SIGOS
testing is our competence

SIGOS App Experience

App Monitoring REST API

App Monitoring 8.11

October 2017

Copyright Notice

TABLE OF CONTENTS

1 Overview

The App Monitoring REST API complements our existing scripting REST API and Java API and is a simple way to retrieve your app monitoring's results. For scripting related API functions, please refer to the Mobile Testing Enterprise REST API user guide.

The App Experience REST API uses a REST architecture. In simple terms, any API query can be executed within a browser using HTTP POST or GET requests.

2 Monitoring API Methods

2.1 Method: establish-api-session

Establish an API session based on username/ password and returns a session id which could be used to fetch monitoring data. This is a POST method which can send/ receive data in both JSON and XML formats.

Required Parameters

Parameter	Description
email	Email of the user
password	Password for the user account

HTTP POST Example URL

```
http://{access-server-hostname}:6232/resource/portal/establish-api-session
```

HTTP POST Example Body

```
{"email": "john.doe@test-domain.com", "password": "test-password"}
```

Results

Parameter	Type	Description
status	String	SUCCESS/ FAILURE
failureReason	String	Reason of the failure
sessionID	String	ID of the current API session
customerId	Integer	Customer ID
muserId	Integer	User ID

HTTP POST Example Body

{"status": "SUCCESS", "failureReason": "", "sessionID": "c9d045f3e21c46d0971db92ebc2ab83c==", "customerId", 999, "muserId": 9999}

2.2 Method: logout-api-session

Close an API session using the sessionID. If the session is not logged out it will automatically timeout in 30 minutes and the account and resources may not be reused until then. This is a POST method which can send/ receive data in both JSON and XML formats.

Required Parameters

Parameter	Description
sessionID	ID of the current API session

HTTP POST Example URL

```
http://{access-server-hostname}:6232/resource/portal/logout-api-session
```

HTTP POST Example Body

```
{"sessionID": "c9d045f3e21c46d0971db92ebc2ab83c=="}
```

Results

Parameter	Type	Description
status	String	SUCCESS/ FAILURE
failureReason	String	Reason of the failure

HTTP POST Example Body

```
{"status": "SUCCESS", "failureReason": ""}
```

2.3 Method: get-projects

Returns information about all projects in your account.

Required Parameters

Parameter	Description
sessionID	Session ID of the user – path parameter represented by {sessionID}

HTTP GET Example URL

```
http://{access-server-hostname}:6232/resource/monitor/{sessionID}/get-projects
```

Results

Parameter	Type	Description
id	Integer	Project ID
name	String	Project name
lastModified	DateTime	Date/time when project was last updated
createdAt	DateTime	Date/time when project was created

2.4 Method: get-monitor-results

Returns all monitor results at a high level (pass/fail status, reason for failure) for each run.

Required Parameters

Parameter	Description
sessionID	Session ID of the user – path parameter represented by {sessionID}
projectId	Project ID
startDate	Start date and time for data collection in the format YYYY-MM-DD HH:MM:SS, e.g., 2016-03-12 23:59:59
endDate	End date and time for data collection in the format YYYY-MM-DD HH:MM:SS, e.g., 2016-03-12 23:59:59
monitorName	Monitor name
location	Location of LiveMonitor server
isTransactionFail	true/false flag to denote if ANY transaction in the monitor run failed

HTTP GET Example URL:

```
http://{access-server-hostname}:6232/resource/monitor/{sessionID}/get-monitor-results?startDate=2016-02-10 00:00:00&endDate=2016-02-12 23:59:59&projectId=36&monitorName=test1
```

Results

Parameter	Type	Description
id	Integer	Unique ID
customerId	Integer	Customer ID
monitorName	String	Monitor name
startTime	DateTime	Start date and time of monitor execution
device	String	Device name

location	String	Location of the monitor (location of the LiveMonitor server)
carrier	String	Device carrier
totalRunTime	Long	Total time the monitor took to complete this particular run
resultData	Boolean	Always set to "true"
countSuccess	Integer	Index of success 1 if success
countFailure	Integer	Index of failure 1 if failure
countSLA	Integer	If a transaction in the run failed, 1, else 0
countException	Integer	If the run failed, index of that failure, i.e., 1st failure, 2nd failure, etc.
deviceIds	String	ID of device (not MCD)
deviceNames	String	Device name
deviceCarriers	String	Device carrier
deviceLocations	String	Device location
descriptionOfResult	String	Result description (what you have provided in the script result description field)
scriptReturnCode	String	Indicates if the monitor was Success or Fail
validResult	Boolean	true/false to flag if result is invalid
excluded	Boolean	true/false to remove and exclude the result
datasetValue	String	Not currently implemented. (SIGOS APPEx Monitoring 4.2 had a feature in which Monitor could be run with a data set.)
testScheduleId	Integer	ID of the monitor script
transactionFailed	Boolean	true/false —true if any transaction in the script failed
deviceMCDs	String	MCD numbers of the devices
deviceSlots	String	Slot name of the devices (Primary, Secondary, etc.)
projectId	Integer	Project ID
errorCategory	String	Error Category of the failure (if defined)
errorType	String	Error type of the failure

runMode	Integer	1 = Development Run 2 = Suppress All Alerts
monitorId	Integer	ID of the monitor (Not used any more. In SIGOS APPEx Monitoring 6.x, we introduced test_schedule_wid.)

2.5 Method: get-monitor-result-incidents

Retrieves incident reports based on policies set in case of timer failure.

Required Parameters

Parameter	Description
sessionID	Session ID of the user – path parameter represented by {sessionID}
projectId	Project ID
startDate	Start date and time for data collection in the format YYYY-MM-DD HH:MM:SS, e.g., 2016-03-12 23:59:59
endDate	End date and time for data collection in the format YYYY-MM-DD HH:MM:SS, e.g., 2016-03-12 23:59:59

HTTP GET Example URL

```
http://{access-server-hostname}:6232/resource/monitor/{sessionID}/get-monitor-result-incidents?startDate=2016-02-10 00:00:00&endDate=2016-02-12 23:59:59&projectId=36
```

Results

Parameter	Type	Description
id	Integer	Unique ID
customerId	Integer	Customer ID
projectId	Integer	Project ID
testScheduleId	Integer	Monitor ID
monitorName	String	Monitor name
policyName	String	Incident policy name
policyType	String	Policy type as defined in Studio
policyDescription	String	Policy description
startMonitorResultId	Integer	ID of the monitor when the policy was started

startTime	DateTime	Date/time of the monitor when the policy was started
endMonitorResultId	Integer	ID of the monitor when the policy was ended
endTime	DateTime	Date/time of the monitor when the policy was ended

2.6 Method: get-monitor-result-indicators

Data of the timers within the monitor

Required Parameters

Parameter	Description
sessionID	Session ID of the user – path parameter represented by {sessionID}
projectId	Project ID
startDate	Start date and time for data collection in the format YYYY-MM-DD HH:MM:SS, e.g., 2016-03-12 23:59:59
endDate	End date and time for data collection in the format YYYY-MM-DD HH:MM:SS, e.g., 2016-03-12 23:59:59
monitorName	Monitor name
monitorResultId	ID of the monitor result that the transaction belongs to
indicatorName	Transaction name

HTTP Get Example URL

```
http://{access-server-hostname}:6232/resource/monitor/{sessionID}/get-monitor-result-indicators?projectId=1&startDate=2016-02-10 00:00:00&endDate=2016-02-12 23:59:59&monitorName=test1
```

Results

Parameter	Type	Description
id	Integer	Unique ID
customerIid		Customer ID
monitorResultId	Integer	ID of the monitor result to which this transaction result belongs.
startTime	DateTime	Start date and time of transaction result
indicatorType	String	Type of timer: Average, One type, Summation
range	Decimal	Timer range; always 0

units	String	Unit of the timer (Milliseconds)
value	Decimal	Value of the timer (time taken to complete the transaction in the units above)
validResult	Boolean	true/false to flag result if invalid
thresholdValue	Integer	Threshold defined for the timer
thresholdConditionId	Integer	1 = Fail if timer < threshold value, 2 = if timer is > threshold value, 3 -> ==, 4 <=, 5 -> >=
thresholdBreached	Boolean	true/false to indicate if the transaction threshold value was breached
transactionFailed	Boolean	true/false to indicate if transaction failed
monitorName	String	Monitor name
projectId	Integer	Project ID
indicatorGroup	String	Name of transaction group (if any)
indicatorName	String	Transaction name
errorCode	String	Error code defined for the transaction failure
errorDetails	String	Details of transaction completion

3 Known Issues and Limitations

N/A.