

i18n

Method Summary

String	getCurrentLanguage() Gets the current language; based on the current locale settings in the Servoy Client Locale preferences.
String	getI18NMessage(i18nKey) Gets the real message (for the clients locale) for a specified message key.
String	getI18NMessage(i18nKey, dynamicValues) Gets the real message (for the clients locale) for a specified message key.
void	setI18NMessage(i18nKey, value) Sets the value of i18n key for client scope, if value null the setting is removed.
void	setLocale(language, country) Set/Overwrite the locale for this client.

Method Details

getCurrentLanguage

String [getCurrentLanguage\(\)](#)

Gets the current language; based on the current locale settings in the Servoy Client Locale preferences.

NOTE: For more information on i18n, see the chapter on Internationalization (i18n) in the Servoy Developer User's Guide, beginning with the Introduction to i18n

Returns

String - a String representing the current language.

Sample

```
var currLang = i18n.getCurrentLanguage();
```

getI18NMessage

String [getI18NMessage\(i18nKey\)](#)

Gets the real message (for the clients locale) for a specified message key.

Parameters

{String} i18nKey - The message key

Returns

String - a String that is the message for the message key.

Sample

```
// returns 'Welcome my_name in my solution'
// if the key 'mykey.username.text' is 'Welcome {0} in my solution'
i18n.getI18NMessage('mykey.username.text', new Array('my_name'))
```

getI18NMessage

String [getI18NMessage\(i18nKey, dynamicValues\)](#)

Gets the real message (for the clients locale) for a specified message key.

You can use parameter substitution by using {n}, where n is a index number of the value thats in the arguments array.

Parameters

{String} i18nKey - The message key

{Object[]} dynamicValues - Arguments array when using parameter substitution.

Returns

String - a String that is the message for the message key.

Sample

```
// returns 'Welcome my_name in my solution'
// if the key 'mykey.username.text' is 'Welcome {0} in my solution'
i18n.getI18NMessage('mykey.username.text',new Array('my_name'))
```

setI18NMessage

void **setI18NMessage** (i18nKey, value)

Sets the value of i18n key for client scope,if value null the setting is removed.
All forms not yet loaded will change (execute this in solution startup or first form)

Parameters

{String} i18nKey - The message key

{String} value - They value for the message key.

Returns

void

Sample

```
//sets the value of i18n key for client scope; if value null the setting is removed
//Warning: already created form elements with i18n text lookup will not change,
//so call this method in the solution startup method or in methods from first form
//this method saves message for current locale, so if locale is changed with setLocale, all messages set from
scripting will be lost

i18n.setI18NMessage('mykey.username.text', 'my_name')
```

setLocale

void **setLocale** (language, country)

Set/Overwrite the locale for this client.
All forms not yet loaded will change (execute this in solution startup or first form).

The language must be a lowercase 2 letter code defined by ISO-639.
see <http://www.ics.uci.edu/pub/ietf/http/related/iso639.txt>
The country must be an upper case 2 letter code defined by IS-3166
see http://www.chemie.fu-berlin.de/diverse/doc/ISO_3166.html

NOTE: For more information on i18n, see the chapter on Internationalization (i18n) in the Servoy Developer User's Guide, beginning with the Introduction to i18n

Parameters

{String} language - The lowercase 2 letter code

{String} country - The upper case 2 letter code.

Returns

void

Sample

```
//Warning: already created form elements with i18n text lookup will not change,
//so call this method in the solution startup method or in methods from first form

i18n.setLocale('en', 'US');
```