Package 'rDVR'

August 29, 2016

Title The rDVR package allows you to start stop and save a video server from within R.

Version 0.1.1

Description The rDVR package allows you to start stop and save a video server from within R. It does this by way of a REST interface to a JAVA service. The jar binary relies on the screen recorder included in the Monte Media Library (CC BY 3.0 licence) developed by Werner Randelshofer (http://http://www.randelshofer.ch/monte/). The REST interface was modified from https://github.com/tuenti/VideoRecorderService which has an Apache licence.

URL http://johndharrison.github.io/rDVR/

BugReports https://github.com/johndharrison/rDVR/issues

Imports methods,RCurl

Depends R (>= 3.0.0)

Suggests knitr,RSelenium

VignetteBuilder knitr

License AGPL-3

LazyData yes

Author ``John Harrison" [aut, cre]

Maintainer ``John Harrison" <johndharrison0@gmail.com>

NeedsCompilation no

Repository CRAN

Date/Publication 2014-04-03 20:20:36

R topics documented:

checkForVServer *Check for Video Server binary*

Description

checkForVServer A utility function to check if the Video Server standalone binary is present.

Usage

```
checkForVServer(jarloc = NULL, update = FALSE)
```

Arguments

jarloc	A directory in which the binary is to be placed. Defaults to the /bin of rDVR package.
update	A boolean indicating whether to update the binary if it is present.

Detail

The Video Server java binary can be found at https://github.com/johndharrison/rDVR. If users would like to create their own please refe to documentation. This convience function downloads the standalone server and places it in the rDVR package directory bin folder by default.

Examples

```
## Not run:
checkForVServer()

## End(Not run)
```

rDVR-class *CLASS rDVR*

Description

class to communicate with the Video Server.

Details

This is class composing of methods to communicate with the Java Video Server. The methods start a video, save a running video and stop a running video without saving. Please note videos are currently limited to 10 minutes in length. If you want to change this limit you will need to compile a custom binary. Videos are encoded using the Apple QuickTime RLE codec. Currently to change the codec requires compiling a custom binary.

Fields

`remoteServerAddr`: Object of class `"character"`, giving the ip of the remote server. Defaults to localhost

`port`: Object of class `"numeric"`, the port of the remote server on which to connect to the Video Server.

`saveDir`: Object of class `"character"`, the location which the videos are saved. This is set by the Video Server. By default it is the OS temp directory. This option should be left NULL unless you set the directory when starting the Video Server.

Methods

`new(...)`: Create a new `rDVR` object. ... is used to define the appropriate slots.

`start(fileName, silent)`: Start a new video recording.

`fileName`: The filename by which to save your video. Defaults to Rtemp. This will be saved as RTemp.mov. YourNAME will be saved as YourNAME.mov.

`silent`: A boolean. If TRUE the method will run silently

`save(silent, replace)`: Save a currently running video to file. The file is given by the fileName used in the `start` method. The directory that the file is written to is the save directory stipulated when the Video Server was started.

`silent`: A boolean. If TRUE the method will run silently

`replace`: A boolean. If TRUE then rDVR will replace a file if it already exists. If false rDVR will append 'copy' to the saveFile name.

`stop(silent)`: Stops a currently running video. Using stop rather then save will result in the video being stopped but not saved.

`silent`: A boolean. If TRUE the method will run silently

`closeServer(silent)`: Stops a currently running Video Server.

`silent`: A boolean. If TRUE the method will run silently

`startVideoServer` *Start the Video Server.*

Description

`startVideoServer` A utility function to start the standalone video server.

Usage

```
startVideoServer(jarloc = NULL, savedir = NULL, port = NULL,
  distmode = FALSE, invisible = TRUE)
```

Arguments

jarloc	A directory in which the standalone video server binary is located. Defaults to the /bin of rDVR package.
savedir	A directory where the user would like videos saved to. If not declared it defaults to the temp folder (which varies depending on the OS).
port	The port on which the video server will listen. Defaults to 9998.
distmode	You can enable a "distribution" mode for the storage of recorded videos that will use the last two characters of the filename requested to save video to place it in a subfolder. By default, if you want to save a video with the name, say, videofile20987 you will end up with the file stored at: /path/to/dest/folder/videofile20987.mov. If you had enabled the distribution mode with distmode = TRUE the video would be stored at: /path/to/dest/folder/87/videofile20987.mov
invisible	Windows Only Show the video server in a shell. By default it is invisible with setting TRUE.

Detail

By default the binary is assumed to be in the rDVR package /bin directory.

Examples

```
## Not run:
startVideoServer()

## End(Not run)
```

Index