Nils Gehlenborg (nils@hms.harvard.edu)
16 February 2011

Nozzle.R1 R Package API Overview

Nozzle reports consist of *structural elements*, such as sections, subsections, lists, etc. and *content elements*, such as paragraphs, figures, tables, etc.. Text in content elements can be formatted with a range of different styles, such as strong or emphasized. Figures, table and citations are numbered automatically and can be referred to in the text.

Conceptually, reports are assembled from the bottom up, i.e. content elements are created first (e.g. a paragraph) and then attached to structural elements (e.g. a subsection), which in turn can be attached to higher level structural elements (e.g. a section) until the full report has been assembled. In practice, these steps can also be mixed.

Sounds complicated? Here is an example of an (incomplete) report:

```
library( "Nozzle.R1" );

# content elements
p <- newParagraph( "This is some ", asEmph( "emphasized" ), " text." );
f <- newFigure( imageFile, imageFileHighRes, "This is the caption with a link to ",
          asLink( url="http://www.google.com", "Google" ), "." );
t <- newTable( tableData, tableFile, "This is a table showing the data of ",
          asReference( f ), "." )

# structural elements
s <- newSubSection( "My Result" );
s <- addTo( s, p, f, t ); # add paragraph, figure and table

r <- newReport( "My Report" );
r <- addToResults( r, s ); # add subsection to results section (created by default)

writeReport( r );
```

All functions used here are described in the following preview of the Nozzle API.

Functions

Notes on nomenclature: Arguments listed in **bold** are required, arguments listed in *italics* are optional and have default values.

1. Generating Structural Elements

---

newReport(**...**)

This function initializes a new report. This step is not required until elements are to be attached to the report.

A Firehose Report contains the following default sections: “Overview”, with subsections “Introduction” and “Summary”, “Results”, and “Methods & Data”, with optional subsections “Input” and “References”.

Arguments:

- **...** - Zero or more strings and scalar variables that will be concatenated and used as title for the report.

Value:

A Report element.

---

newSection(**...**, *protection*=PROTECTION.PUBLIC)
newSubSection(**...**, *protection*=PROTECTION.PUBLIC)
newSubSubSection(**...**, *protection*=PROTECTION.PUBLIC)

These functions create section, subsection or subsubsection elements.

Arguments:

- **...** - Zero or more strings and scalar variables that will be concatenated and used as title for the section, subsection or subsubsection.
- *protection* - Protection level for the new element.

Value:

A Section, Subsection or SubSubSection element.

newList(**...**, *isNumbered*=FALSE, *protection*=PROTECTION.PUBLIC)

This function creates a list, either numbered or with bullets.

Arguments:

- **...** - Zero or more Paragraph, List, Result elements (see below) that make up the list.
- *isNumbered* - If TRUE the list items will be numbered with arabic numerals. If FALSE the list items will have a bullet.
- *protection* - Protection level for the new element.

Value:

A List element.

newResult(**...**, *isSignificant*=FALSE, *protection*=PROTECTION.PUBLIC)

This function creates a new Result element. A Result element is a part of the report describes a particular finding in more detail, for instance, the computations that led to a particular *p*-value. In the report only this p-value (the “result summary”) would be be shown per default. By clicking on the associated result summary - the *p*-value in this case - the full result will be shown. A Result element may contain almost any Nozzle element besides the Report element and further Result elements. Result elements can be marked as significant, which will highlight them in the report.

Arguments:

- **...** - Zero or more strings and scalar variables that will be concatenated and used as summary for the result. This could be a single number of a whole sentence. The summary may not contain any references to figures, tables or citations or other results.
- *isSignificant* - If TRUE the result will be highlighted and add to the total count of significant results in the report.
- *protection* - Protection level for the new element.

Value:

A Result element.

2. Generating Content Elements

newParagraph(**...**, *protection*=PROTECTION.PUBLIC)

This function creates a new Paragraph element, which represents a paragraph of text.

Arguments:

- **...** - Zero or more strings and scalar variables that will be concatenated and used as text for the paragraph.
- *protection* - Protection level for the new element.

Value:

A Paragraph element.

---

newParameterList(**...**, *separator*=" = ", *protection*=PROTECTION.PUBLIC)

This function creates a new ParameterList element. ParameterList elements are essentially lists of key-value pairs. They can be used, for instance, to summarize the settings for a particular method or algorithm.

Arguments:

- **...** - An even number greater than two of strings or scalar variables. Formatting needs to be supplied here (see asParameter, asValue, asFilename below).
- *separator* - String used to separate keys from values.
- *protection* - Protection level for the new element.

Value:

A ParameterList element.

---

newFigure(**file**, **...**, *fileHighRes*=NA**,** *protection*=PROTECTION.PUBLIC)

This function creates a Figure element.

Arguments:

- **file** - Path and filename to the image file in a file format that is natively supported by web browsers, i.e. PNG, JPG or GIF.
- **...** - Zero or more strings and scalar variables that will be concatenated and used as caption for the figure. This may also contain references and Result elements.
- *fileHighRes* - Path and filename to a high-resolution version of the image. File formats such as PDF are permitted.
- *protection* - Protection level for the new element.

Value:

A Figure element.

---

newTable(**table**, **...**, *file*=NA**,** *protection*=PROTECTION.PUBLIC)

This function creates a Table element.

Arguments:

- **table** - A matrix or data.frame R data structure containing the table that should be included in the report. The table header will be created from the column names of the table data structure. The number of lines should be limited to no more than 30 unless there are good reasons to include more.
- **...** - Zero or more strings and scalar variables that will be concatenated and used as caption for the table. This may also contain references and Result elements.
- *tableFilename* - Path and filename to a file containing the full table. Ideally, this should be a tab-delimited text file with a single line header.
- *protection* - Protection level for the new element.

Value:

A Table element.

---

newCitation(*authors*="", **title**, *publication*="", *issue*="", *number*="", *pages*="", *year*="", *url*="")
newJournalCitation(**authors**, **title**, **publication**, **issue**, **number**, **pages**, **year**, *url*="")
newWebCitation(**title**, **url**)

The first function creates a Citation element. The latter two functions are convenience wrappers.

Arguments:

- *authors* - List of authors. The preferred format is “F1. M1. Last1, F2. M2. Last2, ..., F*n.* M*n.* Last*n*”.
- **title** - Title of the paper, book, website, etc..
- *publication* - Title of the publication containing the reference, e.g. a journal title for a paper or book for a book chapter.
- *issue* - Issue or volume of the publication.
- *number* - Number of the publication. Could also be “” if no number is available.
- *pages* - Pages of the publication. The preferred format is “firstpage-lastpage”.
- *year* - Year of the publication.
- *url* - Link to the publication.

Value:

A Citation element.

Examples:

```
citation <- newCitation( authors="Nils Gehlenborg", title="The Nozzle Report Package",
year="2011", url="http://www.gehlenborg.com" ); # e.g. for unpublished

webCitation <- newWebCitation( "The Cancer Genome Atlas Website", "http://
tcga.cancer.gov" );

journalCitation <- newJournalCitation( "Nils Gehlenborg", "Nozzle: An R Package for
Report Generation", "Yet Another Bioinformatics Journal", "14", "44", "783-789",
"2011", "http://wwww.wikipedia.org" );
```

3. Formatting Content

---

asReference(**target**)

This function returns a string representation of a reference to the Table, Figure or Citation element passed into it.

Arguments:

- **target** - A Table, Figure or Citation element that should be refer to.

Value:

A string representing a reference to either a table (e.g. “Table 1”), figure (e.g. “Figure 1”) or citation (e.g. “[4]”).

---

asSummary(**result**)

This function returns a string representation of the summary of the Result element passed into this function as an argument and as a side-effect inserts the Result element into the report right after the current structural element.

Arguments:

- **result** - A result element that has not yet been inserted into the report.

Value:

A string representation of the summary of the Result element.

---

asLink(**url**, ...)

This creates a link to the given URL and returns it as a string representation.

Arguments:

- **url** - The target URL.
- *...* - Zero or more strings and scalar variables that will be concatenated and used as text for the link. This may not contain references and Result elements. If no text is provided the URL itself will be used as text for the link.

Value:

A string representation of the link to the URL.

---

asStrong(**...**)
asEmph(**...**)
asParameter(**...**)
asValue(**...**)
asFilename(**...**)
asCode(**...**)

These functions format the concatenation of the list of strings and scalar variables that is passed into them.

- strong: Information that should be highlighted because it is important and should not be missed.
- emph: Information that should be emphasized.
- parameter: Names of variables that are arguments for a method or algorithm, e.g. the “k” parameter in a k-means clustering.
- value: Values of variables, e.g. the value that “k” is set to.
- filename: Filenames.
- code: Source code or pseudo code used to describe an algorithm. Good reports will avoid this.

Arguments:

- **...** - Zero or more strings and scalar variables that will be concatenated and formatted together.

Value:

A string representation of the concatenated and formatted arguments.

4. Assembling Reports

addTo(**parent**, **...**, row=NA, column=NA)

This function attaches one or more elements to a parent element and is used to assemble the report from the bottom up. For instance, a Paragraph, a Table and a Figure element could be attached to a SubSection element, which in turn can be attached to a Section element.

Arguments:

- **parent** - The parent element. This must be a structural element, i.e. a Report, Section, SubSection, SubSubsection, List or Result element.
- **...** - Zero or more elements that should be attached to the parent element. Not all combinations are allowed, for instance, a Section element cannot be attached to a SubSection element.

Value:

An updated version of the parent element that contains the new child elements. This element can then be passed to further calls of this function or the addToXXX(...) functions to assemble the report.

addToResults(**report**, **...**)
addToMethods(**report**, **...**)
addToIntroduction(**report**, **...**)
addToSummary(**report**, **...**)
addToInput(**report**, **...**)
addToReferences(**report**, **...**)

These functions attach one or more elements directly to one of the predefined report sections (Methods, Results) or report subsections (Introduction, Summary, Input, References).

Arguments:

- **report** - The report element.
- **...** - Zero or more elements that should be attached to the corresponding part of the report.

Value:

An updated version of the report that contains the new elements. This element can then be passed to further calls of these functions or addTo(...) to assemble the report.

5. Exporting Reports

writeReport(**report**, *filename*=DEFAULT.REPORT.FILENAME, *debug*=FALSE,
level=PROTECTION.PUBLIC)

This function writes the report to a file as an HTML fragment for further processing by Firehose. Before the report is written, the optional, predefined “Methods & Data” subsections “Input” and “References” are removed from the report if no elements have been attached to them.

Arguments:

- **report** - The report element.
- *filename* - The filename for the report. The default is “report.html”.
- *debug* - If set to TRUE, a “stand-alone” version of the report will be generated that will work outside Firehose. This is for testing purposes only and this argument must be set to FALSE in production code.
- *level* - Determines which elements should be included in the report. If set to PROTECTION.PUBLIC, only elements marked with PROTECTION.PUBLIC will be included. If set to PROTECTION.TCGA, elements marked with PROTECTION.PUBLIC and PROTECTION.TCGA will be included. If set to PROTECTION.PRIVATE all elements will be included.

Value:

None.