

Service Management

Published 2014-10-28 | (Compatible with SDK 4.5,5.0,5.1 and 2013,2014 models)

This API allows developers to initiate the AllShare Framework (ASF).

Contents

asfservice

Introduction

Summary of Interfaces and Methods

Type Definition

Interfaces

asfservice Introduction

This API allows developers to initiate the AllShare Framework (ASF) which provides convergence services such as media sharing.

The ASF is a component which SAMSUNG devices contains to support it as a service.

Summary of Interfaces and Methods

Interface	Method
WebAPIAllshareObject	
AllShare	
ServiceConnectorObject	
ServiceProviderCreateSuccessCallback	void onsuccess (ServiceProvider serviceProvider)
ServiceProviderErrorCallback	void onerror (WebAPIError error, ServiceState state)
ServiceConnector	void createServiceProvider (ServiceProviderCreateSuccessCallback successCallback, ServiceProviderErrorCallback errorCallback) void deleteServiceProvider (SuccessCallback successCallback, ServiceProviderErrorCallback errorCallback) ServiceProvider getServiceProvider ()
ServiceProvider	DeviceFinder getDeviceFinder () ServiceState getServiceState ()

Type Definition

ServiceState

Specifies the state of ASF service.

Code example

```
enum ServiceState {
    "DISABLED",
    "ENABLED",
    "UNKNOWN"
};
```

DISABLED: The service state is disabled

ENABLED: The service state is enabled

UNKNOWN: The unknown service state

Interfaces

WebAPsAllShareObject

Defines what is instantiated in the AllShareObject.:

Code example

```
[NoInterfaceObject] interface WebAPsAllShareObject {
    readonly attribute AllShare allshare;
};
```

Code example

WebAPs implements WebAPsAllShareObject;

ATTRIBUTES

readonly AllShare allshare

The object will provide interfaces for connecting to services and obtain All Share devices.
This attribute is readonly.

AllShare

The AllShare root API.

Code example

```
[NoInterfaceObject] interface AllShare { };
```

AllShare interfaces will be always available within WebAPs object in the ECMAScript hierarchy (window.webapis.allshare).

ServiceConnectorObject

Defines what is instantiated in the webapis object.

Code example

```
[NoInterfaceObject] interface ServiceConnectorObject {
    readonly attribute ServiceConnector serviceconnector;
};
```

Code example

AllShare implements ServiceConnectorObject;

There will be a webapis.allshare.serviceconnector object that allows accessing the ASF service.

ATTRIBUTES

readonly ServiceConnector serviceconnector

The object will provide interfaces for connecting to ASF service.

This attribute is readonly.

ServiceProviderCreateSuccessCallback

Provides a callback when a ServiceProvider object is created successfully.

Code example

```
[Callback=FunctionOnly, NoInterfaceObject] interface ServiceProviderCreateSuccessCallback {
    void onsuccess(in ServiceProvider serviceProvider);
};
```

This callback interface specifies a success callback with ServiceProvider object as an input argument. It is used in an asynchronous operation ServiceConnector.createServiceProvider(...).

METHODS

onsuccess()

Invoked when ServiceProvider created successfully.

Signature:

Code example

```
void onsuccess(in ServiceProvider serviceProvider);
```

Parameters:

serviceProvider

Optional: No.

Nullable: No.

Type: ServiceProvider.

Description: the ServiceConnector object which is successfully created

ServiceProviderErrorCallback

Provides a error callback when the operation of a ServiceProvider object is failed.

Code example

```
[Callback=FunctionOnly, NoInterfaceObject] interface ServiceProviderErrorCallback {
    void onerror(in WebAPIError error, in ServiceState state);
};
```

METHODS

onerror()

Invoked when a ServiceProvider operation is failed.

Signature:

Code example

```
void onerror(in WebAPIError error, in ServiceState state);
```

Parameters:

error

Optional: No.

Nullable: No.

Type: WebAPIError.

Description: WebAPIError object which indicates error type and message.

state

Optional: No.

Nullable: No.

Type: ServiceState.

Description: Current state of the Service.

ServiceConnector

Provides the connection to the AllShare Framework(ASF) service.

Code example

```
[NoInterfaceObject] interface ServiceConnector {
    void createServiceProvider(in ServiceProviderCreateSuccessCallback successCallback,
        in optional ServiceProviderErrorCallback errorCallback) raises(WebAPIException);
    void deleteServiceProvider(in SuccessCallback successCallback,
        in optional ServiceProviderErrorCallback errorCallback) raises(WebAPIException);
    ServiceProvider getServiceProvider() raises(WebAPIException);
};
```

This interface provides interfaces for creating and deleting ASF service provider.

METHODS

createServiceProvider()

Creates a ServiceProvider object.

Signature:	
------------	--

Code example

```
void createServiceProvider(in ServiceProviderCreateSuccessCallback successCallback,
    in optional ServiceProviderErrorCallback errorCallback);
```

Asynchronous function that connects application to AllShare framework, and returns ServiceProvider object in success callback.

If this API is invoked several times, an exception with error type `AlreadyConnectedError` is synchronously thrown.

The error callback is launched with these error types:

`NotConnectedServiceError`: if the application doesn't connect to allshare framework.

`NetworkNotAvailableError`: if the network is not available.

`TimeOutError`: If the request is timeout.

`UnknownError`: In any other error case.

`InvalidValuesError`: If any input parameter contains invalid values.

`FrameworkNotInstalledError`: If framework is not installed.

Parameters:	
-------------	--

successCallback

Optional: No.

Nullable: No.

Type: `ServiceProviderCreateSuccessCallback`.

Description: callback function called if the request is successful

errorCallback

Optional: Yes.

Nullable: No.

Type: `ServiceProviderErrorCallback`.

Description: callback function called if the request if failed.

Exceptions:	
-------------	--

WebAPIException:

with error type `UnknownError` in any other case.

with error type `NotSupportedError`, if this feature is not supported.

with error type `SecurityError`, if this functionality is not allowed.

with error type `TypeMismatchError`, if any input parameter is not compatible with the expected type for that parameter.

with error type `AlreadyConnectedError`, if a connection with the allshare service already exists.

Code example

```
// Define success callback for creating ServiceProvider
function sProviderCallback(provider) {
    console.log("The service provider has obtained");
}
// Define error callback for creating ServiceProvider
function eProviderCallback(error, state) {
    console.log(error.name);
    console.log("Current ASF service state: " + state);
}
// Try to create ServiceProvider object.
try {
    webapis.allshare.serviceconnector.createServiceProvider(sProviderCallback, eProviderCallback);
} catch(e) {
    console.log(e.message);
}
deleteServiceProvider()
```

Deletes ServiceProvider object.

Signature:	
------------	--

Code example

```
void deleteServiceProvider(in SuccessCallback successCallback,
                             in optional ServiceProviderErrorCallback errorCallback);
```

Asynchronous function, that disconnects application from AllShare framework

The error callback is launched with these error types:

FrameworkNotInstalledError: If framework is not installed.
NotConnectedServiceError: if the application doesn't connect to allshare framework.
NotFoundError: if the serviceProvider does not exist.
NetworkNotAvailableError: if the network is not available.
TimeOutError: If the request is timeout.
UnknownError: In any other error case.

Parameters:	
-------------	--

successCallback

Optional: No.

Nullable: No.

Type: SuccessCallback.

Description: Callback function called if the request is successful

errorCallback

Optional: Yes.

Nullable: No.

Type: ServiceProviderErrorCallback.

Description: Callback function called if the request is failed

Exceptions:	
-------------	--

WebAPIException:

with error type UnknownError in anyother case.
with error type NotSupportedError, if this feature is not supported.

with error type `SecurityError`, if this functionality is not allowed.
with error type `TypeMismatchError`, if any input parameter is not compatible with the expected type for that parameter.

Code example

```
// ServiceProvider object - it is assumed that you have created serviceProvider.
var serviceProvider;
// Define a success callback for deleting ServiceProvider
function sProviderDeleteCallback() {
    console.log("The ASF service provider has deleted properly.");
}
// Define an error callback for creating ServiceProvider
function eProviderDeleteCallback(error, state) {
    console.log(error.name);
    console.log("The current ASF service state: " + state);
}
// Try to delete ServiceProvider object.
try {
    webapis.allshare.serviceconnector.deleteServiceProvider(sProviderDeleteCallback, eProviderDeleteCallback);
} catch(e) {
    console.log(e.message);
}
getServiceProvider()
```

Returns the `ServiceProvider` object that connects to AllShare framework

Signature:	
------------	--

`ServiceProvider getServiceProvider();`

Synchronous function, that returns the `ServiceProvider` object created in the application. If the object is deleted, returns null.

Exceptions:	
-------------	--

WebAPIException:

- with error type `UnknownError` in any other case.
- with error type `NotSupportedError`, if this feature is not supported.
- with error type `SecurityError`, if this functionality is not allowed.

Code example

```
try {
    // it is assumed that createServiceProvider() is successfully invoked
    var serviceProvider = webapis.allshare.serviceconnector.getServiceProvider();
    if (serviceProvider == null) {
        console.log("service provider had not been created properly");
    }
} catch(e) {
    console.log(e.message);
}
```

ServiceProvider

`ServiceProvider` contains methods allowing you to obtain `DeviceFinder` object.

Code example

```
[NoInterfaceObject] interface ServiceProvider {
    DeviceFinder getDeviceFinder() raises(WebAPIException);
    ServiceState getServiceState() raises(WebAPIException);
};
```

METHODS

getDeviceFinder()

Get the DeviceFinder object which provides to find allshare devices.

Signature:

Code example

```
DeviceFinder getDeviceFinder();
```

Return value:

DeviceFinder object.

Exceptions:

WebAPIException:

- with error type UnknownError in any other case.
- with error type NotSupportedError, if this feature is not supported.
- with error type SecurityError, if this functionality is not allowed.

Code example

```
// It is assumed that the ServiceProvider object is successfully obtained.
// For further details, see the createServiceProvider() or getServiceProvider().
var serviceProvider;
try {
    deviceFinder = serviceProvider.getDeviceFinder();
} catch(e) {
    console.log(e.name);
}
```

getServiceState()

Provides the current state of ASF service.

Signature:

Code example

```
ServiceState getServiceState();
```

Return value:

ServiceState value.

Exceptions:

WebAPIException:

- with error type UnknownError in any other case.
- with error type NotSupportedError, if this feature is not supported.
- with error type SecurityError, if this functionality is not allowed.