

JSMethod

Property Summary

[String](#) [code](#)
The full source code of this method (including doc and function declaration).

Method Summary

[Object\[\]](#) [getArguments\(\)](#)
Gets the argument array for this method if that is set for the specific action this method is taken from.

[String](#) [getName\(\)](#)
The name of the method.

[String](#) [getScopeName\(\)](#)
Get scope name

Property Details

code

The full source code of this method (including doc and function declaration).

Returns

[String](#)

Sample

```
var method = form.newMethod('function original() { application.output("Original function."); }');
application.output('original method name: ' + method.getName());
application.output('original method code: ' + method.code);
method.code = 'function changed() { application.output("This is another function."); }';
method.showInMenu = false;
var button = form.newButton('Click me!', 10, 10, 100, 30, method);
```

Method Details

getArguments

[Object\[\]](#) [getArguments\(\)](#)

Gets the argument array for this method if that is set for the specific action this method is taken from.

Will return null by default. This is only for reading, you can't alter the arguments through this array, for that you need to create a new object through `solutionModel.wrapMethodWithArguments(..)` and assign it again.

Returns

[Object\[\]](#) - Array of the arguments, null if not specified.

Sample

```
var frm = solutionModel.getForm("myForm");
var button = frm.getButton("button");
// get the arguments from the button.
// NOTE: string arguments will be returned with quotes (comp.onAction.getArguments()[0] == '\foo\' evals to true)
var arguments = button.onAction.getArguments();
if (arguments && arguments.length > 1 && arguments[1] == 10) {
    // change the value and assign it back to the onAction.
    arguments[1] = 50;
    button.onAction = solutionModel.wrapMethodWithArguments(button.onAction,arguments);
}
```

getName

[String](#) **getName** ()

The name of the method.

Returns

[String](#) - A String holding the name of this method.

Sample

```
var method = form.newMethod('function original() { application.output("Original function."); }');
application.output('original method name: ' + method.getName());
application.output('original method code: ' + method.code);
method.code = 'function changed() { application.output("This is another function."); }';
method.showInMenu = false;
var button = form.newButton('Click me!', 10, 10, 100, 30, method);
```

getScopeName

[String](#) **getScopeName** ()

Get scope name

Returns

[String](#)

Sample

```
var methods = solutionModel.getGlobalMethods();
for (var x in methods)
    application.output(methods[x].getName() + ' is defined in scope ' + methods[x].getScopeName());
```