mobile

Return Types

Method Summary

void	call(telNumber) Initiate a phone call
void	clearLocalData() Clears local data - that is stored in browser local storage.
void	email(emailAddress) Initiate an email sending
void	getCurrentPosition(successCallback, errorHandler, options) Wrapper function around HTML5 getCurrentPosition.
String	getMarkupId(element) Retrieves markup id of an form element
String	getUUIDPKValueAsString(record) UUID pk's are mapped internally to integer to optimize space usage.
Number	getUsedStorageSpace() Retrieves amount of used bytes from browser local storage.
Boolean	isOnline() Returns whether mobile device is online.
void	loadData() (Re)Loads fresh data from the server, discarding the current local changes (asynchronously).
void	loadData(successCallback, errorHandler) (Re)Loads fresh data from the server, discarding the current local changes (asynchronously).
void	remoteSearch(foundset, successCallback, errorHandler) Performs a remote search for the given foundset, that foundset must be in findmode, else an exception will be thrown.
void	syncData() Synchronizes data between Mobile Client and Servoy Application Server (asynchronously).
void	syncData(successCallback, errorHandler) Synchronizes data between Mobile Client and Servoy Application Server (asynchronously).

Method Details

call

void **call** (telNumber)

Initiate a phone call

Parameters

{String} telNumber - telephone number to call

Returns

void

Sample

```
plugins.mobile.call("+490111111")
```

clearLocalData

void **clearLocalData** ()

Clears local data - that is stored in browser local storage.
After calling this method there will be no data in the mobile client until a syncData/loadData operation completes successfully.

Note: It is the caller's responsibility to show a form that doesn't need data or call a sync afterwards. (table structure data is cleared as well)
Use this with care!

Returns

void

Sample

```
// if we want do clear local copy of the data because the user did something really bad with local data
plugins.mobile.clearLocalData();
plugins.mobile.syncData();

// maybe we are done with a task and we don't want to use persistent space on the device
plugins.mobile.clearLocalData();
forms.noDataInitialForm.showWithMessage("Local data discarded."); // a form that displays the message and
shows itself on this call; it also allows doing a sync later for example
```

email

void **email** (emailAddress)

Initiate an email sending

Parameters

{String} emailAddress - destination email address

Returns

void

Sample

```
plugins.mobile.email("support@servoy.com")
```

getCurrentPosition

void **getCurrentPosition** (successCallback, errorHandler, options)

Wrapper function around HTML5 getCurrentPosition. Method will call successCallback method with a Position object argument if GPS position was retrieved. Otherwise will call errorHandler with an error argument. This method has dummy implementation in developer.
NOTE: For options you can specify (a) Accuracy of the returned location information (b) Timeout for retrieving the location information and (c) Use of cached location information

Parameters

{Function} successCallback - Method to be called if the coordinates are retrieved.
{Function} errorHandler - Method to be called if an error occurs while retrieving GPS coordinates. Can be null.
{Object} options - JSON string with different options that are supported by HTML5 function (accuracy, timeout, caching).

Returns

void

Sample

```
plugins.mobile.getCurrentPosition(forms.myform.showLocation,scopes.globals.errorHandler,{timeout:5000});
```

getMarkupId

String **getMarkupId** (element)

Retrieves markup id of an form element

Parameters

{RuntimeComponent} element - to retrive markup id for

Returns

String - the element's markup id

Sample

```
var id = plugins.mobile.getMarkupId(elements.myButton)
```

getUUIDPKValueAsString

String **getUUIDPKValueAsString** (record)

UUID pk's are mapped internally to integer to optimize space usage.
This function gets the real UUID pk value of the record value of pk.

Parameters

{JSRecord} record - the record to get the uuid from

Returns

String
Sample

```
var uuid = plugins.mobile.getUUIDPKValueAsString(record)
```

getUsedStorageSpace

Number **getUsedStorageSpace** ()
Retrieves amount of used bytes from browser local storage.
The storage amount depends on used browser, but, from specification, the browser should allow at least 5MB of local storage.

Note: Some browsers take two bytes per character!
Returns

Number
Sample

```
var usedSpace = plugins.mobile.getUsedStorageSpace()
```

isOnline

Boolean **isOnline** ()
Returns whether mobile device is online. Returns true in developer implementation.
Returns

Boolean
Sample

```
plugins.mobile.isOnline()
```

loadData

void **loadData** ()
(Re)Loads fresh data from the server, discarding the current local changes (asynchronously).
It is equivalent to the "load" part of a sync operation; it doesn't send local changes to the server and discards local changes when it receives new data from the server.

When done, this call might (and most of the time does) need to move to first form, as the (new) local data might be completely different - invalidating the current form/history.
Returns

void
Sample

```
// reload data from server and discard local changes
plugins.mobile.loadData();
```

loadData

void **loadData** (successCallback, errorHandler)
(Re)Loads fresh data from the server, discarding the current local changes (asynchronously).
It is equivalent to the "load" part of a sync operation; it doesn't send local changes to the server and discards local changes when it receives new data from the server.

When done, this call might (and most of the time does) need to move to first form, as the (new) local data might be completely different - invalidating the current form/history.
As callback methods are set, the call back methods will be responsible for doing this themselves - for example using and controller.show() of the first form.
Parameters

{Function} successCallback - Method to be called if the load was successful. Can be null.
First argument of the success callback is the size of received data in bytes.
{Function} errorHandler - Method to be called if an error occurs during load. Can be null.
First argument is the HTTP status code for the request or -1 if unknown.
Second argument is the error message.

Returns

void
Sample

```
function reloadAndDiscardData(event)
{
	plugins.mobile.loadData(onLoadSuccess, onLoadError);
}

function onLoadSuccess(dataBytes)
{
	if (dataBytes > 0) forms.contacts.showContacts(accountmanager_to_companies.companies_to_contacts);
	else forms.noData.showWithMessage("Server has no data to send.");
}

function onLoadError(http_status_code, error_message)
{
	plugins.dialogs.showWarningDialog("Warning", 'Reload of data failed with http status code: ' +
http_status_code + ', and error message: ' + error_message);
	forms.firstForm.showWithMessage("Data reload failed. Please press 'Synchronise' to try again.");
}
```

remoteSearch

void **remoteSearch** (foundset, successCallback, errorHandler)

Performs a remote search for the given foundset, that foundset must be in findmode, else an exception will be thrown.
This will make a call to the server that maps on the offline_data forms ws_create method. With the method param called "search".
In that method make a call to plugins.mobileservice.createRemoteSearchFoundSet(data) with the data that is given as the first argument.
This will make a foundset that is in find mode, where you can perform the seach on and return that through an OfflineDataDescription (like a sync)

Parameters

{JSFoundSet} foundset
{Function} successCallback
{Function} errorHandler

Returns

void
Sample

```
if (foundset.find())
{
    foundset.anumbervalue = ">10";
    plugins.mobile.remoteSearch(foundset,successCallback,errorCallback);
}
```

syncData

void **syncData** ()

Synchronizes data between Mobile Client and Servoy Application Server (asynchronously).
A synchronize will send all local changes to the server, clear the local storage and ask for all the data from the server.

When done, this call might (and most of the time does) need to move to first form, as the (new) local data might be completely different - invalidating the current form/history.

Returns

void
Sample

```
plugins.mobile.syncData()
```

syncData

void **syncData** (successCallback, errorHandler)

Synchronizes data between Mobile Client and Servoy Application Server (asynchronously).
A synchronize will send all local changes to the server, clear the local storage and ask for all the data from the server.
It will be calling "successCallback" on successful synchronization, with first argument the size of the received data in bytes,
or "errorHandler" if an error occurs during synchronization, with first argument the HTTP status code for the request or -1 if unknown, and the second argument
the error message.

When done, this call might (and most of the time does) need to move to first form, as the (new) local data might be completely different - invalidating the current form/history.
As callback methods are set, the call back methods will be responsible for doing this themselves - for example using controller.show() of the first form.

Parameters

{Function} successCallback - Method to be called if the sync was successful. Can be null.
{Function} errorHandler - Method to be called if an error occurs during synchronization. Can be null.

Returns

void

Sample

```
function onSyncAction(event)
{
	plugins.mobile.syncData(onSyncSuccess, onSyncError);
}

function onSyncSuccess(dataBytes)
{
	if (dataBytes > 0) forms.contacts.showContacts(accountmanager_to_companies.companies_to_contacts);
}

function onSyncError(http_status_code, error_message)
{
	plugins.dialogs.showWarningDialog("Warning", 'Sync failed with http status code: ' + http_status_code
+ ', and error message: ' + error_message);
}
```