

i18n



Apr 07, 2024 12:47

Supported Clients

SmartClient WebClient NGClient MobileClient

Methods Summary

Array	getAvailableTimeZoneIds()	Get the list of available time zones.
Array	getCountries()	Gets the list of countries available for localization
String	getCurrencyFormat()	Gets the currency format from client (using client's locale).
String	getCurrentCountry()	Gets the current country; based on the current locale settings in the Servoy Client Locale preferences.
Array	getCurrentExtensions()	Gets the current private extensions; based on the current locale settings in the Servoy Client Locale preferences.
String	getCurrentLanguage()	Gets the current language; based on the current locale settings in the Servoy Client Locale preferences.
String	getCurrentTimeZone()	Gets the current time zone of the client; based on the current locale settings in the Servoy Client Locale preferences.
String	getDateFormat()	Gets the date format from client (using client's locale).
String	getDefaultDateFormat()	Gets the current default date format from server; based on the current admin settings.
String	getDefaultNumberFormat()	Gets the current default number format from server; based on the current admin settings.
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.
String	getI18NMessage(i18nKey, dynamicValues, language, country)	Gets the real message using specified locale for a specified message key.
String	getI18NMessage(i18nKey, language, country)	Gets the real message using the specified locale for a specified message key.
JSDataSet	getLanguages()	Returns a dataset with rows that contains a language key (en) and the displayname (English) column.
String	getNumberFormat()	Gets the number format from client (using client's locale).
JSDataSet	getSystemMessages()	Returns a dataset with rows that contains 3 columns: 'key' (i18n key), 'reference' (reference text for that key) and 'locale' ([CURRENT_LOCALE])' (where [CURRENT_LOCALE] is the current language) - with the system messages of servoy.
Number	getTimeZoneOffset(timeZoneId)	Returns the offset (in milliseconds) of this time zone from UTC for the current date or at the specified date.
Number	getTimeZoneOffset(timeZoneId, date)	Returns the offset (in milliseconds) of this time zone from UTC for the current date or at the specified date.
Array	getTimeZones()	Returns an array of known timezones.
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.
void	setLocale(language, country, extensions)	Set/Overwrite the locale for this client.
void	setTimeZone(timezone)	Sets the current time zone of the client.

Methods Details

getAvailableTimeZoneIds()

Get the list of available time zones.

Returns

[Array](#)

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
//Get the list of available time zones
var timezones = i18n.getAvailableTimeZoneIds();
```

getCountries()

Gets the list of countries available for localization

Returns

[Array](#) a String array containing the available countries.

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
i18n.getCountries()
```

getCurrencyFormat()

Gets the currency format from client (using client's locale).

Returns

[String](#) a String representing the currency format.

Supported Clients

SmartClient, WebClient, NGClient

Sample

```
var currencyFormat = i18n.getCurrencyFormat();
```

getCurrentCountry()

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

NOTE: For more information on i18n, see the chapter on Internationalization in the Servoy Developer User's Guide, and the chapter on Internationalization-I18N in the Programming Guide.

Returns

[String](#) a String representing the current country.

Supported Clients

SmartClient, WebClient, NGClient

Sample

```
var currCountry = i18n.getCurrentCountry();
```

getCurrentExtensions()

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

NOTE: For more information on i18n, see the chapter on Internationalization in the Servoy Developer User's Guide, and the chapter on Internationalization-I18N in the Programming Guide.

Returns

[Array](#) an array of Strings representing the current extensions.

Supported Clients

SmartClient, WebClient, NGClient

Sample

```
var currentExtensions = i18n.getCurrentExtensions();
```

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 in the Servoy Developer User's Guide, and the chapter on Internationalization-I18N in the Programming Guide.

Returns

[String](#) a String representing the current language.

Supported Clients

SmartClient, WebClient, NGClient, MobileClient

Sample

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

getCurrentTimeZone()

Gets the current time zone of the client; based on the current locale settings in the Servoy Client Locale preferences. For Servoy Web Clients the time zone is given by the browser (if it is possible to obtain it).

Returns

[String](#) a String representing the current time zone.

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var currTimeZone = i18n.getCurrentTimeZone();
```

getDateFormat()

Gets the date format from client (using client's locale).

Returns

[String](#) a String representing the date format.

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var dateFormat = i18n.getDateFormat();
```

getDefaultDateFormat()

Gets the current default date format from server; based on the current admin settings.

Returns

[String](#) a String representing the default date format.

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var defaultDateFormat = i18n.getDefaultDateFormat();
```

getDefaultNumberFormat()

Gets the current default number format from server; based on the current admin settings.

Returns

[String](#) a String representing the default number format.

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var defaultNumberFormat = i18n.getDefaultNumberFormat();
```

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.

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

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(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
Array dynamicValues Arguments array when using parameter substitution.

Returns

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

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

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(i18nKey, dynamicValues, language, country)

Gets the real message using specified 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
Array dynamicValues Arguments array when using parameter substitution.
String language The lowercase 2 letter code of the locale
String country The upper case 2 letter code of the locale

Returns

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

Supported Clients

WebClient,NGClient,MobileClient

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'),'en','US')
```

getI18NMessage(i18nKey, language, country)

Gets the real message using the specified locale for a specified message key.

Parameters

String i18nKey The message key
String language The lowercase 2 letter code of the locale
String country The upper case 2 letter code of the locale

Returns

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

Supported Clients

WebClient,NGClient,MobileClient

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'),'en','US')
```

getLanguages()

Returns a dataset with rows that contains a language key (en) and the displayname (English) column.

See <http://www.ics.uci.edu/pub/ietf/http/related/iso639.txt> for a list that could be returned.

Returns

[JSDataSet](#) a JSDataSet with all the languages.

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var set = i18n.getLanguages();
for(var i=1;i<=set.getMaxRowIndex();i++)
{
    application.output(set.getValue(i, 1) + " " + set.getValue(i, 2));
}
```

getNumberFormat()

Gets the number format from client (using client's locale).

Returns

[String](#) a String representing the number format.

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var numberFormat = i18n.getNumberFormat();
```

getSystemMessages()

Returns a dataset with rows that contains 3 columns: 'key' (i18n key), 'reference' (reference text for that key) and 'locale' ([CURRENT_LOCALE])' (where [CURRENT_LOCALE] is the current language) - with the system messages of servoy.

This means all servoy messages, with all available translations.

Returns

[JSDataSet](#) a JSDataSet with all the system messages.

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var set = i18n.getSystemMessages();
for(var i=1;i<=set.getMaxRowIndex();i++)
{
    application.output(set.getValue(i, 1) + " " + set.getValue(i, 2)+ " " + set.getValue(i, 3));
}
```

getTimeZoneOffset(timeZoneId)

Returns the offset (in milliseconds) of this time zone from UTC for the current date or at the specified date.

Parameters

[String](#) timeZoneId The time zone to get the offset for.

Returns

[Number](#) an int representing the time zone's offset from UTC.

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var timeZoneOffset = i18n.getTimeZoneOffset('America/Los_Angeles');
```

getTimeZoneOffset(timeZoneId, date)

Returns the offset (in milliseconds) of this time zone from UTC for the current date or at the specified date.

Parameters

String timeZoneId The time zone to get the offset for.

Date date The date in the time zone (default current date). Needed in case daylight saving time/GMT offset changes are used in the time zone.

Returns

Number an int representing the time zone's offset from UTC.

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var timeZoneOffset = i18n.getTimeZoneOffset('America/Los_Angeles');
```

getTimeZones()

Returns an array of known timezones.

Returns

Array an Array with all the timezones.

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var timeZones = i18n.getTimeZones();
```

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.

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

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(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 ISO 639-1 codes at http://en.wikipedia.org/wiki/List_of_ISO_639-1_code

The country must be an upper case 2 letter code defined by ISO-3166

see ISO-3166-1 codes at http://en.wikipedia.org/wiki/ISO_3166-1_alpha-2

NOTE: For more information on i18n, see the chapter on Internationalization in the Servoy Developer User's Guide, and the chapter on Internationalization-I18N in the Programming Guide.

Parameters

`String` language The lowercase 2 letter code
`String` country The upper case 2 letter code.

Supported Clients

SmartClient, WebClient, NGClient, MobileClient

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');
```

setLocale(language, country, extensions)

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 ISO 639-1 codes at http://en.wikipedia.org/wiki/List_of_ISO_639-1_code

The country must be an upper case 2 letter code defined by ISO-3166
 see ISO-3166-1 codes at http://en.wikipedia.org/wiki/ISO_3166-1_alpha-2

The extensions must be an array of strings indicating for example different industries. Each extension string must consist of only letters and digits with a max length of 8 characters
 see private extensions at <https://docs.oracle.com/javase/tutorial/i18n/locale/extensions.html>

NOTE: For more information on i18n, see the chapter on Internationalization in the Servoy Developer User's Guide, and the chapter on Internationalization-I18N in the Programming Guide.

Parameters

`String` language The lowercase 2 letter code
`String` country The upper case 2 letter code
`Array` extensions array of private extensions strings

Supported Clients

SmartClient, WebClient, NGClient

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');
```

setTimeZone(timezone)

Sets the current time zone of the client.
 The parameter should be a string having the format which can be retrieved via `i18n.getAvailableTimeZones` or can be seen in the SmartClient Edit -> Preferences -> Locale at the "Default Timezone" combobox.
 For instance the time zone for Netherlands is set using the ID "Europe/Amsterdam".

Parameters

`String` timezone the client's time zone

Supported Clients

SmartClient, WebClient

Sample

```
// This will set the default time zone to Central European Time
i18n.setTimeZone("Europe/Amsterdam");
```