APS Session Service Provider

User Guide

Version: 1.0.0

Author: Tommy Svensson

Table of Contents

1 APSSessionService

This service provides session storage functionality. You can create a session, get an existing session by its id, and close a session. Each session can hold any number of named objects.

Why a session service ? To begin with, this is not an HttpSession! That said, it was created to handle a single session among several web applications. This for the APS administration web which are made up of several web applications working toghether. This is explained in detail in the APSAdminWeb documentation.

1.1 APIs

public *interface* **APSSession** [se.natusoft.osgi.aps.api.misc.session] {

This represents an active session.

String getId()

Returns

The id of this session.

boolean isValid()

Returns

true if this session is still valid.

void saveObject(String name, Object object)

Saves an object in the session. Will do nothing if the session is no longer valid.

Parameters

name *- The name to store the object under.*

object *- An object to store in the session.*

Object retrieveObject(String name)

Returns a object stored under the specified name or null if no object is stored under that name.

If isValid() returns false then this will always return null.

Parameters

name *- The name of the object to get.*

}

---

public *interface* **APSSessionService** [se.natusoft.osgi.aps.api.misc.session] {

This is not a http session! It is a simple session that can be used by any code running in the same OSGi server.

APSSession createSession(int timeoutInMinutes)

Creates a new session.

Parameters

timeoutInMinutes *- The timeout in minutes.*

APSSession createSession(String sessionId, int timeoutInMinutes)

Creates a new session.

The idea behind this variant is to support distributed sessions. The implementation must use a session id that is unique enough to support this. The APS implementation uses java.util.UUID.

Parameters

sessionId *- The id of the session to create.*

timeoutInMinutes *- The timeout in minutes.*

APSSession getSession(String sessionId)

Looks up an existing session by its id.

Returns

A valid session having the specified id or null.

Parameters

sessionId *- The id of the session to lookup.*

void closeSession(String sessionId)

Closes the session represented by the specified id. After this call APSSession.isValid() on an *APSSession* representing this session will return false.

Parameters

sessionId *- The id of the session to leaveSyncGroup.*

}
