

Sofia 

APIS SCRIPT

JULIO 2015

Versión 1



indra

1 INDICE

1	INDICE	2
2	INTRODUCCIÓN	4
2.1	REQUISITOS	4
2.2	OBJETIVOS Y ALCANCE DEL PRESENTE DOCUMENTO	4
2.3	APIS SCRIPTING.....	4
3	APIHTTPCONNECTION	6
3.1	HTTPGET(URL)	6
3.2	HTTPGET(URL, TIMEOUT).....	6
3.3	HTTPGET(URL, HEADERS, PARAMETERS)	6
3.4	HTTPPOST(URL, PARAMETERS)	7
3.5	HTTPPOST(URL, PARAMETERS, TIMEOUT).....	7
3.6	HTTPPOST(URL, PARAMETERS, HEADERS, BODY)	8
3.7	HTTPPUT(URL, PARAMETERS, HEADERS, BODY)	9
4	APIJMS	10
4.1	SENDJSON2QUEUE(ONTOLOGY, QUEUE).....	10
4.2	SENDJSON2TOPIC(ONTOLOGY, TOPIC)	10
4.3	SENDXML2QUEUE(ONTOLOGYNAME, ONTOLOGY, QUEUE).....	11
4.4	SENDXML2TOPIC(ONTOLOGYNAME, ONTOLOGY, TOPIC)	11
5	APILOG	13
5.1	LOGDEBUG(MESSAGE)	13
5.2	LOGINFO(MESSAGE)	13
5.3	LOGWARN(MESSAGE)	13
5.4	LOGERROR(MESSAGE).....	14
6	APIMAIL	15
6.1	SENDMAIL(TO, SUBJECT, MSG).....	15
7	APIRTDB	16

7.1	QUERY(ONTOLOGYNAME, QUERY)	16
7.2	INSERT(ONTOLOGYNAME, DATA)	16
7.3	INSERT(EXECUTIONCONTEXT, ONTOLOGYNAME, DATA)	16
7.4	UPDATE(ONTOLOGYNAME, QUERY, DATA).....	17
7.5	DELETE(ONTOLOGYNAME, ID)	18
7.6	FINDINCIRCLE(ONTOLOGYNAME, INDEXNAME, LONGITUDE, LATITUDE, RADIO)	18
7.7	NATIVEQUERY(QUERY).....	18
8	APISCRIPITS	20
8.1	EXECUTERSCRIPT(SCRIPITNAME, PARAM)	20
8.2	EXECUTERSCRIPT(SCRIPITNAME)	20
8.3	EXECUTEMONGOSCRIPT(SCRIPITNAME, PARAM).....	20
9	APISSAP	22
9.1	SENDINSERTMESSAGE(TOKEN, KPINSTANCE, ONTOLOGYNAME, DATA).....	22
10	APITWITTER	23
10.1	SENDTWEET(APIKEY, APIKEYSECRET, TOKEN, TOKENSECRET, PARAMETERS).....	23
11	APIUTILS	24
11.1	GETVALUE(ONTOLOGYNAME, ATRIBUTE)	24
11.2	GETVALUEJSON(INSTANCIA, ATRIBUTE).....	24
11.3	GETVALUEJSONARRAY(INSTANCIA, ATRIBUTE)	25
12	APISOFIA	26



2 INTRODUCCIÓN

2.1 Requisitos

Antes de seguir esta guía se recomienda leer las guías:

- [Primeros Pasos Sofia2](#) (Pdf)
- [Creación Reglas Script](#) (Pdf)

2.2 Objetivos y alcance del presente documento

Esta guía describe las APIS que pueden usarse desde el motor de Reglas Scripting de la Plataforma Sofia2.

En la guía se incluyen:

- Listado de las APIS disponibles
- Listado de los métodos de cada API
- Cometido de cada método
- Parámetros de entrada de cada método
- Resultado que devuelve cada método
- Ejemplo de uso de cada método

2.3 APIS Scripting

Las APIS que disponibiliza la plataforma son:

API	Descripción
APIHttpConnection	API que ofrece diversos métodos para establecer conexiones HTTP desde un script
APIJMS	API para publicar en colas y tópicos JMS desde un script
APILog	API para logear información desde un script
APIMail	API para envío de mails desde un script
APIRTDB	API para hacer operaciones sobre BD Tiempo Real desde un script
APIScripts	API para invocar scripts externos en R, Python,...
APISofia	(DEPRECADA)

	API aglutinadora de todas las operaciones.
APISSAP	API para comunicaciones SSAP desde un script
APITwitter	API para publicar tuits desde un script
APIUtils	API de utilidades



3 APIHttpConnection

3.1 httpGET(url)

<pre> /** * Invoke an URL by GET HTTP method * @param url String * @return content of the URL String */ def httpGET(url) throws ScriptException; </pre>	
Descripción	Hace petición HTTP GET a una url
Parámetros	String url: URL a la que se invoca por GET
Resultado	String contenido: contenido de la URL
Ejemplo uso	<pre> def apihttpconnection = new APIHttpConnection(); def url = "https://www.google.es/search?q=sofia2"; def contenido = apihttpconnection.httpGET(url); println("Contenido httpGET" + contenido); </pre>

3.2 httpGET(url, timeout)

<pre> /** * Invoke an URL by GET HTTP method establishing an timeout in miliseconds * @param url String * @param timeout in ms int * @return content of the URL String */ def httpGET(url, timeout) throws ScriptException; </pre>	
Descripción	Hace petición HTTP GET a una url con timeout definido
Parámetros	<ul style="list-style-type: none"> String url: URL a la que se invoca por GET int timeout: timeout (ms)
Resultado	String contenido: contenido de la URL
Ejemplo uso	<pre> def apihttpconnection = new APIHttpConnection(); def url = "https://www.google.es/search?q=sofia2"; def contenido = apihttpconnection.httpGET(url, 30000); println("Contenido httpGET con timeout" + contenido); </pre>

3.3 httpGET(url, headers, parameters)

<pre> /** * Invoke an URL by GET HTTP method passing parameters and headers * @param url String * @param String[] headers * @param String[] parameters * @return content of the URL String */ def httpGET(url, headers, parameters) throws ScriptException; </pre>	
Descripción	Hace petición HTTP GET a una url pasando headers y parámetros

Parámetros	<ul style="list-style-type: none"> • String url: URL a la que se invoca por GET • String[] headers: cabecera de la petición • String[] parameters: parámetros de la invocación
Resultado	String contenido: contenido de la URL
Ejemplo uso	<pre>def apihttpconnection = new APIHttpConnection(); String [] headers = ["X-SOFIA2- APIKey:9806ed2dfe3f4bdab78c5795c3a38878"]; String [] param=null; String urlSemaforoAlarmaById="http://sofia2.com/sib- api/api/v2/semaforoalarmaes/55818c33e4b032246d12d489"; String resGET = apihttpconnection.httpGET(urlSemaforoAlarmaById,headers,param);</pre>

3.4 httpPOST(url, parameters)

<pre>/** * Invoke an URL by POST HTTP method passing parameters * @param url String * @param parameters Map<String, String> * @return content of the URL String */ def httpPOST(url, parameters) throws ScriptException;</pre>	
Descripción	Hace petición HTTP POST a una url pasando headers y parámetros
Parámetros	<ul style="list-style-type: none"> • String url: URL a la que se invoca por GET • String[] parameters: parámetros de la invocación
Resultado	String contenido: contenido de la URL
Ejemplo uso	<pre>def apihttpconnection = new APIHttpConnection(); String [] headers = ["X-SOFIA2- APIKey:9806ed2dfe3f4bdab78c5795c3a38878"]; String [] param=null; String urlSemaforoAlarma="http://sofia2.com/sib- api/api/v2/semaforoalarmaes"; String resPOST = apihttpconnection.httpPOST(urlSemaforoAlarma,param);</pre>

3.5 httpPOST(url, parameters, timeout)

<pre>/** * Invoke an URL by POST HTTP method passing parameters with a timeout * @param url String * @param parameters Map<String, String> * @param timeout in ms int * @return content of the URL String</pre>	
--	--

<pre>*/ def httpPOST(url, parameters, timeout) throws ScriptException;</pre>	
Descripción	Hace petición HTTP POST a una url pasando headers y parámetros con un timeout
Parámetros	<ul style="list-style-type: none"> • String url: URL a la que se invoca por GET • String[] parameters: parámetros de la invocación • int timeout: timeout (ms)
Resultado	String contenido: contenido de la URL
Ejemplo uso	<pre>def apihttpconnection = new APIHttpConnection(); String [] headers = ["X-SOFIA2- APIKey:9806ed2dfe3f4bdab78c5795c3a38878"]; String [] param=null; String urlSemaforoAlarma="http://sofia2.com/sib- api/api/v2/semaforoalarmaes"; String resPOST2 = apihttpconnection.httpPOST(urlSemaforoAlarma,param,30000);</pre>

3.6 httpPOST(url, parameters, headers, body)

<pre>/** * Invoke an URL by POST HTTP method passing parameters with headers and body * @param url String * @param parameters String [] * @param headers String [] * @param body String * @return content of the URL String */ def httpPOST(url, parameters, headers, body) throws ScriptException;</pre>	
Descripción	Hace petición HTTP POST a una url pasando parámetros con headers y body
Parámetros	<ul style="list-style-type: none"> • String url: URL a la que se invoca por GET • String[] parameters: parámetros de la invocación • String[] headers: cabeceras de la petición • String body: cuerpo de la petición
Resultado	String contenido: contenido de la URL
Ejemplo uso	<pre>def apihttpconnection = new APIHttpConnection(); String [] headers = ["X-SOFIA2- APIKey:9806ed2dfe3f4bdab78c5795c3a38878"]; String [] param=null;</pre>

	<pre>String urlSemaforoAlarma="http://sofia2.com/sib- api/api/v2/semaforoalarmas"; def mydate = new Date(); String instanciaAlarma="{ 'Alarm':{ 'timestamp':{ '<-dollar>date': '+mydate.format("MM-dd- yyyy'T'HH:mm:ss.SSS'Z'")+'' }, 'assetId':1, 'severity':'LOW', 'type':'restPl an'}}"; String resPOST3 = apihttpconnection.httpPOST(urlSemaforoAlarma,param,headers,instanciaAlar ma);</pre>
--	---

3.7 httpPUT(url, parameters, headers, body)

<pre>/** * Invoke an URL by PUT HTTP method passing parameters with headers and body * @param url String * @param parameters String [] * @param headers String [] * @param body String * @return content of the URL String */ def httpPUT(url, parameters, headers, body) throws ScriptException;</pre>	
Descripción	Hace petición HTTP PUT a una url pasando parámetros con headers y body
Parámetros	<ul style="list-style-type: none"> • String url: URL a la que se invoca por GET • String[] parameters: parámetros de la invocación • String[] headers: cabeceras de la petición • String body: cuerpo de la petición
Resultado	String contenido: contenido de la URL
Ejemplo uso	<pre>def apihttpconnection = new APIHttpConnection(); String [] headers = ["X-SOFIA2- APIKey:9806ed2dfe3f4bdab78c5795c3a38878"]; String []param=null; String urlAlarma = "http://sofia2.com/sib- api/api/v1/alarmas/"; String putById= "{ 'Alarma':{ 'mensajeAlarma':'Temperatura Demasiado Alta', 'procedenciaAlarma': 'Casa', 'causa': 'Más de 100 grados', 'timestamp': 0}}"; String resPUT = apihttpconnection.httpPUT(urlAlarma,param,headers,putById);</pre>

4 APIJMS

4.1 sendJson2Queue(ontology, queue)

```
/**
 * Sends instance of the ontology to the JMS queue
 * @param instance of an ontology String
 * @param queue String
 * @return void
 */
def sendJson2Queue(ontology, queue) throws ScriptException;
```

Descripción	Envía una instancia de ontología a la cola JMS
Parámetros	<ul style="list-style-type: none"> String ontology: instancia de la ontología sobre la que se hará la consulta String queue: cola JMS
Resultado	void
Ejemplo uso	Donde la ontology sea delayJustification <pre>def apijms = new APIJMS(); apijms.sendJson2Queue(ontology, "cola.sensor.ambiental");</pre>

4.2 sendJson2Topic(ontology, topic)

```
/**
 * Sends instance of the ontology to the JMS topic
 * @param instance of an ontology String
 * @param topic String
 * @return void
 */
def sendJson2Topic(ontology, topic) throws ScriptException;
```

Descripción	Envía una instancia de ontología al topic JMS
Parámetros	<ul style="list-style-type: none"> String ontology: instancia de la ontología sobre la que se hará la consulta String topic: topic JMS
Resultado	void
Ejemplo uso	Sobre la ontología delayJustification: <pre>def apijms = new APIJMS(); apijms.sendJson2Topic(ontology, "topicPrueba");</pre>

4.3 sendXml2Queue(ontologyName, ontology, queue)

<pre> /** * Sends instance of the ontology as XML to the JMS queue * @param ontologyName String * @param instance of an ontology String * @param queue String * @return void */ def sendXml2Queue(ontologyName, ontology, queue) throws ScriptException; </pre>	
Descripción	Envía una instancia de ontología como XML a la cola JMS
Parámetros	<ul style="list-style-type: none"> String ontologyName: nombre de la ontología sobre la que se hará la consulta String ontology: instancia de la ontología sobre la que se hará la consulta String queue: cola JMS
Resultado	void
Ejemplo uso	<p>Sobre la ontología delayJustification:</p> <pre> def apijms = new APIJMS(); apijms.sendXml2Queue(ontologyName, ontology, "ADIF.ESB.DELAYJUSTIFICATION"); </pre>

4.4 sendXml2Topic(ontologyName, ontology, topic)

<pre> /** * Sends instance of the ontology as XML to the JMS topic * @param ontologyName String * @param instance of an ontology String * @param queue String * @return void */ def sendXml2Topic(ontologyName, ontology, topic) throws ScriptException; </pre>	
Descripción	Envía una instancia de ontología como XML al topic JMS
Parámetros	<ul style="list-style-type: none"> String ontologyName: nombre de la ontología sobre la que se hará la consulta String ontology: instancia de la ontología sobre la que se hará la consulta String topic: topic JMS
Resultado	void
Ejemplo uso	<p>Sobre la ontología delayJustification:</p> <pre> def apijms = new APIJMS(); apijms.sendXml2Topic(ontologyName, ontology, "topicPrueba"); </pre>

--	--



5 APILog

5.1 logDebug(message)

<pre> /** * Logs a debug message on the log4j Appender defined * @param message String * @return void */ def logDebug(message) throws ScriptException; </pre>	
Descripción	Método para logear un mensaje tipo DEBUG en el Appender definido.
Parámetros	String message : mensaje a publicar
Resultado	void
Ejemplo uso	<pre> def apilog = new APILog(); apilog.logDebug("Estamos en modo depuración del bloque THEN"); </pre>

5.2 logInfo(message)

<pre> /** * Logs a info message on the log4j Appender defined * @param message String * @return void */ def logInfo(message) throws ScriptException; </pre>	
Descripción	Método para logear un mensaje tipo INFO en el Appender definido.
Parámetros	String message : mensaje a publicar
Resultado	void
Ejemplo uso	<pre> def apilog = new APILog(); apilog.logInfo("Se cumple la confición if"); </pre>

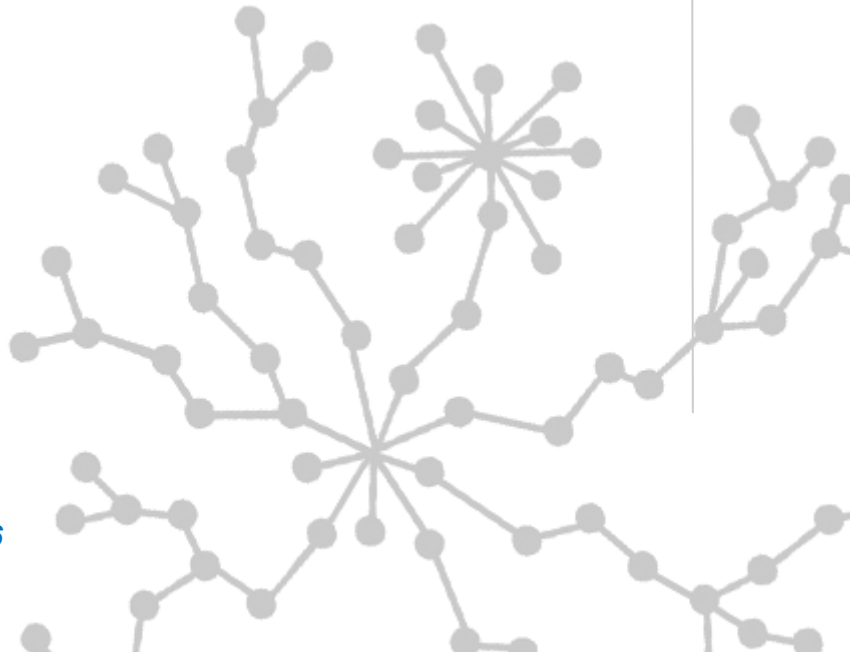
5.3 logWarn(message)

<pre> /** * Logs a warn message on the log4j Appender defined * @param message String * @return void */ def logWarn(message) throws ScriptException; </pre>	
Descripción	Método para logear un mensaje tipo WARN en el Appender definido.
Parámetros	String message : mensaje a publicar
Resultado	void
Ejemplo uso	<pre> def apilog = new APILog(); apilog.logWarn("Atención esto puede probar un Error de división por 0"); </pre>

5.4 logError(message)

```
/**
 * Logs a error message on the log4j Appender defined
 * @param message String
 * @return void
 */
def logError(message) throws ScriptException;
```

Descripción	Método para logear un mensaje tipo ERROR en el Appender definido.
Parámetros	String message : mensaje a publicar
Resultado	void
Ejemplo uso	<pre>def apilog = new APILog(); apilog.logError("Se ha producido un error en la ejecución del script APILog");</pre>



6 APIMail

6.1 sendMail(to, subject, msg)

```

/**
 * send an email to multiple recipients
 * @param to String[]
 * @param subject String
 * @param msg String
 * @return void
 */
def sendMail(to, subject, msg) throws ScriptException;
    
```

Descripción	Método para enviar un correo electrónico a varios destinatarios
Parámetros	<ul style="list-style-type: none"> • String[] to: destinatarios • String subject: asunto • String msg: cuerpo del mensaje
Resultado	void
Ejemplo uso	<pre> def apimail = new APIMail(); api.sendMail(["mymail@mail.com", "mymail2@mail.com"], "Subject of the mail", "Message of the mail") </pre>



7 APIRTDB

7.1 query(ontologyName, query)

```
/**
 * Makes a query on the Real Time DB for the ontology
 * @param name of ontology String
 * @param native query String
 * @return JSON result String
 */
def query(ontologyName, query) throws ScriptException;
```

Descripción	Método para realizar queries sobre la BDTR
Parámetros	<ul style="list-style-type: none"> String ontologyName: nombre de la ontología sobre la que se quiere consultar String query: consulta native
Resultado	String result: resultado de la consulta
Ejemplo uso	http://about.sofia2.com/2015/06/09/comando-de-scripting-getinbdtr/ <pre>def apirtdb = new APIRTDB(); String valor = apirtdb.query("MeteoMiguel10", "{{'MeteoMiguel10.Temperature':{<dollar>gt:99}}}")</pre>

7.2 insert(ontologyName, data)

```
/**
 * Inserts the data passed as a instance of the ontology
 * @param ontologyName String
 * @param data of the instance String
 * @return result String
 */
def insert(ontologyName, data) throws ScriptException;
```

Descripción	Método para realizar insert de datos de una ontología de la Base de Datos de Tiempo Real
Parámetros	<ul style="list-style-type: none"> String ontologyName: nombre de la ontología sobre la que se quiere consultar String data: instancia de ontología a insertar
Resultado	String result: resultado de la insert
Ejemplo uso	<pre>def apirtdb = new APIRTDB(); String insDate= "{<dollar>date': '"+mydate.format("yyyy-MM-dd'T'HH:mm:ss.SSS'Z'")+"}"; String myInstance="{ 'MoteSensor':{ 'mote_id':{ 'id':1, 'serial_number': 'string' }, 'humidity':1, 'temperature':1, 'battery':1, 'timestamp':{<dollar>date ': '"+mydate.format("yyyy-MM-dd'T'HH:mm:ss.SSS'Z'")+"}}}"; def resultInsert=apirtdb.insert("MoteSensor", myInstance);</pre>

7.3 insertInBDTR (executionContext, ontologyName, data)

```
/**
 * Insert de datos de una ontología de la Base de Datos de Tiempo Real
 */
```



```

    * @param ontologyName String
    * @param data String
    */
    def insertInBDTR(executionContext, ontologyName, data) throws
    ScriptException

```

Descripción	Insert de datos en una ontología de la Base de Datos de Tiempo Real
Parámetros	<ul style="list-style-type: none"> • ExecutionContext executionContext: contexto de ejecución. Objeto compuesto de sessionkey y nombre de ontología • String ontologyName: nombre de la ontología sobre la que se quiere consultar • String data: instancia de ontología a insertar
Resultado	String result: resultado de la insert
Ejemplo uso	<pre> def apirtdb = new APIRTDB(); String insertData="{ 'Chat':{ 'autor':'string','mensaje':'string'}}"; /* Para obtener el sessionKey lanzar: {"join": true,"instanceKP": "KP_chat:KP_chat01","token": "e42636d6fb54473db754e2f8d5ea9729"} */ def apiExecution= new ExecutionContext ("c8d20685-ade2-46cb- 915e-4c7037724f79","Chat"); def resultInsert2=apirtdb.insertInBDTR(apiExecution, "Chat", insertData); </pre>

7.4 update(ontologyName, query, data)

```

/**
 * Update the data of the Real Time DB with the query passed
 * @param query String
 * @param data String
 * @return result String
 */
def insert(ontologyName, data) throws ScriptException;

```

Descripción	Método para realizar update de datos de una ontología de la Base de Datos de Tiempo Real
Parámetros	<ul style="list-style-type: none"> • String ontologyName: nombre de la ontología sobre la que se hará el update • String query: consulta • String data: datos a modificar
Resultado	String result: resultado del update
Ejemplo uso	<pre> def apirtdb = new APIRTDB(); String query = "{'MoteSensor.mote_id.serial_number':'PRUEBAAPITEST'}"; String data ="{'<dollar>set':{'MoteSensor' : { 'mote_id' : { 'serial_number' : 'PRUEBAUPDATEAPITEST' }}}}'"; def resultUpdate = apirtdb.update("MoteSensor",query,data); </pre>

7.5 delete(ontologyName, id)

```

/**
 * Delete the record of the id and ontology passed as parameters from the
RTDB
 * @param ontologyName String
 * @param id instance String
 * @return result String
 */
def delete(ontologyName, id) throws ScriptException;

```

Descripción	Borra el registro de una ontología de un ID
Parámetros	<ul style="list-style-type: none"> String ontologyName: ontología String id: identificador OID
Resultado	String result: resultado del delete
Ejemplo uso	<pre> def apirtdb = new APIRTDB(); def resultDelete= apirtdb.delete("MoteSensor",resultInsert); </pre>

7.6 findInCircle(ontologyName, indexName, longitude, latitude, radio)

```

/**
 * Makes a geospatial query finding elements of the ontology on the center
with this ratio
 * Example:
 /
 * @param ontologyName String
 * @param indexName String
 * @param longitude double
 * @param latitude double
 * @param radio in metres int
 * @return
 */
def findInCircle(ontologyName, indexName, longitude, latitude, radio)
throws ScriptException;

```

Descripción	Consulta geoespacial para buscar dentro de un círculo
Parámetros	<ul style="list-style-type: none"> String ontologyName: nombre de la ontología sobre la que se hará la consulta String indexName: nombre del índice sobre el que se hará la consulta Double longitude: coordenada longitud Double latitude: coordenada latitud Int radio: radio expresado en metros
Resultado	String result: resultado de la consulta
Ejemplo uso	<pre> def apirtdb = new APIRTDB(); def attrib1= apirtdb.findInCircle("pt_bus", "MobileElement.geometry", 8.39, 43.37, 100); def var1=!(attrib1.size() > 0); if(var1){ return true; }else{ return false; } </pre>

	http://about.sofia2.com/2015/07/08/como-hacer-busquedas-de-elementos-en-una-zona-concreta/
--	---

Wrapper de la query nearsphere de MongoDB:

<http://docs.mongodb.org/manual/reference/operator/query/nearSphere/>

7.7 nativeQuery(query)

```
/**
 * Makes a native query on the Real Time DB
 * @param native query String
 * @return JSON result String
 */
def nativeQuery(query) throws ScriptException {
    try {
        return api.getNativeInBDTR(query)
    } catch(e) {
        throw new ScriptException(e)
    }
}
```

Descripción	Consultas nativas en bdtr
Parámetros	String query : consulta nativa
Resultado	String result: resultado de la consulta
Ejemplo uso	<pre>def apirtdb = new APIRTDB(); //obtener posicionInicial, último registro insertado en 'SmartDriveOystaCalc' para 'deviceLicense'= '0000DSC' def query = "db.SmartDriveOystaCalc.find({'Feed.deviceLicense': 0000DSC'},{'Feed.geometry.coordinates': 1}).sort({'Feed.timestamp': -1}).limit(1);"; def resQuery = apirtdb.nativeQuery(query);</pre>



8 APIScripts

8.1 executeRScript(scriptName, param)

```
/**
 * Ejecuta un Script R admite parámetros de entrada
 *
 * @param scriptName String
 * @param param String
 */
def executeRScript(scriptName, param);
```

Descripción	Ejecución de un script R parametrizable
Parámetros	<ul style="list-style-type: none"> String scriptName: nombre del script a ejecutar, ubicado por un administrador del sistema en la ruta de scripts del servidor String param: parámetros de ejecución
Resultado	String retorno
Ejemplo uso	<pre>def apiscripts = new APIScripts(); def retorno2 = apiscripts.executeRScript("scriptParamPrueba.R","valid"); if (retorno2.contains("valid")){ return true; } else { return false; }</pre>

8.2 executeRScript(scriptName)

```
/**
 * Ejecuta un Script R sin parámetros de entrada
 *
 * @param scriptName String
 */
def executeRScript(scriptName) throws ScriptException;
```

Descripción	Ejecución de un script R sin parámetros
Parámetros	<ul style="list-style-type: none"> String scriptName: nombre del script R a ejecutar, ubicado por un administrador del sistema en la ruta de scripts del servidor
Resultado	String retorno
Ejemplo uso	<pre>def apiscripts = new APIScripts(); def retorno = apiscripts.executeRScript("scriptPrueba.R"); if (retorno.contains("valid")){ return true; } else { return false; }</pre>

8.3 executeMongoScript(scriptName, param)

```
/**
```

<pre>* Ejecuta un Script Mongo * * @param scriptName String */ def executeMongoScript(scriptName, param) throws ScriptException;</pre>	
Descripción	Ejecución de un script Mongo con parámetros
Parámetros	<ul style="list-style-type: none">• String scriptName: nombre del script R a ejecutar, ubicado por un administrador del sistema en la ruta de scripts del servidor• String param: parámetros de ejecución
Resultado	String retorno
Ejemplo uso	<pre>def apiscripts = new APIScripts(); apiscripts.executeMongoScript("scriptPrueba.js");</pre>



9 APISSAP

9.1 sendInsertMessage(token, kpInstance, ontologyName, data)

```
/**
 * Inserts the data passed as a instance of the ontology passing token an kp
 * @param token String
 * @param instance of KP String
 * @param ontologyName String
 * @param data of the instance String
 * @return result SSAPMessage
 */
def sendInsertMessage(token,kpInstance,ontologyName,data) throws ScriptException;
```

Descripción	Método para enviar un correo electrónico a varios destinatarios
Parámetros	<ul style="list-style-type: none"> String token: token de conexión String kpInstance: instancia del KP String ontologyName: nombre de la ontología String data: JSON con el mensaje SSAP
Resultado	SSAPMessage result
Ejemplo uso	<pre>def apissap = new APISSAP(); def mydate = new Date(); String insDate= "{"<dollar>date': '"+mydate.format("yyyy-MM-dd'T'HH:mm:ss.SSS'Z'")+""}"; String myIsntance="{ 'MoteSensor':{ 'mote_id':{ 'id':1, 'serial_number': 'E STRELLA'}, 'humidity':1, 'temperature':1, 'battery':1, 'timestamp': { '<dollar>date': '" +mydate.format("yyyy-MM-dd'T'HH:mm:ss.SSS'Z'")+""}}}"; String token = "cbc6135f48104195bf10aae6be20c5c9"; String kpInstance = "Kp_Mote:Kp_Mote01"; String nameOntology = "MoteSensor"; def resultsendInsertMessage=apissap.sendInsertMessage(token,kpInsta nce,"MoteSensor", myIsntance);</pre>



10 APITwitter

10.1 sendTweet(apiKey, apiKeySecret, token, tokenSecret, parameters)

```
/**
 * Envía un tweet a la cuenta definida por los parámetros apiKey y token
 * @param apiKey String
 * @param apiKeySecret String
 * @param token String
 * @param tokenSecret String
 * @param parameters Map<String, String>
 * @return void
 */
def sendTweet(apiKey, apiKeySecret, token, tokenSecret, parameters);
```

Descripción	Método para publicar un tweet en la cuenta cuyas credenciales se reciben como parámetro.
Parámetros	<ul style="list-style-type: none"> Claves de la cuenta de twitter <ul style="list-style-type: none"> String apiKey String apiKeySecret String token String tokenSecret Map parameters: parámetros de invocación al API de twitter. Ej: "status": "texto del tweet".
Resultado	String id insertado
Ejemplo uso	<pre>def apiKey = "XXXXXXXXXXXXXXXXXXXXXXXXXXXX"; def apiKeySecret = "XXXXXXXXXXXXXXXXXXXXXXXXXXXX"; def token = "XX"; def tokenSecret = "XX"; def personDetails = [text:"hemos llegado ya a Roma?", password:"XXX",// contraseña de proxy port:"8080", user:"XXXXX", host: "proxy.indra.es"]; def apitwitter = new APITwitter(); String resultado = apitwitter.sendTweet(apiKey, apiKeySecret, token, tokenSecret, personDetails);</pre>

11 APIUtils

11.1 getValue(ontologyName, attribute)

```
/**
 * get the value of an attribute from an ontology
 * @param ontologyName String
 * @param attribute String
 * @return value of the attribute String
 */
def getValue(ontologyName, attribute) throws ScriptException;
```

Descripción	Método para extraer un valor simple de una instancia de ontología.
Parámetros	<ul style="list-style-type: none"> String ontologyName: nombre de la ontología sobre la que se hará la consulta String attribute: atributo del cual se recupera el valor.
Resultado	String: valor del atributo
Ejemplo uso	<p>Sobre la ontología TestSensorTemperatura:</p> <pre>def apiutils = new APIUtils(); String attribute = "Sensor.assetId"; def resultado = apiutils.getValue(ontology, attribute);</pre>

11.2 getValueJson(instancia, attribute)

```
/**
 * get the value of an attribute from an ontology in JSON
 * @param ontologyName String
 * @param attribute String
 * @return value of the attribute in JSON String
 */
def getValueJson(instancia, attribute) throws ScriptException;
```

Descripción	Método para extraer un json de una instancia de ontología. Útil cuando queremos extraer un subdocumento
Parámetros	<ul style="list-style-type: none"> String instancia: nombre de la ontología sobre la que se hará la consulta String attribute: atributo del cual se recupera el valor.
Resultado	String: valor del atributo en JSON
Ejemplo uso	<p>Sobre la ontología TestSensorTemperatura:</p> <pre>def apiutils = new APIUtils(); String attribute2 = "Sensor.geometry"; def resultadoJSON = apiutils.getValueJson(ontology, attribute2);</pre>

11.3 getValueJsonArray(instancia, atribute)

<pre> /** * get the array of values of an attribute from an ontology * @param ontologyName String * @param atribute String * @return value of the attribute as array List<String> */ def getValueJsonArray(instancia, atribute) throws ScriptException; </pre>	
Descripción	Método para extraer un array de valores de un atributo de ontología.
Parámetros	<ul style="list-style-type: none"> String ontologyName: instancia de la ontología sobre la que se hará la consulta String atribute: atributo del cual se recupera el valor.
Resultado	List<String>: valores del array
Ejemplo uso	<p>Sobre la ontología TestSensorTemperatura:</p> <pre> def apiutils = new APIUtils(); String atribute3 = "Sensor.geometry.coordinates"; def resultadoArray = apiutils.getValueJsonArray(ontology, atribute3); </pre>



12 APISofia

Este API es un aglutinador del resto de APIS. Se recomienda utilizar el API correspondiente.

