



SessionService

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

Alexander Küster



Table of contents

Action requests.....	2
Possible actions	2
Action unusual responses.....	2
Action detailed descriptions.....	3
Broadcast.....	3
Required parameters.....	3
Broadcast.Available	Fehler! Textmarke nicht definiert.
Closing	Fehler! Textmarke nicht definiert.
Get.DatabaseURL	3
New	3
Required parameters.....	4
Optional parameters	4
New.Session	4
Required parameters.....	4
Optional parameters	4
Session.Close	Fehler! Textmarke nicht definiert.
Session.GetExists	4
Required parameters.....	4
Session.GetServices.....	5
Optional parameters	5

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 immediately so the `queueAction` call has the same effect as the `performAction`.

To request an action a JSON string must be provided containing the keyword "action" and the action identifier.

Possible actions

Action	Description
Broadcast	Will send a broadcast message to all services in the senders' session
Broadcast.Available	Will send a broadcast message containing: "Service.Connected" as message to all services in the senders' session
Closing	Will deregister the sender from the session and notify others
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 response message
Session.Close	Will shutdown the whole session that the sender service is a part of
Session.GetExists	Will return "true" if the session with the specified ID exists, otherwise returns "false"
Session.GetServices	Will return a list containing all services in the senders' session or a specified session
New	Will register the sender as a new service in a session and return the sessions new UID
New.Session	Will create a new session
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> <i>Between the colon and the error description is a blank space</i>
Warning	WARNING: <Warning description> <i>Between the colon and the warning description is a blank space</i>

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](#) 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 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 unique. 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 existence

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 returned from the specified session