

Application

Return Types

[JSEvent](#) [APPLICATION_TYPES](#)

Method Summary

Number	getApplicationType() Get the application type.
String	getServerURL() Gets the application server URL for mobile client to connect to.
Boolean	isInDeveloper() Returns true if the solution is running in the developer.
void	output(msg) Output something on the out stream.
void	setServerURL(applicationServerURL) Sets the application server URL for mobile client to connect to.
void	setValueListItems(name, displayValues) Fill a custom type valuelist with values from array(s) or dataset.
void	setValueListItems(name, displayValues, realValues) Fill a custom type valuelist with values from array(s) or dataset.

Method Details

getApplicationType

[Number](#) [getApplicationType\(\)](#)

Get the application type.

Returns

[Number](#) - Constant application type

Sample

```
var type = application.getApplicationType();
//see application type constant
```

getServerURL

[String](#) [getServerURL\(\)](#)

Gets the application server URL for mobile client to connect to.

Returns

[String](#) - HTTP server URL

Sample

```
var url = application.getServerURL();
```

isInDeveloper

[Boolean](#) [isInDeveloper\(\)](#)

Returns true if the solution is running in the developer.

Returns

[Boolean](#) - Boolean (true) if the solution is running in the developer, (false) otherwise

Sample

```
var flag = application.isInDeveloper();
```

output

void **output** (msg)

Output something on the out stream. (if running in debugger view output console tab)

Parameters

{[Object](#)} msg - Object to send to output stream

Returns

void

Sample

```
// log level is used to determine how/if to log in servoy_log.txt; for smart client java out and err streams are used
application.output('my very important trace msg');// default log level: info
```

setServerURL

void **setServerURL** (applicationServerURL)

Sets the application server URL for mobile client to connect to.

Parameters

{[String](#)} applicationServerURL - The URL to connect.

Returns

void

Sample

```
// if server URL is not set at mobile solution export, setServerURL should be called from custom login form, before authentication is done
application.setServerURL('http://localhost:8080');
```

setValueListItems

void **setValueListItems** (name, displayValues)

Fill a custom type valuelist with values from array(s) or dataset.

NOTE: if you modify values for checkbox/radio field, note that having one value in valuelist is a special case, so switching between one value and 0 /multiple values after form is created may have side effects

Parameters

{[String](#)} name - Name of the valuelist

{[Object](#)[]} displayValues - Display values array

Returns

void

Sample

```
//set display values (return values will be same as display values)
application.setValueListItems('my_en_types',new Array('Item 1', 'Item 2', 'Item 3'));
//set display values and return values (which are stored in dataprovider)
//application.setValueListItems('my_en_types',new Array('Item 1', 'Item 2', 'Item 3'),new Array
(10000,10010,10456));
//set display values and return values converted to numbers
//application.setValueListItems('my_en_types',new Array('Item 1', 'Item 2', 'Item 3'),new Array
('10000','10010', '10456'), true);
//do query and fill valuelist (see databaseManager for full details of queries/dataset)
//var query = 'select display_value,optional_real_value from test_table';
//var dataset = databaseManager.getDataSetByQuery(databaseManager.getDataSourceServerName(controller.
getDataSource()), query, null, 25);

//application.setValueListItems('my_en_types',dataset);
```

setValueListItems

void **setValueListItems** (name, displayValues, realValues)

Fill a custom type valuelist with values from array(s) or dataset.

NOTE: if you modify values for checkbox/radio field, note that having one value in valuelist is a special case, so switching between one value and 0/multiple values after form is created may have side effects

Parameters

{String} name - Name of the valuelist

{Object[]} displayValues - Display values array

{Object[]} realValues - Real values array

Returns

void

Sample

```
//set display values (return values will be same as display values)
application.setValueListItems('my_en_types',new Array('Item 1', 'Item 2', 'Item 3'));
//set display values and return values (which are stored in dataprovider)
//application.setValueListItems('my_en_types',new Array('Item 1', 'Item 2', 'Item 3'),new Array
(10000,10010,10456));
//set display values and return values converted to numbers
//application.setValueListItems('my_en_types',new Array('Item 1', 'Item 2', 'Item 3'),new Array
('10000','10010', '10456'), true);
//do query and fill valuelist (see databaseManager for full details of queries/dataset)
//var query = 'select display_value,optional_real_value from test_table';
//var dataset = databaseManager.getDataSetByQuery(databaseManager.getDataSourceServerName(controller.
getDataSource()), query, null, 25);

//application.setValueListItems('my_en_types',dataset);
```