

# 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.

These Web services can be used for the following:

- 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 `ServerInformation` object.

Format: `ServerInformationResponse`

```
getServerInformation(ServerInformationRequest request)
```

Request/Response	Parameter	Format	Description
ServerInformationRequest	username	String	The name of the user who has the relevant permissions to access ApplinX server
	password	String	The user's password.
ServerInformationResponse	version	String	The ApplinX server's version.
	startedTime	String	The date and time the server was last initialized.
	activeSessions	int	The number of currently active sessions.
	idleSessions	int	The number of sessions currently idle (connected, but detached).
	processingSessions	int	The number of sessions currently performing actions against the host.
	totalSessions	int	The total number of sessions opened since the server was started.
	startedThreads	int	The number of threads that were activated.
	idleThreads	int	The number of threads that were activated, but are currently not being used.
	allocatedMemory	long	The total amount of memory (in bytes) that is currently allocated to ApplinX server on the Java Virtual Machine.
freeMemory	long	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)`

Request/Response	Parameter	Format	Description
sessionsCountersRequest	username	String	The name of the user who has the relevant permissions to access ApplinX server.
	password	String	The user's password.

Request/Response	Parameter	Format	Description
sessionsCountersResponse	numberOfWebEnablementDisplay	ConnectionsCounter	Relates to Web enablement connections (excluding printer sessions).
	numberOfWebEnablementPrinter	ConnectionsCounter	Relates to the printer sessions. The printer sessions are counted as Web Enablement connections.
	numberOfSOAEnablement	ConnectionsCounter	Relates to the SOA enablement connections (excluding Web integration and RPC connections).
	numberOfWebIntegration	ConnectionsCounter	Relates to Web Integration connections. The Web Integration connections are counted as SOA connections.
	numberOfRPCConnections	ConnectionsCounter	Relates to RPC connections. The RPC connections are counted as SOA connections.
	numberOfUnassignedPoolConnections	ConnectionsCounter	Relates to the unassigned connections in the connection pool.

**Note:**

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. See table below. Note that each of the parameters relates to a specific type of connection.

**ConnectionsCounter Object Attributes**

Attribute	Format	Description
current	int	The number of connections of the same type currently being used.
Max	int	The maximum number of connections of the same type ever connected at one time.
dateOfMax()	java.util.Calendar	Time that the maximum number of connections of the same type was measured.

**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)`

Request/Response	Parameter	Format	Description
GetAllSessionsRequest	username	String	The name of the user who has the relevant permissions to access ApplinX server
	password	String	The user's password.
GetAllSessionsResponse	sessions	Session []	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)`

Request/Response	Parameter	Format	Description
GetApplicationSessionsRequest	username	String	The name of the user who has the relevant permissions to access ApplinX server.
	password	String	The user's password.
	appName	String	The name of the application on ApplinX server.
GetApplicationSessionsResponse	sessions	Session []	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)`

Request/Response	Parameter	Format	Description
GetApplicationSessionsRequest	username	String	The name of the user who has the relevant permissions to access ApplinX server
	password	String	The user's password.
	appName	String	The name of the application on ApplinX server.
	serviceName	String	The name of the Connection Pool.
GetServiceSessionsResponse	sessions	Session []	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)`

Request/Response	Parameter	Format	Description
CancelSessionRequest	username	String	The name of the user who has the relevant permissions to access ApplinX server
	password	String	The user's password.
	sessionId	String	The ID of the session that you would like to cancel.
CancelSessionResponse	isSuccessful	boolean	Returns true or false, to indicate the success or failure of the action.

## Session Object Attributes

Attribute	Format	Description
id	String	The session's ID on the ApplinX server.
description	String	The session's description. For example, this may be the session's computer address.
type	String	The type of session: Display session, Printer session, RPC session, Web Integration session or Development session.
application	String	The name of the application on ApplinX server to which this session is connected.
deviceName	String	Workstation ID, available only in certain protocols.
state	String	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.
idleTime	String	The time period a session has not performed a communication activity with the host.
currentScreen	String	The name of the current screen. When the screen is not identified this will be UNKNOWN.
userAddress	String	The IP address from where the session is connected to ApplinX.
hostAddress	String	The IP address of the host.
replayFile	String	The GCT file name that is working with this session and the screen number in the GCT file.
traceFile	String	The name of the trace file that is tracing the current session.
bytesSent	int	The number of bytes sent to the host.
bytesReceived	int	The number of bytes received from the host
serviceName	String	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 pools.

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)`

Request/Response	Parameter	Format	Description
<code>GetApplicationServicesRequest</code>	<code>username</code>	String	The name of the user who has the relevant permissions to access ApplinX server
	<code>password</code>	String	The user's password.
	<code>appName</code>	String	The name of the application on ApplinX server.
<code>GetApplicationServicesResponse</code>	<code>services</code>	Service []	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)`

Request/Response	Parameter	Format	Description
<code>GetFolderServicesRequest</code>	<code>username</code>	String	The name of the user who has the relevant permissions to access ApplinX server
	<code>password</code>	String	The user's password.
	<code>appName</code>	String	The name of the application on ApplinX server.
	<code>folder</code>	String	The name of the folder where the connection pools are located.
<code>GetFolderServicesResponse</code>	<code>services</code>	Service []	An array of Service objects which contain the session information.

## Service Object Attributes

Attribute	Format	Description
name	String	The name of the connection pool.
folder	String	The folder in which the connection pool is placed.
status	String	The connection pool's status: Not Started, Initializing, Active, In standby, Suspended, Stopping or Stopped.
activeConnections	int	The number of connections currently held by a session (user).
readyConnections	int	The number of connections ready for use.
processingConnections	int	The number of connections currently in the Processing state.
averageWaitTime	long	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.
percentOfWaiting	int	The percent of sessions that did not immediately get a connection when trying to connect to ApplinX.
connectionCount	int	The total number of connections in the connection pool (since the last time the connection pool was started), ignoring broken connections.
maxConnections	int	The maximum number of connections that were connected concurrently since the connection pool started.
sessionCount	int	The total number of sessions that connected to the host connection pool since the connection pool started.
maxConcurrentSessions	int	The maximum number of sessions that were connected concurrently since the connection pool started.
connectionInfoName	int	The name of the information set used by this connection pool.
numberOfCurrentlyWaiting	int	The number of users currently waiting for a connection.
numberOfTimeouts	int	the number of users who received a timeout after a connection was not assigned to them.
numberOfWaitedUsers	int	the total number of users who waited for a connection since the connection pool was last started.
maxWaitTime	long	Maximum time, since the session started, that a user waited for a connection.

## Method cancelConnection

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

Format: `CancelConnectionResponse cancelConnection(CancelConnectionRequest request)`

Request/Response	Parameter	Format	Description
CancelConnectionRequest	username	String	The name of the user who has the relevant permissions to access ApplinX server.
	password	String	The user's password.
	appName	String	The name of the application on ApplinX server.
	serviceName	String	The name of the connection pool.
	folder	String	The name of the folder where the connection pools are located.
	connectionId	String	The ID of the connection which you would like to cancel.
CancelConnectionResponse	isSuccessful	boolean	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)`

Request/Response	Parameter	Format	Description
GetServiceConnectionsRequest	username	String	The name of the user who has the relevant permissions to access ApplinX server.
	password	String	The user's password.
	appName	String	The name of the application on ApplinX server.
	serviceName	String	The name of the connection pool.
GetServiceConnectionsResponse	connections	ServiceConnection []	An array of ServiceConnection objects which contain the session information.

### ServiceConnection Object Attributes

Attribute	Format	Description
ConnectionId	String	The identifier of the connection.
Status	String	The current status of the connection. Possible values: Active, Ready, Initializing, Recycling, Keep-alive, Terminating or Broken.
SessionId	String	Active connections only - displays the ID of the session holding the displayed connection.
elapsedTime	String	The amount of time that has elapsed since the session status last changed.
errorMessage	String	Displays error messages for broken connections.
createdTime	String	The time the connection was created.
statusChangedTime	String	The time the status was last changed.
sessionServed	int	The number of sessions this specific connection served so far.
connectionInformation	String	The connection information set row number used by the connection.
currentScreen	String	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)`

Request/Response	Parameter	Format	Description
StartServiceRequest	username	String	The name of the user who has the relevant permissions to access ApplinX server.
	password	String	The user's password.
	appName	String	The name of the application on ApplinX server.
	serviceName	String	The name of the connection pool.
	folderName	String	The name of the folder where the connection pool is located. By default, this is the root folder.
StartServiceResponse	isSuccessful	boolean	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)`

Request/Response	Parameter	Format	Description
StopServiceRequest	username	String	The name of the user who has the relevant permissions to access ApplinX server.
	password	String	The user's password.
	appName	String	The name of the application on ApplinX server.
	serviceName	String	The name of the connection pool.
	folderName	String	The name of the folder where the connection pool is located. By default, this is the root folder.
StopServiceResponse	isSuccessful	boolean	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)

Request/Response	Parameter	Format	Description
StopAllServicesRequest	username	String	The name of the user who has the relevant permissions to access ApplinX server.
	password	String	The user's password.
	appName	String	The name of the application on ApplinX server.
StopAllServicesResponse	isSuccessful	boolean	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)

Request/Response	Parameter	Format	Description
SuspendServiceRequest	username	String	The name of the user who has the relevant permissions to access ApplinX server.
	password	String	The user's password.
	appName	String	The name of the application on ApplinX server.
	serviceName	String	The name of the connection pool.
	folderName	String	The name of the folder where the connection pool is located. By default, this is the root folder.
SuspendServiceResponse	isSuccessful	boolean	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)`

Request/Response	Parameter	Format	Description
standbyServiceRequest	username	String	The name of the user who has the relevant permissions to access ApplinX server.
	password	String	The user's password.
	appName	String	The name of the application on ApplinX server.
	serviceName	String	The name of the connection pool.
	folderName	String	The name of the folder where the connection pool is located. By default, this is the root folder.
StandbyServiceResponse	isSuccessful	boolean	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)`

Request/Response	Parameter	Format	Description
ResumeServiceRequest	username	String	The name of the user who has the relevant permissions to access ApplinX server.
	password	String	The user's password.
	appName	String	The name of the application on ApplinX server.
	serviceName	String	The name of the connection pool.
	folderName	String	The name of the folder where the connection pool is located. By default, this is the root folder.
ResumeServiceResponse	isSuccessful	boolean	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)`

Request/Response	Parameter	Format	Description
GetConnectionInfoSetRequest	username	String	The name of the user who has the relevant permissions to access ApplinX server.
	password	String	The user's password.
	appName	String	The name of the application on ApplinX server.
	connectionInfoName	String	The name of the connection information set.
GetConnectionInfoSetResponse	connectionInfoSet	ConnInfoSet	Objects which contain the connection information set parameters.

**Note:**

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

## ConnInfoSet Object Attributes

Attribute	Format	Description
columns	ConnInfoColumn[ ]	Array of the connection information columns, not including the ID and Repeat columns.
rows	ConnInfoRow[ ]	Array of the connection information set rows.

## ConnInfoColumns Object Attributes

Attribute	Format	Description
hidden	boolean	Indicates whether the column is defined as a password column.
type	int	Indicates the column type: Variables (0), Application fields (1) and Application and connection parameters (2).
name	String	Column name.

## ConnInfoRow Object Attributes

Attribute	Format	Description
cells	ConnInfoCell[ ]	Array of the cells in a specific row in the connection information set not including the ID and Repeat cells.
Repeat	int	The value of the repeat cell.
id	int	The value of the ID cell.

## ConnInfoCell Object Attributes

Attribute	Format	Description
value	String	The cells's value.
columnName	String	The cell's column name.
columnType	int	Indicates the cell's column type: Variables (0), Application fields (1) and Application and connection parameters (2).

## Method updateConnectionInfoRow

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

Format: UpdateConnectionInfoRowResponse

updateConnectionInfoRow(UpdateConnectionInfoRowRequest request)

Request/Response	Parameter	Format	Description
UpdateConnectionInfoRowRequest	username	String	The name of the user who has the relevant permissions to access ApplinX server.
	password	String	The user's password.
	appName	String	The name of the application on ApplinX server.
	connectionInfoName	String	The name of the connection information set.
	row	ConnectionInfoRow	<p>The row object that contains the information that we wish to update in the connection information set. The following parameters must be set:</p> <ul style="list-style-type: none"> <li>● The ID of the row as it appears in the server.</li> <li>● The repeat number: when either updating the repeat number or preserving the number currently set on the server (when this number is not zero).</li> <li>● The value of each cell in the ConnInfoCell[ ]. Caution: Null values will replace existing values on the server.</li> </ul>
UpdateConnectionInfoRowResponse	isSuccessful	boolean	Returns true or false, to indicate the success or failure of the action.

### Method addConnectionInfoRow

Returns a AddConnectionInfoRowResponse object which contains a boolean indicating whether the add operation succeeded.

Format: AddConnectionInfoRowResponse

addConnectionInfoRow(AddConnectionInfoRowRequest request)

Request/Response	Parameter	Format	Description
AddConnectionInfoRowRequest	username	String	The name of the user who has the relevant permissions to access ApplinX server.
	password	String	The user's password.
	appName	String	The name of the application on ApplinX server.
	connectionInfoName	String	The name of the connection information set.
	row	ConnectionInfoRow	<p>The row object that contains the information that we wish to update in the connection information set. The following parameters must be set:</p> <ul style="list-style-type: none"> <li>● The ID of the row as it appears in the server.</li> <li>● The repeat number: when either updating the repeat number or preserving the number currently set on the server (when this number is not zero).</li> <li>● The value of each cell in the ConnInfoCell[ ]. Caution: Null values will replace existing values on the server.</li> </ul>
AddConnectionInfoRowResponse	isSuccessful	boolean	Returns true or false, to indicate the success or failure of the action.

### Method `removeConnectionInfoRow`

Returns a `RemoveConnectionInfoRowResponse` object which contains a boolean indicating whether the remove operation succeeded.

Format: `RemoveConnectionInfoRowResponse`  
`removeConnectionInfoRow(RemoveConnectionInfoRowRequest request)`

Request/Response	Parameter	Format	Description
RemoveConnectionInfoRowRequest	username	String	The name of the user who has the relevant permissions to access ApplinX server.
	password	String	The user's password.
	appName	String	The name of the application on ApplinX server.
	connectionInfoName	String	The name of the connection information set.
	rowId	int	The row ID.
RemoveConnectionInfoRowResponse	isSuccessful	boolean	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)`

Request/Response	Parameter	Format	Description
ProgramPoolRequest	username	String	The name of the user who has the relevant permissions to access ApplinX server.
	password	String	The user's password.
	appName	String	The name of the application on ApplinX server.
PoolInformationResponse	availableConnectionsCount	int	The number of available connections in the pool.
	maxPoolSize	int	The maximum number of connections that can exist in the pool concurrently.
	excludedCount	int	The number of resources that were invalidated but are still in use.
	minPoolSize	int	The minimum number of connections that must exist in a pool.
	poolSize	int	The number of connections that currently exist in the pool.
	awaitingCheckinCount	int	The number of used resources (including the excluded resources).

## Method resetPool

This method resets the pool.

Format: `resetPool()`