



Manual de *RESTServices* de PortaFIB

**Manual de Integració emprant Serveis REST
disponibles a PortaFIB.**



G VICEPRESIDÈNCIA
O I CONSELLERIA
I INNOVACIÓ,
B RECERCA I TURISME
/ DIRECCIÓ GENERAL
DESENVOLUPAMENT
TECNOLÒGIC



G VICEPRESIDÈNCIA
O I CONSELLERIA
I INNOVACIÓ,
B RECERCA I TURISME
/ FUNDACIÓ BIT

Informació general del document.

Descripció.

Títol: Manual de RESTServices de PortaFIB
Estat: Esborrany/Aprovat
Versió: 1.0
Autor/s: Antoni Nadal Bennasar
Creat: 04/09/2017
Modificat 05/09/2017
Fitxer: Manual_de_RESTServices_de_PortaFIB.odt

Històric de modificacions.

Comentari:	Autor/s:	Data:
Serveis REST	A. Nadal	04/09/2017

Font documental.

Index de Contingut

1.-Introducció.....	4
2.-Consulta de TipusDocument REST.....	4
3.-Consulta d'Avisos de Peticions de Firma REST.....	5
4.-Cridada Echo REST.....	6
5.-Exemple.....	6
5.1.-Introducció.....	6
5.2.-Dependències.....	7
5.3.-Desenvolupament Android.....	7
5.4.-Codi Font Exemple.....	8

1.- Introducció

Nou a 1.1.3

Usar l'API de WS en certes consultes i en certs entorns pot ser inadequat o impossible. Per això, per certs serveis també els oferirem en una versió REST-JSON amb la finalitat de fer més oberts els serveis que ofereix PortaFIB.

Teniu exemples de cridades REST de Client en el projecte [portafib-1.1]\ws\portafib_client_rest_example. Més detalls en el punt "5.-Exemple".


2.- Consulta de TipusDocument REST

Nou a 1.1.3

URL BASE	<code>http://HOST:PORT/portafib/public/rest/tipusdocument/v1/list</code>
Tipus Cridada	GET
Autenticat	NO
Descripció	
Retorna una llista del tipus de documents base definits a PortaFIB en l'idioma definit per defecte a PortaFIB.	
Paràmetres:	
<ul style="list-style-type: none"> • lang: Opcional. Permet canviar l'idioma del nom del tipus de documents. En principi només hi ha disponibles "es" i "ca" • appuser: Opcional. Permet definir l'identificador d'un Usuari-Aplicació per afegir al llistat base, els tipus documentals definits per aquest Usuari-Aplicació. 	
Resposta:	
Retorna una llista on cada element esta format per tres camps: <ul style="list-style-type: none"> • tipusDocumentID(long): és l'identificador del tipus • nom(String): és el nom del tipus de document en l'idioma elegit • tipusDocumentNTI(Long): Representa l'identificador al Tipus de Document definit a les NTI i associat a aquest tipus. Si val null significa que aquest tipus forma part del tipus NTI base. 	
Exemple Cridada:	
<ul style="list-style-type: none"> ■ <code>http://localhost:8080/portafib/public/rest/tipusdocument/v1/list?lang=ca&appuser=fundaciobit_app</code> ■ <code>http://localhost:8080/portafib/public/rest/tipusdocument/v1/list?lang=ca</code> 	
Exemple Format Retorn	
<pre>[{"tipusDocumentID":1,"nom":"Resolució","tipusDocumentNTI":null}, {"tipusDocumentID":2,"nom":"Acord","tipusDocumentNTI":null}, ...]</pre>	

3.- Consulta d'Avisos de Peticions de Firma REST

Nou a 1.1.3

URL BASE	<code>http://HOST:PORT/common/rest/usuarientitat/avisos/v1/list</code>
Tipus Cridada	GET
Autenticat	SI
Descripció	
Retorna una llista de les Peticions agrupades per ROL, que tenen algun avis per l'usuari que fa la cridada. En principi són els mateixos avisos que pareixen via web a l'usuari.	
	
Paràmetres:	
<ul style="list-style-type: none"> entitatID: Opcional. Si l'usuari està donat d'alta a múltiples entitats llavors defineix de la qual volem els avisos. Si val null o no està definida, llavors recull els avisos de totes les entitats a les que esta donat d'alta. 	
Resposta:	
<p>Retorna una llista on cada element esta format per dos camps: el primer un String que conté el nom del rol i el segon una llista de sencers que conté els identificadors de les Peticions de Firma origen dels avisos.</p> <ul style="list-style-type: none"> rol(String): és el nom del rol (ROLE_SOLI, ROLE_DELE, ROLE_COLA, ROLE_DES, ROLE_ADEN, ...) peticions(long[]): és el nom del tipus de document en l'idioma elegit tipusDocumentNTI(Long): llista de sencers que conté els identificadors de les Peticions de Firma origen dels avisos 	
Exemple Cridada:	
<ul style="list-style-type: none"> ■ <code>http://10.215.216.175:8080/portafib/common/rest/usuarientitat/avisos/v1/list</code> ■ <code>http://10.215.216.175:8080/portafib/common/rest/usuarientitat/avisos/v1/list?entitatID=uib</code> 	
Exemple Format Retorn	
<pre>[{"rol":"ROLE_SOLI","peticions":[182129,181786,182153,177417,182086,181961,181939]}, {"rol":"ROLE_ADEN","peticions":[177417]}, {"rol":"ROLE_DELE","peticions":[182361]}, {"rol":"ROLE_DEST","peticions":[182361]}]</pre>	

4.- Cridada Echo REST

Nou a 1.1.3

URL BASE	<code>http://HOST:PORT/portafib/common/rest/usuariantitat/avisos/v1/list</code>
Tipus Cridada	GET
Autenticat	SI
Descripció	
Mètode per comprovar la comunicació i si els valors de les credencials són correctes. Retorna un String amb el valor passat per paràmetre.	
Paràmetres:	
<ul style="list-style-type: none"> • echo: Opcional. Missatge que vol que es torni. 	
Resposta:	
Retorna un simple String amb el valor del paràmetre "echo". Si aquest no es defineix llavors retorna un String amb el següent valor: "Parametre echo buit".	
Exemple Cridada:	
<ul style="list-style-type: none"> ■ <code>http://10.215.216.175:8080/portafib/common/rest/echo</code> ■ <code>http://10.215.216.175:8080/portafib/common/rest/echo=HolaCaracola</code> 	
Exemple Format Retorn	
<code>"HolaCaracola"</code>	

5.- Exemple

Nou a 1.1.3

5.1.- Introducció

Com el lector hauria de saber, les cridades REST o Serveis REST són simplement cridades HTTP, GET o POST, que retornen informació estructurada d'una forma específica. En aquest cas mostrarem exemples en codi java de dues formes diferents: emprant la classe `URLConnection` (pròpia del JDK de java) o emprant la llibreria Jersey.

Podeu trobar aquest exemple complet dins del repositori de PortaFIB a la següent ubicació [portafib-1.1]\ws\portafib_client_rest_example. La ruta al repositori de GitHub és la següent https://github.com/GovernIB/portafib/tree/portafib-1.1/ws/portafib_client_rest_example.

5.2.- Dependències

Si en el nostre projecte Java volen emprar `HttpURLConnection` llavors no fa falta importar cap llibreria addicional. En cas de voler usar Jersey, que incorpora mapejadors de JSON a POJOs, llavors al nostre `pom.xml` haurem d'afegir les següent dependències:

```
<dependency>
  <groupId>commons-io</groupId>
  <artifactId>commons-io</artifactId>
  <version>2.2</version>
</dependency>

<dependency>
  <groupId>com.sun.jersey</groupId>
  <artifactId>jersey-client</artifactId>
  <version>1.8</version>
</dependency>

<dependency>
  <groupId>com.sun.jersey</groupId>
  <artifactId>jersey-json</artifactId>
  <version>1.8</version>
  <exclusions>

    <exclusion>
      <groupId>stax</groupId>
      <artifactId>stax-api</artifactId>
    </exclusion>

    <exclusion>
      <groupId>javax.xml.stream</groupId>
      <artifactId>stax-api</artifactId>
    </exclusion>

    <exclusion>
      <groupId>com.sun.xml.bind</groupId>
      <artifactId>jaxb-impl</artifactId>
    </exclusion>

  </exclusions>
</dependency>
```

5.3.- Desenvolupament Android

La llibreria Jersey no és compatible amb Android, per la qual cosa s'haurà d'usar la classe `HttpURLConnection` per la cridada i recepció de dades i després emprar les classes `org.json.JSONObject` i `org.json.JSONArray`, pròpies d'Android, per processar el resultat. El següent codi mostra com processar el resultat d'una cridada a Avisos (Veure 3.-Consulta d'Avisos de Peticions de Firma REST)

```
JSONArray jArray = new JSONArray(json);
for (int i = 0; i < jArray.length(); i++) {
    JSONObject jObject = jArray.getJSONObject(i);
    String rol = jObject.getString("rol");
    System.out.println(" ===== ");
    System.out.println(" ROL: " + rol);
    JSONArray petitions = jObject.getJSONArray("peticions");
    System.out.print(" PETICIONS: ");
    for (int p = 0; p < petitions.length(); p++) {
        System.out.print(" " + petitions.getString(p));
    }
    System.out.println();
}
```

5.4.- Codi Font Exemple

Podeu trobar una còpia més actualitzada d'aquest codi font junt amb tot el projecte en la següent ruta La ruta al repositori de GitHub és la següent https://github.com/GovernIB/portafib/tree/portafib-1.1/ws/portafib_client_rest_example.

```
import java.io.FileInputStream;
import java.net.HttpURLConnection;
import java.net.URL;
import java.net.URLEncoder;
import java.util.Arrays;
import java.util.List;
import java.util.Properties;

import org.apache.commons.io.IOUtils;
import org.codehaus.jackson.map.ObjectMapper;
import org.codehaus.jackson.type.TypeReference;
import org.junit.Test;

import com.sun.jersey.api.client.Client;
import com.sun.jersey.api.client.ClientResponse;
import com.sun.jersey.api.client.WebResource;
import com.sun.jersey.api.client.filter.HTTPBasicAuthFilter;
import com.sun.jersey.core.util.Base64;

/**
 * @author anadal
 */
public class TestPortaFIBRestServices {

    private static Properties testProperties = new Properties();

    static {
        // Propietats del Servidor
        try {
            testProperties.load(new FileInputStream("test.properties"));
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public static String getURLBase() {
        return testProperties.getProperty("urlbase");
    }
}
```



```

}

public static String getUsername() {
    return testProperties.getProperty("username");
}

public static String getPassword() {
    return testProperties.getProperty("password");
}

public static void main(String[] args) {
    TestPortaFIBRestServices tester = new TestPortaFIBRestServices();
    try {
        // Public
        tester.testTipusDocumentJersey();

        // Autenticated
        tester.testEchoHttpURLConnection();
        tester.testAvisosJersey();
    } catch (Exception e) {
        e.printStackTrace();
    }
}

/**
 * Cridada a recuperar TipusDeDocument emprant llibreria Jersey
 */
@Test
public void testTipusDocumentJersey() throws Exception {

    Client client = Client.create();

    client.addFilter(new HTTPBasicAuthFilter(getUsername(), getPassword()));

    StringBuffer params = new StringBuffer("?");

    String lang = testProperties.getProperty("lang");
    if (lang != null && lang.trim().length() != 0) {
        params.append("lang=").append(lang);
    }

    String appuser = testProperties.getProperty("appuser");
    if (appuser != null && appuser.trim().length() != 0) {
        if (params.length() != 1) {
            params.append("&");
        }
        params.append("appuser=").append(appuser);
    }

    WebResource webResource = client.resource(getURLBase()
        + "/public/rest/tipusdocument/v1/list"
        + ((params.length() == 1) ? "" : params.toString()));

    ClientResponse response = webResource.type("application/json").get(ClientResponse.class);

    if (response.getStatus() != 200) {
        throw new Exception("Error " + response.getEntity(String.class) + "(Error code = "
            + response.getStatus() + ")");
    }

    String json = response.getEntity(String.class);

```

```
System.out.println("JSON TIPUS DOCUMENT " + json);

ObjectMapper mapper = new ObjectMapper();
List<TipusDocumentRest> listTipus = mapper.readValue(json,
    new TypeReference<List<TipusDocumentRest>>() {
    });

System.out.println("Resposta cridada REST a mètode tipusdocument():");

for (TipusDocumentRest tipus : listTipus) {
    System.out.println(" * '" + tipus.getTipusDocumentID() + "'\t" + tipus.getNom()
        + "\t[NTI:" + tipus.getTipusDocumentNTI() + "]"");
}
}

/**
 * @author anadal
 */
public static class TipusDocumentRest {

    protected long tipusDocumentID;

    protected String nom;

    protected Long tipusDocumentNTI;

    public TipusDocumentRest() {
        super();
    }

    public TipusDocumentRest(long tipusDocumentID, String nom, Long tipusDocumentNTI) {
        super();
        this.tipusDocumentID = tipusDocumentID;
        this.nom = nom;
        this.tipusDocumentNTI = tipusDocumentNTI;
    }

    public long getTipusDocumentID() {
        return tipusDocumentID;
    }

    public void setTipusDocumentID(long tipusDocumentID) {
        this.tipusDocumentID = tipusDocumentID;
    }

    public String getNom() {
        return nom;
    }

    public void setNom(String nom) {
        this.nom = nom;
    }

    public Long getTipusDocumentNTI() {
        return tipusDocumentNTI;
    }

    public void setTipusDocumentNTI(Long tipusDocumentNTI) {
        this.tipusDocumentNTI = tipusDocumentNTI;
    }
}
```

```

}

/**
 * Prova d'echo (Emprant HttpURLConnection)
 *
 * @throws Exception
 */
@Test
public void testEchoHttpURLConnection() throws Exception {

    final String basicAuth = "Basic "
        + new String(Base64.encode((getUsername() + ":" + getPassword()).getBytes()));

    @SuppressWarnings("deprecation")
    URL url = new URL(getURLBase() + "/common/rest/echo?echo=" +
        URLEncoder.encode(testProperties.getProperty("echo")));
    HttpURLConnection conn = (HttpURLConnection) url.openConnection();
    conn.setRequestMethod("GET");
    conn.setRequestProperty("Accept", "application/json");
    conn.setRequestProperty("Authorization", basicAuth);

    if (conn.getResponseCode() != 200) {
        throw new Exception("Error " + conn.getResponseMessage() + "(Error code = "
            + conn.getResponseCode() + ")");
    }

    String json = IOUtils.toString(conn.getInputStream());
    System.out.println("JSON ECHO = " + json);
    conn.disconnect();
}

/**
 * Recupera Peticions amb Avisos (Emprant llibreria Jersey)
 *
 * @throws Exception
 */
@Test
public void testAvisosJersey() throws Exception {

    Client client = Client.create();

    client.addFilter(new HTTPBasicAuthFilter(getUsername(), getPassword()));

    final String params;
    String entitatID = testProperties.getProperty("entitatID");
    if (entitatID != null && entitatID.trim().length() != 0) {
        params = "?entitatID=" + entitatID.trim();
    } else {
        params = "";
    }

    WebResource webResource = client.resource(getURLBase()
        + "/common/rest/usuari/entitat/avisos/v1/list" + params);

    ClientResponse response = webResource.type("application/json").get(ClientResponse.class);

    if (response.getStatus() != 200) {
        throw new Exception("Error " + response.getEntity(String.class) + "(Error code = "
            + response.getStatus() + ")");
    }

    String json = response.getEntity(String.class);

```

```
System.out.println("JSON " + json);

ObjectMapper mapper = new ObjectMapper();
List<NotificacioRest> avisos = mapper.readValue(json,
    new TypeReference<List<NotificacioRest>>() {
    });

System.out.println("Resposta cridada REST a mètode avisos():");

for (NotificacioRest notificacioRest : avisos) {
    System.out.println(" + " + notificacioRest.getRol() + " => "
        + Arrays.toString(notificacioRest.getPeticions().toArray()));
}
}

/**
 * @author anadal
 */
public static class NotificacioRest {

    String rol;

    List<Long> peticions;

    public NotificacioRest() {
        super();
    }

    public NotificacioRest(String rol, List<Long> peticions) {
        super();
        this.rol = rol;
        this.peticions = peticions;
    }

    public String getRol() {
        return rol;
    }

    public void setRol(String rol) {
        this.rol = rol;
    }

    public List<Long> getPeticions() {
        return peticions;
    }

    public void setPeticions(List<Long> peticions) {
        this.peticions = peticions;
    }
}
}
```