Package ‘BCPublish’

December 29, 2013

Type Package

Title Bouncingdata.com remote publish utility

Version 1.0

Date 2013-12-28

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

Imports RCurl

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:	2013-12-28
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"
                 , 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"
`description`	Description of the analysis. Default "Analysis description"
`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 returned to the user. If any error occur, a corresponding message will be returned to the user: "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",
    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:")
    params = list(file1 = file, user = user, title = title, description = description)
    ret <- postForm(uri = uri, .params = params, curl = curl,
        style = style, .encoding = .encoding, binary = binary,
        .checkParams = .checkParams)
    if (ret == "Upload failed" || ret == "Execution failed") {
        print(ret)
    }
    else {
        print(paste("Analysis publishing succeeded. Your analysis url is ",
            base_uri, "/anls/", ret, 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 = ""
                , 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
`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 returned 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 = "", 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:")
    params = list(file1 = file, user = user, title = title, description = description,
        tags = tags)
    ret <- postForm(uri = uri, .params = params, curl = curl,
        style = style, .encoding = .encoding, binary = binary,
        .checkParams = .checkParams)
    if (ret == "Upload failed" || ret == "Execution failed") {
        print(ret)
    }
    else {
        print(paste("Dataset publishing succeeded. Your dataset url is ",
            base_uri, "/dataset/view/", ret, sep = ""))
    }
    return(ret)
  }
```