Package: ottrpal (via r-universe)

October 1, 2024

Type Package

Title Companion Tools for Open-Source Tools for Training Resources (OTTR)

Version 1.3.0

Description Tools for converting Open-Source Tools for Training Resources (OTTR) courses into Leanpub or Coursera courses. 'ottrpal' is for use with the OTTR Template repository to create courses.

License GPL-3

URL https://github.com/jhudsl/ottrpal

BugReports https://github.com/jhudsl/ottrpal/issues

Depends R (>= 3.5.0)

Imports bookdown, curl, dplyr, fs, httr, janitor, jsonlite, knitr (>= 1.33), magrittr, openssl, purrr, R.utils, readr, rmarkdown (>= 2.10), rprojroot, rvest, stringr, webshot2, xml2, yaml

Suggests testthat (>= 3.0.0), tibble, utils

Remotes jhudsl/cow

VignetteBuilder knitr

ByteCompile true

Encoding UTF-8

LazyData true

RoxygenNote 7.3.1

Config/testthat/edition 3

Repository https://jhudsl.r-universe.dev

RemoteUrl https://github.com/jhudsl/ottrpal

RemoteRef HEAD

RemoteSha 6fe8e03969e2d8b6b7d08ade3d4c78dbc6cc870b

Contents

authorize *Authorize R package to access the Google Slides API*

Description

This is a function to authorize the R package to access the Google Slides API interactively.

Usage

```
authorize(token = NULL, cache = FALSE, ...)
```

Arguments

token	An output from oauth2.0_token to set as the authentication token.
cache	Should the token be cached as an .httr-oauth file?
...	Additional arguments to send to oauth2.0_token

Value

OAuth token saved to the environment so the package can use the users' Google data

Examples

```
## Not run:

authorize()

## End(Not run)
```

auth_from_secret *Use secrets to authorize R package to access Google Slides API*

Description

This is a function to authorize the R package to access the Google Slides API. If no client.id and client.secret is provided, the package would provide predefined values.

Usage

```
auth_from_secret(access_token = NULL, refresh_token = NULL)
```

Arguments

access_token	Access token can be obtained from running authorize() interactively: token <- authorize(); token$credentials$access_token
refresh_token	Refresh token can be obtained from running authorize() interactively: token <- authorize(); token$credentials$refresh_token

Value

OAuth token saved to the environment so the package can use the users' Google data

Examples

```
## Not run:

token <- authorize()

auth_from_secret(
  token$credentials$access_token,
  token$credentials$refresh_token
)

## End(Not run)
```

`bad_quiz_path` *Path to bad example quiz*

Description

Path to bad example quiz

Usage

```
bad_quiz_path()
```

Value

The file path to an example bad quiz included in the package that will fail the quiz checks.

Examples

```
quiz_path <- bad_quiz_path()
```

`check_all_questions` *Check all quiz questions*

Description

Takes output from [ottrpal::parse_quiz] and runs checks on each question in a quiz by calling [ottrpal::check_question] for each question. First splits questions into their own data frame. Returns a list of messages/warnings about each question's set up.

Usage

```
check_all_questions(
  quiz_specs,
  quiz_name = NA,
  verbose = TRUE,
  ignore_coursera = TRUE
)
```

Arguments

quiz_specs	quiz_specs which is output from [ottrpal::parse_quiz].
quiz_name	The name of the quiz being checked.
verbose	Whether progress messages should be given.
ignore_coursera	Coursera doesn't like '¡ or ':' in the quizzes. Do not convert quizzes to coursera and ignore ! and : in question prompts that would not be allowed in Leanpub quizzes when converted to a Coursera quiz. Default is to ignore Coursera compatibility.

Value

A list of the output from [ottrpal::check_question] with messages/warnings regarding each question and each check.

Examples

```
## Not run:

# Using good quiz md example

quiz_path <- good_quiz_path()
good_quiz <- readLines(quiz_path)
good_quiz_specs <- parse_quiz(good_quiz)
good_quiz_checks <- check_all_questions(good_quiz_specs)

# Using bad quiz md example

bad_quiz <- readLines(bad_quiz_path())
bad_quiz_specs <- parse_quiz(bad_quiz)
bad_quiz_checks <- check_all_questions(bad_quiz_specs)

## End(Not run)
```

check_question *Check Quiz Question Set Up*

Description

Check quiz question set up to see if it is compliant with Leanpub and Coursera needs. Based off of [Markua guide](https://leanpub.com/markua/read#leanpub-auto-quizzes-and-exercises). Is called by [ottrpal::check_all_questions] and run for each question.

Usage

```
check_question(
  question_df,
  quiz_name = NA,
  verbose = TRUE,
  ignore_coursera = TRUE
)
```

Arguments

question_df	Which is an individual question's data frame after being parse from
quiz_name	The name of the quiz the question is from
verbose	Whether progress messages should be given
ignore_coursera	Coursera doesn't like '¡ or ':' in the quizzes. Do not convert quizzes to coursera and ignore ! and : in question prompts that would not be allowed in Leanpub quizzes when converted to a Coursera quiz. Default is to ignore Coursera compatibility

Value

A list of messages/warnings regarding each check for the given question.

Examples

```
## Not run:

# Use readLines to read in a quiz
quiz_path <- good_quiz_path()
quiz_lines <- readLines(quiz_path)

# Use group_split to get the questions
questions_df <- parse_quiz(quiz_lines)$data %>%
  dplyr::group_split(question)

good_quiz_checks <- check_question(questions_df[[2]])

## End(Not run)
```

check_quiz *Check Quiz*

Description

For a file path to a quiz, check whether it is properly formatted for Leanpub.

Usage

```
check_quiz(quiz_path, verbose = TRUE, ignore_coursera = TRUE)
```

Arguments

quiz_path	A file path to a quiz markdown file
verbose	print diagnostic messages? TRUE/FALSE
ignore_coursera	Coursera doesn't like '¡ or ':' in the quizzes. Do not convert quizzes to coursera and ignore ! and : in question prompts that would not be allowed in Leanpub quizzes when converted to a Coursera quiz. Default is to ignore Coursera compatibility

Value

A list of checks. "good" means the check passed. Failed checks will report where it failed.

Examples

```
## Not run:

# Take a look at a good quiz's checks:
quiz_path <- good_quiz_path()
good_checks <- check_quiz(quiz_path)

# Take a look at a failed quiz's checks:
quiz_path <- bad_quiz_path()
failed_checks <- check_quiz(quiz_path)

## End(Not run)
```

check_quizzes *Check all quizzes in a directory*

Description

Check the formatting of all quizzes in a given directory.

Usage

```
check_quizzes(
  quiz_dir = "quizzes",
  write_report = TRUE,
  verbose = TRUE,
  ignore_coursera = TRUE
)
```

Arguments

quiz_dir	A path to a directory full of quizzes that should all be checked with [ottrpal::check_all_quizzes].
write_report	TRUE/FALSE save warning report to a CSV file?
verbose	print diagnostic messages
ignore_coursera	Coursera doesn't like '¡ or ':' in the quizzes. Do not convert quizzes to coursera and ignore ! and : in question prompts that would not be allowed in Leanpub quizzes when converted to a Coursera quiz. Default is to ignore Coursera compatibility

Value

A list checks performed on each quiz

Examples

```
## Not run:


## Make a temporary quiz directory
quiz_dir <- dirname(good_quiz_path())

## Now check the quizzes in that directory
all_quiz_results <- check_quizzes(quiz_dir = quiz_dir)

## End(Not run)
```

check_quiz_attributes *Check Quiz Attributes*

Description

Check Quiz Attributes

Usage

```
check_quiz_attributes(quiz_specs, quiz_name = NULL, verbose = TRUE)
```

Arguments

quiz_specs	The output from [ottrpal::parse_quiz].
quiz_name	A character string indicating the name of the quiz being checked.
verbose	Would you like progress messages? TRUE/FALSE

Value

A logical

check_quiz_question_attributes *Check a question's attributes*

Description

This is ran automatically by [ottrpal::check_all_questions] for all questions. It checks that the attributes specified are accepted ones by Leanpub.

Usage

```
check_quiz_question_attributes(question_df, quiz_name = NULL, verbose = TRUE)
```

Arguments

question_df	a data.frame obtained from [ottrpal::parse_quiz_df] and dplyr::group_split(question).
quiz_name	inherited from parse
verbose	print diagnostic messages

Value

Will return a warning for any quiz question attributes used that are not supported.

convert_coursera_quizzes *Convert Leanpub md quiz to Coursera yaml quiz*

Description

Convert Leanpub md quiz to Coursera yaml quiz

Usage

```
convert_coursera_quizzes(
  input_quiz_dir = "quizzes",
  output_quiz_dir = "coursera_quizzes",
  verbose = TRUE
)
```

Arguments

input_quiz_dir	A path to a directory of leanpub formatted quiz md files. By default assumes "quizzes" and looks in current directory.
output_quiz_dir	A folder (existing or not) that the new coursera converted quizzes should be saved to. By default saves to "coursera_quizzes".
verbose	Would you like the progress messages: TRUE/FALSE?

Value

A folder of coursera ready quiz files saved to the output directory specified as a yamls.

Examples

```
# Set up a directory with a quiz in it for this example
tdir <- tempfile()
dir.create(tdir, showWarnings = FALSE, recursive = TRUE)

file.copy(
  from = good_quiz_path(),
  to = file.path(tdir, basename(good_quiz_path()))
)

# Provide path to directory of quizzes
convert_coursera_quizzes(tdir)

system("rm -r coursera_quizzes")
```

convert_quiz *Convert Leanpub md quiz to Coursera yaml quiz*

Description

Convert a Leanpub-formatted md quiz file to a Coursera-formatted yaml quiz file in preparation for uploading to Coursera.

Usage

```
convert_quiz(quiz_path, output_quiz_dir = dirname(quiz_path), verbose = TRUE)
```

Arguments

quiz_path	A path to a quiz .md file to be converted.
output_quiz_dir	An existing folder where you would like the new version of the quiz to be saved. Default is the directory of the quiz_path provided
verbose	Would you like the progress messages?

Value

A Coursera-ready quiz file saved to the output directory specified as a yaml.

Examples

```
## Not run:

quiz_path <- good_quiz_path()

# Provide path to quiz to convert
convert_quiz(quiz_path)

## End(Not run)
```

course_path *Find main course git directory*

Description

Find main course git directory

Usage

```
course_path(path = ".")
```

Arguments

path	Where to look for the file. By default looks in current directory.

Value

Returns the directory where the .git folder is contained.

---

course_to_book_txt *Create Book.txt file from files existing in quiz directory*

Description

Create Book.txt file from files existing in quiz directory

Usage

```
course_to_book_txt(
  path = ".",
  md_files = NULL,
  output_dir = "manuscript",
  quiz_dir = "quizzes",
  verbose = TRUE
)
```

Arguments

path	path to the bookdown or quarto course repository, must have a '_bookdown.yml' or '_quarto.yml' file
md_files	vector of file path of the md's to be included
output_dir	output directory to put files. It should likely be relative to path
quiz_dir	Where are the quizzes stored? Default looks for folder called "quizzes".
verbose	print diagnostic messages

Value

A list of quiz and chapter files in order in a file called Book.txt – How Leanpub wants it.

download_ottr_template *Download files from main OTTR_Template to test*

Description

Download files from main OTTR_Template to test

Usage

```
download_ottr_template(dir = "inst/extdata", type = "rmd")
```

Arguments

dir	What relative file path should the files be downloaded
type	Which OTTR repo are we downloading? Options are "rmd", "quarto", "rmd_website", "quarto_website"

Value

This downloads the main branch repo files from the respective repo for testing purposes

encrypt_creds_path *Get file path to an encrypted credentials RDS*

Description

Get file path to an encrypted credentials RDS

Usage

```
encrypt_creds_path()
```

encrypt_creds_user_path *Get file path to an default credentials RDS*

Description

Get file path to an default credentials RDS

Usage

```
encrypt_creds_user_path()
```

`extract_meta` *Extract meta fields from a tag*

Description

Extract meta fields from a tag

Usage

```
extract_meta(tags)
```

Arguments

`tags`	A single tag or vector of tags to extract the fields from.

Value

A named vector indicating the field and entry associated with it.

Examples

```
### Simple example
tag <- "{quiz, id: quiz_name_here, attempts: 10}"

# Extract metadata tags
meta <- extract_meta(tag)

### Example using a file
quiz_path <- good_quiz_path()
quiz_lines <- readLines(quiz_path)

# Put this in a data.frame so we can identify the content
quiz_df <- parse_quiz_df(quiz_lines)

# Extract the tags
tags <- quiz_df %>%
  dplyr::filter(type == "tag") %>%
  dplyr::pull("original")

# Extract metadata tags
meta <- extract_meta(tags)
```

`extract_object_id` *Extract Object IDs using Google Slides API*

Description

Performs a HTTP GET method to request the IDs of every slide in a Google Slides presentation. The ID of the first slide is always 'p'.

Usage

```
extract_object_id(
  slide_url,
  token = NULL,
  access_token = NULL,
  refresh_token = NULL
)
```

Arguments

`slide_url`	URL whose 'General access' is set to 'Anyone with the link'
`token`	OAuth 2.0 Access Token. If you don't have a token, use [authorize()] to obtain an access token from Google's OAuth 2.0 server.
`access_token`	Access token can be obtained from running authorize() interactively (token <- authorize(); token$credentials$access_token). This allows it to be passed in using two secrets.
`refresh_token`	Refresh token can be obtained from running authorize() interactively (token <- authorize(); token$credentials$refresh_token). This allows it to be passed in using two secrets.

Value

Character vector of object ID(s)

Examples

```
## Not run:
# First, obtain access token and store token for extract_object_id() to use
authorize(client_id = "MY_CLIENT_ID", client_secret = "MY_CLIENT_SECRET")
# Use stored token to talk to Google Slides API
extract_object_id(slide_url = "https://docs.google.com/presentation/d/1H5aF_ROKVxE-H
                               FHhoOy9vU2Y-y2M_PiV0q-JBL17Gss/edit?usp=sharing")

## End(Not run)
```

get_chapters *Make Leanpub file that has embed webpage of a chapter*

Description

Make Leanpub file that has embed webpage of a chapter

Usage

```
get_chapters(html_page = file.path("docs", "index.html"), base_url = ".")
```

Arguments

html_page	The file path of the rendered index.html file. It can be a url
base_url	The base url of where the chapters are published – the url to provide to the iframe in Leanpub e.g. https://jhudatascience.org/OTTR_Template/coursera

Value

A data.frame of the chapter urls and their titles that are to be ported to Leanpub. This can be passed to

get_gs_pptx *Download Google Slides pptx file*

Description

Download Google Slides pptx file

Usage

```
get_gs_pptx(id)
```

Arguments

id	Identifier of Google slides presentation, passed to get_slide_id

Value

Downloaded file (in temporary directory)

Note

This downloads presentations if they are public and also try to make sure it does not fail on large files

get_object_id_notes *Retrieve Speaker Notes and their corresponding Object (Slide) IDs from a Google Slides presentation*

Description

Google Slides API calls a presentation slide ID as an 'object ID'.

Usage

```
get_object_id_notes(slide_url)
```

Arguments

slide_url	URL whose 'General access' is set to 'Anyone with the link'

Value

Data frame of Object IDs and Speaker notes.

Examples

```
## Not run:
get_object_id_notes("https://docs.google.com/presentation/d/
                     1H5aF_ROKVxE-HFHhoOy9vU2Y-y2M_PiV0q-JBL17Gss/edit?usp=sharing")

## End(Not run)
```

get_pages_url *Retrieve pages url for a repo*

Description

Given an repository on GitHub, retrieve the pages URL for it.

Usage

```
get_pages_url(repo_name, git_pat = NULL, verbose = FALSE, keep_json = FALSE)
```

Arguments

repo_name	The full name of the repo to get bookdown chapters from. e.g. "jhudsl/OTTR_Template"
git_pat	If private repositories are to be retrieved, a github personal access token needs to be supplied. If none is supplied, then this will attempt to grab from a git pat set in the environment with usethis::create_github_token(). Authorization handled by get_git_auth
verbose	TRUE/FALSE do you want more progress messages?
keep_json	verbose TRUE/FALSE keep the json file locally?

Value

a data frame with the repository with the following columns: data_level, data_path, chapt_name, url, repository name

Examples

```
## Not run:

usethis::create_github_token()

get_pages_url("jhudsl/Documentation_and_Usability")

## End(Not run)
```

---

get_slide_id *Get Slide ID from URL*

Description

Get Slide ID from URL

Usage

```
get_slide_id(x)
```

Arguments

x	URL of slide

Value

A character vector

Examples

```
x <- paste0(
  "https://docs.google.com/presentation/d/",
  "1Tg-GTGnUPduOtZKYuMoelqUNZnUp3vvg_7TtpUPL7e8",
  "/edit#slide=id.g154aa4fae2_0_58"
)
get_slide_id(x)
```

get_yaml_spec *Load in yaml specifications from _bookdown.yml or _quarto.yml*

Description

Load in yaml specifications from _bookdown.yml or _quarto.yml

Usage

```
get_yaml_spec(path = ".")
```

Arguments

path	Where to look for the yaml spec file. By default looks in current directory.

Value

The yaml contents using yaml::yaml.load_file()

good_quiz_path *Path to good example quiz*

Description

Path to good example quiz

Usage

```
good_quiz_path()
```

Value

The file path to an example good quiz included in the package that should pass the quiz checks.

gs_id_from_slide *Google Slides Helper Functions*

Description

Google Slides Helper Functions

Usage

```
gs_id_from_slide(file)

get_image_link_from_slide(file)

get_image_from_slide(file)
```

Arguments

file	markdown file for manuscript

Value

A scalar character vector

gs_png_url *Get Google Slide PNG URL*

Description

Get Google Slide PNG URL

Usage

```
gs_png_url(url)

get_slide_page(url)

gs_png_download(url, output_dir = ".", overwrite = TRUE)

include_slide(
  url,
  output_dir = knitr::opts_chunk$get("fig.path"),
  overwrite = TRUE,
  ...
)
```

Arguments

url	URL to Google Slide
output_dir	path to output png
overwrite	should the slide PNG be overwritten?
...	for include_slide, options passed to [knitr::include_graphics()]

Value

A character vector of URLs

Examples

```
url <- paste0(
  "https://docs.google.com/presentation/d/",
  "12DPZgPteQBwgal6kSPP58zhPhjZ7QSPZLe3NkA8M3eo/edit",
  "#slide=id.gc8648f14c3_0_397&t=4"
)
id <- get_slide_id(url)
gs_png_url(url)
```

---

key_encrypt_creds_path *Get file path to an key encryption RDS*

Description

Get file path to an key encryption RDS

Usage

```
key_encrypt_creds_path()
```

---

make_embed_markdown *Make Leanpub file that has embed webpage of a chapter*

Description

Make Leanpub file that has embed webpage of a chapter

Usage

```
make_embed_markdown(
  url,
  chapt_title,
  width_spec = 800,
  height_spec = 600,
  img_path,
  output_dir = "manuscript",
  verbose = TRUE,
  footer_text = ""
)
```

Arguments

url	The url to the chapter that is to be embed
chapt_title	Title of chapter to be used as file name and printed on iframe
width_spec	How wide should the iframe be in pixels?
height_spec	How high should the iframe be in pixels?
img_path	File path to image to use for poster
output_dir	output directory to put files. It should likely be relative to path
verbose	print diagnostic messages
footer_text	Optionally can add a bit of text that will be added to the end of each file before the references section.

Value

A markdown file with an iframe of the provided chapter

---

make_screenshots *A function to make screenshots from an OTTR bookdown website*

Description

This function creates screenshots of course chapters that are stored in a created output directory

Usage

```
make_screenshots(
  git_pat,
  repo,
  output_dir = "resources/chapt_screen_images",
  base_url = NULL
)
```

Arguments

git_pat	required argument; a Git secret – see https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/managing-your-personal-access-tokens for more info
repo	required argument; GitHub repository name, e.g., jhudsl/OTTR_Template
output_dir	default is "resources/chapt_screen_images"; Output directory where the chapter's screen images should be stored. For OTTR courses, don't change this unless you've changed the downstream functions accordingly.
base_url	default is NULL; rendered bookdown URL where screenshots are taken from, if NULL, the function will use the repo_name and and git_pat to find the base_url

Value

the file path for file where chapter urls are saved

Author(s)

Candace Savonen

Examples

```
## Not run:

 make_screenshots(Sys.getenv("secrets.GH_PAT"), "jhudsl/OTTR_Template")


## End(Not run)
```

---

output_destination *Declare file path to docs/ folder*

Description

Declare file path to docs/ folder

Usage

```
output_destination(path = ".")
```

Arguments

path	Where to look for the _bookdown.yml or _quarto.yml file. Passes to get_yaml_spec() function. By default looks in current directory

Value

The file paths to rmds or qmds listed in the _bookdown.yml or _quarto.ymlfile.

parse_quiz *Parse Quiz and Other Checking Functions*

Description

Parse Quiz and Other Checking Functions

Extract lines of the quiz

Usage

```
parse_quiz(quiz_lines, quiz_name = NULL, verbose = FALSE)

extract_quiz(quiz_lines)
```

Arguments

quiz_lines	A quiz's contents read in with readLines()
quiz_name	A character vector indicating the name of the quiz.
verbose	Would you like progress messages? TRUE/FALSE

Value

A list of elements, including a 'data.frame' and metadata for questions

the lines of the quiz that actually contain of the content of the quiz.

Examples

```
quiz_lines <- c(
  "{quiz, id: quiz_00_filename}",
  "### Lesson Name quiz",
  "{choose-answers: 4}",
  "? What do you think?",
  "",
  "C) The answer to this one",
  "o) Not the answer",
  "o) Not the answer either",
  "C) Another correct answer",
  "m) Mandatory different answer",
  "",
  "{/quiz}"
)
quiz_specs <- parse_quiz(quiz_lines)
check_quiz_attributes(quiz_specs)
```

parse_quiz_df *Parse quiz into a data.frame*

Description

Parse quiz into a data.frame

Usage

```
parse_quiz_df(quiz_lines, remove_tags = FALSE)
```

Arguments

quiz_lines	A character vector of the contents of the markdown file obtained from readLines()
remove_tags	TRUE/FALSE remove tags and empty lines?

Value

A data frame containing a type column which indicates what type of line each is.

Examples

```
## Not run:

# Use readLines() to read in a quiz
quiz_path <- good_quiz_path()
quiz_lines <- readLines(quiz_path)

# Can use this to parse the quiz into a data.frame
quiz_df <- parse_quiz_df(quiz_lines)

## End(Not run)
```

parse_q_tag *Parse apart a tag*

Description

Parse apart a tag

Usage

```
parse_q_tag(tag)
```

Arguments

tag	A single tag to extract from

Value

A named vector indicating the field and entry associated with it.

Examples

```
tag <- "{quiz, id: quiz_name_here, attempts: 10}"
parse_q_tag(tag)
```

---

pptx_notes *Get Notes from a PowerPoint (usually from Google Slides)*

Description

Get Notes from a PowerPoint (usually from Google Slides)

Usage

```
pptx_notes(file, ...)

pptx_slide_text_df(file, ...)

pptx_slide_note_df(file, ...)

unzip_pptx(file)
```

Arguments

file	Character. Path for ‘PPTX‘ file
...	additional arguments to pass to xml_notes, particularly xpath

Value

Either a character vector or ‘NULL‘

Examples

```
## Not run:
pptx_notes(ex_file)
pptx_slide_note_df(ex_file)
pptx_slide_text_df(ex_file)

## End(Not run)
```

qrmd_files *Get file paths to all qmds or rmds in the course website directory*

Description

Get file paths to all qmds or rmds in the course website directory

Usage

```
qrmd_files(path = ".")
```

Arguments

path	Where to look for the _bookdown.yml or _quarto.yml file. Passes to get_yaml_spec() function. By default looks in current directory

Value

The file paths to rmds or wmds listed in the _bookdown.yml or _quarto.yml file.

render_without_toc *Create TOC-less course website for use in Coursera or Leanpub*

Description

Create a version of the course that does not have a TOC and has quizzes in the Coursera yaml format. This is only needed to be used on Bookdown courses. Quarto has a simple command for this.

Usage

```
render_without_toc(
  output_dir = file.path("docs", "no_toc"),
  output_yaml = "_output.yml",
  convert_quizzes = FALSE,
  input_quiz_dir = "quizzes",
  output_quiz_dir = "coursera_quizzes",
  verbose = TRUE
)
```

Arguments

output_dir	A folder (existing or not) that the TOC-less Bookdown for Coursera files should be saved. By default is file.path("docs", "coursera")
output_yaml	A output.yml file to be provided to bookdown. By default is "_output.yml"
convert_quizzes	TRUE/FALSE whether or not to convert quizzes. Default is TRUE
input_quiz_dir	A path to a directory of Leanpub-formatted quiz md files. By default assumes "quizzes" and looks in current directory.
output_quiz_dir	A folder (existing or not) where the coursera quizzes should be saved. By default is "coursera_quizzes".
verbose	Would you like the progress messages? TRUE/FALSE

Value

A folder of coursera ready quiz files and html chapter files saved to output directories specified.

---

set_knitr_image_path *Set image path for 'knitr'*

Description

Set image path for 'knitr'

Usage

```
set_knitr_image_path(verbose = FALSE)
```

Arguments

verbose	print out what the figure path is

Value

When used inside a knitted R Markdown document, will set the image path to a place compatible with 'ottrpal' output folders.

`set_up_leanpub` *Set up Manuscript folder for Leanpub publishing*

Description

Set up Manuscript folder for Leanpub publishing

Usage

```
set_up_leanpub(
  path = ".",
  clean_up = FALSE,
  render = NULL,
  output_dir = "manuscript",
  make_book_txt = FALSE,
  quiz_dir = "quizzes",
  run_quiz_checks = FALSE,
  remove_resources_start = FALSE,
  verbose = TRUE,
  footer_text = NULL
)
```

Arguments

`path`	path to the top of course repository (looks for .git folder)
`clean_up`	TRUE/FALSE the old output directory should be deleted and everything created fresh.
`render`	If NULL will not be run. If "quarto" or "bookdown" then the respective render type will be run
`output_dir`	output directory to put files. It should likely be relative to path
`make_book_txt`	Should [ottrpal::course_to_book_txt()] be run to create a ‘Book.txt‘ in the output directory?
`quiz_dir`	directory that contains the quiz .md files that should be checked and incorporated into the Book.txt file. If you don’t have quizzes, set this to NULL
`run_quiz_checks`	TRUE/FALSE run quiz checks
`remove_resources_start`	remove the word ‘resources/‘ at the front of any image path.
`verbose`	print diagnostic messages
`footer_text`	Optionally can add a bit of text that will be added to the end of each file before the references section.

Value

A list of output files and diagnostics

website_to_embed_leanpub *Convert Website Course to Leanpub*

Description

Convert Website Course to Leanpub

Usage

```
website_to_embed_leanpub(
  path = ".",
  chapt_img_key = NULL,
  render = NULL,
  html_page = file.path(base_url, "index.html"),
  base_url = NULL,
  default_img = NULL,
  output_dir = "manuscript",
  make_book_txt = FALSE,
  quiz_dir = "quizzes",
  run_quiz_checks = FALSE,
  remove_resources_start = FALSE,
  verbose = TRUE,
  footer_text = ""
)
```

Arguments

path	path to the bookdown or quarto course repository, must have a '_bookdown.yml' or '_quarto.yml' file
chapt_img_key	File path to a TSV whose contents are the chapter urls ('url'), the chapter titles ('chapt_title'), the file path to the image to be used for the chapter ('img_path'). Column names 'url', 'chapt_title', and 'img_path' must be used. If no chapter title column supplied, the basename of the url will be used, If no image column supplied, default image used.
render	if 'TRUE', then [bookdown::render_book()] will be run on each Rmd.
html_page	The file path of the rendered index.html file
base_url	The base url of where the chapters are published – the url to provide to the iframe in Leanpub e.g. https://jhudatascience.org/OTTR_Template/coursera
default_img	A google slide link to the default image to be used for all chapters
output_dir	output directory to put files. It should likely be relative to path
make_book_txt	Should [ottrpal::course_to_book_txt()] be run to create a 'Book.txt' in the output directory?
quiz_dir	directory that contains the quiz .md files that should be checked and incorporated into the Book.txt file. If you don't have quizzes, set this to NULL

run_quiz_checks
: TRUE/FALSE run quiz checks

remove_resources_start
: remove the word 'resources/' at the front of any image path.

verbose
: print diagnostic messages

footer_text
: Optionally can add a bit of text that will be added to the end of each file before the references section.

Value

A directory of output files in a folder 'manuscript' for publishing on Leanpub.

Examples

```
## Not run:

ottrpal::website_to_embed_leanpub(
  base_url = "https://jhudatascience.org/OTTR_Template/",
  make_book_txt = TRUE,
  quiz_dir = NULL
)

## End(Not run)
```

---

xml_notes *Get Notes from XML*

Description

Get Notes from XML

Usage

```
xml_notes(file, collapse_text = TRUE, xpath = "//a:r//a:t")
```

Arguments

file
: XML file from a PPTX

collapse_text
: should text be collapsed by spaces?

xpath
: xpath to pass to [xml2::xml_find_all()]

Value

A character vector

Index