

invicti.auth

Contains form authentication helper methods.

Methods

`applyOtp(otpopt)`

Tries to find an OTP form and fill the specified password.

Parameters:

Name	Type	Attributes	Description
otp	string	<optional>	The OTP token to fill.

`click(e1, delayopt)`

Simulates a click for the specified e1.

Parameters:

Name	Type	Attributes	Description
e1	string HTMLElement		The element to click.
delay	number	<optional>	The number of milliseconds (thousandths of a second) that the click should be delayed by. Note that all delays are timed from the beginning of the script execution and does not work in sequence if you have several function calls with delay.

Example

```
// Click element by id  
invicti.auth.click('LoginButton');  
// Click element by id after 2 seconds  
invicti.auth.click('LoginButton', 2000);  
// Click element by DOM element reference  
invicti.auth.click(document.getElementsByTagName('BUTTON')[0]);
```

clickByQuery(query, delay_{opt})

Simulates a click for the element specified by the CSS query.

Parameters:

Name	Type	Attributes	Description
query	string		The CSS query that locates the element.
delay	number	<optional>	The number of milliseconds (thousandths of a second) that the click should be delayed by. Note that all delays are timed from the beginning of the script execution and does not work in sequence if you have several function calls with delay.

Example

```
// Click element by using a complex CSS query
invicti.auth.clickByQuery('#loginForm > div:nth-child(2) > button');
// Click element by id
invicti.auth.clickByQuery('#LoginButton');
// Click element by id after 2 seconds
invicti.auth.clickByQuery('#LoginButton', 2000);
```

clickByQueryAsync(query, delay_{opt})

Simulates a click for the element specified by the CSS with waiting query selector.

Parameters:

Name	Type	Attributes	Description
query	string		The CSS query that locates the element.
delay	number	<optional>	The number of milliseconds (thousandths of a second) that the click should be delayed by. Note that all delays are timed from the beginning of the script execution and does not work in sequence if you have several function calls with delay.

Example

```
// Click element by using a complex CSS query
await invicti.auth.clickByQueryAsync('#loginForm > div:nth-child(2) > button');
// Click element by id
await invicti.auth.clickByQueryAsync('#LoginButton');
// Click element by id after 2 seconds
await invicti.auth.clickByQueryAsync('#LoginButton', 2000);
```

`executeInFrame(frameEl, code, delayopt)`

Executes the supplied code in specified frame element.

Parameters:

Name	Type	Attributes	Description
frameEl	string HTMLElement		The frame element id or DOM reference to execute code.
code	string		The code to execute.
delay	number	<optional>	The number of milliseconds (thousandths of a second) that the code execution should be delayed by. Note that all delays are timed from the beginning of the script execution and does not work in sequence if you have several function calls with delay.

Example

```
// Clicks an element inside a frame  
var frame = document.getElementsByTagName('IFRAME')[1];  
invicti.auth.executeInFrame(frame, 'invicti.auth.click("LoginButton");');
```

`log(message)`

Logs a message.

Parameters:

Name	Type	Description
message	string	The message.

Example

```
// Log a simple message  
invicti.auth.log('Hello world!');
```

`login(usernameopt, passwordopt, otpopt, delayopt)`

Tries to find a login form and fill the specified credentials.

Parameters:

Name	Type	Attributes	Description
username	string	<optional>	The username to fill.
password	string	<optional>	The password to fill.
otp	string	<optional>	The one time password to fill.
delay	number	<optional>	The number of milliseconds (thousandths of a second) that the login should be delayed by. Note that all delays are timed from the beginning of the script execution and does not work in sequence if you have several function calls with delay.

Example

```
// Login using hard-coded values
invicti.auth.login('john.doe', 'p4ssw0rd', '543326');
// Login using hard-coded values after 2 seconds
invicti.auth.login('john.doe', 'p4ssw0rd', '543326', 2000);
// Login using implicit credentials (current persona)
invicti.auth.login();
// Login using implicit credentials (current persona) after 2 seconds
invicti.auth.login(2000);
```

`request(uriopt, requestObjectopt)`

Sends HTTP requests and returns response details.

There are three usages of this function:

1. If called with a string parameter, sends a GET request to the given URI and returns the response (async).
2. If called with no parameters, returns a requestObject to be passed to the function itself (async).
3. If called with requestObject parameter, sends a request which is defined by requestObject.

This function returns a response object which is explained at the latest example.

Parameters:

Name	Type	Attributes	Description
uri	string	<optional>	The target uri to send GET request.
requestObject	object	<optional>	Object containing request details.

Examples

```
var response = await invicti.auth.request("http://example.com");
```

```
var requestObject = invicti.auth.request();
requestObject.uri = "http://example.com";
requestObject.method = "POST";
requestObject.parameters.add("parameterKey", "parameterValue", 1); // 0 - GET (Queue)
requestObject.headers.add("headerName", "headerValue");
var response = await invicti.auth.request(requestObject);
```

```
var response = invicti.auth.request("http://example.com");
response.body; // the body of the response.
response.headers; // collection of response headers.
response.responseTime; // response time.
response.statusCode; // HTTP status code of the response.
response.statusDescription; // Human readable description of the HTTP response code.
response.uri; // The uri that HTTP request has sent.
```

setCurrentPersona(username, password, otp)

Sets the current credentials.

Parameters:

Name	Type	Description
username	string	The username.
password	string	The password.
otp	string	The OPT token.

Example

```
// Basic sample.  
invicti.auth.setCurrentPersona('john.doe', 'password', '123456');
```

setInputValue(el, value, delay_{opt})

Sets the value of specified el.

Parameters:

Name	Type	Attributes	Description
el	string HTMLElement		The element to click.
value	string		The value to set.
delay	number	<optional>	The number of milliseconds (thousandths of a second) that this set value operation should be delayed by. Note that all delays are timed from the beginning of the script execution and does not work in sequence if you have several function calls with delay.

Example

```
// Set value by id  
invicti.auth.setInputValue('Username', 'john.doe');  
// Set value after 2 seconds  
invicti.auth.setInputValue('Username', 'john.doe', 2000);  
// Set value by DOM element reference  
invicti.auth.setInputValue(document.forms[0].elements[0], 'john.doe');
```

setOtpField(query, value, isExactMatch)

Tries to find OTP field using the CSS query and sets its value.

Parameters:

Name	Type	Description
query	string	The CSS query that locates the element.
value	string	The value to set.
isExactMatch	boolean	The boolean value whether the CSS query is matches OTP field exactly.

Example

```
// Set OTP Field  
invicti.auth.setOtpField('#loginForm > div:nth-child(2) > input', '123456', true);
```

setOtpFieldAsync(query, value, isExactMatch)

Tries to find OTP field using the CSS query asynchronously and sets its value.

Parameters:

Name	Type	Description
query	string	The CSS query that locates the element.
value	string	The value to set.
isExactMatch	number	The number of milliseconds (thousandths of a second) that this set value operation should be delayed by. Note that all delays are timed from the beginning of the script execution and does not work in sequence if you have several function calls with delay.

Example

```
// Set OTP Field  
await invicti.auth.setOtpFieldAsync('#loginForm > div:nth-child(2) > input', '1234
```

setValueByQuery(query, value, delay_{opt})

Finds input element using the CSS query and sets its value.

Parameters:

Name	Type	Attributes	Description
query	string		The CSS query that locates the element.
value	string		The value to set.
delay	number	<optional>	The number of milliseconds (thousandths of a second) that this set value operation should be delayed by. Note that all delays are timed from the beginning of the script execution and does not work in sequence if you have several function calls with delay.

Example

```
// Query complex paths
invicti.auth.setValueByQuery('#loginForm > div:nth-child(2) > input', 'john.doe');
// Query by id
invicti.auth.setValueByQuery('#Username', 'john.doe');
// Set value after 2 seconds
invicti.auth.setValueByQuery('#Username', 'john.doe', 2000);
```

setValueByQueryAsync(query, value, delay_{opt})

Finds input element using the CSS query asynchronously and sets its value with waiting query selector.

Parameters:

Name	Type	Attributes	Description
query	string		The CSS query that locates the element.
value	string		The value to set.
delay	number	<optional>	The number of milliseconds (thousandths of a second) that this set value operation should be delayed by. Note that all delays are timed from the beginning of the script execution and does not work in sequence if you have several function calls with delay.

Example

```
// Query complex paths
await invicti.auth.setValueByQueryAsync('#loginForm > div:nth-child(2) > input', 'john.doe');
// Query by id
await invicti.auth.setValueByQueryAsync('#Username', 'john.doe');
// Set value after 2 seconds
await invicti.auth.setValueByQueryAsync('#Username', 'john.doe', 2000);
```


waitForSelector(query)

Waits until the dom element loaded then returns the element using the CSS query asynchronously.

Parameters:

Name	Type	Description
query	string	The CSS query that locates the element.

Example

```
// Query complex paths  
await invicti.auth.waitForSelector('#loginForm > div:nth-child(2) > input');  
// Query by id
```

waitTimeoutAsync(delay_{opt})

Waits for the specified number of milliseconds.

Parameters:

Name	Type	Attributes	Description
delay	number	<optional>	The number of milliseconds (thousandths of a second) that this set value operation should be delayed by.

Example

```
// Wait timeout  
invicti.auth.waitTimeoutAsync(5000);
```