

Package ‘Rbonaut2’

December 4, 2015

Type Package
Title CLIP2's Rbonaut
Version 0.2
Date 2015-12-04
Author Cavorit
Maintainer Harald Fiedler <harald.fiedler@cavorit.de>
Depends data.table
Description The CLIP2-Version of CLIP1's Rbonaut-Package
License This package is private and internal of Cavorit Consulting GmbH
LazyData TRUE

R topics documented:

Rbonaut2-package	1
getAdrWAlsListe	2
getFirstAdrW	2
getNachname	3
getSessionTimeStamp	4
getVorname	4
isMultiTarget	5
istFormatNachnameKommaVorname	6
readItemBank	6
SQL2DF	7
Index	8

Rbonaut2-package	<i>Rbonaut2</i>
------------------	-----------------

Description

CLIP2-Paket

Author(s)

Harald Fiedler (c) Cavorit

getAdrWAlsListe	<i>getAdrWAlsListe</i>
-----------------	------------------------

Description

Hilfsfunktion von SQL2DF()

Usage

```
getAdrWAlsListe(adrW)
```

Arguments

adrW	character
------	-----------

Details

In den DB-Abfragen von CGoal findet sich die Variable adrW für die Zielfelder. Beim Umstellen von Single-Target auf Multi-Target wurde aus einer Zahl nun einen String, der einen JSON-Vektor darstellt. Wenn also Früher nur das Zielfeld 7 angegeben war, kann bei Multitarget nun der Ausdruck "7, 2, 21" angegeben sein. Die hier vorliegende Funktion arbeitet Vektorwertig und macht beispielsweise aus den Tabelleneinträgen c("1, 2, 3, 4", "11, 12, 13, 14") eine List der Form list(c(1, 2, 3, 4), c(11, 12, 13, 14))

Value

list mit numerischen Elementen

Author(s)

Harald Fiedler

Examples

```
adrW <- c("{1, 2, 3, 4}", "{11, 12, 13, 14}")
getAdrWAlsListe(adrW = adrW)
```

getFirstAdrW	<i>getFirstArdW</i>
--------------	---------------------

Description

Hilfsfunktion von SQL2DF(): Gibt erstes Ziel in adrW im numerischen Format

Usage

```
getFirstAdrW(adrW)
```

Arguments

adrW	character Vektor, etwa c("2, 4, 5", "12,19", "4")
------	---

Details

Bei der Umstellung von Single-Target auf Multi-Target wurden die Einträge in der FBN-Datenbank stark abgeändert. Wo früher beispielsweise eine Zahl 7 für das Zielfeld mit der Adresse 7 stand, ist nun "3, 5, 15" ein String, der die unterschiedlichen Zielfelder darstellt. Unabhängig davon, ob in adrW ein multiTarget oder singleTarget-Design hinterlegt wird, liefert diese Funktion nur das erste Ziel zurück, und zwar als Zahl.

Value

numeric

Author(s)

Harald Fiedler

Examples

```
getFirstAdrW(adrW=c("{2, 4, 5}", "{12, 19}", "{4}"))
```

getNachname

getNachname

Description

Hilfsfunktion von SQL2DF(): Gibt aus einem Spielernamen den Vornamen

Usage

```
getNachname(Spielername)
```

Arguments

Spielername character Vektor von beliebiger Länge

Details

Spielernamen können in SQL-Abfragen des FBN beispielsweise "Dogan, Isa" sein. Es wird "Isa" zurückgegeben.

Value

character Vektor der gleichen Länge wie der an die Funktion übergebene Vektor

Author(s)

Harald Fiedler

Examples

```
Spielername <- c("Fiedler, Harald", "Mayer, Jan", "A-Team")
getNachname(Spielername = Spielername)
```

getSessionTimeStamp	<i>getSessionTimeStamp</i>
---------------------	----------------------------

Description

Hilfsfunktion von SQL2DF(): ermittelt Sessionstart

Usage

```
getSessionTimeStamp(DatumString)
```

Arguments

DatumString String, etwa "2015-08-27 18:59:25.328383+02"

Details

Macht aus 2015-08-27 18:59:25.328383+02 den String 18:59:25

Value

Ein String, etwas "18:59:25"

Author(s)

Harald Fiedler

Examples

```
DatumString <- c("2015-08-27 18:59:25.328383+02", "2015-08-27 18:59:25.328383+02", "2015-08-27 18:59:25.328383+02")
```

getVorname	<i>getVorname</i>
------------	-------------------

Description

Hilfsfunktion von SQL2DF(): Gibt aus einem Spielernamen den Vornamen

Usage

```
getVorname(Spielername)
```

Arguments

Spielername character Vektor von beliebiger Länge

Details

Spielernamen können in SQL-Abfragen des FBN beispielsweise "Dogan, Isa" sein. Es wird "Isa" zurückgegeben.

Value

character Vektor der gleichen Länge wie der an die Funktion übergebene Vektor

Author(s)

Harald Fiedler

Examples

```
Spielername <- c("Fiedler, Harald", "Mayer, Jan", "A-Team")  
getVorname(Spielername = Spielername)
```

<code>isMultiTarget</code>	<i>isMultiTarget</i>
----------------------------	----------------------

Description

Hilfsfunktion von SQL2DF(): ist adrW multitargetting?

Usage

```
isMultiTarget(adrW)
```

Arguments

adrW character Array, etwa `c("22, 33, 44, 55", "11, 22222, 11111", "99")`

Details

Sagt, ob 11, 21, 16 oder 23 unter adrW abgespeichert wurde

Value

boolescher Vektor

Author(s)

Harald Fiedler

```
istFormatNachnameKommaVorname
      istFormatNachnameKommaVorname
```

Description

Hilfsfunktion von SQL2DF()

Usage

```
istFormatNachnameKommaVorname(Spielername)
```

Arguments

Spielername String

Details

In den SQL-Auszügen des FBN finden sich Spielername vom Format "Fiedler, Harald", aber auch "A_TEST_Forschung". Die Funktion testet komponentenweise, ob zwei Strings kommasetrennt gepastet sind.

Value

Boolescher Wert, der angibt, ob das Format Name, Vorname (mutmaßlich) vorliegt

Author(s)

Harald Fiedler

Examples

```
Spielername <- c("Fiedler, Harald", "Mayer, Jan", "A-Team")
istFormatNachnameKommaVorname(Spielername = Spielername)
```

```
readItemBank                      readItemBank
```

Description

Liest die ItemBank ein

Usage

```
readItemBank(file = NA)
```

Arguments

Pfad character der Länge 1, der den Pfadname zu einer .csv-Datei darstellt. Die Datei muss eine gültige ItemBank im Sinne des 4-PL-Modells sein. Als Default-Wert für den Pfad fungiert ein Pfad zu einer Pakte-Datei, die in der Lib installiert wurde (was der eigentliche Clou dieser Funktion ist).

Details

Es wird die ItemBank im 1:4PL-Modell eingelesen.

Value

data.frame für das 4PL-Modell

Author(s)

Harald Fiedler

Examples

```
ItemBank <- readItemBank()
head(ItemBank)
```

SQL2DF

SQL2DF

Description

SQL zu data.frame

Usage

```
SQL2DF(SQL)
```

Arguments

SQL data.frame, dass per read.csv eingelesen wurde

Details

Mit shinySQL erhalten wir von CGoal SQL-Abfragen händisch als .csv-Files zurück. Diese werden in ein data.frame umgewandelt

Value

data.frame

Author(s)

Harald Fiedler

Examples

```
message("Ich lade den R-Paket-internen RAW-Datensatz: Footbonaut_Datenabfrage_RicoWehrle.csv")
Pfad <- system.file("extdata", package="Rbonaut2", "Footbonaut_Datenabfrage_RicoWehrle.csv")
SQL <- read.csv2(file=Pfad, sep = ",", stringsAsFactors = FALSE, encoding = "utf8")
DF <- SQL2DF(SQL=SQL)
head(DF)
```

Index

*Topic **package**

Rbonaut2-package, [1](#)

getAdrWAlisListe, [2](#)

getFirstAdrW, [2](#)

getNachname, [3](#)

getSessionTimeStamp, [4](#)

getVorname, [4](#)

isMultiTarget, [5](#)

istFormatNachnameKommaVorname, [6](#)

Rbonaut2 (Rbonaut2-package), [1](#)

Rbonaut2-package, [1](#)

readItemBank, [6](#)

SQL2DF, [7](#)