Alexander Küster

SessionService

Description, Command syntax overview for the communication with the session service and additional information

Table of contents

[Action requests 2](#_Toc58335852)

[Possible actions 2](#_Toc58335853)

[Action unusual responses 2](#_Toc58335854)

[Action detailed descriptions 3](#_Toc58335855)

[Broadcast 3](#_Toc58335856)

[Required parameters 3](#_Toc58335857)

[Broadcast.Available **Fehler! Textmarke nicht definiert.**](#_Toc58335858)

[Closing **Fehler! Textmarke nicht definiert.**](#_Toc58335859)

[Get.DatabaseURL 3](#_Toc58335860)

[New 3](#_Toc58335861)

[Required parameters 4](#_Toc58335862)

[Optional parameters 4](#_Toc58335863)

[New.Session 4](#_Toc58335864)

[Required parameters 4](#_Toc58335865)

[Optional parameters 4](#_Toc58335866)

[Session.Close **Fehler! Textmarke nicht definiert.**](#_Toc58335867)

[Session.GetExists 4](#_Toc58335868)

[Required parameters 4](#_Toc58335869)

[Session.GetServices 5](#_Toc58335870)

[Optional parameters 5](#_Toc58335871)

# Action requests

An action will be executed when the performAction() or queueAction() function is called by the microservice framework. In the session service actions will always be executed immidiatly so the queueAction call has the same effect as the performAction.

To reqeuest an action a JSON string must be provided containing the keyword “action” and the action identifier.

# Possible actions

|  |  |
| --- | --- |
| Action | Description |
| [Broadcast](#_Broadcast) | Will send a broadcast message to all services in the senders’ session |
| [Broadcast.Available](#_Broadcast.Available) | Will send a broadcast message containing:  “Service.Connected“  as message to all services in the senders’ session |
| [Closing](#_Closing) | Will deregister the sender from the session and notify others |
| [Debug.Off](#_Debug.Off) | Will turn the debug mode off |
| [Debug.On](#_Debug.On) | Will turn the debug mode on |
| [Get.DatabaseURL](#_Get.DatabaseURL) | Will return the global database URL as reponse message |
| [Session.Close](#_Session.Close) | Will shutdown the whole session that the sender service is a part of |
| [Session.GetExists](#_Session.GetExists) | Will return “true” if the session with the specified ID exists, otherwise returns “false” |
| [Session.GetServices](#_Get.Services) | Will return a list containing all services in the senders’ session or a specified session |
| [New](#_New) | Will register the sender as a new service in a session and return the sessions new UID |
| [New.Session](#_New.Session) | Will create a new session |
| [Session.Close](#_Session.Close) | Will shutdown the whole session that the sender service is a part of |
|  |  |
|  |  |

# Action unusual responses

|  |  |
| --- | --- |
| Case | Response syntax |
| Error | ERROR: <Error description> | *Between the colon and the error description is a blank space* |
| Warning | WARNING: <Warning description> | *Between the colon and the warning description is a blank space* |

# Action detailed descriptions

## Broadcast

Will send a broadcast message to all services in the senders’ session.

The actual string that will be send will look like this:

{“Message”:”<Actual message>”, “Sender.Name”:”<Sender name>”, ”Sender.IP”:”<Sender IP>”}

### Required parameters

|  |  |  |
| --- | --- | --- |
| Parameter | Value type | Description |
| Session.ID | String | The service UID |
| Message | String | The message that will be send |

## Broadcast.Available

Will perform the Broadcast action with the parameter:

Message = “Service.Connected”

### Required parameters

|  |  |  |
| --- | --- | --- |
| Parameter | Value type | Description |
| Session.ID | String | The service UID |

## Closing

Will deregister the sender from the session and notify other services in the same session about it.

### Required parameters

|  |  |  |
| --- | --- | --- |
| Parameter | Value type | Description |
| Session.ID | String | The service UID |

## Debug.Off

Will turn the debug mode off.

## Debug.On

Will turn the debug mode on.

## Get.DatabaseURL

Will return the global database URL as reponse message.

## New

Will register the sender as a new service in the specified session. This step is mandatory to be able to use the [Broadcast](#_Broadcast) commands or other session related actions. The response will contain the new assigned UID (UID datatype = unsigned long long) as a string.

### Required parameters

|  |  |  |
| --- | --- | --- |
| Parameter | Value type | Description |
| Session.ID | String | The ID of the session the service should be registered at |
| Service.Name | String | The name of the service that is registering |
| Service.Type | String | The type of the service |
| Port | String | The services port |

### Optional parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Value type | Default | Description |
| Notify.Connected | Boolean | False | If true is provided a [Broadcast.Available](#_Broadcast.Available) action for this service will be performed |

## New.Session

Will create a new session with the provided parameters.

### Required parameters

|  |  |  |
| --- | --- | --- |
| Parameter | Value type | Description |
| Session.ID | String | The ID of the session. The ID must be uniquqe. Two sessions with the same ID may not exist at the same time |
| User.Name | String | The name of the user that created this session |
| Project.Name | String | The name of the project this session is created for |
| Collection.Name | String | The collection where the data of this session is stored |
| Session.Type | String | The type of the session |

### Optional parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Value type | Default | Description |
| Start.Mandatory | Boolean | True | If true is provided the session will ensure that all mandatory services will be started |

## Session.Close

Will shutdown the whole session that the sender service is a part of. Every other service in the session will get a message with the parameter:

Message = “Service.ShutdownRequested”

### Required parameters

|  |  |  |
| --- | --- | --- |
| Parameter | Value type | Description |
| Session.ID | String | The service UID |

## Session.GetExists

Will return “true” if the session with the specified ID exists, otherwise returns “false”. The return value is a string.

### Required parameters

|  |  |  |
| --- | --- | --- |
| Parameter | Value type | Description |
| Session.ID | String | The ID of the session to check for existance |

## Session.GetServices

Will return a list containing all services in the specified session.

### Required parameters

|  |  |  |
| --- | --- | --- |
| Parameter | Value type | Description |
| Session.ID | String | The services will be returen from the specified session |