

Package ‘exams.mylearn’

October 13, 2022

Title Question Generation in the 'MyLearn' XML Format

Version 1.4

Description Randomized multiple-select and single-select question generation for the 'MyLearn' teaching and learning platform. Question templates in the form of the R/exams package (see <<http://www.r-exams.org/>>) are transformed into XML format required by 'MyLearn'.

License GPL-3

Depends R (>= 4.0.0)

Imports exams (>= 2.3-4), glue (>= 1.4.0), stringr (>= 1.4.0), stringi (>= 1.4.6), pkgbuild (>= 1.1.0), xml2 (>= 1.2.5)

Suggests knitr, rmarkdown

URL <https://github.com/hdarjus/exams.mylearn>

BugReports <https://github.com/hdarjus/exams.mylearn/issues>

Encoding UTF-8

LazyData true

LazyDataCompression bzip2

VignetteBuilder knitr

RoxygenNote 7.1.1

NeedsCompilation no

Author Darjus Hosszejni [aut, cre] (<<https://orcid.org/0000-0002-3803-691X>>)

Maintainer Darjus Hosszejni <darjus.hosszejni@wu.ac.at>

Repository CRAN

Date/Publication 2021-04-19 17:10:02 UTC

R topics documented:

exams.mylearn-package	2
example_paths	2
exams2mylearn	2
special_characters	4

Index**5**

exams.mylearn-package *Question Generation in the 'MyLearn' XML Format*

Description

Randomized multiple-select and single-select question generation for the 'MyLearn' platform. Question templates in the form of the R/exams package are transformed into XML format required by 'MyLearn'.

example_paths *Paths to Examples*

Description

Paths to Examples

Usage

```
example_paths()
```

Value

Named sequence of characters. Each character string points at an example file in the package.

Examples

```
ex_files <- example_paths()
ex_files
```

exams2mylearn *Exam Generation for the 'MyLearn' Platform*

Description

The 'MyLearn' distance learning and teaching platform has a special XML format. exams2mylearn transforms input files in the R/exams format to XML files and zips them. The resulting zip file can be directly uploaded to the 'MyLearn' platform.

Usage

```
exams2mylearn(
  filename,
  n,
  dir,
  tdir = NULL,
  name = NULL,
  outfile = NULL,
  dontask = !base::interactive(),
  distort.shortname = FALSE,
  ...
)
```

Arguments

filename	(character) absolute or relative path to the exercise template. Usually simply a file name pointing at a .Rmd file in the working directory
n	(integer) number of random variants to create
dir	(character) output directory, will be created if non-existent
tdir	(character, <i>optional</i>) temporary directory; will use tempdir() if unspecified
name	(character, <i>optional</i>) unique name prefix of temporary and output files, defaults to filename without the non-alphabetic characters
outfile	(character, <i>optional</i>) output file name (not a path), defaults to name.zip
dontask	(logical, <i>optional</i>) if TRUE and the output zip file exists then
distort.shortname	(logical, <i>optional</i>) should the shortname include a random ending? Defaults to FALSE
...	forwarded to exams2html

Value

If dir is invalid or unspecified, the function returns with an error. Otherwise the function produces a zip file in directory dir. The exact path to the zip file is returned invisibly.

Note

The development team has to turn on the upload functionality on a per course basis.

Examples

```
# Get the examples provided with the package
ex_files <- example_paths()
if (interactive()) {
  # Produce 40 exam questions in the current directory
  # using the example that contains a plot; the
  # output is "final_exam.zip", and we want to
```

```

# distort the shortname used by 'MyLearn'
exams2mylearn(ex_files["plot"], 40, dir = ".",
              outfile = "final_exam_question_1.zip",
              distort.shortname = TRUE)
}

# For special characters, please check
?special_characters

# Takes some time:
## Not run:
# Produce 500 exam questions in the current
# directory using a different example with more
# verbose output from exams::exams2html
exams2mylearn(ex_files["single_choice"], 500,
              dir = ".", name = "final_exam_question_1",
              verbose = TRUE)

## End(Not run)

```

special_characters *List of Special Characters that Users Have to Deal with Manually*

Description

Some special characters are used both in writing and in programming. Unfortunately, the HTML converter (the first step of the XML file creation) sometimes mis-interprets these unless special care is taken. In case of the characters listed in this data frame, the user has to manually substitute the code for the character in the Rmarkdown source.

Details

If something does not work that is not listed here, please try to find the the HTML code for your character and open an issue on GitHub for the maintainer with an example.

Examples

```

# Table containing the characters
data("special_characters")

# For usage examples, please open and see the file under
example_paths()["special-characters"]

```

Index

`example_paths`, [2](#)
`exams.mylearn` (`exams.mylearn-package`), [2](#)
`exams.mylearn-package`, [2](#)
`exams2mylearn`, [2](#)

`special_characters`, [4](#)