R documentation

of ‘allWorklog2.Rd’ etc.

October 2, 2016

`allWorklog2` *R style wich returns the worklog of all the users.*

Description

R style wich returns the worklog of all the users.

Usage

```
allWorklog2(worklogsMatrix, start, end)
```

`allWorklog` *Returns the worklog of all the users.*

Description

Returns the worklog of all the users.

Usage

```
allWorklog(worklogsMatrix, start, end)
```

buildDFissues *Creates a data frame with basic data of the sprint's users.*

Description

Creates a data frame with basic data of the sprint's users.

Usage

```
buildDFissues(issues)
```

Arguments

issues	A list with the json of the issues in the sprint

Value

Data frame with columns "Issue key", "assignee key", "assignee name", "status", "type", "original estimate (h)", "time spent (h)", "difference (h)", "story points", "story points filled", "parent", "time judgement"

canPingJira *Tests if the Jira ip address is reachable by ping.*

Description

Tests if the Jira ip address is reachable by ping.

Usage

```
canPingJira()
```

`getAllWorklogs` *Creates a matrix with the issues from a list.*

Description

Creates a matrix with the issues from a list.

Usage

```
getAllWorklogs(issueKeys)
```

Arguments

`issuesKeys`	A list of factors with the keys of the issues

Value

Data frame with columns "Issue key", "assignee key", "assignee name", "status", "type", "original estimate (h)", "time spent (h)", "difference (h)", "story points", "story points filled", "parent", "time judgement"

`getIndividualIssuesJson` *Creates a matrix with the issues from a list.*

Description

Creates a matrix with the issues from a list.

Usage

```
getIndividualIssuesJson(issueKeys)
```

Arguments

`issuesKeys`	A list of factors with the keys of the issues

Value

matrix with 7 columns "issue", "assignee", "expand", "id", "self", "field". The column "assignee" is the key. The column "fields" is the original json from the issue.

`getIssuesJson` *Issues download and processing. Get the json of jira users.*

Usage

```
getIssuesJson()
```

`getSprintData` *Gets the start and end date of the sprint.*

Description

Gets the start and end date of the sprint.

Usage

```
getSprintData()
```

`getUserChoices` *Returns a vector with key and displayName of users.*

Description

Returns a vector with key and displayName of users.

Usage

```
getUserChoices(u)
```

`getUserDF` *Returns a data.frame with key and displayName of users.*

Description

Returns a data.frame with key and displayName of users.

Usage

```
getUserDF(u)
```

`getUsers` *Get the users in the jira instance. Retrives the users from Jira.*

Usage

```
getUsers()
```

`havingIP` *Tests if the machine has a valid IP.*

Description

Tests if the machine has a valid IP.

Usage

```
havingIP()
```

`issueListWorklog` *Gets the worklogs of a issue list keys.*

Description

Gets the worklogs of a issue list keys.

Usage

```
issueListWorklog(assignee, worklogs, start, end)
```

Arguments

`worklogs`	A matrix with all the worklogs of the sprint
`issueList`	A vector with keys of lists

Examples

```
issueListWorklog(vector(”PGR-64”, ”CADMF-219”, ”BDATA-19”), wl)
```

jiraIp *Parameters needed by the application.*

Usage

```
jiraIp
```

Format

An object of class `character` of length 1.

jiraProject *jiraProject*

Description

Used only to get the users of the Jira

Usage

```
jiraProject
```

Format

An object of class `character` of length 1.

packageLoading *Prepare packages and general functions Packages installing and loading and the functions general to the application Package installing and loading.*

Usage

```
packageLoading()
```

`removeEmptyUsers` *Remove from the users.*

Description

Remove from the users.

Usage

```
removeEmptyUsers(issuesDF, usersDF)
```

Arguments

`x`	A number
`y`	A number

Value

The sum of `x` and `y`

Examples

```
add(1, 1)
add(10, 1)
```

`setStoryPoints` *Assigns the same story points to all subtask which does not have.*

Description

Assigns the same story points to all subtask which does not have.

Usage

```
setStoryPoints(issues)
```

`storyPoints` *Assigns the same story points to a subtask which does not have.*

Description

Assigns the same story points to a subtask which does not have.

Usage

```
storyPoints(key, i)
```

`testingname` *app.R. Some to does Packages importation*

`worklogOfAssignee` *Returns the worklog of a user.*

Description

Returns the worklog of a user.

Usage

```
worklogOfAssignee(assigneeKey, issuesMatrix, worklogsMatrix, start, end)
```

Index