

mobileservice

Return Types

[OfflineDataDescription](#)

Method Summary

[OfflineDataDescr](#) [createOfflineDataDescription\(\)](#)
 iption Create a descriptive model for offline data.
[OfflineDataDescr](#) [createOfflineDataDescription\(prefix\)](#)
 iption Create a descriptive model for offline data.
[JSFoundSet](#) [createRemoteSearchFoundSet\(data\)](#)
 This should be called by inside the offline_data.
[Array](#) [getRowDescriptions\(datasource, pks\)](#)
 This method returns the description of rows as a json array object.
 void [performSync\(data, version, authenticateResult\)](#)
 If all the changes for a mobile client needs to come as once so that they can be in 1 transaction.

Method Details

createOfflineDataDescription

[OfflineDataDescription](#) [createOfflineDataDescription \(\)](#)

Create a descriptive model for offline data.

Returns

[OfflineDataDescription](#)

Sample

```
//plugins.mobile.getOfflineFoundSetData(foundset, null);
var data = plugins.mobileservice.createOfflineDataDescription('data_');
```

createOfflineDataDescription

[OfflineDataDescription](#) [createOfflineDataDescription \(prefix\)](#)

Create a descriptive model for offline data.

Parameters

{[String](#)} prefix - specified formname prefix used to retrieve row data from REST wise

Returns

[OfflineDataDescription](#)

Sample

```
//plugins.mobile.getOfflineFoundSetData(foundset, null);
var data = plugins.mobileservice.createOfflineDataDescription('data_');
```

createRemoteSearchFoundSet

[JSFoundSet](#) [createRemoteSearchFoundSet \(data\)](#)

This should be called by inside the offline_data.ws_create method of a mobile service solution.

That method is called by the the plugins.mobile.remoteSearch(foundset) that can be done on the mobile client.

This will return a FoundSet that is in findmode with the exact same FindStates that the mobile client had when remoteSearch is called.

On this FoundSet search() should be called and then this foundset should be added to a OfflineDataDescription.addFoundSet(fs) and that OfflineDataDescription object should be returned.

Parameters

{[Object](#)} data - The data that is given into the offline_data.ws_create method

Returns

[JSFoundSet](#) - The FoundSet that is in find mode or null if no FoundSet could be created.

Sample

```
function ws_create(data,version,method,authenticateResult) {
// Test if it is a remoteSearch call
if (method == "search") {
// Create the FoundSet from the data that is send over from the mobile client
var fs = plugins.mobileservice.createRemoteSearchFoundSet(data);
// This FoundSet is in find mode, more stuff could be added to it if you want to filter even more. Then
call search()
fs.search();
// Create the OfflineDataDescription that will be returned for this remoteSearch
var retval = plugins.mobileservice.createOfflineDataDescription('data_');
retval.addFoundSet(fs);
return retval;
}
}
```

getRowDescriptions

[Array](#) **getRowDescriptions** (datasource, pks)

This method returns the description of rows as a json array object.

Parameters

[{String}](#) datasource - the pks belong to
[{Object\[\]}](#) pks - the array of pks

Returns

[Array](#)

Sample

```
/** @type {Array<Object>} */
var idsArray = new Array(100,200,300);
var json = plugins.mobileservice.getRowDescriptions('db:/example_data/contacts', idsArray)
```

performSync

void **performSync** (data, version, authenticateResult)

If all the changes for a mobile client needs to come as once so that they can be in 1 transaction. Then the offline_data form should define a ws_update ((data,version,authenticateResult)) method.

This method will then get the complete data package which should be give to this function.

This will redispach all the inserts,updates or deletes to the various data forms that has the real rest methods to handle the actual insert.

If something fails then an Exception will be thrown and you can do a rollback so that everything of this sync is reverted.

Parameters

[{Object}](#) data - The data package that is given to the offline_data form ws_update method.
[{Number}](#) version - The version number that is given to the offline_data form ws_update method.
[{Object}](#) authenticateResult - The authenticateResult object that is given to the offline_data form ws_update method. (generated by the offline_data. ws_authenticate)

Returns

void

Sample

```
try {
databaseManager.startTransaction();
plugins.mobileservice.performSync(data,version,authenticateResult);
databaseManager.commitTransaction();
} catch (e) {
databaseManager.rollbackTransaction();
// log the error and return false to that the mobile client will know the sync did fail.
application.output(e,LOGGINGLEVEL.ERROR);
return false;
}
```