

# Package ‘xlsxSimple’

August 29, 2018

**Title** A Simple Interface to the xlsx Package

**Version** 2.0.0

**Description** A simple way to save tabular data and plots as XLSX workbooks.

**Depends** R (>= 3.5.1)

**Imports** xlsx (>= 0.6.1)

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.0.1

## R topics documented:

xlsxSimple-package . . . . .	1
addHyperLinks . . . . .	2
createWorkbook . . . . .	3
getColumnValue . . . . .	4
saveWorkbook . . . . .	4
toSheet . . . . .	5
<b>Index</b>	<b>8</b>

---

xlsxSimple-package	<i>xlsx Simple</i>
--------------------	--------------------

---

## Description

A simple interface to the **xlsx** package, simplifying the task of sending a data frame or a graph to Excel.

## Details

This requires the **xlsx** package and the Java run time environment. Which means that it will only run with the version of Java that you have installed and running. You may need to install with one of:

- `install.packages(file.choose(), INSTALL_opts="--no-multiarch")`
- `install(path, args = "--no-multiarch")`
- `install_github("But-I-Digress/xlsxSimple", args = "--no-multiarch")`

## Changes With This Release

- The “book” parameter has been renamed “workbook” to more closely follow the usage in the **xlsx** package and has been made optional. When **xlsxSimple** is loaded a default workbook is created. This workbook is then used wherever the “workbook” parameter is omitted.
- Error trapping has been added for the toSheet method. If there is no defined method for the object that you are trying to send to a sheet then an error is now thrown. *Id est*, toSheet(NULL).
- saveWorkbook no longer requires “file.path”, “title” or “subject” parameters. If “file.path” is omitted then the workbook will be saved with the same path as the script but with an XLSX extension.

## Author(s)

**Maintainer:** Charles Scheid <chharleshscheid@gmail.com>

## References

<https://poi.apache.org/apidocs/org/apache/poi/>

## See Also

[xlsx](#), [rJava](#)

## Examples

```
## Not run:
createWorkbook() -> book
toSheet(book, mtcars, "mtcars") -> sheet
addHyperLinks(sheet, "Google", "http://google.com");
saveWokbook(book, start = TRUE)

library(ggplot2)
(mtcars %>% ggplot(aes(mpg, cyl)) + geom_point()) %>% toSheet
saveWokbook(start = TRUE)

## End(Not run)
```

---

addHyperLinks

*Add Hyper Links*

---

## Description

addHyperLinks adds hyperlinks to an **xlsx** or **xlsxSimple**, Excel worksheet.

## Usage

```
addHyperLinks(sheet, col.name, urls)

## S4 method for signature 'jobRef'
addHyperLinks(sheet, col.name, urls)
```

**Arguments**

sheet	An <b>xlsx</b> or <b>xlsxSimple</b> , worksheet.
col.name	A character scalar, the name of the column where the hyperlinks are to be added. The column will be created if it doesn't exist.
urls	A character vector, the list of the hyperlink targets.

**Value**

The worksheet invisibly.

**Examples**

```
## Not run:
toSheet(mtcars) -> sheet
addHyperLinks(sheet, "Google", "http://google.com")

## End(Not run)
```

---

createWorkbook	<i>Create an Empty Workbook object</i>
----------------	--

---

**Description**

Create an Empty Workbook object

**Usage**

```
createWorkbook(type = "xlsx")
```

**Arguments**

... Arguments to pass through to `xlsx::createWorkbook`

**Value**

The worksheet object.

**See Also**

[createWorkbook](#)

**Examples**

```
## Not run:
createWorkbook() -> book

## End(Not run)
```

---

getColumnValue	<i>Get Column Value</i>
----------------	-------------------------

---

### Description

Gets the values from column in an xlsx or xlsxsimple, Excel worksheet.

### Usage

```
getColumnValue(sheet, col.name)
```

```
## S4 method for signature 'jobRef'
getColumnValue(sheet, col.name)
```

### Arguments

sheet	An xlsx or xlsxsimple worksheet.
col.name	A character scalar, the name of the column to return.

### Value

A vector.

### Examples

```
## Not run:
sheet <- toSheet(mtcars)
getColumnValue(sheet, "mpg")

## End(Not run)
```

---

saveWorkbook	<i>Save Work Book</i>
--------------	-----------------------

---

### Description

saveWorkbook saves an Excel workbook on the file system, setting the metadata. Optionally, the file is opened in Excel.

### Usage

```
saveWorkbook(workbook = getWorkbook(), file.path = thisFile(), title = NA,
  subject = NA, creator = Sys.getenv("USERNAME"), start = FALSE)
```

**Arguments**

workbook	An optional xlsxSimple workbook.
file.path	An optional character scalar, the path and filename for the file.
title	An optional character scalar, the title of the workbook.
subject	An optional character scalar, the subject of the workbook.
creator	An optional character scalar, the creator of the workbook. Defaults to the logged-on user name.
start	An optional logical scalar, should the workbook be started in Excel.

**Value**

A character scalar, the path of the file, invisibly.

**Note**

If the file.path does not end with “.xlsx” then the file extension will be set to “.xlsx”. If file.path is omitted then the script file path will be used, but that extension substituted. But an error is thrown if file.path is omitted in interactive mode.

**Examples**

```
book <- createWorkbook()
saveWorkbook(book, "foo.xlsx", "Foo", "Examples", start=TRUE)
```

---

toSheet	<i>Send something to a sheet</i>
---------	----------------------------------

---

**Description**

toSheet sends tabular data or an image to a new xlsx work sheet.

**Usage**

```
toSheet(x, workbook, title, ...)

## S3 method for class 'data.frame'
toSheet(d, workbook = getWorkbook(),
  sheetName = getSheetName(workbook), auto.filter = TRUE,
  freeze.pane = TRUE, row.names = FALSE, startRow = 1, ...)

## S3 method for class 'tbl_df'
toSheet(d, workbook = getWorkbook(),
  sheetName = getSheetName(workbook), ...)

## S3 method for class 'tbl'
toSheet(d, workbook = getWorkbook(),
  sheetName = getSheetName(workbook), ...)
```

```
## S3 method for class 'matrix'
toSheet(d, workbook = getWorkbook(),
  sheetName = getSheetName(workbook), row.names = TRUE, ...)

## S3 method for class 'character'
toSheet(x, workbook = getWorkbook(),
  sheetName = getSheetName(workbook), landscape = TRUE)

## S3 method for class 'ggplot'
toSheet(p, workbook = getWorkbook(),
  sheetName = getSheetName(workbook), filename = tempfile(fileext =
  ".jpeg"), landscape = TRUE, height = if (landscape) 7 else 9.7,
  width = if (landscape) 8.5 else 5.99, ...)
```

### Arguments

<code>x</code>	A data frame, plot or path to a saved plot.
<code>workbook</code>	An optional xlsx work book.
<code>title</code>	Character, an optional title for the new sheet.
<code>...</code>	Other parameters to pass to <code>addDataFrame</code> .
<code>auto.filter</code>	A boolean, should the columns be filtered.
<code>freeze.pane</code>	A boolean, should the top row be frozen.
<code>row.names</code>	A boolean, should the row names of <code>x</code> be written along with <code>x</code> to the file.
<code>startRow</code>	A scalar integer, the row to start the data frame.
<code>landscape</code>	A boolean, should the sheet be oriented horizontally.
<code>filename</code>	Optional file path and name for the temporary file for the image of the plot.
<code>height</code>	Numeric, optional height of the image in inches.
<code>width</code>	Numeric, optional width of the image in inches.

### Value

A worksheet object, invisibly.

### Methods (by class)

- `data.frame`: Send a data frame to a work sheet.
- `tbl_df`: Send a tibble to a work sheet.
- `tbl`: Send a data frame to a work sheet.
- `matrix`: Send a matrix to a work sheet.
- `character`: send a saved image to a work sheet.
- `ggplot`: Send a ggplot object to a work sheet.

### Examples

```
## Not run:
book <- createWorkbook()
toSheet(mtcars, book, "mtcars")

library(ggplot2)
```

```
p <- ggplot(mtcars, aes(mpg))  
p <- p + geom_bar()  
toSheet(p)  
  
## End(Not run)
```

# Index

`addHyperLinks`, [2](#)  
`addHyperLinks`, `jobjRef`-method  
    (`addHyperLinks`), [2](#)  
  
`createWorkbook`, [3](#), [3](#)  
  
`getColumnValue`, [4](#)  
`getColumnValue`, `jobjRef`-method  
    (`getColumnValue`), [4](#)  
  
`rJava`, [2](#)  
  
`saveWorkbook`, [4](#)  
  
`toSheet`, [5](#)  
  
`xlsx`, [2](#)  
`xlsxSimple` (`xlsxSimple-package`), [1](#)  
`xlsxSimple-package`, [1](#)