

# headlessclient

## Return Types

[JSCClient](#)

## Method Summary

[JSCClient](#) [#createClient](#)(solutionName, username, password, solutionOpenMethodArgs)  
Creates a headless client that will open the given solution.

[JSCClient](#) [#getClient](#)(clientID)  
Gets an existing headless client for the given client uuid.

## Method Details

createClient

[JSCClient](#) **createClient**(solutionName, username, password, solutionOpenMethodArgs)

Creates a headless client that will open the given solution.

### Parameters

solutionName

username

password

solutionOpenMethodArgs

### Returns

[JSCClient](#)

### Sample

```
// Creates a headless client that will open the given solution.
var headlessClient = plugins.headlessclient.createClient("someSolution", "user", "pass", null);
if (headlessClient != null && headlessClient.isValid()) {
    var x = new Object();
    x.name = 'remotel';
    x.number = 10;
    headlessClient.queueMethod(null, "remoteMethod", [x], callback);
}
```

getClient

[JSCClient](#) **getClient**(clientID)

Gets an existing headless client for the given client uuid.

### Parameters

clientID

### Returns

[JSCClient](#)

### Sample

```
// Gets an existing headless client for the given client uuid.
var headlessClient = plugins.headlessclient.getClient("clientID");
if (headlessClient != null && headlessClient.isValid()) {
    headlessClient.queueMethod(null, "someRemoteMethod", null, callback);
}
```