Package 'CSS'

July 2, 2014

Type Package

Title Extract information from an html document with CSS selectors

Version 1.0.2

Date 2013-01-31

Author Francois Guillem <guillem.francois@gmail.com>

Maintainer Francois Guillem <guillem.francois@gmail.com>

Description The package provides functions that facilitate extraction of information from an html document by using css selectors instead of xpath queries.

License GPL (>= 2)

Depends XML, stringr

Suggests testthat

NeedsCompilation no

Repository CRAN

Date/Publication 2013-11-08 07:08:46

R topics documented:

CSS-package *Extract information from an html document with CSS selectors*

Description

The package provides functions that facilitate extraction of information from an html document by using css selectors instead of xpath queries.

Details

Package:	CSS
Type:	Package
Version:	1.0
Date:	2013-01-31
License:	GPL (>= 2)

The functions of this packages are actually wrapper functions for the package XML. For instance `cssApply` converts a css path in an xpath query and then invokes `xpathSApply`.

Author(s)

Francois Guillem <guillem.francois@gmail.com>

Examples

```
doc <- "<html>
<head></head>
<body>
  <div id='character1' class='character'>
    <span class='name'>Mike</span>
    <span class='level digit'>10</span>
  </div>
  <div id='character2' class='character'>
    <span class='name'>Stan</span>
  </div>
</body>
</html>"

doc <- htmlParse(doc)

# Names of the characters
cssApply(doc, ".character>.name", cssCharacter)

# Name of character1
cssApply(doc, "#character1>.name", cssCharacter)

# Level of characters
```

```
cssApply(doc, ".character>.level", cssNumeric)

# character 2 does not have level, we would want to have a NA value instead of nothing
cssApplyInNodeSet(doc, ".character", ".level", cssNumeric)
```

`cssApply` *Apply a function to elements identified by a CSS path.*

Description

This function is a wrapper function for `xpathSApply`. It selects nodes inside an html document corresponding to a CSS path and then applies an arbitrary function to theses nodes. It is useful to extract data frome an html page.

Usage

```
cssApply(doc, path, fun, ...)
```

Arguments

`doc`	An html document parsed with `htmlParse`.
`path`	CSS path
`fun`	Function to extract data from the selected nodes. See `cssExtract`
`...`	Parameters passed to "`fun`"

Value

The result may differ based on the function used. It will generally be a character or numeric vector with length equal to the number of nodes selected.

If no element in the html document corresponds to the path provided, the function will return an empty list.

Author(s)

Francois Guillem <guillem_francois@gmail.com>

See Also

`cssApplyInNodeSet`, `cssNumeric`

Examples

```
doc <- "<html>
<head></head>
<body>
  <div id='character1' class='character'>
    <span class='name'>Mike</span>
    <span class='level digit'>10</span>
  </div>
  <div id='character2' class='character'>
    <span class='name'>Stan</span>
  </div>
</body>
</html>"

doc <- htmlParse(doc)

# Names of the characters
cssApply(doc, ".character>.name", cssCharacter)

# Name of character1
cssApply(doc, "#character1>.name", cssCharacter)
```

`cssApplyInNodeSet` *Find a set of elements and inside each of them apply a function to some element.*

Description

Consider the following case : on an html page you have information about several people. Several informations are included in a div of class "people", but for some people some information is missing.

With this function, you can first select all divs of class "people" and then search inside them if the information is available. If not, the function will return a NA value for the person.

Usage

```
cssApplyInNodeSet(doc, path, relPath, fun, prefix = "./", ...)
```

Arguments

`doc`	An html document parsed with `htmlParse`.
`path`	Character. It CSS path used to identify elements where to search the information
`relPath`	Character. CSS path used to select elements in the elements selected with "path"
`fun`	Function to apply to the selected nodes.
`prefix`	Should be "./" if the first element in relPath has to be the child of the elements selected by "path", or ".//" if it may be any descendent (and not necessarily a direct child).
`...`	Parameters passed to "`fun`"

Value

If no element in the html document corresponds to "path", the function will return an empty list. Else it will return a list of length equal to the number of elements selected.

Author(s)

Francois Guillem <guillem.francois@gmail.com>

See Also

cssApply, cssNumeric

Examples

```
doc <- "<html>
<head></head>
<body>
  <div id='character1' class='character'>
    <span class='name'>Mike</span>
    <span class='level digit'>10</span>
  </div>
  <div id='character2' class='character'>
    <span class='name'>Stan</span>
  </div>
</body>
</html>"

doc <- htmlParse(doc)

# Level of characters
cssApply(doc, ".character>.level", cssNumeric)

# character 2 does not have level, we would want to have a NA value instead of nothing
cssApplyInNodeSet(doc, ".character", ".level", cssNumeric)
```

---

cssExtract *Extract information from an html element*

Description

These function has to be used in cssApply and cssApplyInNodeSet. They aim to facilitate extraction of different kind of information.

cssNumeric extracts the numeric value of an element, cssCharacter extracts text. cssLink extracts the url of a link.

cssSrc, cssId, cssClass, cssName and cssValue are less usefull and extract respectively the source of an element, its id, its css class, its name and its value (may be usefull for input elements).

Usage

```
cssLink(node)
cssNumeric(node, ...)
cssCharacter(node, ...)
cssSrc(node)
cssId(node)
cssClass(node)
```

Arguments

node	An html element
...	Additional arguments passed to xmlValue

Value

All these function return a chracter string except cssNumeric which returns a Numeric value

Author(s)

Francois Guillem <guillem.francois@gmail.com>

See Also

cssApply, cssApplyInNodeSet

Examples

```
doc <- "<html>
<head></head>
<body>
  <div id='character1' class='character'>
    <span class='name'>Mike</span>
    <span class='level digit'>10</span>
    <a href='http://someurl.com'>Complete profile</a>
  </div>
  <div id='character2' class='character'>
    <span class='name'>Stan</span>
    <a href='http://someurl2.com'>Complete profile</a>
  </div>
</body>
</html>"

doc <- htmlParse(doc)

# Names of the characters
cssApply(doc, ".character>.name", cssCharacter)

# Name of character1
cssApply(doc, "#character1>.name", cssCharacter)

# Urls of the profiles
```

```
cssApply(doc, ".character>a", cssLink)

# Level of characters
cssApply(doc, ".character>.level", cssNumeric)

# character 2 does not have level, we would want to have a NA value instead of nothing
cssApplyInNodeSet(doc, ".character", ".level", cssNumeric)
```

`cssToXpath` *Translate a cssPath to an xpath.*

Description

this function translates a CSS path in an xpath query. It is used by `cssApply` and you generally won't have to use it directly. Nevertheless, it may be useful for debug if `cssApply` does unexpected things.

Usage

```
cssToXpath(cssPath, prefix = "//")
```

Arguments

`cssPath`	Character. A CSS path
`prefix`	Character string appended at the beggining of the xpath query. Valid options are "/", "//", "./" and ".//"

Value

A character string representing an xpath query.

Author(s)

Francois Guillem <guillem.francois@gmail.com>

See Also

`cssApply`, `cssApplyInNodeSet`

Examples

```
cssToXpath(".character>.name")
cssToXpath("#character1 .name")
```

Index