Updates to the R Documentation System

Andrew Redd, R documentation Task Force

UseR!2017
History

- UseR!2016
- July 2016, R Consortium Funding
- September-December 2016
- January 2017 - Now
Full class system for documentation
- Can document anything
- stored in `attr(., 'documentation')` for traditional objects;
  - functions, data.frames, etc.
- Stored as a meta object similar to class definitions for S4 and reference classes.
- Everything can be accessed or set from the documentation and `documentation<- generics`.
- Formats for input and output support decided
- Independent of but supports the package system
Many in - Many Out

Figure 1: Many in - Many out approach.
Current State

- parsetools
- documentation
- documentation-md
- documentation-yaml
```r
library(parsetools)
options(keep.source=TRUE)
pd <- get_parse_data(parse(text="x <- rnorm(100)"))
kable(pd[c(-1,-3)])
```

<table>
<thead>
<tr>
<th>col1</th>
<th>col2</th>
<th>id</th>
<th>parent</th>
<th>token</th>
<th>terminal</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>SYMBOL</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>14</td>
<td>expr</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>15</td>
<td>14</td>
<td>0</td>
<td>expr</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>14</td>
<td>LEFT_ASSIGN</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>10</td>
<td>4</td>
<td>6</td>
<td>SYMBOL_FUNCTION_CALL</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>10</td>
<td>6</td>
<td>12</td>
<td>expr</td>
</tr>
<tr>
<td>12</td>
<td>6</td>
<td>15</td>
<td>12</td>
<td>14</td>
<td>expr</td>
</tr>
<tr>
<td>5</td>
<td>11</td>
<td>11</td>
<td>5</td>
<td>12</td>
<td>‘(’</td>
</tr>
<tr>
<td>7</td>
<td>12</td>
<td>14</td>
<td>7</td>
<td>8</td>
<td>NUM_CONST</td>
</tr>
<tr>
<td>8</td>
<td>12</td>
<td>14</td>
<td>8</td>
<td>12</td>
<td>expr</td>
</tr>
<tr>
<td>9</td>
<td>15</td>
<td>15</td>
<td>9</td>
<td>12</td>
<td>‘)’</td>
</tr>
</tbody>
</table>
```
get_pd_assign_value(pd) %>% select(-starts_with('line')) %>% kable

<table>
<thead>
<tr>
<th>col1</th>
<th>col2</th>
<th>id</th>
<th>parent</th>
<th>token</th>
<th>terminal</th>
<th>text</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>6</td>
<td>10</td>
<td>4</td>
<td>6</td>
<td>SYMBOL_FUNCTION_CALL</td>
<td>TRUE</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>10</td>
<td>6</td>
<td>12</td>
<td>expr</td>
<td>FALSE</td>
</tr>
<tr>
<td>12</td>
<td>6</td>
<td>15</td>
<td>12</td>
<td>14</td>
<td>expr</td>
<td>FALSE</td>
</tr>
<tr>
<td>5</td>
<td>11</td>
<td>11</td>
<td>5</td>
<td>12</td>
<td>'('</td>
<td>TRUE</td>
</tr>
<tr>
<td>7</td>
<td>12</td>
<td>14</td>
<td>7</td>
<td>8</td>
<td>NUM_CONST</td>
<td>TRUE</td>
</tr>
<tr>
<td>8</td>
<td>12</td>
<td>14</td>
<td>8</td>
<td>12</td>
<td>expr</td>
<td>FALSE</td>
</tr>
<tr>
<td>9</td>
<td>15</td>
<td>15</td>
<td>9</td>
<td>12</td>
<td>')'</td>
<td>TRUE</td>
</tr>
</tbody>
</table>

get_pd_assign_variable(pd) %>% select(-starts_with('line')) %>% kable

<table>
<thead>
<tr>
<th>col1</th>
<th>col2</th>
<th>id</th>
<th>parent</th>
<th>token</th>
<th>terminal</th>
<th>text</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>SYMBOL</td>
<td>TRUE</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>14</td>
<td>expr</td>
<td>FALSE</td>
</tr>
</tbody>
</table>
```


documentation

```r
library(documentation)
x <- new('Documentation'
  , title = "Example documentation"
  , author = person('Andrew', 'Redd')
  , description = "This is used to demonstrate the class "system for the documentation package"
  , keywords = "documentation"
)
```
```r
str(x)

## Formal class 'Documentation' [package "documentation"] with 10 slots
## ..@ author :Class 'person' hidden list of 1
## .. @ title : chr "Example documentation"
## .. @ description: chr "This is used to demonstrate the class system for the documentation package in R"
## .. @ references : list()
## ..@ seealso : chr(0)
## ..@ examples : list()
## ..@ keywords : chr "documentation"
## ..@ concepts : chr(0)
## ..@ sections : list()
toRd(x) %>% cat(sep='\n')

## \author{Andrew Redd}
## \title{Example documentation}
## \description{This is used to demonstrate the class system}
## \keyword{documentation}
Something a bit more interesting

```r
hw <- function(){
  print("hello world")
}
documentation(hw) <- function_documentation('hw', title = 'the standard Hello world')
```

```
## \name{hw}
## \usage{hw()}
## \value{NA}
## \title{the standard Hello world}
```
Can your documentation do this?

```r
make_hello <- function(what = 'world'){
  documented(function(){
    print(paste("hello", what))
  }, name = paste0("hello_", what)
  , title = sprintf("the dynamic hello_%s example", what)
  , description = "Prints hello" %\% what %\%"to the console.
                 "Used to show the power of dynamic documentation."
}
hello_world <- make_hello()
hello_belgium <- make_hello('Belgium')

hello_belgium()

## [1] "hello Belgium"
```
```stopcode

## \name{hello_Belgium}
## \usage{hello_Belgium()}
## \value{NA}
## \title{the dynamic hello_Belgium example}
## \description{Prints hello Belgium to the console.
Used to show the power of dynamic documentation.}
```

```
Test extraction

Example R code in "./R/hello.R"

```r
hello_belgium <- make_hello('Belgium')
if(FALSE){
    expect_output(hello_belgium(), "hello Belgium")
}
```
We run extract_tests() then ...
in “./tests/testthat/test-hello.R” we will have

```r
#! This file was automatically produced by documentation on 2017-07-05 11:00:00
#! changes will be overwritten.
context('tests extracted from file `./R/hello.R`')
test_that("hello_belgium", {
  #@testing
  expect_output(hello_belgium(), "hello Belgium")
})
```
Something still to finish

Relational Tags

```r
hw <-
function( greet = "Hello" #< A greeting
 , who   = "World"   #< who to greet.
 ){
  #! A more complicated hello world
  print(paste(greet, who))
  #< Called for the side effect of printing,
  #^ but returns the pasted arguments invisibly.
}
```
State in summary

<table>
<thead>
<tr>
<th>Documentation Classes</th>
<th>done..ish</th>
</tr>
</thead>
<tbody>
<tr>
<td>parsetools</td>
<td>functionally Complete</td>
</tr>
<tr>
<td>test extraction</td>
<td>done</td>
</tr>
<tr>
<td>Relational Tags</td>
<td>Infrastructure in place</td>
</tr>
<tr>
<td></td>
<td>Needs to integrate with Roxygen</td>
</tr>
<tr>
<td></td>
<td>Lots to do</td>
</tr>
</tbody>
</table>

**Input Forms**

<table>
<thead>
<tr>
<th>Rd</th>
<th>in place, could use improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>markdown</td>
<td>lots to do</td>
</tr>
</tbody>
</table>

**Output Forms**

| Others                  | not started                      |
how people can help and get involved.

- Make a github request for a feature.
- Code one of the missing functionalities
  - code it up
  - test the heck out of it
  - make a pull request showing

https://github.com/RDocTaskForce/

- Join the mailing list

https://lists.r-consortium.org/mailman/listinfo/rconsortium-wg-dtf
Acknowledgements

- Special Thanks to the R Consortium for the funding for this project
- Thanks to the members and participants of the RDTF.
  - Richard Calaway
  - Sarah Goslee
  - Michael Lawrence
  - Martin Maechler
  - Duncan Murdoch
  - Kirill Muller