eastnets

Eastnets

SafeWatch Screening

JAVA API

December 2022

COPYRIGHTS

TRADEMARKS

DISCLAIMER

TABLE OF CONTENTS

1 INTRODUCTION

The Eastnets SafeWatch Screening Java API allows third party companies and customers to develop applications that can interact with the SafeWatch Screening services. The API Adapter interacts with all its clients using an XML protocol on TCP/IP connections. The Java API has been developed to reduce the complexity of the communication. All the communication aspects and the XML protocol buffer creation/parsing are included in the API.

The Java API method described in this document applies for the Enterprise version of Eastnets SafeWatch Screening.

This document provides detailed instructions for architects and developers who need to develop new applications interacting with the Eastnets SafeWatch Screening server to install and use JAVA API for Eastnets SafeWatch Screening application.

2 INSTALLATION REQUIREMENTS

Operating System:

The Eastnets SafeWatch Screening Java API is available on the following platforms:

- Windows Server 2008 R2
- IBM AIX 5L version 6.x

Java Runtime Library:

The Eastnets SafeWatch Screening Java API has been developed and tested with JRE/JDK version 11.

Delivery:

The Java API is supplied as a jar file named “jswapi.jar”.

3 SAFEWATCH JAVA API CLASSES

The SafeWatch Java API is composed of a set of Container classes and a Master class; “SafeWatchApi” class, available in the package named “com.side.ofac.jswapi”. The goal of the Container classes is to provide the needed set of parameters for the SafeWatchApi class methods, and to retrieve a set of the data sent back by the server after the methods are called.

To use the Java API method; you should fill the Java container class fields, and call the API method, and then check the result available in the response container class. For each business method, there is a request “Container” class containing a set of fields with the data to supply the SafeWatch Screening application, and a response “Container” class containing fields filled with the server response data.

3.1 Container Classes

Each container class has a set of “SetXXX” and “GetXXX” methods allowing setting/ retrieving values of variables. Each container class has also a “Reset” method, this method is to reset the content of all class variables.

The following is a list of the Container Classes:

SafeWatchApiAddGGRequest	SafeWatchApiAddGGResponse
SafeWatchApiCheckAlertRequest	SafeWatchApiCheckAlertResponse
SafeWatchApiCheckDetectionRequest	SafeWatchApiCheckDetectionResponse
SafeWatchApiCreateAlertRequest	SafeWatchApiCreateAlertResponse
SafeWatchApiLoginRequest	SafeWatchApiLoginResponse
SafeWatchApiLogoutRequest	SafeWatchApiLogoutResponse
SafeWatchApiScanRequest	SafeWatchApiScanResponse
SafeWatchApiReport	SafeWatchApiReportEntity
SafeWatchApiReportEntityAddress	SafeWatchApiUpdateAlertRequest
SafeWatchApiUpdateAlertResponse	SafeWatchApiRepeatModelRequest
SafeWatchApiRepeatModelResponse	SafeWatchApiReportContextCondition

Each of the above classes will be described in the following sections.

3.1.1 SafeWatchApiAddGGRequest

This class is used to add Good Guy information and pass it to the server.

Method	Parameter	Type	Flags	Description
Reset	-	-	-	Reset all variables content.
setEntityId	EntityId	Int	Mandatory	Entity identifier.
setAcceptedString	acceptedString	String	Mandatory	Accepted string to be used for scanning.
setComments	comments	String	Optional	Set comments.
setCondition	condition	String	Optional	Set conditions.
setEntityName	entityName	String	Mandatory	Set the entity name.
setReportViolations	reportViolations	Boolean	Optional	Whether to display the related violations or not.
setZoneId	zoneId	String	Optional	Zone ID.

3.1.2 SafeWatchApiAddGGResponse

This class is filled with the server response data.

Method	Parameter	Type	Description
Reset	-	-	Reset all variables content.
getGGListId	/	Int	Return the Good Guy list identifier.

3.1.3 SafeWatchApiCheckAlertRequest

This class is used to store the “Check Alert” information and pass it to the server.

Method	Parameter	Type	Flags	Description
Reset	-	-	-	Resets all variables content.
setDetectionId	DetectionId	Int	Optional	Detection Identifier.
setAlertId	AlertId	Int	Optional	Alert Identifier.

3.1.4 SafeWatchApiCheckAlertResponse

This class is filled with the server response data.

Method	Parameter	Type	Description
Reset	-	-	Resets all variables content.
getAlertId	/	Int	Returns the alert identifier.
getStatus	/	String	Returns the alert status.
getAssignedTo	-	String	Returns Assigned To value.

3.1.5 SafeWatchApiCheckDetectionRequest

This class is used to store the “Check Detection” information and pass it to the server.

Method	Parameter	Type	Flags	Description
Reset	-	-	-	Resets all variables content.
setDetectionId	DetectionId	Int	Optional	Detection Identifier.

3.1.6 SafeWatchApiCheckDetectionResponse

This class is filled with the server response data.

Method	Parameter	Type	Description
Reset	-	-	Resets all variables content.
getDetectionId	/	Int	Returns the Detection Identifier.
getGlobalStatus	/	String	Returns the global detection status (from its alerts).
getAlertCount	/	Int	Returns the number of alerts associated with the detection.
getAlertId	Int	Int	Returns the Id of the specified alert.
getStatus	Int	String	Returns the status of the specified alert.

Note:

- *Use the Alert Count to loop the list of available alerts and get the status of each.*

3.1.7 SafeWatchApiCreateAlertRequest

This class is used to store the “Create Alert” information and pass it to the server.

Method	Parameter	Type	Flags	Description
Reset	-	-	-	Resets all variables content.
setDetectionId	DetectionId	Int	Mandatory	Detection Identifier.
setDetectionPosition	Detection position	Int	Mandatory	Detection Position.
setAssignTo	AssignedTo	String	Mandatory	The name of the user to which the alert is assigned to.
setAssignToGroup	AssignedToGroup	String	Optional	The name of the group to which the alert is assigned to.
setComment	Comment	String	Optional	The comment associated with the alert creation.

3.1.8 SafeWatchApiCreateAlertResponse

This class is filled with the server response data.

Method	Parameter	Type	Description
Reset	-	-	Resets all variables content.
getAlertId	/	int	Returns the Alert Identifier.

3.1.9 SafeWatchApiLoginRequest

This class is used to store the login information and pass it to the server.

Method	Parameter	Type	Flags	Description
Reset	-	-	-	Resets all variables content.
setUser	User	String	Optional *	User name.
setPassword	Password	String	Optional *	User password.
setApplication	Application	String	Optional	SafeWatch application name.
setLoginFile	Filename	String	Optional *	SafeWatch login file name.
setServerIp	IpAddress	String	Mandatory	TCP/IP address of the API Adapter.
setServerPort	Port	Int	Mandatory	TCP/IP port of the API Adapter.
setZoneId	Zone Id	String	Mandatory	User Zone ID.
setConnectionTimeout	ConnectionTimeout	Int	Optional *	Socket Connection Timeout (milliseconds)
setReadTimeout	ReadTimeout	Int	Optional *	Socket Read Timeout (milliseconds)

** Note: You should provide a combination of the user name and password, or a login file.*

3.1.10 SafeWatchApiLoginResponse

This class is filled with the server response data.

Method	Parameter	Type	Description
Reset	-	-	Resets all variables content.
getDefaultDetectCountry	-	Int	Zero = false. 1 = true. If true, scanning will detect countries by default.
getDefaultDetectVessels	-	Int	Zero = false. 1= true. If true, scanning will detect vessels by default.
getDefaultRank	-	Int	Default rank value.
getProfileName (Deprecated)	-	String	The profile name of the assigned user.
getServerEdition	-	String	Server edition.
getServerVersion	-	String	Server version.
getUserZoneID	-	Int	User zone ID.
getListSetID	-	Int	List set ID.
getProfilesCount	-	Int	Number of profiles the user is associated with.
getGroupsCount	-	Int	Number of groups the user is member of.
getProfileID	index	Int	ID of the specified profile.
getProfileName	index	Int	Name of the specified profile.
getGroupID	index	Int	ID of the specified group.
getGroupName	index	Int	Name of the specified group.

Notes:

- *The values of the Default Detect Countries, Detect Vessels and Rank options, are set from the AMLUI List Set details screen.*
- *You can override the values of these options using the parameters passed through class "SafeWatchApiScanRequest" using methods setRank, setCheckVessels and setCheckCountry.*
- *Use Profile and Group Count values in a loop to get the list of profiles and groups IDs and the names for the logged-in users.*

3.1.11 SafeWatchApiLogoutRequest

This class is used to store the logout information and pass it to the server.

Method	Parameter	Type	Description
Reset	-	-	Resets all variables content.

3.1.12 SafeWatchApiLogoutResponse

This class is filled with the server response data.

Method	Parameter	Type	Description
Reset	-	-	Reset all variables content.

3.1.13 SafeWatchApiScanRequest

This class is used to store the scan information and pass it to the server.

Method	Parameter	Type	Flags	Description
Reset	-	-	-	Resets all variables content.
setData	Data	String	Mandatory	Data buffer to scan.
setListSetId	List set Id	String	Mandatory	The list set to be used for scanning.
setAddress	Address	String	Optional	Address.
setBic	BIC	String	Optional	BIC Code.
setCity	City	String	Optional	City Name.
setContext	Context	String	Optional	Scan context string.
setCountry	Country	String	Optional	Country Name.
setFormat	Format	String	Mandatory	Format of the data to be scanned (can be TEXT, RJE, ISO20022, NAME or USER).
setRecordId	Record Id	String	Optional	Record Identifier.
setRecordLocation	Record location	String	Optional	Information to specify the data location (for example a file name).
setScanSessionId	Session Id	String	Optional	Session Identifier specified by the client.

setRank	Rank Value	Int	Optional	Rank used for the scan.
setCheckVessels	Vessel Detection Flag	Int	Optional	Zero = false, 1 = True. If true, checks vessels.
setCheckCountry	Country Detection Flag	Int	Optional	Zero = false, 1 = True. If true, checks countries.
setAutoCreateAlert	Auto create alert flag	Boolean	Optional	If true, creates alert when violations are detected.
setPositiveDetection	Create positive detections flag	Boolean	Optional	If true, creates a detection even when no violations are detected.
setFullReport	Full report flag	Boolean	Optional	If true, the engine will return all entity information.
setRecordXPath	Record XPath	String	Optional	XML file record XPath

Notes:

- *The setdata function parameter depends on the selected format type. Example: If (text): you can pass any generic text. If (user): the data passed will depend on the created customer format.*
- *If the setRank is set to zero, it will be ignored and the rank of the List Set will be used.*
- *You need to start a scan session when scanning with a user-defined format. If the record delimiter of the format, which is set from the "Format Manager" screen in the AMLUI, is XML, this means you are going to scan XML files. In this case, use setRecordXpath method to pass the XPath value returned in the SafeWatchApiStartScanSessionResponse.*

3.1.14 SafeWatchApiScanResponse

This class is filled with the server response data:

Method	Parameter	Type	Description
Reset	-	-	Resets all variables content.
getAcceptedCount	-	Int	Number of "Accepted" violations (Good Guys list).
getDetectionId	-	Int	Detection Identifier (= key of detection record).
getExternalCount	-	Int	Number of "External" violations (Matching Rules).

getReport	-	Vector	Returns a vector of "SafeWatchApiReport" objects containing all violations information.
getViolationCount	-	Int	Number of "real" violations.

3.1.15 SafeWatchApiReport

This class is filled with the report information:

Method	Parameter	Type	Description
Reset	-	-	Resets all variables content.
getBeginPosition	-	Int	Start position in the scanned buffer of data that match.
getCategory	-	String	Matching category.
getData	-	String	Matching data.
getEndPosition	-	Int	End position in the scanned buffer of data that match.
getEntity	-	Vector	Returns a vector of "SafeWatchApiReportEntity" objects containing all entities information.
getEntityId	-	Int	Matched entity Identifier.
getField	-	String	Field name (RJE format only).
getLine	-	Int	Matching line number.
getListDate	-	String	Matching list date.
getListName	-	String	Matching list name.
getMatch	-	String	Matching data.
getInputBic	-	String	Scanned BIC.
getInputAddress	-	String	Expanded BIC address.
getInputCity	-	String	Expanded BIC city.
getInputCountry	-	String	Expanded BIC country.
getProgram	-	String	Program name.
getExternalId	-	String	Matched entity external Identifier.
getRank	-	Int	Matching rank.
getRemark	-	String	Matching remark.
getStatus	-	String	Status (For example Reported).
getTitle	-	String	Title.
getDOB	-	String	Date of birth.
getPOB	-	String	Place of birth.
getGender	-	String	Gender of entity in case it was an individual.

getDataSource	-	Vector	Data sources. This information will be returned from the server only if full report option found in SafeWatchApiScanRequest class is set to true via setFullReport function.
getStatusCause	-	String	This information will only be returned from the server if violation status is not reported, i.e. accepted or external.
getEntityNationalities	-	Vector	Returns a vector of string objects containing all entity's nationalities.
getEntityBirthDates		Vector	Returns a vector of string objects containing all entity's birth dates.

3.1.16 SafeWatchApiReportEntity

This class is filled with the report entity information:

Method	Parameter	Type	Description
Reset	-	-	Resets all variables content.
getName	-	String	Entity name.
getNameType	-	String	Entity name type.

3.1.17 SafeWatchApiReportEntityAddress

This class is filled with the report entity address:

Method	Parameter	Type	Description
Reset	-	-	Resets all variables content.
getAddress	-	String	Address.
getCity	-	String	City.
getCountry	-	String	Country.

3.1.18 SafeWatchApiReportContextCondition

This class is filled with the report context condition:

Method	Parameter	Type	Description
Reset	-	-	Resets all variables content.
getContextName	-	String	Context condition name.
getContextValue	-	String	Context condition value.

3.1.19 SafeWatchApiUpdateAlertRequest

This class is used to store the “Update Alert” information and pass it to the server.

Method	Parameter	Type	Flags	Description
Reset	-	-	-	Resets all variables content.
setAlertId	AlertId	Int	Mandatory	Alert Identifier.
setStatus	Status	String	Mandatory	Alert Status.
setAssignedTo	AssignedTo	String	Optional	Alert Assigned To.
setComments	Comments	String	Optional	Alert Comments.

Notes:

Predefined values are supplied by the API for the possible status values:

```
public static final String API_ALERT_STATUS_NEW = ”NEW”;
public static final String API_ALERT_STATUS_INVESTIGATING = ”INVESTIGATING”;
public static final String API_ALERT_STATUS_FALSE_POSITIVE=”FALSE POSITIVE”;
public static final String API_ALERT_STATUS_REAL_VIOLATION =”REAL VIOLATION”;
public static final String API_ALERT_STATUS_DONT_KNOW = ”DONT KNOW”;
```

3.1.20 SafeWatchApiUpdateAlertResponse

This class is filled with the server response data.

Method	Parameter	Type	Description
Reset	-	-	Resets all variables content.
getStatus	-	String	Returns the action status. If no errors occurred, the value of this field will be “Success”.

3.1.21 SafeWatchApiStartScanSessionRequest

This class is used to store the data needed to start a scan session.

Method	Parameter	Type	Description
Reset	-	-	Resets all variables content.
setFormatName	Format	String	Predefined custom format name to be used for file scanning.
setDataSource	Source	String	Name of file to be scanned.

Notes:

- *Setting the name of the file to be scanned using the API does not mean the file will be scanned automatically, you need to provide your own code to open the file, read its contents and send it to the Scan service for scanning using the appropriate API methods.*

- *The value of the data source will be viewed in the Detection Manager screen under the "Data Source" column of the detections table. If you did not start a session and used the API to scan free text, the value of this field under "Data Source" column in Detection Manager will be "SafeWatch API".*

3.1.22 SafeWatchApiStartScanSessionResponse

This class is filled with server response data.

Method	Parameter	Type	Description
Reset	-	-	Resets all variables content.
getSessionID	-	String	Returns the newly created session ID.
getRecordXPath	-	String	Returns the XPath of the format in case the specified format in the SafeWatchApiStartScanSessionRequest class was XML user defined format.

3.1.23 SafeWatchApiRepeatModelRequest

This class is used to store the data needed to add a Repeat (Repeat Data Model).

Method	Parameter	Type	Description
setDetectionId	detectionId	String	Detection Identifier.
setZoneId	zoneID	String	Zone Identifier.
setComments	Comment	String	Remarks on repeat configuration data element.
setExpiryDate	Expiration date	Date	Expiration date for repeat data.

3.1.24 SafeWatchApiRepeatModelResponse

This class is filled with server response data.

Method	Parameter	Type	Description
getStatus	-	String	Returns the action status. If no errors occurred, the value of this field will be "Success".

3.1.25 SafeWatchApiReplayRequest

This class is used to store the data needed to add a Replay Violation (Replay Data Model).

Method	Parameter	Type	Description
setAlertId	alertID	Integer	Identifier of Alert wanted to be replay-able.
setInvestigator	investigator	String	Identifier of Investigator to make violation replay-able.
setDecision	values allowed (FALSE POSITIVE, REAL VIOLATION)	String	Save Replay as False Positive, or Real violation.

3.1.26 SafeWatchApiReplayResponse

This class is filled with server response data.

Method	Parameter	Type	Description
getStatus	-	String	Returns the action status. If no errors occurred, the value of this field will be “Success”.

4 SAFEWATCHAPI CLASS

This is the main class of the Eastnets SafeWatch Screening Java API. This is the class which the developers need to instantiate in their application.

4.1 Methods List

The following is the list of methods available in the SafeWatchAPI Java Class:

- Login
- Logout
- StartScanSession
- EndScanSession
- Scan
- CreateAlert
- CheckAlert
- CheckDetection
- UpdateAlert
- AddGoodGuy
- AddRepeatModel
- CreateReplayViolation
- GetLastErrorText
- GetLastErrorCode

These methods interact with the SafeWatch Screening services to perform the requested operations.

Note:

- *The 4-Eyes principle does not work with API connectors.*

4.2 Methods Description

The following sub-sections describe the methods available in the SafeWatchAPI Java Class.

4.2.1 GetLastErrorText

Description:
This function returns the last error reason text.

Prototype:
Public String getLastErrorText ().

Parameters:
/

Return Code:
This function returns a string containing the last error reason.

4.2.2 GetLastErrorCode

Description:
This function returns the last error reason code.

Prototype:
Public int getLastErrorCode ()

Parameters:
/

Return Code:
This function returns the code of the last error reason. The following is a list of the possible error codes:

Code	Description
0	Success.
-9001	Invalid parameter supplied to method.
-9002	Connection to API Adapter failed.
-9003	Send request to API Adapter failed.
-9004	Receive response from API Adapter failed.
-9005	Unable to extract XML tag value from response.
-9006	Generic error.
-9007	Bad login file.
-9008	Host name mismatch (while login).
-9009	Application name mismatch (while login).

4.2.3 Login

Description:

This function is to connect and login to the SafeWatch Screening application.

Prototype:

Public int Login (SafeWatchApiLoginRequest request, SafeWatchApiLoginResponse response).

Parameters:

Request: A filled *SafeWatchApiLoginRequest* object.

Response: An allocated *SafeWatchApiLoginResponse* object.

Return Code:

This function returns 0 in case of success, different from 0 in case of error. In case it returns a value other than 0, you can check the last error code and last error text using the GetLastErrorText, GetLastErrorCode methods.

4.2.4 Logout

Description:

This function is to disconnect and log out from the SafeWatch Screening application.

Prototype:

Public int Logout (SafeWatchApiLogoutRequest request, SafeWatchApiLogoutResponse response).

Parameters:

Request: A filled *SafeWatchApiLogoutRequest* object.

Response: An allocated *SafeWatchApiLogoutResponse* object.

Return Code:

This function returns 0 in case of success, different from 0 in case of error. In case it returns a value other than 0, you can check the last error code and last error text using the GetLastErrorText, GetLastErrorCode methods.

4.2.5 StartScanSession

Description:

This function initiates a scan session within the API Adapter.

Prototype:

Public int StartScanSession (SafeWatchApiStartScanSessionRequest request, SafeWatchApiStartScanSessionResponse response)

Parameters:

Request: a filled SafeWatchApiStartScanSessionRequest object.

Response: an allocated SafeWatchApiStartScanSessionResponse object.

Return Code:

This function returns 0 in case of success, different from 0 in case of error. In case it returns a value other than 0, you can check the last error code and last error text using the GetLastErrorText, GetLastErrorCode methods.

4.2.6 EndScanSession

Description:

This function terminates a scan session within the API Adapter.

Prototype:

Public int EndScanSession (String application, int sessionId)

Parameters:

Application: A String object that contains the application name.

SessionId: An integer value of the session Id that need to be terminated.

Return Code:

This function returns 0 in case of success, different from 0 in case of error.

4.2.7 Scan

Description:

This function performs a scan operation within the API Adapter.

Prototype:

Public int Scan (SafeWatchApiScanRequest request, SafeWatchApiScanResponse response)

Parameters:

Request: A filled *SafeWatchApiScanRequest* object.

Response: An allocated *SafeWatchApiScanResponse* object.

Return Code:

This function returns 0 in case of success, different from 0 in case of error. In case it returns a value other than 0, you can check the last error code and last error text using the GetLastErrorText, GetLastErrorCode methods.

4.2.8 CreateAlert

Description:

This function creates an alert for a detection.

Prototype:

Public int CreateAlert (SafeWatchApiCreateAlertRequest request, SafeWatchApiCreateAlertResponse response).

Parameters:

Request: A filled *SafeWatchApiCreateAlertRequest* object.

Response: An allocated *SafeWatchApiCreateAlertResponse* object.

Return Code:

This function returns 0 in case of success, different from 0 in case of error. In case it returns a value other than 0, you can check the last error code and last error text using the GetLastErrorText, GetLastErrorCode methods.

4.2.9 CheckAlert

Description:

This function checks the status of an alert.

Prototype:

Public int CheckAlert (SafeWatchApiCheckAlertRequest request, SafeWatchApiCheckAlertResponse response)

Parameters:

Request: A filled *SafeWatchApiCheckAlertRequest* object.

Response: An allocated *SafeWatchApiCheckAlertResponse* object.

Return Code:

This function returns 0 in case of success, different from 0 in case of error. In case it returns a value other than 0, you can check the last error code and last error text using the GetLastErrorText, GetLastErrorCode methods.

4.2.10 CheckDetection

Description:

This function checks the status of a detection; it checks all alerts, and returns the global status and the details of each alert (Id and Status).

Prototype:

Public int CheckDetection (SafeWatchApiCheckDetectionRequest request, SafeWatchApiCheckDetectionResponse response).

Parameters:

Request: A filled *SafeWatchApiCheckDetectionRequest* object.

Response: An allocated *SafeWatchApiCheckDetectionResponse* object.

Return Code:

This function returns 0 in case of success, different from 0 in case of error. In case it returns a value other than 0, you can check the last error code and last error text using the GetLastErrorText, GetLastErrorCode methods.

4.2.11 UpdateAlert

Description:

This function updates the status of an alert.

Prototype:

Public int UpdateAlert (SafeWatchApiUpdateAlertRequest request, SafeWatchApiUpdateAlertResponse response).

Parameters:

Request: A filled *SafeWatchApiUpdateAlertRequest* object.

Response: An allocated *SafeWatchApiUpdateAlertResponse* object.

Return Code:

This function returns 0 in case of success, different from 0 in case of error. In case it returns a value other than 0, you can check the last error code and last error text using the GetLastErrorText, GetLastErrorCode methods.

4.2.12 AddGoodGuy

Description:

This function adds a Good Guy entry in the Good Guys list.

Prototype:

Public int AddGoodGuy (SafeWatchApiAddGGRequest request, SafeWatchApiAddGGResponse response).

Parameters:

Request: A filled *SafeWatchApiAddGGRequest* object.

Response: An allocated *SafeWatchApiAddGGResponse* object.

Return Code:

This function returns 0 in case of success, different from 0 in case of error. In case it returns a value other than 0, you can check the last error code and last error text using the GetLastErrorText, GetLastErrorCode methods.

4.2.13 AddRepeatModel

Description:

This function stores Repeat Data Model based on the detection to the database for comparison purposes once the repeat message is detected.

Prototype:

Public int AddRepeatModel(SafeWatchApiRepeatModelRequest request, SafeWatchApiRepeatModelResponse response).

Parameters:

Request: A filled *SafeWatchApiRepeatModelRequest* object.

Response: An allocated *SafeWatchApiRepeatModelResponse* object.

Return Code:

This function returns 0 in case of success, different from 0 in case of error. In case it returns a value other than 0, you can check the last error code and last error text using the GetLastErrorText, GetLastErrorCode methods.

4.2.14 CreateReplayViolation

Description

This function creates Replay Violation for specific Alert with certain Investigator and defines if the Replay is “False Positive” or “Real Violation”.

Prototype:

public int CreateReplayViolation(SafeWatchApiReplayRequest replayRequest, SafeWatchApiReplayResponse replayResponse)

Parameters:

Request: A filled *SafeWatchApiReplayRequest* Object.

Response: An allocated *SafeWatchApiReplayResponse* Object.

Return Code:

This function returns 0 in case of success, different from 0 in case of error. In case it returns a value other than 0, you can check the last error code and last error text using the GetLastErrorText, GetLastErrorCode methods

5 APPENDIX A: EXAMPLE

The following is an example of a code using the Eastnets SafeWatch Screening Java API:

```
import com.side.ofac.jswapi.*;

public class JClientDemo {

    // ----------------------------------------------------------------------------
    public static void main(String[] args) {
        JClientDemo demo = new JClientDemo();

        demo.testClientApi(args);
        demo.scanUsingFormatName();
    }

    // ----------------------------------------------------------------------------

    public void testClientApi(String[] args) {
        SafeWatchApi test = new SafeWatchApi();
        SafeWatchApiLoginRequest loginRequest = new SafeWatchApiLoginRequest();
        SafeWatchApiLoginResponse loginResponse = new SafeWatchApiLoginResponse();
        SafeWatchApiLogoutRequest logoutRequest = new SafeWatchApiLogoutRequest();
        SafeWatchApiLogoutResponse logoutResponse = new SafeWatchApiLogoutResponse();
        SafeWatchApiScanRequest ScanRequest = new SafeWatchApiScanRequest();
        SafeWatchApiScanResponse ScanResponse = new SafeWatchApiScanResponse();
        SafeWatchApiCreateAlertRequest AlertRequest = new SafeWatchApiCreateAlertRequest();
        SafeWatchApiCreateAlertResponse AlertResponse = new SafeWatchApiCreateAlertResponse();
        SafeWatchApiCheckAlertRequest CheckRequest = new SafeWatchApiCheckAlertRequest();
        SafeWatchApiCheckAlertResponse CheckResponse = new SafeWatchApiCheckAlertResponse();
        SafeWatchApiCheckDetectionRequest CheckDetRequest = new
SafeWatchApiCheckDetectionRequest();
        SafeWatchApiCheckDetectionResponse CheckDetResponse = new
SafeWatchApiCheckDetectionResponse();
        SafeWatchApiUpdateAlertRequest UpdateRequest = new SafeWatchApiUpdateAlertRequest();
        SafeWatchApiUpdateAlertResponse UpdateResponse = new SafeWatchApiUpdateAlertResponse();
        SafeWatchApiAddGGRequest AddGGRequest = new SafeWatchApiAddGGRequest();
        SafeWatchApiAddGGResponse AddGGResponse = new SafeWatchApiAddGGResponse();
        SafeWatchApiRepeatModelrepeatModelRequest repeatModelRequest = new
        SafeWatchApiRepeatModelrepeatModelRequest();
        SafeWatchApiRepeatModelrepeatModelResponse repeatModelResponse = new
        SafeWatchApiRepeatModelrepeatModelResponse();
        SafeWatchApiReplayRequest replayRequest = new SafeWatchApiReplayRequest();
        SafeWatchApiReplayResponse replayResponse = new SafeWatchApiReplayResponse();


        SafeWatchApiReportEntity entity = null;
        SafeWatchApiReportEntityAddress entityAddress = null;
        SafeWatchApiReport report = null;
        int i = 0;
        int j = 0;
        int k = 0;

        /* Set Connection and Read Timeout. (Optionnal)  */
        loginRequest.setConnectionTimeout (5000);
        loginRequest.setReadTimeout (5000);

        /* If ReadTimeout is reached, call soDisconnect() method who disconnects you from the API
Adapter */


        System.out.println("Test Client API starts");

        /* login */
        if (args[0] != null)
            loginRequest.setServerIp(args[0]);
        if (args[1] != null)
            loginRequest.setServerPort(Integer.parseInt(args[1]));
        if (args[2] != null)
            loginRequest.setLoginFile(args[2]);
```

```
if (test.Login(loginRequest, loginResponse) != 0) {
    System.err.println("A problem has occured while log in on en.SafeWatch  Server:\n"
                    + test.getLastErrorText()
                    + " code : "
                    + test.getLastErrorCode());
    return;
}

System.out.println("Login successfull");

/* scan */
ScanRequest.setFormat("TEXT");
ScanRequest.setData("Saddam Hussein");
ScanRequest.setContext("<test>true</test>");
ScanRequest.setListSetId("1");

if (test.Scan(ScanRequest, ScanResponse) != 0) {
    System.err.println("A problem has occured while scanning on en.SafeWatch  Server "
                    + test.getLastErrorText()
                    + " code : "
                    + test.getLastErrorCode());
    return;
}

/* display scan result */
System.out.println("** scan result **\n\n");
System.out.println("violation : " + ScanResponse.getViolationCount());
System.out.println("accept    : " + ScanResponse.getAcceptCount());
System.out.println("external  : " + ScanResponse.getExternalCount());
System.out.println("detec.id : " + ScanResponse.getDetectionId());

for (i = 0; i < ScanResponse.getReport().size(); i++) {
    report = (SafeWatchApiReport) ScanResponse.getReport().elementAt(i);
    System.out.println("\n");
    System.out.println("Status  : " + report.getStatus());
    System.out.println("Data    : " + report.getData());
    System.out.println("Match   : " + report.getMatch());
    if (report.getInputBic() != "") {
        System.out.println("   Scanned BIC : " + report.getInputBic());
        System.out.println("   BIC Address : "
                + report.getInputAddress());
        System.out.println("   BICCity     : " + report.getInputCity());
        System.out.println("   BIC Country : "
                + report.getInputCountry());
    }
    System.out.println("Rank    : " + report.getRank());
    System.out.println("List    : " + report.getListName());
    System.out.println("Listdate: " + report.getListDate());
    System.out.println("ent.id : " + report.getEntityId());
    System.out.println("Category: " + report.getCategory());
    System.out.println("Remarks : " + report.getRemark());
    System.out.println("Title   : " + report.getTitle());
    System.out.println("Position: From " + report.getBeginPosition()
            + " to " + report.getEndPosition());
    System.out.println("Field   : " + report.getField());
    System.out.println("Line    : " + report.getLine());
    System.out.println("Programs: " + report.getProgram());
    System.out.println("Date of birth: " + report.getDOB());
    System.out.println("Place of birth: " + report.getPOB());

    System.out.println("Ext. ID : " + report.getExternalId());

    for (j = 0; j < report.getEntity().size(); j++) {
        entity = (SafeWatchApiReportEntity) report.getEntity().elementAt(j);
        System.out.println("   Name Type : " + entity.getNameType());
        System.out.println("   Name      : " + entity.getName());
    }

    for (j = 0; j < report.getEntityAddresses().size(); j++) {
        entityAddress = (SafeWatchApiReportEntityAddress) report.getEntityAddresses()
                .elementAt(j);
        System.out.println("   Address : " + entityAddress.getAddress());
        System.out.println("   City    : " + entityAddress.getCity());
```

```
                System.out.println("    Country : " + entityAddress.getCountry());
            }

        }

        /* create an alert */
        AlertRequest.setDetectionId(ScanResponse.getDetectionId());
        AlertRequest.setDetectionPosition(1);
        AlertRequest.setAssignTo(loginRequest.getUser());
        AlertRequest.setComment("what a wonderfull J API !");

        if (test.CreateAlert(AlertRequest, AlertResponse) != 0) {
            System.err.println("A problem has occured while creating an alert on en.SafeWatch
Server "
                            + test.getLastErrorText()
                            + " code : "
                            + test.getLastErrorCode());
            return;
        } else
            System.out.println("Alert successfully created : id : "
                    + AlertResponse.getAlertId());

        /* check an alert status */
        CheckRequest.setDetectionId(ScanResponse.getDetectionId());
        CheckRequest.setAlertId(AlertResponse.getAlertId());

        if (test.CheckAlert(CheckRequest, CheckResponse) != 0) {
            System.err.println("A problem has occured while checking an alert on en.SafeWatch
Server "
                            + test.getLastErrorText()
                            + " code : "
                            + test.getLastErrorCode());
            return;
        } else
            System.out.println("Check Alert successfully done : status : "
                    + CheckResponse.getStatus());

        /* Update an alert status */
        UpdateRequest.setAlertId(AlertResponse.getAlertId());
        UpdateRequest.setAssignedTo(loginRequest.getUser());
        UpdateRequest.setComments("Done by JSWAPI");
        UpdateRequest.setAlertStatus(SafeWatchApi.API_ALERT_STATUS_INVESTIGATING);

        if (test.UpdateAlert(UpdateRequest, UpdateResponse) != 0) {
            System.err.println("A problem has occured while updating an alert on en.SafeWatch
Server "
                            + test.getLastErrorText()
                            + " code : "
                            + test.getLastErrorCode());
            return;
        } else
            System.out.println("Update Alert successfully done : status : "
                    + UpdateResponse.getStatus());

        /* check a detection for all its alerts */
        CheckDetRequest.setDetectionId(ScanResponse.getDetectionId());

        if (test.CheckDetection(CheckDetRequest, CheckDetResponse) != 0) {
            System.err.println("A problem has occured while checking a detection on en.SafeWatch
Server "
                            + test.getLastErrorText()
                            + " code : "
                            + test.getLastErrorCode());
            return;
        } else
            System.out.println("Check detection successfully done : global status : "
                            + CheckDetResponse.getGlobalStatus());

        System.out.println("Detection ID:" + CheckDetResponse.getDetectionId());
        System.out.println("Global Status:"
                + CheckDetResponse.getGlobalStatus());

        for (k = 0; k < CheckDetResponse.getAlertCount(); k++) {
            System.out.println("Alert ID:" + CheckDetResponse.getAlertId(k));
```

```
            System.out.println("Status:" + CheckDetResponse.getStatus(k));
        }

        /*
         * add an entry in the good guy list
         * the entity ID used here is just for demonstration purposes
         * you should use the entity ID which can be found in
         * SafeWatchApiReport class
         */
        AddGGRequest.setEntityId(76639);
        AddGGRequest.setEntityName("Saddam Hussein");
        AddGGRequest.setAcceptedString("Saddam Hussein");
        AddGGRequest.setReportViolations(true);

        /*
         * private String entityName; private String comments; private String
         * condition;
         */

        if (test.AddGoodGuy(AddGGRequest, AddGGResponse) != 0) {
            System.err.println("A problem has occured while adding an entry in GGList on
en.SafeWatch Server "
                            + test.getLastErrorText()
                            + " code : "
                            + test.getLastErrorCode());
            return;
        } else
            System.out.println("Add GG list successfully created : id : "
                    + AddGGResponse.getGGListId());

        /*
        * Testing adding repeat model base on detection
        */

        repeatModelRequest.setDetectionId("307");
        repeatModelRequest.setZoneId("1");
        repeatModelRequest.setComments("added using API test");
        repeatModelRequest.setExpiryDate(new Date());

        int i = test.AddRepeatModel(repeatModelRequest, repeatModelResponse);

        if (i==0)
                System.out.println("success");
        else
                System.out.println(repeatModelResponse.getStatus());


        /*
        * Test adding replay Violation
        */
        int alertId = 355;
        replayRequest.setAlertId(alertId);
        replayRequest.setInvestigatorId(41);
        replayRequest.setDecision("FALSE POSITIVE");

        if(test.CreateReplayViolation(replayRequest, replayResponse)!= 0){
                System.err.println("A problem has occurred while creating a replay for  alert with
                ID "+ alertId +", on en.SafeWatch  Server: "
                                                    + test.getLastErrorText()
                                                    + " code : "
                                                    + test.getLastErrorCode());
        }else{
                System.out.println("Replay Created successfully for Alert with ID: " +  alertId);
        }

        /* disconnect from server */
        if (test.Logout(logoutRequest, logoutResponse) != 0) {
            System.out.println("logout refused by en.SafeWatch  server : result : "
                            + test.getLastErrorText()
                            + " code : "
                            + test.getLastErrorCode());
        } else
            System.out.println("logout successfull ");
```

```
        System.out.println("Test Client API done");
    }

    // --------------------------------------------------------------------------

    private void scanUsingFormatName() {

        SafeWatchApi safewatchApi = new SafeWatchApi();

        SafeWatchApiLoginRequest loginRequest = new SafeWatchApiLoginRequest();
        SafeWatchApiLoginResponse loginResponse = new SafeWatchApiLoginResponse();

        loginRequest.setServerIp("localhost");
        loginRequest.setServerPort(8401);
        loginRequest.setUser("user");
        loginRequest.setPassword("password");
        loginRequest.setZoneId("1");

        if (safewatchApi.Login(loginRequest, loginResponse) != 0) {
            System.err.println("A problem has occured while log in on en.SafeWatch  Server:\n"
                    + safewatchApi.getLastErrorText()
                    + " code : "
                    + safewatchApi.getLastErrorCode());
            return;
        }

        /* Test Format is the name of a predefined user format*/
        SafeWatchApiStartScanSessionRequest sessionRequest = new
SafeWatchApiStartScanSessionRequest();
        sessionRequest.setFormatName("Test_Format");

        SafeWatchApiStartScanSessionResponse sessionResponse = new
SafeWatchApiStartScanSessionResponse();

        if (safewatchApi.StartScanSession(sessionRequest, sessionResponse) != 0) {
            System.err.println("A problem has occured while starting scan session:\n"
                    + safewatchApi.getLastErrorText()
                    + " code : "
                    + safewatchApi.getLastErrorCode());
            return;
        }

        /*
         * Assuming that 'Test_Format' has the following structure:
         * FULL_NAME;COUNTRY
         */
        SafeWatchApiScanRequest scanRequest = new SafeWatchApiScanRequest();
        scanRequest.setFormat("USER");
        scanRequest.setScanSessionId(sessionResponse.getSessionID()) ;
        scanRequest.setData("Saddam Hussein;IRAQ");
        scanRequest.setRank(60);
        scanRequest.setListSetId("2");

        SafeWatchApiScanResponse scanResponse = new SafeWatchApiScanResponse();

        if (safewatchApi.Scan(scanRequest, scanResponse) != 0) {
            System.err.println("A problem has occured while scanning on en.SafeWatch Server: "
                    + safewatchApi.getLastErrorText()
                    + " code : "
                    + safewatchApi.getLastErrorCode());
        }
        else
            System.out.println("ViolationCount :: " + scanResponse.getViolationCount());


        /* disconnect from server */
        SafeWatchApiLogoutRequest logoutRequest = new SafeWatchApiLogoutRequest();
        SafeWatchApiLogoutResponse logoutResponse = new SafeWatchApiLogoutResponse();

        if (safewatchApi.Logout(logoutRequest, logoutResponse) != 0) {
            System.out.println("logout refused by en.SafeWatch  server : result : "
                    + safewatchApi.getLastErrorText()
                    + " code : "
                    + safewatchApi.getLastErrorCode());
```

```
            } else
                System.out.println("logout successfull ");

        }
    }
```

About

Eastnets ensures peace of mind by securing a safer future for everyone.

A global provider of compliance and payment solutions for the financial services sector, our experience and expertise help ensure trust at 750 financial institutions across the world, including 11 of the top 50 banks. For more than 35 years, we have worked to keep the world safe and secure from financial crime. We do it by helping our partners manage risk through Sanction Screening, Transactions Monitoring, analysis, and reporting, plus state-of-the-art consultancy, and customer support.

Learn more at www.eastnets.com

Belgium | Egypt | Hong Kong | Jordan | Luxembourg | Pakistan | Qatar | UAE | United Kingdom | USA