Package 'moodef'

May 9, 2026

Type Package

Title Defining 'Moodle' Elements from R

Version 1.2.0

Description The main objective of this package is to support the definition of 'Moodle' elements taking advantage of the power that R offers. In this first version, it allows the definition of quizzes to be included in the question bank.

License MIT + file LICENSE

URL https://josesamos.github.io/moodef/, https://github.com/josesamos/moodef

BugReports https://github.com/josesamos/moodef/issues

Imports blastula, dplyr, glue, magick, readr, readxl, snakecase, tibble, tidyr, tools, xlsx, xml2

Suggests knitr, pander, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

Language en-GB

LazyData true

RoxygenNote 7.3.2

Depends R (>= 3.5)

NeedsCompilation no

Author Jose Samos [aut, cre] (ORCID: <https://orcid.org/0000-0002-4457-3439>), Universidad de Granada [cph]

Maintainer Jose Samos <jsamos@ugr.es>

Repository CRAN

Date/Publication 2025-01-26 13:10:02 UTC

Contents

`create_question_csv` *Create a question csv file*

Description

Creates an empty question csv file.

Usage

```
create_question_csv(file, sep = ",", extended = FALSE)
```

Arguments

`file`	A string, name of a text file.
`sep`	Column separator character.
`extended`	A Boolean, use extended question definition.

Value

A string.

See Also

Other support functions: `create_question_data_frame()`, `create_question_excel()`, `read_question_csv()`, `read_question_excel()`, `vector_to_string()`

Examples

```
file <- create_question_csv(file = tempfile(fileext = '.csv'))
```

create_question_data_frame *Create a question data frame*

Description

Creates an empty question data frame.

Usage

```
create_question_data_frame(extended = FALSE)
```

Arguments

extended	A Boolean, use extended question definition.

Value

A data frame.

See Also

Other support functions: `create_question_csv()`, `create_question_excel()`, `read_question_csv()`, `read_question_excel()`, `vector_to_string()`

Examples

```
df <- create_question_data_frame()
```

create_question_excel *Create a question Excel file*

Description

Creates an empty question Excel file.

Usage

```
create_question_excel(file, extended = FALSE)
```

Arguments

file	A string, name of a text file.
extended	A Boolean, use extended question definition.

Value

A string.

See Also

Other support functions: `create_question_csv()`, `create_question_data_frame()`, `read_question_csv()`, `read_question_excel()`, `vector_to_string()`

Examples

```
file <- create_question_excel(file = tempfile(fileext = '.xlsx'))
```

`define_extended_question` *Define an extended question*

Description

This function allows users to define an extended question, including metadata, feedback and optional image data.

Usage

```
define_extended_question(
  qc,
  category,
  type,
  fraction,
  id,
  name,
  author,
  fb_general,
  fb_correct,
  fb_partially,
  fb_incorrect,
  question,
  image,
  image_alt,
  answer,
  a_1,
  a_2,
  a_3,
  a_4,
  a_5,
```

```
  a_6,
  a_7,
  fb_answer,
  fb_a_1,
  fb_a_2,
  fb_a_3,
  fb_a_4,
  fb_a_5,
  fb_a_6,
  fb_a_7,
  tag_1,
  tag_2,
  tag_3,
  tag_4,
  tag_5,
  tag_6,
  tag_7,
  tag_8,
  tag_9
)

## S3 method for class 'question_category'
define_extended_question(
  qc,
  category = "",
  type = "",
  fraction = 0,
  id = "",
  name = "",
  author = "",
  fb_general = "",
  fb_correct = "",
  fb_partially = "",
  fb_incorrect = "",
  question = "",
  image = "",
  image_alt = "",
  answer = "",
  a_1 = "",
  a_2 = "",
  a_3 = "",
  a_4 = "",
  a_5 = "",
  a_6 = "",
  a_7 = "",
  fb_answer = "",
  fb_a_1 = "",
  fb_a_2 = "",
```

```
  fb_a_3 = "",
  fb_a_4 = "",
  fb_a_5 = "",
  fb_a_6 = "",
  fb_a_7 = "",
  tag_1 = "",
  tag_2 = "",
  tag_3 = "",
  tag_4 = "",
  tag_5 = "",
  tag_6 = "",
  tag_7 = "",
  tag_8 = "",
  tag_9 = ""
)
```

Arguments

qc	A question category object. It should have a questions data frame where new questions will be added.
category	A character string specifying the category of the question.
type	A character string indicating the type of the question.
fraction	A number between 0 and 1.
id	A unique identifier for the question.
name	A character string representing the name of the question.
author	The name of the author of the question.
fb_general	General feedback for the question.
fb_correct	Feedback displayed when the correct answer is selected.
fb_partially	Feedback displayed for partially correct answers.
fb_incorrect	Feedback displayed for incorrect answers.
question	The text of the question.
image	Path to an image file associated with the question.
image_alt	Alternative text describing the image for accessibility. Required if an image is provided.
answer	The correct answer to the question.
a_1	Additional possible answer.
a_2	Additional possible answer.
a_3	Additional possible answer.
a_4	Additional possible answer.
a_5	Additional possible answer.
a_6	Additional possible answer.
a_7	Additional possible answer.

fb_answer	Feedback for the correct answer.
fb_a_1	Feedback for additional answer.
fb_a_2	Feedback for additional answer.
fb_a_3	Feedback for additional answer.
fb_a_4	Feedback for additional answer.
fb_a_5	Feedback for additional answer.
fb_a_6	Feedback for additional answer.
fb_a_7	Feedback for additional answer.
tag_1	Tag to categorize the question.
tag_2	Tag to categorize the question.
tag_3	Tag to categorize the question.
tag_4	Tag to categorize the question.
tag_5	Tag to categorize the question.
tag_6	Tag to categorize the question.
tag_7	Tag to categorize the question.
tag_8	Tag to categorize the question.
tag_9	Tag to categorize the question.

Details

Parameter values that are not defined are taken from the category definition, if they are defined there.

The `fraction` attribute is used in various question types to determine how a specific answer impacts the question's score. Specifically, for incorrect answers in the `multichoice` and `truefalse` types, the value calculated by dividing `fraction` by the number of incorrect answers available is considered as the amount deducted in case of an incorrect response.

In the example provided, we have intentionally used the same structure as in the `define_question()` function to demonstrate that any parameters not needed do not need to be explicitly defined.

Value

Returns the updated question category object.

See Also

Other question definition functions: `define_question()`, `define_questions_from_csv()`, `define_questions_from_dat`
`define_questions_from_excel()`, `generate_xml()`, `generate_xml_file()`, `question_category()`

Examples

```
qc <- question_category(category = 'Initial test') |>
  define_extended_question(
    question = 'What are the basic arithmetic operations?',
    answer = 'Addition, subtraction, multiplication and division.',
    a_1 = 'Addition and subtraction.',
    a_2 = 'Addition, subtraction, multiplication, division and square root.'
  )
```

`define_question` *Define a question*

Description

Define a question and the possible answers. The type of question is deduced.

Usage

```
define_question(qc, type, question, image, image_alt, answer, ...)

## S3 method for class 'question_category'
define_question(
  qc,
  type = "",
  question = "",
  image = "",
  image_alt = "",
  answer = "",
  ...
)
```

Arguments

`qc`	A `question_category` object.
`type`	A string, question type (if needed).
`question`	A string, statement of the question.
`image`	A string, optional, image file to include in the question.
`image_alt`	A string, description of the image to include in the question.
`answer`	A string, correct answer to the question.
`...`	A string, rest of the answers to the question.

Details

If we include an image in the question, we must also include text in the `alt` field associated with it.

After the correct answer, we can indicate as many answers as we want, if we do not indicate all the parameters, we have to give each answer a parameter name different from the rest of the parameter names.

Value

A `question_category`.

See Also

Other question definition functions: `define_extended_question()`, `define_questions_from_csv()`, `define_questions_from_data_frame()`, `define_questions_from_excel()`, `generate_xml()`, `generate_xml_file()`, `question_category()`

Examples

```
qc <- question_category(category = 'Initial test') |>
  define_question(
    question = 'What are the basic arithmetic operations?',
    answer = 'Addition, subtraction, multiplication and division.',
    a_1 = 'Addition and subtraction.',
    a_2 = 'Addition, subtraction, multiplication, division and square root.'
  )
```

`define_questions_from_csv` *Define questions from a csv file*

Description

Each row in the text file is interpreted as a question. We only have to define the columns that we are going to use, the rest of the columns are taken by default.

Usage

```
define_questions_from_csv(qc, file, sep)

## S3 method for class 'question_category'
define_questions_from_csv(qc, file, sep = ",")
```

Arguments

`qc`	A `question_category` object.
`file`	A string, name of a text file.
`sep`	Column separator character.

Details

For answers where a vector is required, "<|>" is used as a separator of the vector elements.

Value

A question_category.

See Also

Other question definition functions: define_extended_question(), define_question(), define_questions_from_data
define_questions_from_excel(), generate_xml(), generate_xml_file(), question_category()

Examples

```
file <- system.file("extdata", "questions.csv", package = "moodef")
qc <-
  question_category(category = 'Initial test', adapt_images = TRUE) |>
  define_questions_from_csv(file = file)
```

---

define_questions_from_data_frame

Define questions from a data frame

Description

Each row in the text data frame is interpreted as a question. We only have to define the columns that we are going to use, the rest of the columns are taken by default.

Usage

```
define_questions_from_data_frame(qc, df)

## S3 method for class 'question_category'
define_questions_from_data_frame(qc, df)
```

Arguments

qc	A question_category object.
df	A data frame.

Details

For answers where a vector is required, "<|>" is used as a separator of the vector elements.

Value

A question_category.

See Also

Other question definition functions: define_extended_question(), define_question(), define_questions_from_csv(
define_questions_from_excel(), generate_xml(), generate_xml_file(), question_category()

Examples

```
file <- system.file("extdata", "questions.csv", package = "moodef")
df <- read_question_csv(file = file)

qc <-
  question_category(category = 'Initial test', adapt_images = TRUE) |>
  define_questions_from_data_frame(df)
```

`define_questions_from_excel`

Define questions from a Excel file

Description

Each row in the Excel file is interpreted as a question. We only have to define the columns that we are going to use, the rest of the columns are taken by default.

Usage

```
define_questions_from_excel(qc, file, sheet_index, sheet_name)

## S3 method for class 'question_category'
define_questions_from_excel(qc, file, sheet_index = NULL, sheet_name = NULL)
```

Arguments

qc	A question_category object.
file	A string, name of an Excel file.
sheet_index	A number, sheet index in the workbook.
sheet_name	A string, sheet name.

Details

In addition to the file, we can indicate the sheet by its name or index. If we do not indicate anything, it considers the first sheet.

For answers where a vector is required, "<|>" is used as a separator of the vector elements.

Value

A question_category.

See Also

Other question definition functions: define_extended_question(), define_question(), define_questions_from_csv(
define_questions_from_data_frame(), generate_xml(), generate_xml_file(), question_category()

Examples

```
file <- system.file("extdata", "questions.xlsx", package = "moodef")
qc <-
  question_category(category = 'Initial test', adapt_images = TRUE) |>
  define_questions_from_excel(file = file)
```

---

generate_xml *Generate questions xml string*

Description

Generate questions xml string

Usage

```
generate_xml(qc)

## S3 method for class 'question_category'
generate_xml(qc)
```

Arguments

qc	A question_category object.

Value

A string.

See Also

Other question definition functions: define_extended_question(), define_question(), define_questions_from_csv(
define_questions_from_data_frame(), define_questions_from_excel(), generate_xml_file(),
question_category()

Examples

```
qc <- question_category(category = 'Initial test') |>
  define_question(
    question = 'What are the basic arithmetic operations?',
    answer = 'Addition, subtraction, multiplication and division.',
    a_1 = 'Addition and subtraction.',
    a_2 = 'Addition, subtraction, multiplication, division and square root.'
  )

xml <- qc |>
  generate_xml()
```

---

generate_xml_file *Generate questions xml file*

Description

Generate questions xml file

Usage

```
generate_xml_file(qc, file)

## S3 method for class 'question_category'
generate_xml_file(qc, file = NULL)
```

Arguments

qc	A question_category object.
file	A string, file name.

Value

A question_category.

See Also

Other question definition functions: define_extended_question(), define_question(), define_questions_from_csv(
define_questions_from_data_frame(), define_questions_from_excel(), generate_xml(),
question_category()

Examples

```
qc <- question_category(category = 'Initial test') |>
  define_question(
    question = 'What are the basic arithmetic operations?',
    answer = 'Addition, subtraction, multiplication and division.',
    a_1 = 'Addition and subtraction.',
    a_2 = 'Addition, subtraction, multiplication, division and square root.'
  ) |>
  generate_xml_file(file = tempfile(fileext = '.xml'))
```

`question_category` `question_category` *S3 class*

Description

Creates a `question_category` object.

Usage

```
question_category(
  category = "Default category",
  first_question_number = 1,
  copyright = "",
  license = "",
  correct_feedback = "Correct.",
  partially_correct_feedback = "Partially correct.",
  incorrect_feedback = "Incorrect.",
  adapt_images = FALSE,
  width = 800,
  height = 600,
  author = "",
  fraction = 0
)
```

Arguments

`category`	A string, category name.
`first_question_number`	An integer, first number to compose the question names.
`copyright`	A string, copyright text to be included in each question that is defined.
`license`	A string, license text to be included in each question that is defined.
`correct_feedback`	A string, feedback on correct answers to each question.
`partially_correct_feedback`	A string, feedback on partially correct answers to each question.

incorrect_feedback	A string, feedback on incorrect answers to each question.
adapt_images	A boolean, adapt the images so that they are a similar size.
width	A integer, width of each image.
height	A integer, height of each image.
author	A string, author name to be included in each question that is defined.
fraction	A number between 0 and 1.

Details

Defines a category of questions to be included in the *Moodle* question bank.

It allows us to define the name of the category, the copyright and license literals that will be added to each question, and the feedback literals for correct, partially correct and incorrect questions.

Each question can include an image after the text. We can also configure if we want to automatically transform the images so that they have a standard size that we can also indicate.

The `fraction` attribute is used in various question types to determine how a specific answer impacts the question's score. Specifically, for incorrect answers in the `multichoice` and `truefalse` types, the value calculated by dividing `fraction` by the number of incorrect answers available is considered as the amount deducted in case of an incorrect response.

Value

A `question_category` object.

See Also

Other question definition functions: `define_extended_question()`, `define_question()`, `define_questions_from_csv()`, `define_questions_from_data_frame()`, `define_questions_from_excel()`, `generate_xml()`, `generate_xml_file()`

Examples

```
qc <- question_category(category = 'Initial test')
```

---

read_question_csv *Read a question csv file*

Description

Reads a csv file of questions and returns a data frame.

Usage

```
read_question_csv(file, sep = ",")
```

Arguments

file	A string, name of a text file.
sep	Column separator character.

Value

A data frame.

See Also

Other support functions: create_question_csv(), create_question_data_frame(), create_question_excel(), read_question_excel(), vector_to_string()

Examples

```
file <- system.file("extdata", "questions.csv", package = "moodef")
df <- read_question_csv(file = file)
```

---

read_question_excel *Read a question Excel file*

Description

Reads an Excel file of questions and returns a data frame.

Usage

```
read_question_excel(file, sheet_index = NULL, sheet_name = NULL)
```

Arguments

file	A string, name of a text file.
sheet_index	A number, sheet index in the workbook.
sheet_name	A string, sheet name.

Details

In addition to the file, we can indicate the sheet by its name or index. If we do not indicate anything, it considers the first sheet.

Value

A data frame.

See Also

Other support functions: create_question_csv(), create_question_data_frame(), create_question_excel(), read_question_csv(), vector_to_string()

Examples

```
file <- system.file("extdata", "questions.xlsx", package = "moodef")
df <- read_question_excel(file = file)
```

---

vector_to_string *Transforms a vector of strings into a string*

Description

Insert the separator that we consider to later perform the reverse operation.

Usage

```
vector_to_string(vector)
```

Arguments

vector	A vector of strings.

Value

A string.

See Also

Other support functions: create_question_csv(), create_question_data_frame(), create_question_excel(), read_question_csv(), read_question_excel()

Examples

```
s <- vector_to_string(c('Addition', '+'))
```

Index