

Search: 

[Libraries](#) » [jnicklas/capybara \(master\)](#) » [Index \(M\)](#) » [Capybara](#) » [Node](#) » [Matchers](#)  
([no frames](#))

[Class List](#) [Method List](#) [File List](#)



# Module: Capybara::Node::Matchers

Included in:

[Base](#), [Simple](#)

Defined in:

lib/capybara/node/matchers.rb

## Instance Method Summary ([collapse](#))

- [- \(Boolean\) has\\_button?\(locator\)](#)

Checks if the page or current node has a button with the given text, value or id.

- [- \(Boolean\) has\\_checked\\_field?\(locator\)](#)

Checks if the page or current node has a radio button or checkbox with the given label, value or id, that is currently checked.

- [- \(Boolean\) has\\_css?\(path, options = {}\)](#)

Checks if a given CSS selector is on the page or current node.

- [- \(Boolean\) has\\_field?\(locator, options = {}\)](#)

Checks if the page or current node has a form field with the given label, name or id.

- [- \(Boolean\) has\\_link?\(locator, options = {}\)](#)

Checks if the page or current node has a link with the given text or id.

- [- \(Boolean\) has\\_no\\_button?\(locator\)](#)

Checks if the page or current node has no button with the given text, value or id.

- [- \(Boolean\) has\\_no\\_checked\\_field?\(locator\)](#)

Checks if the page or current node has no radio button or checkbox with the given label, value or id, that is currently checked.

- [- \(Boolean\) \*\*has\\_no\\_css?\*\*\(path, options = {}\)](#)

Checks if a given CSS selector is not on the page or current node.

- [- \(Boolean\) \*\*has\\_no\\_field?\*\*\(locator, options = {}\)](#)

Checks if the page or current node has no form field with the given label, name or id.

- [- \(Boolean\) \*\*has\\_no\\_link?\*\*\(locator, options = {}\)](#)

Checks if the page or current node has no link with the given text or id.

- [- \(Boolean\) \*\*has\\_no\\_select?\*\*\(locator, options = {}\)](#)

Checks if the page or current node has no select field with the given label, name or id.

- [- \(Boolean\) \*\*has\\_no\\_selector?\*\*\(\\*args\)](#)

Checks if a given selector is not on the page or current node.

- [- \(Boolean\) \*\*has\\_no\\_table?\*\*\(locator, options = {}\)](#)

Checks if the page or current node has no table with the given id or caption.

- [- \(Boolean\) \*\*has\\_no\\_text?\*\*\(content\)](#) (also: `#has_no_content?`)

Checks if the page or current node does not have the given text content, ignoring any HTML tags and normalizing whitespace.

- [- \(Boolean\) \*\*has\\_no\\_unchecked\\_field?\*\*\(locator\)](#)

Checks if the page or current node has no radio button or checkbox with the given label, value or id, that is currently unchecked.

- [- \(Boolean\) \*\*has\\_no\\_xpath?\*\*\(path, options = {}\)](#)

Checks if a given XPath expression is not on the page or current node.

- [- \(Boolean\) \*\*has\\_select?\*\*\(locator, options = {}\)](#)

Checks if the page or current node has a select field with the given label, name or id.

- [- \(Boolean\) \*\*has\\_selector?\*\*\(\\*args\)](#)

Checks if a given selector is on the page or current node.

- [- \(Boolean\) \*\*has\\_table?\*\*\(locator, options = {}\)](#)

Checks if the page or current node has a table with the given id or caption.

- - (Boolean) **has\_text?(content)** (also: #has\_content?)

Checks if the page or current node has the given text content, ignoring any HTML tags and normalizing whitespace.

- - (Boolean) **has\_unchecked\_field?(locator)**

Checks if the page or current node has a radio button or checkbox with the given label, value or id, that is currently unchecked.

- - (Boolean) **has\_xpath?(path, options = {})**

Checks if a given XPath expression is on the page or current node.

## Instance Method Details

[permalink](#) - (Boolean) **has\_button?(locator)**

Checks if the page or current node has a button with the given text, value or id.

### Parameters:

- locator (String) —

The text, value or id of a button to check for

### Returns:

- (Boolean) —

Whether it exists

[\[View source\]](#)

```
# File 'lib/capybara/node/matchers.rb', line 255
```

```
255 def has_button?(locator)
256   has_selector?(:button, locator)
257 end
```

[permalink](#) - (Boolean) **has\_checked\_field?(locator)**

Checks if the page or current node has a radio button or checkbox with the given label, value or id, that is currently checked.

### Parameters:

- locator (String) —

The label, name or id of a checked field

## Returns:

- (Boolean) —

Whether it exists

[\[View source\]](#)

```
# File 'lib/capybara/node/matchers.rb', line 312
312 def has_checked_field?(locator)
313   has_selector?(:field, locator, :checked => true)
314 end
```

[permalink](#) - (Boolean) **has\_css?**(path, options = {})

Checks if a given CSS selector is on the page or current node.

```
page.has_css?('p#foo')
```

By default it will check if the selector occurs at least once, but a different number can be specified.

```
page.has_css?('p#foo', :count => 4)
```

This will check if the selector occurs exactly 4 times.

It also accepts all options that [Finders#all](#) accepts, such as :text and :visible.

```
page.has_css?('li', :text => 'Horse', :visible => true)
```

## Parameters:

- path (String) —  
A CSS selector
- options (Hash) (*defaults to: {}*) —  
a customizable set of options

## Options Hash (options):

- :count (Integer) — default: nil —  
Number of times the selector should occur

## Returns:

- (Boolean) —

If the selector exists

[\[View source\]](#)

```
# File 'lib/capybara/node/matchers.rb', line 159

159 def has_css?(path, options={})
160   has_selector?(:css, path, options)
161 end
```

[permalink](#) - (Boolean) **has\_field?**(locator, options = {})

Checks if the page or current node has a form field with the given label, name or id.

For text fields and other textual fields, such as textareas and HTML5 email/url/etc. fields, it's possible to specify a `:with` option to specify the text the field should contain:

```
page.has_field?('Name', :with => 'Jonas')
```

### Parameters:

- `locator (String)` —  
The label, name or id of a field to check for
- `options (Hash)` (*defaults to: {}*) —  
a customizable set of options

### Options Hash (**options**):

- `:with (String)` —  
The text content of the field

### Returns:

- (Boolean) —  
Whether it exists

[\[View source\]](#)

```
# File 'lib/capybara/node/matchers.rb', line 286

286 def has_field?(locator, options={})
287   has_selector?(:field, locator, options)
288 end
```

[permalink](#) - (Boolean) **has\_link?**(locator, options = {})

Checks if the page or current node has a link with the given text or id.

### Parameters:

- `locator (String)` —

The text or id of a link to check for

- `options` (*defaults to: {}*)

### Options Hash (`options`):

- `:href (String)` —

The value the href attribute must be

### Returns:

- `(Boolean)` —

Whether it exists

[\[View source\]](#)

```
# File 'lib/capybara/node/matchers.rb', line 231
231 def has_link?(locator, options={})
232   has_selector?(:link, locator, options)
233 end
```

[permalink](#) - `(Boolean)` **has\_no\_button?**(locator)

Checks if the page or current node has no button with the given text, value or id.

### Parameters:

- `locator (String)` —

The text, value or id of a button to check for

### Returns:

- `(Boolean)` —

Whether it doesn't exist

[\[View source\]](#)

```
# File 'lib/capybara/node/matchers.rb', line 267
267 def has_no_button?(locator)
268   has_no_selector?(:button, locator)
269 end
```

[permalink](#) - `(Boolean)` **has\_no\_checked\_field?**(locator)

Checks if the page or current node has no radio button or checkbox with the given label, value or id, that is currently checked.

### Parameters:

- `locator (String)` —

The label, name or id of a checked field

### Returns:

- `(Boolean)` —

Whether it doesn't exists

[\[View source\]](#)

```
# File 'lib/capybara/node/matchers.rb', line 325
325 def has_no_checked_field?(locator)
326   has_no_selector?(:field, locator, :checked => true)
327 end
```

[permalink](#) - `(Boolean)` **has\_no\_css?**(path, options = {})

Checks if a given CSS selector is not on the page or current node. Usage is identical to `Capybara::Node::Matchers#has_css?`

### Returns:

- `(Boolean)`

[\[View source\]](#)

```
# File 'lib/capybara/node/matchers.rb', line 171
171 def has_no_css?(path, options={})
172   has_no_selector?(:css, path, options)
173 end
```

[permalink](#) - `(Boolean)` **has\_no\_field?**(locator, options = {})

Checks if the page or current node has no form field with the given label, name or id. See [#has\\_field?](#).

### Parameters:

- `locator (String)` —

The label, name or id of a field to check for

- `options (Hash)` (*defaults to: {}*) —

a customizable set of options

### Options Hash (**options**):

- `:with (String)` —

The text content of the field

### Returns:

- `(Boolean)` —

Whether it doesn't exist

[\[View source\]](#)

```
# File 'lib/capybara/node/matchers.rb', line 299
299 def has_no_field?(locator, options={})
300   has_no_selector?(:field, locator, options)
301 end
```

[permalink](#) - `(Boolean)` **has\_no\_link?**(locator, options = {})

Checks if the page or current node has no link with the given text or id.

### Returns:

- `(Boolean)` —

Whether it doesn't exist

[\[View source\]](#)

```
# File 'lib/capybara/node/matchers.rb', line 243
243 def has_no_link?(locator, options={})
244   has_no_selector?(:link, locator, options)
245 end
```

[permalink](#) - `(Boolean)` **has\_no\_select?**(locator, options = {})

Checks if the page or current node has no select field with the given label, name or id. See [#has\\_select?](#).

### Parameters:

- `locator (String)` —

The label, name or id of a select box

- `options (Hash)` (*defaults to: {}*) —



a customizable set of options

## Returns:

- (Boolean) —

Whether it doesn't exist

[\[View source\]](#)

```
# File 'lib/capybara/node/matchers.rb', line 390
```

```
390 def has_no_select?(locator, options={})
391   has_no_selector?(:select, locator, options)
392 end
```

[permalink](#) - (Boolean) **has\_no\_selector?(\*args)**

Checks if a given selector is not on the page or current node. Usage is identical to Capybara::Node::Matchers#has\_selector?

## Returns:

- (Boolean)

[\[View source\]](#)

```
# File 'lib/capybara/node/matchers.rb', line 67
```

```
67 def has_no_selector?(*args)
68   options = if args.last.is_a?(Hash) then args.last else {} end
69   wait_until do
70     results = all(*args)
71
72     case
73     when results.empty?
74       true
75     when options[:between]
76       not(options[:between] === results.size)
77     when options[:count]
78       not(options[:count].to_i == results.size)
79     when options[:maximum]
80       not(options[:maximum].to_i >= results.size)
81     when options[:minimum]
82       not(options[:minimum].to_i <= results.size)
83     else
84       results.empty?
85     end or raise ExpectationNotMet
86   end
87 rescue Capybara::ExpectationNotMet
88   return false
89 end
```

[permalink](#) - (Boolean) **has\_no\_table?(locator, options = {})**

Checks if the page or current node has no table with the given id or caption. See [#has\\_table?](#).

## Parameters:

- `locator (String)` —

The id or caption of a table

## Returns:

- `(Boolean)` —

Whether it doesn't exist

[\[View source\]](#)

```
# File 'lib/capybara/node/matchers.rb', line 422
422 def has_no_table?(locator, options={})
423   has_no_selector?(:table, locator, options)
424 end
```

[permalink](#) - `(Boolean)` **has\_no\_text?**(content) Also known as: `has_no_content?`

Checks if the page or current node does not have the given text content, ignoring any HTML tags and normalizing whitespace.

Unlike `has_content` this only matches displayable text and specifically excludes text contained within non-display nodes such as `script` or `head` tags.

## Parameters:

- `content (String)` —

The text to check for

## Returns:

- `(Boolean)` —

Whether it exists

[\[View source\]](#)

```
# File 'lib/capybara/node/matchers.rb', line 209
209 def has_no_text?(content)
210   Checks if the page or current node has no radio button or checkbox
211   with the given label, value or id that is currently unchecked.
212   wait_until do
213     !normalize_whitespace(text).include?(normalized_content) or
214     false
215   end
216 rescue Capybara::ExpectationNotMet
217   return false
218 end
```

The label, name or id of an unchecked field

## Returns:

- (Boolean) —

Whether it doesn't exists

[\[View source\]](#)

```
# File 'lib/capybara/node/matchers.rb', line 351
351 def has_no_unchecked_field?(locator)
352   has_no_selector?(:field, locator, :unchecked => true)
353 end
```

[permalink](#) - (Boolean) **has\_no\_xpath?**(path, options = {})

Checks if a given XPath expression is not on the page or current node. Usage is identical to Capybara::Node::Matchers#has\_xpath?

## Returns:

- (Boolean)

[\[View source\]](#)

```
# File 'lib/capybara/node/matchers.rb', line 132
132 def has_no_xpath?(path, options={})
133   has_no_selector?(:xpath, path, options)
134 end
```

[permalink](#) - (Boolean) **has\_select?**(locator, options = {})

Checks if the page or current node has a select field with the given label, name or id.

It can be specified which option should currently be selected:

```
page.has_select?('Language', :selected => 'German')
```

For multiple select boxes, several options may be specified:

```
page.has_select?('Language', :selected => ['English', 'German'])
```

It's also possible to check if a given set of options exists for this select box:

```
page.has_select?('Language', :options => ['English', 'German'])
```

## Parameters:

- locator (String) —

The label, name or id of a select box

- `options (Hash)` (*defaults to: {}*) —  
a customizable set of options

### Options Hash (`options`):

- `:options (Array)` —  
Options which should be contained in this select box
- `:selected (String, Array)` —  
Options which should be selected

### Returns:

- `(Boolean)` —  
Whether it exists

[\[View source\]](#)

```
# File 'lib/capybara/node/matchers.rb', line 378
378 def has_select?(locator, options={})
379   has_selector?(:select, locator, options)
380 end
```

[permalink](#) - `(Boolean)` **has\_selector?**(\*args)

Checks if a given selector is on the page or current node.

```
page.has_selector?('p#foo')
page.has_selector?(:xpath, '//*[@id="foo"]')
page.has_selector?(:foo)
```

By default it will check if the expression occurs at least once, but a different number can be specified.

```
page.has_selector?('p#foo', :count => 4)
```

This will check if the expression occurs exactly 4 times.

It also accepts all options that [Finders#all](#) accepts, such as `:text` and `:visible`.

```
page.has_selector?('li', :text => 'Horse', :visible => true)
```

`has_selector?` can also accept XPath expressions generated by the XPath gem:

```
xpath = XPath.generate { |x| x.descendant(:p) }
page.has_selector?(:xpath, xpath)
```

## Parameters:

- options (Hash) —  
a customizable set of options

## Returns:

- (Boolean) —  
If the expression exists

[\[View source\]](#)

```
# File 'lib/capybara/node/matchers.rb', line 35
```

```

35 def has_selector?(*args)
36   options = if args.last.is_a?(Hash) then args.last else {} end
37   wait_until do
38     results = all(*args)
39
40     case
41     when results.empty?
42       false
43     when options[:between]
44       options[:between] === results.size
45     when options[:count]
46       options[:count].to_i == results.size
47     when options[:maximum]
48       options[:maximum].to_i >= results.size
49     when options[:minimum]
50       options[:minimum].to_i <= results.size
51     else
52       results.size > 0
53     end or raise ExpectationNotMet
54   end
55 rescue Capybara::ExpectationNotMet
56   return false
57 end

```

[permalink](#) - (Boolean) **has\_table?**(locator, options = {})

Checks if the page or current node has a table with the given id or caption.

If the options :rows is given, it will check that the table contains the rows and columns given:

```
page.has_table?('People', :rows => [['Jonas', '24'], ['Peter', '32']])
```

Note that this option is quite strict, the order needs to be correct and the text needs to match exactly.

## Parameters:

- locator (String) —  
The id or caption of a table

## Returns:

- (Boolean) —

Whether it exist

[\[View source\]](#)

```
# File 'lib/capybara/node/matchers.rb', line 410
410 def has_table?(locator, options={})
411   has_selector?(:table, locator, options)
412 end
```

[permalink](#) - (Boolean) **has\_text?**(content) Also known as: has\_content?

Checks if the page or current node has the given text content, ignoring any HTML tags and normalizing whitespace.

Unlike has\_content this only matches displayable text and specifically excludes text contained within non-display nodes such as script or head tags.

## Parameters:

- content (String) —

The text to check for

## Returns:

- (Boolean) —

Whether it exists

[\[View source\]](#)

```
# File 'lib/capybara/node/matchers.rb', line 186
186 def has_text?(content)
187   normalized_content = normalize_whitespace(content)
188
189   wait_until do
190     normalize_whitespace(text).include?(normalized_content) or
191     raise ExpectationNotMet
192   end
193 rescue Capybara::ExpectationNotMet
194   return false
195 end
```

[permalink](#) - (Boolean) **has\_unchecked\_field?**(locator)

Checks if the page or current node has a radio button or checkbox with the given label, value or id, that is currently unchecked.

## Parameters:

- `locator (String)` —

The label, name or id of an unchecked field

## Returns:

- `(Boolean)` —

Whether it exists

[\[View source\]](#)

```
# File 'lib/capybara/node/matchers.rb', line 338
338 def has_unchecked_field?(locator)
339   has_selector?(:field, locator, :unchecked => true)
340 end
```

[permalink](#) - `(Boolean)` **has\_xpath?**(path, options = {})

Checks if a given XPath expression is on the page or current node.

```
page.has_xpath?(' ./p[@id="foo"] ')
```

By default it will check if the expression occurs at least once, but a different number can be specified.

```
page.has_xpath?(' ./p[@id="foo"] ', :count => 4)
```

This will check if the expression occurs exactly 4 times.

It also accepts all options that [Finders#all](#) accepts, such as `:text` and `:visible`.

```
page.has_xpath?(' ./li ', :text => 'Horse', :visible => true)
```

`has_xpath?` can also accept XPath expressions generate by the XPath gem:

```
xpath = XPath.generate { |x| x.descendant(:p) }
page.has_xpath?(xpath)
```

## Parameters:

- `path (String)` —

An XPath expression

- `options (Hash)` (*defaults to: {}*) —

a customizable set of options

## Options Hash (options):

- `:count` (Integer) — default: nil —

Number of times the expression should occur

## Returns:

- (Boolean) —

If the expression exists

[\[View source\]](#)

```
# File 'lib/capybara/node/matchers.rb', line 120
```

```
120 def has_xpath?(path, options={})  
121   has_selector?(:xpath, path, options)  
122 end
```

Generated on Tue Jan 10 17:07:33 2012 by [yard](#) 0.7.4 (ruby-1.9.3).