APIS SCRIPT

JULY 2015

Version 1

indra

1 INDEX

2 INTRODUCTION

2.1 Requirements

We recommend reading the following guides:

- **First steps with Sofia2** (Pdf)
- **Script Rules creation** (Pdf)

2.2 Current document´s goals and scope

This guide describes the APIS that can be used from the Scripting Rules engine of the Sofia2 Platform.

This guide includes:

- List of the available APIS
- List of the methods of each API
- Tasks of each method
- Input parameters of each method
- Results returned by each method
- Example of use of each method

2.3 APIS Scripting

The APIS that are available are:

API	Description
APIHttpConnection	API that offers various methods for establishing HTTP connections from a script
APIJMS	API to publish on JMS queues and topics from a script
APILog	API to log information from a script
APIMail	API to send mails from a script
APIRTDB	API to do operations on the real time DB from a script
APIScripts	API to invoke external scripts in R, Python,…
APISofia	**(DEPRECATED)** API unifying all operations.

APISSAP	API for SSAP communications from a script
APITwitter	API to publish tweets from a script
APIUtils	API of utilities

3 APIHttpConnection

3.1 httpGET(url)

```
/**
 * Invoke an URL by GET HTTP method
 * @param url String
 * @return content of the URL String
 */
def httpGET(url) throws ScriptException;
```

Description	HTTP petition GET to a url
Parameters	String **url**: URL which is invoked through GET
Results	String contenido: content of the URL

Use example	def apihttpconnection = new APIHttpConnection();
	def url = "https://www.google.es/search?q=sofia2";
	def contenido = apihttpconnection.httpGET(url);
	println("Contenido httpGET" + contenido);

3.2 httpGET(url, timeout)

```
/**
 * 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;
```

Description	HTTP petition GET to a url with defined timeout
Parameters	• String **url**: URL which is invoked through GET • int **timeout**: timeout (ms)
Results	String contenido: content of the URL

Use example	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);

3.3 httpGET(url, headers, parameters)

```
/**
 * 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;
```

Description	HTTP petiton GET to a url passing headers and parameters

Parameters	• String **url**: URL which is invoked through GET • String[] **headers**: headers of the petition • String[] **parameters**: parameters of the invocation
Results	String contenido: content of the URL
Use example	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);

3.4 httpPOST(url, parameters)

```
/**
 * 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;
```

Description	HTTP petition POST to a url passing headers and parameters
Parameters	• String **url**: URL which is invoked through GET • String[] **parameters**: parameters of the invocation
Results	String contenido: contenido de la URL
Use example	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);

3.5 httpPOST(url, parameters, timeout)

```
/**
 * 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
```

```
        */
       def httpPOST(url, parameters, timeout) throws ScriptException;
```

Description	HTTP petition POST to a url passing headers and parameters with timeout
Parameters	• String **url**: URL which is invoked through GET • String[] **parameters**: parameters of the invocation • int **timeout**: timeout (ms)
Results	String contenido: content of the URL
Use example	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);

.

3.6 httpPOST(url, parameters, headers, body)

```
      /**
         * 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;
```

Description	HTTP petition POST to a url passing parameters with headers and body
Parameters	• String **url**: URL which is invoked through GET • String[] **parameters**: parameters of the invocation • String[] **headers**: headers of the petition • String **body**: body of the petition
Results	String contenido: content of the URL
Use example	def apihttpconnection = new APIHttpConnection(); String [] headers = ["X-SOFIA2-APIKey:9806ed2dfe3f4bdab78c5795c3a38878"]; String [] param=null;

	String urlSemaforoAlarma="http://sofia2.com/sib-api/api/v2/semaforoalarmaes"; 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':'restPlan'}}"; String resPOST3 = apihttpconnection.httpPOST(urlSemaforoAlarma,param,headers,instanciaAlarma);

3.7 httpPUT(url, parameters, headers, body)

```
        /**
           * 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;
```

Description	HTTP petition PUT to a url passing parameters with headers and body
Parameters	• String **url**: URL which is invoked through GET • String[] **parameters**: parameters of the invocation • String[] **headers**: headers of the petition • String **body**: body of the petition
Results	String contenido: content of the URL
Use example	def apihttpconnection = new APIHttpConnection(); String [] headers = ["X-SOFIA2-APIKey:9806ed2dfe3f4bdab78c5795c3a38878"]; String [] param=null; String urlAlarma = "http://sofia2.com/sib-api/api/v1/alarmaes/"; String putById= "{'Alarma':{'mensajeAlarma':'Temperatura Demasiado Alta','procedenciaAlarma': 'Casa','causa': 'Más de 100 grados', 'timestamp': 0}}"; String resPUT = apihttpconnection.httpPUT(urlAlarma,param,headers,putById);

.

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;
```

Description	Sends an Ontology Instance to a JMS queue
Parameters	• String **ontology**: ontology instance on which on which we will query • String **queue**: JMS queue
Results	void

Use example	Where the ontology is delayJustification
	
	def apijms = new APIJMS();
	apijms.sendJson2Queue(ontology, "cola.sensor.ambiental");

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;
```

Description	Sends an ontology instance to JMS topic
Parameters	• String **ontology**: ontology instance on which we will query • String **topic**: topic JMS
Results	void

Use example	On the ontology delayJustification:
	
	def apijms = new APIJMS();
	apijms.sendJson2Topic(ontology, "topicPrueba");

4.3 sendXml2Queue(ontologyName, ontology, queue)

```
/**
   * 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;
```

Description	Sends an ontology instance as XML to a JMS queue
Parameters	• String **ontologyName**: name of the ontology on which we will query • String **ontology**: ontology instance on which we will query • String **queue**: JMS queue
Results	void

Use example	On the ontology delayJustification:
	
	def apijms = new APIJMS();
	
	apijms.sendXml2Queue(ontologyName, ontology, "ADIF.ESB.DELAYJUSTIFICATION");

4.4 sendXml2Topic(ontologyName, ontology, topic)

```
/**
   * 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;
```

Description	Sends an ontology instance as XML to the topic JMS
Parameters	• String **ontologyName**: name of the ontology on which we will query • String **ontology**: ontology instance on which we will query • String **topic**: topic JMS
Results	void

Use example	On the ontology delayJustification:
	
	def apijms = new APIJMS();
	apijms.sendXml2Topic(ontologyName, ontology,"topicPrueba");

5 APILog

5.1 logDebug(message)

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

Description	Method to log a DEBUG type message in the defined Appdender.
Parameters	String **message**: message to publish
Results	void
Use example	def apilog = new APILog(); apilog.logDebug("We are debugging the block THEN");

5.2 logInfo(message)

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

Description	Method to log an INFO type message in the defined Appender.
Parameters	String **message**: message to publish
Results	void
Use example	def apilog = new APILog(); apilog.logInfo("The condition if is met");

5.3 logWarn(message)

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

Description	Method to log a WARN type message in the defined Appender.
Parameters	String **message**: message to publish
Results	void
Use example	def apilog = new APILog(); apilog.logWarn("Attention this can provoke an Error of division by 0");

5.4 logError(message)

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

Description	Method to log an ERROR type message in the defined Appender.
Parameters	String **message**: message to publish
Results	void
Use example	def apilog = new APILog(); apilog.logError("An error has occured in the execution of the script APILog");

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;
```

Description	Method to send emails to different recipients
Parameters	• String[] **to:** recipients • String **subject:** subject • String **msg:** body of the message
Results	void
Use example	def apimail = new APIMail(); api.sendMail(["mymail@mail.com","mymail2@mail.com"],"Subject of the mail","Message of the mail")

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;
```

Description	Method to make queries on the RTDB
Parameters	• String **ontologyName:** name of the ontology on which we will query • String **query:** native query
Results	String result: result of the query
Use example	http://about.sofia2.com/2015/06/09/comando-de-scripting-getinbdtr/ def apirtdb = new APIRTDB(); String valor = apirtdb.query("MeteoMiguel10 ", "{{'MeteoMiguel10.Temperature':{<dollar>gt:99}}}");

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;
```

Description	Method to perform an insert of data of an ontology of the Real Time Database
Parameters	• String **ontologyName:** name of the ontology we will query • String **data:** instance on the ontology to insert
Results	String result: result of the insert

Use example	def apirtdb = new APIRTDB();
	
	String insDate= "{'<dollar>date': '"+mydate.format("yyyy-MM-dd'T'HH:mm:ss.SSS'Z'")+"'}";
	
	String myIsntance="{'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", myIsntance);

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	
Description	Data Insert in an ontology of the Real Time Database
Parameters	• ExecutionContext **executionContext:** Object composed of sessionKey and ontology name • String **ontologyName:** name of the ontology on which we will query • String **data:** ontology instance to insert
Results	String result: result of the insert

Use example	def apirtdb = new APIRTDB();
	
	String insertData="{'Chat':{ 'autor':'string','mensaje':'string'}}";
	
	/* Para obtemer 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);

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;	
Description	Method to perform update of data of an ontology of the Real Time Database
Parameters	• String ontologyName: name of the ontology we want to update • String query: query • String data: data to modify
Results	String result: result of the update

Use example	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);

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;
```

Description	Deletes the ontology record of an ID
Parameters	• String ontologyName: ontology • String id: identifier OID
Results	String result: result of the delete

Use example	def apirtdb = new APIRTDB();
	
	def resultDelete= apirtdb.delete("MoteSensor",resultInsert);

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;
```

Description	Geoespatial query to find inside a circule
Parameters	• String **ontologyName**: name of the ontology on which we will query • String **indexName**: name of the indexo n which we will query • Double **longitude**: longitude coordinate • Double **latitude**: latitude coordinate • Int **radio**: radius expressed in meters
Results	String result: result of the query

Use example	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;
	}

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

Wrapper of the query nearsphere of MongoDB: http://docs.mongodb.org/manual/reference/operator/query/nearSphere/

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);
```

Description	Execution of a parameterizable R script
Parameters	• String **scriptName**: name of the script to execute, located by a system administrator in the server´s script route • String **param**: execution parameters
Results	String retorno

Use example	def apiscripts = new APIScripts();
	
	def retorno2 = apiscripts.executeRScript("scriptParamPrueba.R","valid");
	if (retorno2.contains("valid")){
	return true;
	}
	else {
	return false;
	}

8.2 executeRScript(scriptName)

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

Description	Execution of a R script without parameters
Parameters	• String **scriptName**: name of the R script to execute, located by a system administrator in the server´s script route
Results	String return

Use example	def apiscripts = new APIScripts();
	def retorno = apiscripts.executeRScript("scriptPrueba.R");
	
	if (retorno.contains("valid")){
	return true;
	} else {
	return false;
	}

8.3 executeMongoScript(scriptName, param)

```
/**
```

```
        * Ejecuta un Script Mongo
        *
        * @param scriptName String
        */
            def executeMongoScript(scriptName, param) throws ScriptException;
```

Description	Execution of a Mongo script with parameters
Parameters	• String **scriptName**: name of the Mongo script to execute, located by a system administrator in the server´s script route • String **param**: Execution parameters
Results	String return

Use example	def apiscripts = new APIScripts();
	
	apiscripts.executeMongoScript("scriptPrueba.js");

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;
```

Description	Méthod to send an email to different recipients
Parameters	• String **token**: connection token • String **kpInstance**: KP instance • String **ontologyName**: name of the ontology • String **data**: JSON with the SSAP message
Results	SSAPMessage result

Use example	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':'ESTRELLA'},'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,kpInstance,"MoteSensor", myIsntance);

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);
```

Description	Method to publish a tweet in the account recived in the parameters.

Parameters	• Twitter account keys
	  ○ String **apiKey**
	  ○ String **apiKeySecret**
	  ○ String **token**
	  ○ String **tokenSecret**
	• Map **parameters**: Invocation parameters of twitter API. Ej: "status": "text of the tweet".

Results	String id inserted
Use example	def apiKey = "XXXXXXXXXXXXXXXXXXXXXXXXXX"; def apiKeySecret = "XXXXXXXXXXXXXXXXXXXXXXXXXX"; def token = "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"; def tokenSecret = "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"; def personDetails = [text:"have we reached Rome?", 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);

11 APIUtils

11.1 getValue(ontologyName, atribute)

```
/**
   * 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;
```

Description	Method to extract a simple value from an ontology instance.
Parameters	• String **ontologyName**: name of the ontology on which we will query • String **attribute**: attribute of which we will recover the value.
Results	String: attribute value

Use example	On the ontology TestSensorTemperatura:
	
	def apiutils = new APIUtils();
	String atribute = "Sensor.assetId";
	def resultado = apiutils.getValue(ontology, atribute);

11.2 getValueJson(instancia, atribute)

```
/**
   * 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, atribute) throws ScriptException;
```

Description	Method to extract the json on an ontology instance. Useful when we want to extract a subdocument
Parameters	• String **instancia**: name of the ontology on which we will query • String **atribute**: attribute which we will recover the value.
Results	String: value of the attribute in JSON

Use example	On the ontology TestSensorTemperatura:
	
	def apiutils = new APIUtils();
	String atribute2 = "Sensor.geometry";
	def resultadoJSON = apiutils.getValueJson(ontology, atribute2);

11.3 getValueJsonArray(instancia, atribute)

```
    /**
       * get the array of values of an attribute from an ontology
       * @param ontologyName String
       * @param attribute String
       * @return value of the attribute as array List<String>
       */
  def getValueJsonArray(instancia, atribute) throws ScriptException;
```

Description	Method to extract an array of values of the attribute of an ontology.
Parameters	• String **ontologyName**: Ontology instance on which we will query • String **atribute**: attribute which we will recover the value.
Results	List<String>: values of the array

Use example	On the ontology TestSensorTemperatura:
	
	def apiutils = new APIUtils();
	String atribute3 = "Sensor.geometry.coordinates";
	def resultadoArray = apiutils.getValueJsonArray(ontology, atribute3);

12 APISofia

This API unifies the other APIs. We recommend using the appropriate API.