Administrative Web Services API

ApplinX Administrative Web Services API provides the developer the capability to retrieve data and perform actions based on information received from ApplinX server in runtime, without accessing the ApplinX Designer or Administrator but rather using standard Web services.

Using these Web services you can:

- Retrieving Server Information
- Retrieving Session Information
- Managing Connection Pools
- Managing Connection Pools Connection Information Sets
- Managing RPC Connection Pools

Note:
The user name required in this method refers to a user who has ApplinX server Administrator permissions.

Retrieving Server Information

This Web service retrieves data regarding server information.

URL: http://localhost:2380/wsstack/services/ServerManager?wsdl

Note:
This address is relevant when working with the WS-Stack in local mode. When working in remote mode, you need to update the server address to reflect the WS-Stack server address.

Method: getServerInformation

Returns a `ServerInformationResponse` object which contains the ServerInfomation object.

Format: `ServerInformationResponse getServerInformation(ServerInformationRequest request)`

ServerInformationRequest parameters

String username

The name of the user who has the relevant permissions to access ApplinX server.

String password

The user’s password

ServerInformationResponse parameters:

String version

The ApplinX server's version.

String startedTime

The date and time the server was last initialized.

int activeSessions

The number of currently active sessions.

int idleSessions

The number of sessions currently idle (connected, but detached).

int processingSessions

The number of sessions currently performing actions against the host.

int totalSessions

The total number of sessions opened since the server was started.

int startedThreads

The number of threads that were activated.

int idleThreads

The number of threads that were activated, but are currently not being used.

long allocatedMemory

The total amount of memory (in bytes) that is currently allocated to ApplinX server on the Java Virtual Machine.

long freeMemory

The amount of memory (in bytes) currently available for ApplinX to use.

Method: getSessionsCounters

Returns a `sessionsCountersResponse` object which contains the ConnectionsPerLicense object.

Format: `sessionsCountersResponse getSessionsCounters(sessionsCountersRequest request)`

sessionsCountersRequest parameters:

String username

The name of the user who has the relevant permissions to access ApplinX server.

String password

The user's password

sessionsCountersReponse parameters:

Each of the parameters includes the number of connections currently being used, the maximum number of connections ever connected at one time and the date and time that this occurred. Notice that each of the parameters relates to a specific type of connection.

numberOfWebEnablementDisplay

Relates to Web enablement connections (excluding printer sessions).

numberOfWebEnablementPrinter

Relates to the printer sessions. The printer sessions are counted as Web Enablement connections.

numberOfSOAEnablement

Relates to the SOA enablement connections (excluding Web integration and RPC connections).

numberOfWebIntegration

Relates to Web Integration connections. The Web Integration connections are counted as SOA connections.

numberOfRPCConnections

Relates to RPC connections. The RPC connections are counted as SOA connections.

numberOfUnassignedPoolConnections

Relates to the unassigned connections in the connection pool.

Retrieving Session Information

This Web service retrieves data regarding session information.

URL: http://localhost:2380/wsstack/services/SessionManager?wsdl

Note:
This address is relevant when working with the WS-Stack in local mode. When working in remote mode, you need to update the server address to reflect the WS-Stack server address.

Method: getAllSessions

Returns a object which contains a list of all the sessions that are open on the server.

Format: `GetAllSessionsResponse getAllSessions(GetAllSessionsRequest request) getServerInformation(ServerInformationRequest request).`

GetAllSessionsRequest parameters

String username

The name of the user who has the relevant permissions to access ApplinX server.

String password

The user's password

GetAllSessionsResponse Parameters

Session [ ] sessions

An array of Session objects which contain the session information.

Method: getApplicationSessions

Returns a `GetApplicationSessionsResponse` object which contains a list of sessions that are connected to a specific application.

Format: `GetApplicationSessionsResponse getApplicationSessions(GetApplicationSessionsRequest request)`

GetApplicationSessionsRequest parameters

String username

The name of the user who has the relevant permissions to access ApplinX server.

String password

The user's password

String appName

The name of the application on ApplinX server

Session Information Response Parameters

Session [ ] sessions

An array of Session objects which contain the session information.

Method: getServiceSessions

Returns a `GetServiceSessionsResponse` object which contains a list of sessions that are connected to a specific Connection pool.

Format: `GetServiceSessionsResponse getServiceSessions(GetServiceSessionsRequest request)`

GetServiceSessionsRequest parameters

String username

The name of the user who has the relevant permissions to access ApplinX server.

String password

The user's password

String appName

The name of the application on ApplinX server

String serviceName

The name of the Connection Pool

GetServiceSessionsResponse Parameters

Session [ ] sessions

An array of Session objects which contain the session information.

Method: cancelSession

Returns a `CancelSessionResponse` object which contains a boolean that indicates that the session has been canceled.

Format: `CancelSessionResponse cancelSession(CancelSessionRequest request)`

CancelSessionRequest parameters

String username

The name of the user who has the relevant permissions to access ApplinX server.

String password

The user's password

String sessionId

The ID of the session that you would like to cancel.

CancelSessionResponse Parameters

boolean isSuccessful

Returns true or false, to indicate the success or failure of the action.

Session Object Attributes

String id

The session's ID on the ApplinX server.

String description

The session's description. For example, this may be the session's computer address.

String type

The type of session: Display session, Printer session, RPC session, Web Integration session or Development session.

String application

The name of the application on ApplinX server to which this session is connected.

String deviceName

Workstation ID, available only in certain protocols.

String state

The current communication status between ApplinX server and the host. Can be either: Idle (connected, not attached), Initializing, Processing (executing an action), Active (attached), or Disconnecting.

String idleTime

The time period a session has not performed a communication activity with the host.

String currentScreen

The name of the current screen. When the screen is not identified this will be UNKNOWN.

String userAddress

The IP address from where the session is connected to ApplinX.

String hostAddress

The IP address of the host.

String replayFile

The GCT file name that is working with this session and the screen number in the GCT file.

String traceFile

The name of the trace file that is tracing the current session.

int bytesSent

The number of bytes sent to the host.

int bytesReceived

The number of bytes received from the host

String serviceName

The name of the connection pool used by the current session. Will return an empty value when no connection pool is used by the session.

Managing Connection Pools

This Web service retrieves runtime data regarding existing connection pools and enables starting and stopping these services.

URL: http://localhost:2380/wsstack/services/ServiceManager?wsdl

Note:
This address is relevant when working with the WS-Stack in local mode. When working in remote mode, you need to update the server address to reflect the WS-Stack server address.

Method: getApplicationServices

Returns a `GetApplicationServicesResponse` object which contains a list of connection pools and their status, for a specific application.

Format:
```
GetApplicationServicesResponse
getApplicationServices(GetApplicationServicesRequest request)
```

GetApplicationServicesRequest parameters

String username

The name of the user who has the relevant permissions to access ApplinX server.

String password

The user’s password

String appName

The name of the application on ApplinX server

GetApplicationServicesResponse Parameters

Service [ ] services

An array of Service objects which contain the session information.

Method: getFolderServices

Returns a `GetFolderServicesResponse` object which contains a list of names and status of connection pools that are in a specific folder of an application.

Format: `GetFolderServicesResponse getFolderServices(GetFolderServicesRequest request)`

GetFolderServicesRequest parameters

String username

The name of the user who has the relevant permissions to access ApplinX server.

String password

The user's password

String appName

The name of the application on ApplinX server

String folder

The name of the folder where the connection pools are located.

GetFolderServicesResponse Parameters

Service [ ] services

An array of Service objects which contain the session information.

Service Object Attributes

Service attributes:

String name

The name of the connection pool.

String folder

The folder in which the connection pool is placed.

String status

The connection pool's status: Not Started, Initializing, Active, In standby, Suspended, Stopping or Stopped.

int activeConnections

The number of connections currently held by a session (user).

int readyConnections

The number of connections ready for use.

int processingConnections

The number of connections currently in the Processing state.

long averageWaitTime

The average time (in milliseconds) sessions waited for a READY connection (calculated only among those sessions that waited) multiplied by the percentage of waiting sessions. For example: if 8% sessions had to wait, and in average each of those waited 1000 milliseconds, the overall average wait time was: 0.08 * 1000 = 80 milliseconds.

int percentOfWaiting

The percent of sessions that did not immediately get a connection when trying to connect to ApplinX.

int connectionCount

The total number of connections in the connection pool (since the last time the connection pool was started), ignoring broken connections.

int maxConnections

The maximum number of connections that were connected concurrently since the connection pool started.

int sessionCount

The total number of sessions that connected to the host service since the connection pool started.

int maxConcurrentSessions

The maximum number of sessions that were connected concurrently since the connection pool started.

String connectionInfoName

The name of the information set used by this connection pool.

int numberOfCurrentlyWaiting

The number of users currently waiting for a connection.

int numberOfTimeouts

the number of users who received a timeout after a connection was not assigned to them.

int numberOfWaitedUsers

the total number of users who waited for a connection since the connection pool was last started.

long maxWaitTime

Maximum time, since the session started, that a user waited for a connection.

CancelConnectionResponse Parameters

boolean isSuccessful

Returns true or false, to indicate the success or failure of the action.

Method: cancelConnection

Returns a `CancelConnectionResponse` object which contains a boolean that indicates that the connection has been canceled.

Format: `CancelConnectionResponse cancelConnection(CancelConnectionRequest request)`

CancelConnectionRequest parameters

username

The name of the user who has the relevant permissions to access ApplinX server.

password

The user’s password

appName

The name of the application on ApplinX server

serviceName

The name of the connection pool

folder

The name of the folder where the connection pools are located.

connectionId

The ID of the connection which you would like to cancel.

CancelConnectionResponse Parameters

boolean isSuccessful

Returns true or false, to indicate the success or failure of the action.

Method: getServiceConnections

Returns a `GetServiceConnectionsRequest` object which contains a list of connections for a specific Connection Pool.

Format: `GetServiceConnectionsRequest getServiceConnections(GetServiceConnectionsRequest request)`

GetServiceConnectionsRequest parameters

String username

The name of the user who has the relevant permissions to access ApplinX server.

String password

The user's password

String appName

The name of the application on ApplinX server

String serviceName

The name of the connection pool

GetServiceConnectionsResponse parameters

ServiceConnection [ ] connections

An array of ServiceConnection objects which contain the session information.

ServiceConnection object attributes:

String ConnectionId

The identifier of the connection.

String status

The current status of the connection. Possible values: Active, Ready, Initializing, Recycling, Keep-alive, Terminating or Broken.

String sessionId

Active connections only—displays the ID of the session holding the displayed connection.

String elapsedTime

The amount of time that has elapsed since the session status last changed.

String errorMessage

Displays error messages for broken connections.

String createdTime

The time the connection was created.

String statusChangedTime

The time the status was last changed.

int sessionServed

The number of sessions this specific connection served so far.

String connectionInformation

The connection information set row number used by the connection.

String currentScreen

The name of the current screen.

Method: startService

Starts the connection pool specified in the request and returns a response object with a boolean indicating success or failure.

Format: `StartServiceResponse startService(StartServiceRequest request)`

StartServiceRequest parameters

String username

The name of the user who has the relevant permissions to access ApplinX server.

String password

The user’s password

String appName

The name of the application on ApplinX server

String serviceName

The name of the connection pool

String folderName

The name of the folder where the connection pool is located. By default, this is the root folder.

StartServiceResponse Parameters

boolean isSuccessful

Returns true or false, to indicate the success or failure of the action.

Method: stopService

Stops the connection pool specified in the request and returns a `StopServiceResponse` object with a boolean indicating success or failure.

Format: `StopServiceResponse stopService(StopServiceRequest request)`

StopServiceRequest parameters

String username

The name of the user who has the relevant permissions to access ApplinX server.

String password

The user's password

String appName

The name of the application on ApplinX server

String serviceName

The name of the connection pool

String folderName

The name of the folder where the connection pool is located. By default, this is the root folder.

StopServiceResponse Parameters

boolean isSuccessful

Returns true or false, to indicate the success or failure of the action.

Method: stopAllServices

Stops all the connection pools of the application and returns a `StopAllServicesResponse` object with a boolean indicating success or failure.

Format: `StopAllServicesResponse stopAllServices(StopAllServicesRequest request)`

StopAllServicesRequest parameters

String username

The name of the user who has the relevant permissions to access ApplinX server.

String password

The user's password

String appName

The name of the application on ApplinX server

StopAllServicesResponse Parameters

boolean isSuccessful

Returns true or false, to indicate the success or failure of the action.

Method: suspendService

Suspends the available connection pool of the application and returns a `SuspendServiceResponse` object with a boolean indicating success or failure.

Format: `SuspendServiceResponse suspendService(SuspendServiceRequest request)`

SuspendServiceRequest parameters

String username

The name of the user who has the relevant permissions to access ApplinX server.

String password

The user's password

String appName

The name of the application on ApplinX server

String serviceName

The name of the connection pool

String folderName

The name of the folder where the connection pool is located. By default, this is the root folder.

SuspendServiceResponse Parameters

boolean isSuccessful

Returns true or false, to indicate the success or failure of the action.

Method: StandbyService

Suspends the available connection pool of the application and returns a `standbyServiceResponse` object with a boolean indicating success or failure.

Format: `StandbyServiceResponse standbyService(standbyServiceRequest request)`

standbyServiceRequest parameters

String username

The name of the user who has the relevant permissions to access ApplinX server.

String password

The user's password

String appName

The name of the application on ApplinX server

String serviceName

The name of the connection pool

String folderName

The name of the folder where the connection pool is located. By default, this is the root folder.

StandbyServiceResponse Parameters

boolean isSuccessful

Returns true or false, to indicate the success or failure of the action.

Method: resumeService

Resumes the possibility to use a suspended connection pool of the application and returns a `ResumeServiceResponse` object with a boolean indicating success or failure.

Format: `ResumeServiceResponse resumeService(ResumeServiceRequest request)`

ResumeServiceRequest parameters

String username

The name of the user who has the relevant permissions to access ApplinX server.

String password

The user's password

String appName

The name of the application on ApplinX server

String serviceName

The name of the connection pool

String folderName

The name of the folder where the connection pool is located. By default, this is the root folder.

ResumeServiceResponse Parameters

boolean isSuccessful

Returns true or false, to indicate the success or failure of the action.

Managing Connection Pools Connection Information Sets

This Web service updates the connection information set cell values.

URL: http://localhost:2380/wsstack/services/ConnectionInfoManager?wsdl

Note:
This address is relevant when working with the WS-Stack in local mode. When working in remote mode, you need to update the server address to reflect the WS-Stack server address.

Method: getConnectionInfoSet

Returns a `GetConnectionInfoSetResponse` object which contains a `ConnInfoSet` object.

Format: `GetConnectionInfoSetResponse getConnectionInfoSet (GetConnectionInfoSetRequest request)`

GetConnectionInfoSetRequest parameters

String username

The name of the user who has the relevant permissions to access ApplinX server.

String password

The user’s password

String appName

The name of the application on ApplinX server

String connectionInfoName

The name of the connection information set

GetConnectionInfoSetResponse parameters

ConnInfoSet connectionInfoSet

Objects which contain the connection information set parameters

ConnInfoSet parameters

ConnInfoColumn[ ] columns

Array of the connection information columns, not including the ID and Repeat columns.

ConnInfoRow[ ] rows

Array of the connection information set rows.

ConnInfoColumn parameters

Boolean hidden

Indicates whether the column is defined as a password column.

int type

Indicates the column type: Variables (0), Application fields (1) and Application and connection parameters (2).

String name

Column name.

ConnInfoRow parameters

ConnInfoCell[] cells

Array of the cells in a specific row in the connection information set not including the ID and Repeat cells.

int Repeat

The value of the repeat cell.

int id

The value of the ID cell.

ConnInfoCell parameters

String value

The cells's value.

String columnName

The cell's column name.

int columnType

Indicates the cell's column type: Variables (0), Application fields (1) and Application and connection parameters (2).

Note:
The user name required in this method refers to a user who has ApplinX server Administrator permissions.

Method: updateConnectionInfoRow

Returns a `UpdateConnectionInfoRowResponse` object which contains a boolean indicating whether the update operation succeeded.

Format: `UpdateConnectionInfoRowResponse updateConnectionInfoRow(UpdateConnectionInfoRowRequest request)`

UpdateConnectionInfoRowRequest parameters

String username

The name of the user who has the relevant permissions to access ApplinX server.

String password

The user's password

String appName

The name of the application on ApplinX server

String connectionInfoName

The name of the connection information set

ConnInfoRow row

The row object that contains the information that we wish to update in the connection information set. The following parameters must be set:

- The ID of the row as it appears in the server.
- The repeat number: when either updating the repeat number or preserving the number currently set on the server (when this number is not zero).
- The value of each cell in the ConnInfoCell[]. Caution: Null values will replace existing values on the server.

UpdateConnectionInfoRowResponse parameters

boolean isSuccessful

Returns true or false, to indicate the success or failure of the action.

Method: addConnectionInfoRow

Format: `AddConnectionInfoRowResponse addConnectionInfoRow(AddConnectionInfoRowRequest request)`

AddConnectionInfoRowRequest parameters

String username

The name of the user who has the relevant permissions to access ApplinX server.

String password

The user's password

String appName

The name of the application on ApplinX server

String connectionInfoName

The name of the connection information set

ConnInfoRow row

The row object that contains the information that we wish to add to the connection information set. See the following guidelines for setting the object parameters:

- The ID of the row as it appears in the server.
- The repeat number: when either updating the repeat number or preserving the number currently set on the server (when this number is not zero).
- The value of each cell in the ConnInfoCell[]. Caution: Null values will replace existing values on the server.

AddConnectionInfoRowResponse parameters

boolean isSuccessful

Returns true or false, to indicate the success or failure of the action.

Method: removeConnectionInfoRow

Format: `RemoveConnectionInfoRowResponse removeConnectionInfoRow(RemoveConnectionInfoRowRequest request)`

RemoveConnectionInfoRowRequest parameters

String username

The name of the user who has the relevant permissions to access ApplinX server.

String password

The user's password

String application

The name of the application on ApplinX server

String connectionInfoName

The name of the connection information set

int rowed

The row ID

RemoveConnectionInfoRowResponse parameters

boolean isSuccessful

Returns true or false, to indicate the success or failure of the action.

Managing RPC Connection Pools

This Web service retrieves runtime data regarding existing RPC connection pools and enables resetting the pool.

URL:http://localhost:2380/wsstack/services/ProgramPoolManager?wsdl

Note:
This address is relevant when working with the WS-Stack in local mode. When working in remote mode, you need to update the server address to reflect the WS-Stack server address.

Method: getPoolInformation

Returns a `PoolInformationResponse` object which contains runtime information regarding the pool manager. `resetPool(ProgramPoolRequest request)` resets the pool.

Format: `PoolInformationResponse getPoolInformation(ProgramPoolRequest request)`

ProgramPoolRequest parameters

String username

The name of the user who has the relevant permissions to access ApplinX server.

String password

The user's password

String appName

The name of the application on ApplinX server

PoolInformationResponse parameters

int availableConnectionsCount

The number of available connections in the pool.

int maxPoolSize

The maximum number of connections that can exist in the pool concurrently.

int excludedCount

The number of resources that were invalidated but are still in use

int minPoolSize

The minimum number of connections that must exist in a pool.

int poolSize

The number of connections that currently exist in the pool.

int awaitingCheckinCount

The number of used resources (including the excluded resources).

Method: resetPool

This method restarts the pool.

Format: `resetPool( )`