute it and/or modify
    it under the terms of the GNU Affero General Public License as
    published by the Free Software Foundation, either version 3 of the
    License, or (at your option) any later version.

    Refer to the COPYING file distributed with this package.

-->

<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-2.5.xsd" default-
autowire="no">

    <!-- WURFLManager, automatically exported in ServletContext, if you need outside
    Spring. See Javadoc -->
    <bean class="net.sourceforge.wurfl.spring.SpringWurflManager">
        <!--Ubicació per defecte-->
        <!--<property name="wurfl" value="/WEB-INF/wurfl.zip"/>-->

        <!--Si només s'aplica un patchfile-->
        <property name="wurflPatch" value="/WEB-INF/web_browsers_patch.xml"/>

        <!-- Si s'apliquen 2 o més fitxers patchfiles -->
        <!--
        <property name="wurflPatches">
            <list>
                <value>/WEB-INF/web_browsers_patch.xml</value>
                <value>/WEB-INF/web_browsers_patch2.xml</value>
            </list>
        </property>
        -->
    </bean>
</beans>
```

Utilització

A continuació es detallarà com utilitzar aquest servei des d'un controlador de l'aplicació web.

Per utilitzar el servei s'ha de definir un atribut SpringWurflManager a la classe que el vulgui utilitzar. Cal decorar-lo amb @Autowired per tal que s'inicialitzi automàticament dins del context d'Spring

```
@Autowired
private SpringWurflManager wurflService;
```

(Application context).

Per poder identificar el dispositiu es requereix conèixer l'"user-agent" de la petició.

```
String myUserAgent = FacesContext.getCurrentInstance()
    .getExternalContext().getRequestHeaderMap().get("user-agent");
```

Per obtenir la informació del dispositiu que s'està connectant cal fer una crida al servei indicant-li l'user-agent recuperat.

```
Device dispositiu;
dispositiu = wurflService.getDeviceForRequest(myUserAgent);
```

Servei Característiques dispositius mòbils (WURFL) amb Canigó3

A partir del dispositiu es pot obtenir el valor d'una característica en concret "getCapability(String Capability)" o bé obtenir un Mapa amb totes les propietats del dispositiu "getCapabilities()".

Totes les propietats sempre retornaran un valor.

Exemple:

Situació: Es vol pintar una imatge que ocupi tota l'amplada de la pantalla.

S'inicia consultant al WURFL l'amplada de la pantalla del dispositiu i amb el servei de re dimensió de Canigo3 obtindrem la imatge amb amplada definida.

```
@Controller("testWurfl")
@Lazy(true)
public class TestWurflBean {

    @Autowired
    private SpringWurflManager wurflService;

    @Autowired
    @Qualifier("resizeImg")
    ResizeImgService resizeImgService;

    public byte[] ampladaPantalla() {
        Device dispositiu;
        String myUserAgent = FacesContext.getCurrentInstance()
            .getExternalContext().getRequestHeaderMap().get("user-agent");

        dispositiu = wurflService.getDeviceForRequest(myUserAgent);

        //Obté el valor de l'amplada maxima de la pantalla
        String maxWidth = dispositiu.getCapability("max_image_width");
        //Obtenir la imatge logo.png a la maxima amplada de pantalla
        Return resizeImage("../images/canigo/logo.png", new Integer(maxWidth)
    }

    //Redimensiona la imatge a la ampada definida
    //Retornan el binari de la nova imatge
    private byte[] resizeImage(String pathImatge, Integer width) {
        byte[] imatgeDesti = null;
        try {
            byte[] imatgeJpg = LoadImageFile(pathImatge);
            imatgeDesti = testResizeImg(imatgeJpg, width, null, "jpg");
        } catch (ResizeImgModuleException e) {
            e.printStackTrace();
        }
        return imatgeDesti;
    }

    //Retorna el array de bytes de la imatge
    private byte[] LoadImageFile(String filename)
    {
        byte[] fileContent=null;
        File file = new File(filename);
        try {
            FileInputStream fin = new FileInputStream(file);
            fileContent = new byte[(int) file.length()];
            fin.read(fileContent);
        } catch (FileNotFoundException e) {
            System.out.println("File not found" + e);
        } catch (IOException ioe) {
            System.out.println("Exception while reading the file " + ioe);
        }
        return fileContent;
    }
}
```

Servei Característiques dispositius mòbils (WURFL) amb Canigó3

Per no haver d'estar consultant aquest servei en cada petició, es pot emmagatzemar la informació més rellevant del dispositiu (per exemple la resolució de la pantalla) en algun tipus de cache (per exemple la HttpSession). Cada aplicació, en la fase de disseny, haurà de considerar quins atributs es consultaran amb més freqüència i si cal emmagatzemar-los en una cache temporal per agilitzar-ne la consulta.

Principals característiques

A continuació es detallaran un llistat de les principals característiques que es poden emprar en les diferents aplicacions. Per obtenir informació de la resta de característiques que informa el servei cal consultar la documentació oficial del WURFL.

Definició de les característiques: http://wurfl.sourceforge.net/help_doc.php

Característica	Definició
ajax_support_events	Compatibilitat amb els events onload, onclick, onsubmit i onselect
ajax_support_javascript	Un dispositiu es pot dir activar Javascript només si les següents característiques són compatibles amb fiabilitat: alert, confirm, accés a elements de forma (establir dinàmicament / modificar els valors), setTimeout, setInterval, document.location. Si un dispositiu falla una d'aquestes proves, marcar com falses (és a dir, paralitzat Javascript no és suficient per ser marcat com Javascript habilitat)
cookie_support	Indica si l'explorador admet galetes (cookie)
https_support	suport genèric per al protocol HTTPS (connexions SSL).
pdf_support	Pot llegir fitxers PDF de forma nativa
post_method_support	Si és true que el telèfon compatible amb el mètode HTTP POST
preferred_markup	
resolution_height	Aquest camp representa l'alçada de la pantalla expressada en píxels
resolution_width	Aquest camp representa l'amplada de la pantalla expressada en píxels
rss_support	El dispositiu suporta RSS de forma nativa.
sms_enabled	El dispositiu suporta SMS
streaming_video	Si és true, el dispositiu suporta streaming de vídeo.
total_cache_disable_support	possibilitat de desactivar la memòria cau del navegador completament
video	Si es true, el dispositiu pot descarregar video clips
xhtml_can_embed_video	Indica si el dispositiu es compatibles amb la incrustació de vídeo a través de l'etiqueta <object>