Package 'BCPublish'

January 5, 2014

Type Package

Title What the package does (short line)

Version 1.0

Date 2014-01-05

Author Techburg JSC

Maintainer Techburg JSC <ha@techburgcorp.com>

Description An until package used to publish directly analysis and dataset by R command to the bouncingdata.com system

License BSD

LazyLoad yes

Depends R (>= 2.7.0), RCurl, rjson

Imports RCurl, rjson

R topics documented:

`BCPublish-package` *Bouncingdata.com remote publish utility*

Description

An until package used to publish directly analysis/dataset by R command to the bouncingdata.com system

Details

Package:	BCPublish
Type:	Package
Version:	1.0
Date:	2014-01-05
License:	BSD

Author(s)

Techburg JSC

Maintainer: Techburg JSC

`bcAnalysisPublish` *Publish an analysis to the bouncingdata.com system*

Description

Publish an R source file to the bouncingdata.com system. The bouncingdata.com will execute the uploaded R file to generate an analysis.

Usage

```
bcAnalysisPublish(filename
                , title = "Analysis Title"
                , description = "Analysis description"
                , public = "false"
                , base_uri = "http://bouncingdata.com")
```

Arguments

`filename`	Name of the R source file to be upload to generate analysis
`title`	Title of the analysis. Default "Analysis title"
`public`	Make the analysis public or not. Default "false", i.e. not public yet
`base_uri`	The url of the SERVER to which the R source file to be uploaded. Default "http://bouncingdata.com". For debug purpose only

Value

If this process terminates successfully, the URL of the newly created analysis will be shown to the user. If any error occur, a corresponding message will be returned to the user. For example, "Upload failed" in the case of failed uploading, "Execution failed" in the case of failed execution.

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (filename, title = "Analysis Title", description = "Analysis description",
    public = "false", base_uri = "http://bouncingdata.com")
{
    contentType <- "text/plain"
    file <- fileUpload(filename = filename, contentType = contentType)
    uri <- paste(base_uri, "/public/ide/publish", sep = "")
    curl <- getCurlHandle()
    style <- "HTTPPOST"
    .encoding <- "UTF-8"
    binary <- NA
    .checkParams <- TRUE
    user <- readline("Input user name:")
    password <- readline("Input password:")
    .opts <- curlOptions(url = uri, password = password)
    params = list(file1 = file, user = user, password = password,
        title = title, description = description, isPublic = public)
    retJSON <- postForm(uri = uri, .params = params, .opts = .opts,
        curl = curl, style = style, .encoding = .encoding, binary = binary,
        .checkParams = .checkParams)
    ret <- fromJSON(retJSON)
    if (ret["code"] != 0) {
        print(ret["message"])
    }
    else {
        print(paste("Analysis publishing succeeded. Your analysis url is ",
            base_uri, "/anls/", ret["message"], sep = ""))
    }
    return(ret)
  }
```

`bcDatasetPublish` *Publish an dataset to the bouncingdata.com system*

Description

Publish a data file in csv, txt or xls format to the bouncingdata.com system. The bouncingdata.com will parse the data schema and generate a dataset from the uploaded data.

Usage

```
bcDatasetPublish(filename
                , title = "Dataset Title"
                , description = "Dataset description"
```

```
                , tags = ""
                , public = "false"
                , base_uri = "http://bouncingdata.com")
```

Arguments

filename	Name of the data file
title	Title of the dataset. Default "Dataset Title"
description	Description of the dataset. Default "Dataset description"
tags	Tags will be attached to the dataset (the delimeter is comma). Default empty
public	Make the dataset public or not. Default "false", i.e. not public yet
base_uri	The url of the SERVER to which the data file to be uploaded. Default "http://bouncingdata.com". For debug purpose only

Value

If this process terminates successfully, the URL of the newly created analysis will be shown to the user. If any error occur, a corresponding message will be returned to the user: "Upload failed" in the case failed uploading, "Parsing failed" in the case of failed data schema parsing, "Insert to DB failed" in the case of failed dataset persisting.

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (filename, title = "Dataset Title", description = "Dataset description",
    tags = "", public = "false", base_uri = "http://bouncingdata.com")
{
    contentType <- "text/plain"
    file <- fileUpload(filename = filename, contentType = contentType)
    uri <- paste(base_uri, "/public/ide/publish_dataset", sep = "")
    curl <- getCurlHandle()
    style <- "HTTPPOST"
    .encoding <- "UTF-8"
    binary <- NA
    .checkParams <- TRUE
    user <- readline("Input user name:")
    password <- readline("Input password:")
    .opts <- curlOptions(url = uri, password = password)
    params = list(file1 = file, user = user, password = password,
        title = title, description = description, tags = tags,
        isPublic = public)
    retJSON <- postForm(uri = uri, .params = params, .opts = .opts,
        curl = curl, style = style, .encoding = .encoding, binary = binary,
        .checkParams = .checkParams)
    ret <- fromJSON(retJSON)
    if (ret["code"] != 0) {
```

```
        print(ret["message"])
    }
    else {
        print(paste("Dataset publishing succeeded. Your dataset url is ",
            base_uri, "/dataset/view/", ret["message"], sep = ""))
    }
    return(ret)
}
```