

Package ‘rqti’

March 21, 2024

Title Create Tests According to QTI 2.1 Standard

Version 0.1.1

Description Create tests and tasks compliant with the Question & Test Interoperability (QTI) information model version 2.1. Input sources are Rmd/md description files or S4-class objects. Output formats include standalone zip or xml files. Supports the generation of basic task types (single and multiple choice, order, pair association, matching tables, filling gaps and essay) and provides a comprehensive set of attributes for customizing tests.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.2.3

Imports htmltools, xml2, yaml, rmarkdown, servr, rstudioapi, fs, stringr, methods, lubridate, magrittr, httr2, curl, digest, knitr, getPass, keyring, zip, kableExtra

Suggests covr, dplyr, testthat (>= 3.0.0), XML, readr

Config/testthat/edition 3

Config/testthat/parallel false

URL <https://github.com/shevandrin/rqti>,
<https://shevandrin.github.io/rqti/>

BugReports <https://github.com/shevandrin/rqti/issues>

Collate 'rqti.R' 'ModalFeedback.R' 'AssessmentItem.R'
'AssessmentSection.R' 'AssessmentTest.R' 'AssessmentTestOpal.R'
'Choice.R' 'CorrectFeedback.R' 'MatchTable.R' 'DirectedPair.R'
'Entry.R' 'Essay.R' 'Gap.R' 'InlineChoice.R' 'MultipleChoice.R'
'MultipleChoiceTable.R' 'NumericGap.R' 'OneInColTable.R'
'OneInRowTable.R' 'Order.R' 'SingleChoice.R' 'TextGap.R'
'TextGapOpal.R' 'WrongFeedback.R' 'api_opal.R' 'character.R'
'extract_results.R' 'helpers.R' 'knit_functions.R'
'object_builder.R' 'qti_task.R' 'qti_test.R'
'response_processing.R' 'rqti_project.R' 'section_builder.R'
'utils-pipe.R' 'zzz.R'

Depends R (>= 2.10)

NeedsCompilation no

Author Andrey Shevandrin [aut, cre, cph]

(<<https://orcid.org/0000-0003-0807-2546>>),

Petr Bondarenko [ctb] (<<https://orcid.org/0000-0002-5154-9664>>),

Ivonne Ojeda [ctb],

Johannes Titz [aut, cph] (<<https://orcid.org/0000-0002-1102-5719>>),

Brian Mottershead [cph] (Author of QTIJS library),

Stiftung für Innovation in der Hochschullehre [fnd]

Maintainer Andrey Shevandrin <shevandrin@gmail.com>

Repository CRAN

Date/Publication 2024-03-21 12:10:02 UTC

R topics documented:

AssessmentItem-class	3
AssessmentSection-class	4
AssessmentTest-class	6
AssessmentTestOpal-class	7
auth_opal	9
buildAssessmentSection	10
Choice-class	11
CorrectFeedback-class	11
createAssessmentTest	12
createItemBody	13
createOutcomeDeclaration	14
createQtiTask-methods	14
createQtiTest-methods	15
createResponseDeclaration	16
createResponseProcessing	17
createText	18
createZip	19
create_assessment_item	19
create_qti_task	20
create_qti_test	20
DirectedPair-class	21
dropdown	23
Entry-class	24
Essay-class	25
extract_results	27
Gap-class	28
gap_numeric	29
gap_text	30
getAssessmentItems	31
getCalculator-methods	31
getFiles-methods	32
getIdentifier-methods	33

getObject-methods	33
getPoints-methods	34
getResponse	35
get_resources	35
get_resource_url	36
InlineChoice-class	37
MatchTable-class	37
mdlist	39
ModalFeedback-class	40
MultipleChoice-class	40
MultipleChoiceTable-class	42
NumericGap-class	44
OneInColTable-class	45
OneInRowTable-class	47
Order-class	48
prepareQTIJSFiles-methods	50
prepare_renderer	50
qtjs_path	51
render_opal	51
render_qtjs	52
render_xml	53
render_zip	53
rmd2xml	54
rmd2zip	54
section	55
SingleChoice-class	56
start_server	58
stop_server	59
test	59
test4opal	61
TextGap-class	63
TextGapOpal-class	64
upload2opal	65
WrongFeedback-class	66
Index	67

AssessmentItem-class *Class AssessmentItem*

Description

Abstract class `AssessmentItem` is responsible for creating a root element 'assessmentItem' in XML task description according to QTI 2.1. This class is not meant to be instantiated directly; instead, it serves as a base for derived classes.

Slots

- identifier** A character representing the unique identifier of the assessment task. By default, it is generated as 'id_task_ddd', where dddd represents random digits.
- title** A character representing the title of the XML file associated with the task. By default, it takes the value of the identifier.
- content** A list of character content to form the text of the question, which can include HTML tags. For tasks of the [Entry](#) type, it must also contain at least one instance of Gap objects, such as [TextGap](#), [NumericGap](#), or [InlineChoice](#).
- prompt** An optional character representing a simple question text, consisting of one paragraph. This can supplement or replace content in the task. Default is "".
- points** A numeric value, optional, representing the number of points for the entire task. Default is 1, but pay attention:
- For tasks of the [Entry](#) type, it is calculated as the sum of the gap points by default.
 - For tasks of the [MatchTable](#) type, it can also be calculated as the sum of points for individual answers, when provided.
 - For tasks of the [MultipleChoice](#) type, points is numeric vector and required. Each number in this vector determines the number of points that will be awarded to a candidate if they select the corresponding answer. The order of the scores must match the order of the choices. It is possible to assign negative values to incorrect answers. All answers with a positive score are considered correct.
- feedback** A list containing feedback messages for candidates. Each element of the list should be an instance of either [ModalFeedback](#), [CorrectFeedback](#), or [WrongFeedback](#) class.
- calculator** A character, optional, determining whether to show a calculator to the candidate. Possible values:
- "simple-calculator"
 - "scientific-calculator"
- files** A character vector, optional, containing paths to files that will be accessible to the candidate during the test/exam.
- qti_version** A character representing the QTI (Question and Test Interoperability) information model version. Default is 'v2p1'.

AssessmentSection-class

Class "AssessmentSection"

Description

Class `AssessmentSection` is responsible for forming a section in the test XML specification according to QTI 2.1.

Slots

- identifier** A character representing the unique identifier of the assessment section. By default, it is generated as 'id_section_dddd', where dddd represents random digits.
- title** A character representing the title of the section in the test. By default, it takes the value of the identifier.
- time_limit** A numeric value, optional, controlling the amount of time *in minutes* a candidate is allowed for this part of the test.
- visible** A boolean value, optional. If TRUE, it shows this section in the hierarchy of the test structure. Default is TRUE.
- assessment_item** A list containing [AssessmentSection](#) and/or Assessment item objects, such as [SingleChoice](#), [MultipleChoice](#), [Essay](#), [Entry](#), [Order](#), [OneInRowTable](#), [OneInColTable](#), [MultipleChoiceTable](#), and [DirectedPair](#).
- shuffle** A boolean value, optional, responsible for randomizing the order in which the assessment items and subsections are initially presented to the candidate. Default is FALSE.
- selection** A numeric value, optional, defining how many children of the section are delivered in the test.
- max_attempts** A numeric value, optional, enabling the maximum number of attempts a candidate is allowed to pass in this section.
- allow_comment** A boolean value, optional, enabling to allow the candidate to leave comments in each question of the section. Default is TRUE.

See Also

[section\(\)](#), [test\(\)](#), [test4opal\(\)](#), [AssessmentTest](#), [AssessmentTestOpal](#).

Examples

```
sc1 <- new("SingleChoice", prompt = "Example task 1", title = "SC1",
  identifier = "q1", choices = c("a", "b", "c"))
sc2 <- new("SingleChoice", prompt = "Example task 2", title = "SC2",
  identifier = "q2", choices = c("A", "B", "C"))
sc3 <- new("SingleChoice", prompt = "Example task 3", title = "SC3",
  identifier = "q3", choices = c("aa", "bb", "cc"))
exam_section <- new("AssessmentSection",
  identifier = "sec_id",
  title = "Section",
  time_limit = 20,
  visible = FALSE,
  assessment_item = list(sc1, sc2, sc3),
  shuffle = FALSE,
  selection = 1,
  max_attempts = 1,
  allow_comment = FALSE)
```

AssessmentTest-class *Class "AssessmentTest"*

Description

Class AssessmentTest is responsible for creating XML exam files according to the QTI 2.1 standard.

Details

Test consists of one or more sections. Each section can have one or more questions/tasks and/or one or more sub sections.

Slots

identifier A character representing the unique identifier of the assessment test. By default, it is generated as 'id_test_ddd', where dddd represents random digits.

title A character representing the title of the test. By default, it takes the value of the identifier.

points Do not use directly; the maximum number of points for the exam/test. It is calculated automatically as a sum of points of included tasks.

test_part_identifier A character representing the identifier of the test part.

navigation_mode A character value, optional, determining the general paths that the candidate may have during the exam. Possible values:

- "linear" - candidate is not allowed to return to the previous questions.
- "nonlinear" - candidate is free to navigate. This is used by default.

submission_mode A character value, optional, determining when the candidate's responses are submitted for response processing. Possible values:

- "individual" - submit candidates' responses on an item-by-item basis. This is used by default.
- "simultaneous" - candidates' responses are submitted all together by the end of the test.

section A list containing one or more [AssessmentSection](#) objects.

qti_version A character value, optional, representing the QTI information model version. Default is 'v2p1'.

time_limit A numeric value, optional, controlling the amount of time in minutes which a candidate is allowed for this part of the test.

max_attempts A numeric value, optional, enabling the maximum number of attempts that a candidate is allowed to pass.

allow_comment A boolean value, optional, enabling to allow candidates to leave comments in each question.

rebuild_variables A boolean value, optional, enabling to recalculate variables and reshuffle the order of choices for each item-attempt.

academic_grading A boolean value, optional, enabling to show to candidates at the end of the testing a grade according to a 5-point academic grade system as feedback. Default is FALSE.

`grade_label` A character value, optional, representing a short message to display with a grade in the final feedback. For multilingual usage, it has to be a named vector with two-letter ISO language codes as names (e.g., `c(en="Grade", de="Note")`); during test creation, it takes the value for the language of the operating system. Default is `c(en="Grade", de="Note")`.

`table_label` A character value, optional, representing a concise message to display as the column title of the grading table in the final feedback. For multilingual usage, it has to be a named vector with two-letter ISO language codes as names (e.g., `c(en="Grade", de="Note")`); during test creation, it takes the value for the language of the operating system. Default is `c(en="Grade", de="Note")`.

See Also

[AssessmentSection](#), [AssessmentTestOpal](#), [test\(\)](#), [test4opal\(\)](#), [section\(\)](#).

Examples

```
# This example creates test 'exam' with one section 'exam_section' which
# consists of two questions/tasks: essay and single choice types
task1 <- new("Essay", prompt = "Test task", title = "Essay",
            identifier = "q1")
task2 <- new("SingleChoice", prompt = "Test task", title = "SingleChoice",
            choices = c("A", "B", "C"), identifier = "q2")
exam_section <- new("AssessmentSection", identifier = "sec_id",
                  title = "section", assessment_item = list(task1, task2))
exam <- new("AssessmentTest",
          identifier = "id_test_1234",
          title = "Example of Exam",
          navigation_mode = "linear",
          submission_mode = "individual",
          section = list(exam_section),
          time_limit = 90,
          max_attempts = 1,
          academic_grading = TRUE,
          grade_label = "Preliminary grade")
```

AssessmentTestOpal-class

Class "AssessmentTestOpal"

Description

Class `AssessmentTestOpal` is responsible for creating XML exam files according to the QTI 2.1 standard for LMS Opal.

Details

Test consists of one or more sections. Each section can have one or more questions/tasks and/or one or more sub sections.

Slots

identifier A character representing the unique identifier of the assessment test. By default, it is generated as 'id_test_ddd', where dddd represents random digits.

title A character representing the title of the test. By default, it takes the value of the identifier.

points Do not use directly; the maximum number of points for the exam/test. It is calculated automatically as a sum of points of included tasks.

test_part_identifier A character representing the identifier of the test part.

navigation_mode A character value, optional, determining the general paths that the candidate may have during the exam. Possible values:

- "linear" - candidate is not allowed to return to the previous questions.
- "nonlinear" - candidate is free to navigate. This is used by default.

submission_mode A character value, optional, determining when the candidate's responses are submitted for response processing. Possible values:

- "individual" - submit candidates' responses on an item-by-item basis. This is used by default.
- "simultaneous" - candidates' responses are submitted all together by the end of the test.

section A list containing one or more [AssessmentSection](#) objects.

qti_version A character value, optional, representing the QTI information model version. Default is 'v2p1'.

time_limit A numeric value, optional, controlling the amount of time in minutes which a candidate is allowed for this part of the test.

max_attempts A numeric value, optional, enabling the maximum number of attempts that a candidate is allowed to pass.

allow_comment A boolean value, optional, enabling to allow candidates to leave comments in each question.

rebuild_variables A boolean value, optional, enabling to recalculate variables and reshuffle the order of choices for each item-attempt.

academic_grading A boolean value, optional, enabling to show to candidates at the end of the testing a grade according to a 5-point academic grade system as feedback. Default is FALSE.

grade_label A character value, optional, representing a short message to display with a grade in the final feedback. For multilingual usage, it has to be a named vector with two-letter ISO language codes as names (e.g., c(en="Grade", de="Note")); during test creation, it takes the value for the language of the operating system. Default is c(en="Grade", de="Note").

table_label A character value, optional, representing a concise message to display as the column title of the grading table in the final feedback. For multilingual usage, it has to be a named vector with two-letter ISO language codes as names (e.g., c(en="Grade", de="Note")); during test creation, it takes the value for the language of the operating system. Default is c(en="Grade", de="Note").

show_test_time A boolean value, optional, determining whether to show the candidate elapsed processing time without time limit. Default is FALSE.

calculator A character value, optional, determining whether to show a calculator to the candidate. Possible values:

- "simple-calculator"
- "scientific-calculator".

mark_items A boolean value, optional, determining whether to allow candidate marking of questions. Default is TRUE.

keep_responses A boolean value, optional, determining whether to save candidate's answers from the previous attempt. Default is FALSE.

files A character vector, optional, containing paths to files that will be accessible to the candidate during the test/exam.

See Also

[AssessmentSection](#), [AssessmentTest](#), [test\(\)](#), [test4opal\(\)](#), [section\(\)](#).

Examples

```
# This example creates test 'exam' with one section 'exam_section' which
# consists of two questions/tasks: essay and single choice types
task1 <- new("Essay", prompt = "Test task", title = "Essay",
            identifier = "q1")
task2 <- new("SingleChoice", prompt = "Test task", title = "SingleChoice",
            choices = c("A", "B", "C"), identifier = "q2")
exam_section <- new("AssessmentSection", identifier = "sec_id",
                  title = "section", assessment_item = list(task1, task2))
exam <- new("AssessmentTestOpal",
          identifier = "id_test_1234",
          title = "Example of Exam",
          navigation_mode = "linear",
          submission_mode = "individual",
          section = list(exam_section),
          time_limit = 90,
          max_attempts = 1,
          academic_grading = TRUE,
          grade_label = "Preliminary grade",
          show_test_time = TRUE,
          calculator = "scientific-calculator",
          mark_items = TRUE,
          files = "text_book.pdf")
```

Description

Function `auth_opal()` performs the necessary authentication steps in OPAL API. If the authentication is successful, the function sets the token value in the system environment and returns the user's identity key in OPAL. The token value is required to access the OPAL API system.

Usage

```
auth_opal(api_user = NULL, api_password = NULL, endpoint = NULL)
```

Arguments

api_user	A character value of the username in the OPAL.
api_password	A character value of the password in the OPAL.
endpoint	A string of endpoint of LMS Opal; by default it is got from environment variable QTI_API_ENDPOINT. To set a global environment variable, you need to call <code>Sys.setenv(QTI_API_ENDPOINT='xxxxxxxxxxxxxxxx')</code> or you can put these command into <code>.Renviron</code> .

Value

A character string with Opal user id

Authentication

To use OPAL API, you need to provide your OPAL-username and password. This package uses system credential store 'keyring' to store user's name and password. After the first successful access to the OPAL API, there is no need to specify the username and password again, they will be extracted from the system credential store

Examples

```
auth_opal()
```

```
buildAssessmentSection
```

Build tags for AssessmentSection in assessmentTest

Description

Generic function for tags that contains assesmentSection in assessnetTest

Usage

```
buildAssessmentSection(object, folder = NULL, verify = FALSE)
```

```
## S4 method for signature 'AssessmentItem'
```

```
buildAssessmentSection(object, folder)
```

```
## S4 method for signature 'AssessmentSection'
```

```
buildAssessmentSection(object, folder = NULL, verify = FALSE)
```

```
## S4 method for signature 'character'
```

```
buildAssessmentSection(object, folder = NULL, verify = FALSE)
```

Arguments

object	an instance of the S4 object (AssessmentSection and all types of AssessmentItem)
folder	string; a folder to store xml file
verify	boolean, optional; check validity of xml file, default FALSE

Choice-class *Class "Choice"*

Description

Abstract class Choice is not meant to be instantiated directly; instead, it serves as a base for derived classes [SingleChoice](#) and [MultipleChoice](#).

Slots

- choices A character vector defining a set of answer options in the question.
- choice_identifiers A character vector, optional, containing a set of identifiers for answers. By default, identifiers are generated automatically according to the template "ChoiceD", where D is a letter representing the alphabetical order of the answer in the list.
- shuffle A boolean value indicating whether to randomize the order in which the choices are initially presented to the candidate. Default is TRUE.
- orientation A character, determining whether to place answers in vertical or horizontal mode. Possible values:
- "vertical" - Default.
 - "horizontal"

CorrectFeedback-class *Class "CorrectFeedback"*

Description

Class CorrectFeedback is responsible for delivering feedback messages to the candidate in case of a correct answer on the entire exercise.

Slots

`outcome_identifier` A character representing the unique identifier of the outcome declaration variable that relates to feedback. Default is "FEEDBACKMODAL".

`show` A boolean value, optional, determining whether to show (TRUE) or hide (FALSE) the modal feedback. Default is TRUE.

`title` A character value, optional, representing the title of the modal feedback window.

`content` A list of character content to form the text of the modal feedback, which can include HTML tags.

`identifier` A character value representing the identifier of the modal feedback item. Default is "correct". `cfb <- new("CorrectFeedback", title = "Right answer", content = list("Some demonstration"))`

`createAssessmentTest` *Create an element `assessmentTest` of a `qti-xml` document for test*

Description

Generic function for creating `assessmentTest` element for XML document of specification the test following the QTI schema v2.1

Usage

```
createAssessmentTest(object, folder, verify = FALSE)
```

```
## S4 method for signature 'AssessmentTest'
createAssessmentTest(object, folder, verify = FALSE)
```

```
## S4 method for signature 'AssessmentTestOpal'
createAssessmentTest(object, folder, verify = FALSE)
```

Arguments

<code>object</code>	an instance of the S4 object AssessmentTest or AssessmentTestOpal
<code>folder</code>	string, optional; a folder to store xml file; working directory by default
<code>verify</code>	boolean, optional; to check validity of xml file, default FALSE

createItemBody	<i>Create an element itemBody of a qti-xml document</i>
----------------	---

Description

Generic function for creating itemBody element for XML document of specification the question following the QTI schema v2.1

Usage

```
createItemBody(object)

## S4 method for signature 'DirectedPair'
createItemBody(object)

## S4 method for signature 'Entry'
createItemBody(object)

## S4 method for signature 'Essay'
createItemBody(object)

## S4 method for signature 'MultipleChoice'
createItemBody(object)

## S4 method for signature 'MultipleChoiceTable'
createItemBody(object)

## S4 method for signature 'OneInColTable'
createItemBody(object)

## S4 method for signature 'OneInRowTable'
createItemBody(object)

## S4 method for signature 'Order'
createItemBody(object)

## S4 method for signature 'SingleChoice'
createItemBody(object)
```

Arguments

object	an instance of the S4 object (SingleChoice , MultipleChoice , Essay , Entry , Order , OneInRowTable , OneInColTable , MultipleChoiceTable , DirectedPair)
--------	---

createOutcomeDeclaration

Create an element outcomeDeclaration of a qti-xml document

Description

Generic function for creating outcomeDeclaration element for XML document of specification the question following the QTI schema v2.1

Usage

```
createOutcomeDeclaration(object)

## S4 method for signature 'AssessmentItem'
createOutcomeDeclaration(object)

## S4 method for signature 'AssessmentTest'
createOutcomeDeclaration(object)

## S4 method for signature 'Entry'
createOutcomeDeclaration(object)

## S4 method for signature 'Gap'
createOutcomeDeclaration(object)
```

Arguments

object an instance of the S4 object ([SingleChoice](#), [MultipleChoice](#), [Essay](#), [Entry](#), [Order](#), [OneInRowTable](#), [OneInColTable](#), [MultipleChoiceTable](#), [DirectedPair](#), [TextGap](#), [NumericGap](#), [InlineChoice](#))

createQtiTask-methods *Create XML file for question specification*

Description

Create XML file for question specification

Usage

```
createQtiTask(object, dir = NULL, verification = FALSE)

## S4 method for signature 'AssessmentItem'
createQtiTask(object, dir = NULL, verification = FALSE)
```

Arguments

object	An instance of the S4 object (SingleChoice , MultipleChoice , Essay , Entry , Order , OneInRowTable , OneInColTable , MultipleChoiceTable , DirectedPair).
dir	A character value, optional; a folder to store xml file; working directory is used by default.
verification	A boolean value, optional; to check validity of xml file. Default is FALSE.

Value

A path to xml file.

Examples

```
essay <- new("Essay", prompt = "Test task", title = "Essay")
## Not run:
# creates folder with zip (side effect)
createQtiTask(essay, "result", TRUE)
## End(Not run)
```

createQtiTest-methods *Create zip-archive of the qti test specification*

Description

Create zip-archive of the qti test specification

Usage

```
createQtiTest(object, dir = NULL, verification = FALSE, zip_only =
  FALSE)

## S4 method for signature 'AssessmentItem'
createQtiTest(object, dir = ".", verification = FALSE, zip_only = FALSE)

## S4 method for signature 'AssessmentTest'
createQtiTest(object, dir = getwd(), verification = FALSE, zip_only = FALSE)

## S4 method for signature 'character'
createQtiTest(object, dir = getwd())
```

Arguments

object	An instance of the AssessmentTest , AssessmentTestOpal or AssessmentItem S4 object.
dir	A character value, optional; a folder to store xml file; working directory is used by default.

verification A boolean value, optional; to check validity of xml files. Default is FALSE.
zip_only A boolean value, optional; returns only zip file in case of TRUE or zip, xml and downloads files in case of FALSE value. Default is FALSE.

Value

A path to zip and xml files.

Examples

```
essay <- new("Essay", prompt = "Test task", title = "Essay",
            identifier = "q1")
sc <- new("SingleChoice", prompt = "Test task", title = "SingleChoice",
         choices = c("A", "B", "C"), identifier = "q2")
exam_section <- new("AssessmentSection", identifier = "sec_id",
                  title = "section", assessment_item = list(essay, sc))
exam <- new("AssessmentTestOpal", identifier = "id_test",
          title = "some title", section = list(exam_section))
## Not run:
# creates folder with zip (side effect)
createQtiTest(exam, "exam_folder", "TRUE")

## End(Not run)
```

createResponseDeclaration

Create an element responseDeclaration of a qti-xml document

Description

Generic function for creating responseDeclaration element for XML document of specification the question following the QTI schema v2.1

Usage

```
createResponseDeclaration(object)

## S4 method for signature 'AssessmentItem'
createResponseDeclaration(object)

## S4 method for signature 'MatchTable'
createResponseDeclaration(object)

## S4 method for signature 'Entry'
createResponseDeclaration(object)

## S4 method for signature 'Essay'
createResponseDeclaration(object)
```



```
## S4 method for signature 'InlineChoice'
createResponseDeclaration(object)

## S4 method for signature 'MultipleChoice'
createResponseDeclaration(object)

## S4 method for signature 'MultipleChoiceTable'
createResponseDeclaration(object)

## S4 method for signature 'NumericGap'
createResponseDeclaration(object)

## S4 method for signature 'Order'
createResponseDeclaration(object)

## S4 method for signature 'SingleChoice'
createResponseDeclaration(object)

## S4 method for signature 'TextGap'
createResponseDeclaration(object)
```

Arguments

object an instance of the S4 object ([SingleChoice](#), [MultipleChoice](#), [Entry](#), [Order](#), [OneIn-RowTable](#), [OneInColTable](#), [MultipleChoiceTable](#), [DirectedPair](#), [TextGap](#), [NumericGap](#), [InlineChoice](#))

createResponseProcessing

Create an element responseProcessing of a qti-xml document

Description

Generic function for creating responseProcessing element for XML document of specification the question following the QTI schema v2.1

Usage

```
createResponseProcessing(object)

## S4 method for signature 'AssessmentItem'
createResponseProcessing(object)

## S4 method for signature 'Entry'
createResponseProcessing(object)
```

```

## S4 method for signature 'Essay'
createResponseProcessing(object)

## S4 method for signature 'Gap'
createResponseProcessing(object)

## S4 method for signature 'NumericGap'
createResponseProcessing(object)

## S4 method for signature 'Order'
createResponseProcessing(object)

## S4 method for signature 'SingleChoice'
createResponseProcessing(object)

## S4 method for signature 'TextGapOpal'
createResponseProcessing(object)

```

Arguments

object an instance of the S4 object ([SingleChoice](#), [MultipleChoice](#), [Essay](#), [Entry](#), [Order](#), [OneInRowTable](#), [OneInColTable](#), [MultipleChoiceTable](#), [DirectedPair](#), [TextGap](#), [NumericGap](#), [InlineChoice](#))

createText	<i>Compose a set of html elements to display question in qti-xml document</i>
------------	---

Description

Generic function for creating a set of html elements to display question for XML document of specification the question following the QTI schema v2.1

Usage

```

createText(object)

## S4 method for signature 'Gap'
createText(object)

## S4 method for signature 'InlineChoice'
createText(object)

## S4 method for signature 'character'
createText(object)

```

Arguments

object an instance of the S4 object (Gap, InlineChoice, character)

createZip *Create an Zip archive of QTI test*

Description

Generic function for creating zip archive with set of XML documents of specification the test following the QTI schema v2.1

Usage

```
createZip(object, input, output, file_name, zip_only)
```

```
## S4 method for signature 'AssessmentTest'
createZip(object, input, output, file_name, zip_only)
```

```
## S4 method for signature 'AssessmentTestOpal'
createZip(object, input, output, file_name, zip_only)
```

Arguments

object an instance of the S4 object [AssessmentTest](#) or [AssessmentTestOpal](#)

input string, optional; a source folder with xml files

output string, optional; a folder to store zip and xml files; working directory by default

file_name string, optional; file name of zip archive

zip_only boolean, optional; returns only zip file in case of TRUE or zip, xml and downloads files in case of FALSE value

create_assessment_item *Compose a root element AssessmentItem of xml task*

Description

create_assessment_item() creates html structure with AssessmentItem root element (shiny.tag) for xml qti task description according QTI 2.1

Usage

```
create_assessment_item(object)
```

Arguments

object an instance of the S4 object

Value

A list() with a shiny.tag class

create_qti_task *Create XML file for question specification*

Description

Create XML file for question specification

Usage

```
create_qti_task(object,
                 dir = NULL,
                 verification = FALSE)
```

Arguments

object an instance of the S4 object ([SingleChoice](#), [MultipleChoice](#), [Essay](#), [Entry](#), [Order](#), [OneInRowTable](#), [OneInColTable](#), [MultipleChoiceTable](#), [DirectedPair](#)).

dir string, optional; a folder to store xml file; working directory by default

verification boolean, optional; to check validity of xml file, default FALSE

Value

xml document.

create_qti_test *Create XML file for exam test specification*

Description

Create XML file for exam test specification

Usage

```
create_qti_test(object, path = ".", verification = FALSE, zip_only
                = FALSE)
```

Arguments

object	an instance of the AssessmentTest S4 object
path	string, optional; a path to folder to store zip file with possible file name; working directory by default
verification	boolean, optional; to check validity of xml file, default FALSE
zip_only	boolean, optional; returns only zip file in case of TRUE or zip, xml and downloads files in case of FALSE value

Value

xml document.

DirectedPair-class *Class "DirectedPair"*

Description

Class `DirectedPair` is responsible for creating assessment tasks according to the QTI 2.1 standard, where a candidate has to make binary associations between answer options.

Slots

- `identifier` A character representing the unique identifier of the assessment task. By default, it is generated as `'id_task_ddd'`, where `ddd` represents random digits.
- `title` A character representing the title of the XML file associated with the task. By default, it takes the value of the identifier.
- `content` A list of character content to form the text of the question, which can include HTML tags. For tasks of the [Entry](#) type, it must also contain at least one instance of `Gap` objects, such as [TextGap](#), [NumericGap](#), or [InlineChoice](#).
- `prompt` An optional character representing a simple question text, consisting of one paragraph. This can supplement or replace content in the task. Default is `""`.
- `points` A numeric value, optional, representing the number of points for the entire task. Default is 1, but pay attention:
- For tasks of the [Entry](#) type, it is calculated as the sum of the gap points by default.
 - For tasks of the [MatchTable](#) type, it can also be calculated as the sum of points for individual answers, when provided.
 - For tasks of the [MultipleChoice](#) type, `points` is numeric vector and required. Each number in this vector determines the number of points that will be awarded to a candidate if they select the corresponding answer. The order of the scores must match the order of the choices. It is possible to assign negative values to incorrect answers. All answers with a positive score are considered correct.
- `feedback` A list containing feedback messages for candidates. Each element of the list should be an instance of either [ModalFeedback](#), [CorrectFeedback](#), or [WrongFeedback](#) class.

- calculator A character, optional, determining whether to show a calculator to the candidate. Possible values:
- "simple-calculator"
 - "scientific-calculator"
- files A character vector, optional, containing paths to files that will be accessible to the candidate during the test/exam.
- qti_version A character representing the QTI (Question and Test Interoperability) information model version. Default is 'v2p1'.
- rows A character vector specifying answer options as row names in the table or the first elements in couples in [DirectedPair](#).
- rows_identifiers A character vector, optional, specifying identifiers for answer options defined in rows of the table or identifiers of the first elements in couples in [DirectedPair](#).
- cols A character vector specifying answer options as column headers in the table or the second elements in couples in [DirectedPair](#).
- cols_identifiers A character vector, optional, specifying identifiers for answer options defined in columns of the table or identifiers of the second elements in couples in [DirectedPair](#).
- answers_identifiers A character vector specifying couples of identifiers that combine the correct answers.
- answers_scores A numeric vector, optional, where each number determines the number of points awarded to a candidate if they select the corresponding answer. If not assigned, the individual values for correct answers are calculated from the task points and the number of correct options.
- shuffle A boolean value, optional, determining whether to randomize the order in which the choices are initially presented to the candidate. Default is TRUE.
- shuffle_rows A boolean value, optional, determining whether to randomize the order of the choices only in rows. Default is TRUE.
- shuffle_cols A boolean value, optional, determining whether to randomize the order of the choices only in columns. Default is TRUE.
- orientation A character, optional, determining whether to place answers in vertical or horizontal mode. Possible values:
- "vertical" - Default.
 - "horizontal"

Examples

```
dp <- new("DirectedPair",
  identifier = "id_task_1234",
  title = "Directed pair",
  content = list("<p>\\"Directed pairs\\" task</p>"),
  points = 5,
  rows = c("row1", "row2", "row3"),
  rows_identifiers = c("a", "b", "c"),
  cols = c("alfa", "beta", "gamma"),
  cols_identifiers = c("k", "l", "m"),
  answers_identifiers = c("a k", "b l", 'c m'),
  shuffle = TRUE,
  orientation = "vertical")
```

dropdown *Create YAML string for InlineChoice object (dropdown list)*

Description

Create YAML string for InlineChoice object (dropdown list)

Usage

```
dropdown(  
  choices,  
  solution_index = 1,  
  points = 1,  
  shuffle = TRUE,  
  response_identifier = NULL  
)
```

Arguments

choices	A numeric or character vector; contains values of possible answers. If you use a named vector, the names will be used as identifiers.
solution_index	An integer value, optional; the number of right answer in the choices vector. Default is 1.
points	A numeric value, optional; the number of points for this gap. Default is 1.
shuffle	A boolean, optional; is responsible to randomize the order in which the choices are initially presented to the candidate. Default is TRUE.
response_identifier	A character string, optional; an identifier for the answer.

Value

A character string mapped as yaml.

See Also

[gap_text\(\)](#), [gap_numeric\(\)](#), [mdlist\(\)](#)

Examples

```
dropdown(c("Option A", "Option B"), response_identifier = "task_dd_list")
```

Entry-class

Class "Entry"

Description

Class Entry is responsible for creating assessment tasks according to the QTI 2.1 standard. These tasks include one or more instances of text input fields (with numeric or text answers) or dropdown lists.

Slots

identifier A character representing the unique identifier of the assessment task. By default, it is generated as 'id_task_ddd', where dddd represents random digits.

title A character representing the title of the XML file associated with the task. By default, it takes the value of the identifier.

content A list of character content to form the text of the question, which can include HTML tags. For tasks of the [Entry](#) type, it must also contain at least one instance of Gap objects, such as [TextGap](#), [NumericGap](#), or [InlineChoice](#).

prompt An optional character representing a simple question text, consisting of one paragraph. This can supplement or replace content in the task. Default is "".

points A numeric value, optional, representing the number of points for the entire task. Default is 1, but pay attention:

- For tasks of the [Entry](#) type, it is calculated as the sum of the gap points by default.
- For tasks of the [MatchTable](#) type, it can also be calculated as the sum of points for individual answers, when provided.
- For tasks of the [MultipleChoice](#) type, points is numeric vector and required. Each number in this vector determines the number of points that will be awarded to a candidate if they select the corresponding answer. The order of the scores must match the order of the choices. It is possible to assign negative values to incorrect answers. All answers with a positive score are considered correct.

feedback A list containing feedback messages for candidates. Each element of the list should be an instance of either [ModalFeedback](#), [CorrectFeedback](#), or [WrongFeedback](#) class.

calculator A character, optional, determining whether to show a calculator to the candidate. Possible values:

- "simple-calculator"
- "scientific-calculator"

files A character vector, optional, containing paths to files that will be accessible to the candidate during the test/exam.

qti_version A character representing the QTI (Question and Test Interoperability) information model version. Default is 'v2p1'.

See Also

[NumericGap](#), [TextGap](#), [InlineChoice](#)

Examples

```

entry_gaps <- new("Entry", content = list("<p>In mathematics, the common
logarithm is the logarithm with base", new("NumericGap",
      response_identifier = "numeric_1",
      solution = 10,
      placeholder = "it is a number"),
". It is also known as the decimal", new("TextGap",
      response_identifier = "text_1",
      solution = "logarithm",
      placeholder = "it is a text"),
".</p>"),
      title = "entry with number and text in answers",
      identifier = "entry_example")
entry_dropdown <- new("Entry", content = list("<p>In mathematics, the common
logarithm is the logarithm with base", new("InlineChoice",
      response_identifier = "numeric_1",
      choices = c("10", "7", "11")),
". It is also known as the decimal", new("InlineChoice",
      response_identifier = "text_1",
      choices = c("logarithm", "limit")),
".</p>"),
      title = "entry with dropdown lists for answers",
      identifier = "entry_example")

```

Essay-class

Class "Essay"

Description

Class Essay is responsible for creating essay type of assessment task according to QTI 2.1.

Slots

identifier A character representing the unique identifier of the assessment task. By default, it is generated as 'id_task_ddd', where dddd represents random digits.

title A character representing the title of the XML file associated with the task. By default, it takes the value of the identifier.

content A list of character content to form the text of the question, which can include HTML tags. For tasks of the [Entry](#) type, it must also contain at least one instance of Gap objects, such as [TextGap](#), [NumericGap](#), or [InlineChoice](#).

prompt An optional character representing a simple question text, consisting of one paragraph. This can supplement or replace content in the task. Default is "".

points A numeric value, optional, representing the number of points for the entire task. Default is 1, but pay attention:

- For tasks of the [Entry](#) type, it is calculated as the sum of the gap points by default.
- For tasks of the [MatchTable](#) type, it can also be calculated as the sum of points for individual answers, when provided.

- For tasks of the [MultipleChoice](#) type, `points` is numeric vector and required. Each number in this vector determines the number of points that will be awarded to a candidate if they select the corresponding answer. The order of the scores must match the order of the choices. It is possible to assign negative values to incorrect answers. All answers with a positive score are considered correct.

`feedback` A list containing feedback messages for candidates. Each element of the list should be an instance of either [ModalFeedback](#), [CorrectFeedback](#), or [WrongFeedback](#) class.

`calculator` A character, optional, determining whether to show a calculator to the candidate. Possible values:

- "simple-calculator"
- "scientific-calculator"

`files` A character vector, optional, containing paths to files that will be accessible to the candidate during the test/exam.

`qti_version` A character representing the QTI (Question and Test Interoperability) information model version. Default is 'v2p1'.

`expected_length` A numeric, optional. Responsible for setting the size of the text input field in the content delivery engine.

`expected_lines` A numeric, optional. Responsible for setting the number of rows of the text input field in the content delivery engine.

`words_max` A numeric, optional. Responsible for setting the maximum number of words that a candidate can write in the text input field.

`words_min` A numeric, optional. Responsible for setting the minimum number of words that a candidate should write in the text input field.

`data_allow_paste` A logical, optional. Determines whether it is possible for a candidate to copy text into the text input field. Default is FALSE.

Note

If 'ModalFeedback' is given, default values for slots related to the text input field are calculated automatically.

Examples

```
es <- new("Essay",
  identifier = "id_task_1234",
  title = "Essay Task",
  content = list("<p>Develop some idea and write it down in
                the text field</p>"),
  prompt = "Write your answer in text field",
  points = 1,
  feedback = list(),
  calculator = "scientific-calculator",
  files = "text_book.pdf",
  expected_length = 100,
  expected_lines = 5,
  words_max = 200,
  words_min = 10,
  data_allow_paste = FALSE)
```

extract_results	<i>Create data frame with test results</i>
-----------------	--

Description

The function `extract_results()` takes Opal zip archive "Export results" or xml file and creates two kinds of data frames (according to parameter 'level'), see the 'Details' section.

Usage

```
extract_results(file, level = "exercises", hide_filename = TRUE)
```

Arguments

<code>file</code>	A string with a path of the xml test result file.
<code>level</code>	A string with two possible values: exercises and items.
<code>hide_filename</code>	A boolean value, TRUE to hide original file names by default.

Value

A dataframe with attributes of the candidates outcomes and result variables.

Note

1. With option `level = "exercises"` data frame consists of columns:

- 'file' - name of the xml file with test results (to identify candidate)
- 'date' - date and time of test
- 'id_question' - question item identifier
- 'duration' - time in sec. what candidate spent on this item
- 'score_candidate' - points that were given to candidate after evaluation
- 'score_max' - max possible score for this question
- 'question_type' - the type of question
- 'is_answer_given' - TRUE if candidate gave the answer on question, otherwise FALSE
- 'title' - the values of attribute 'title' of assessment items

2. With option `level = "items"` data frame consists of columns:

- 'file' - name of the xml file with test results (to identify candidate)
- 'date' - date and time of test
- 'id_question' - question item identifier
- 'base_type' - type of answer (identifier, string or float)
- 'cardinalities' - defines whether this question is single, multiple or ordered -value
- 'qti_type' - specifies the type of the task

- 'id_answer' - identifier of each response variable
- 'expected_response' - values that considered as right responses for question
- 'candidate_response' - values that were given by candidate
- 'score_candidate' - points that were given to candidate after evaluation
- 'score_max' - max possible score for this question item
- 'is_response_correct' - TRUE if candidate gave the right response, otherwise FALSE
- 'title' - the values of attribute 'title' of assessment items

Examples

```
file <- system.file("test_results.zip", package='rqt')
df <- extract_results(file, level = "items")
```

Gap-class

Class "Gap"

Description

Abstract class Gap is not meant to be instantiated directly; instead, it serves as a base for derived classes such as [NumericGap](#), [TextGap](#), [TextGapOpal](#) and [InlineChoice](#).

Slots

`response_identifier` A character value representing an identifier for the answer. By default, it is generated as 'id_gap_ddd', where dddd represents random digits.

`points` A numeric value, optional, representing the number of points for this gap. Default is 1.

`placeholder` A character value, optional, responsible for placing helpful text in the text input field in the content delivery engine.

`expected_length` A numeric value, optional, responsible for setting the size of the text input field in the content delivery engine.

See Also

[NumericGap](#), [TextGap](#), [TextGapOpal](#) and [InlineChoice](#).

gap_numeric	<i>Create YAML string for NumericGap object</i>
-------------	---

Description

Create YAML string for NumericGap object

Usage

```
gap_numeric(
  solution,
  tolerance = 0,
  tolerance_type = "absolute",
  points = 1,
  response_identifier = NULL,
  include_lower_bound = TRUE,
  include_upper_bound = TRUE,
  expected_length = size_gap(solution),
  placeholder = NULL
)
```

Arguments

solution	A numeric value; contains right answer for this numeric entry.
tolerance	A numeric value, optional; specifies the value for up and low boundaries of tolerance rate for candidate answer. Default is 0.
tolerance_type	A character string, optional; specifies tolerance mode; possible values:"exact", "absolute" (by default), "relative".
points	A numeric value, optional; the number of points for this gap. Default is 1.
response_identifier	A character string, optional; an identifier for the answer.
include_lower_bound	A boolean, optional; specifies whether or not the lower bound is included in tolerance rate.
include_upper_bound	A boolean, optional; specifies whether or not the upper bound is included in tolerance rate.
expected_length	An integer value, optional; is responsible to set a size of text input field in content delivery engine.
placeholder	A character string, optional; is responsible to place some helpful text in text input field in content delivery engine.

Value

A character string mapped as yaml.

See Also

[gap_text\(\)](#), [dropdown\(\)](#), [mdlist\(\)](#)

Examples

```
gap_numeric(5.0, tolerance = 10, tolerance_type = "relative")
```

gap_text

Create YAML string for TextGap object

Description

Create YAML string for TextGap object

Usage

```
gap_text(
  solution,
  tolerance = NULL,
  case_sensitive = FALSE,
  points = 1,
  response_identifier = NULL,
  expected_length = size_gap(solution),
  placeholder = NULL
)
```

Arguments

solution	A character vector containing values considered as correct answers.
tolerance	An integer value, optional; defines the number of characters to tolerate spelling mistakes in evaluating candidate answers.
case_sensitive	A boolean, optional; determines whether the evaluation of the correct answer is case sensitive. Default is FALSE.
points	A numeric value, optional; the number of points for this gap. Default is 1.
response_identifier	A character string (optional) representing an identifier for the answer.
expected_length	An integer value, optional; sets the size of the text input field in the content delivery engine.
placeholder	A character string, optional; places helpful text in the text input field in the content delivery engine.

Value

A character string mapped as yaml.

See Also

[gap_numeric\(\)](#), [dropdown\(\)](#), [mdlist\(\)](#)

Examples

```
gap_text(c("Solution", "Solutions"), tolerance = 2)
```

getAssessmentItems *Get list of AssessmentItems for AssessmentSection*

Description

Generic function for

Usage

```
getAssessmentItems(object)

## S4 method for signature 'AssessmentItem'
getAssessmentItems(object)

## S4 method for signature 'AssessmentSection'
getAssessmentItems(object)

## S4 method for signature 'character'
getAssessmentItems(object)
```

Arguments

object an instance of the S4 object ([AssessmentSection](#), [AssessmentItem](#))

getCalculator-methods *Get value of the slot 'calculator'*

Description

Get value of the slot 'calculator'

Usage

```

getCalculator(object)

## S4 method for signature 'AssessmentItem'
getCalculator(object)

## S4 method for signature 'AssessmentSection'
getCalculator(object)

## S4 method for signature 'character'
getCalculator(object)

```

Arguments

object an instance of the S4 object ([SingleChoice](#), [MultipleChoice](#), [Essay](#), [Entry](#), [Order](#), [OneInRowTable](#), [OneInColTable](#), [MultipleChoiceTable](#), [DirectedPair](#), [TextGap](#), [NumericGap](#), [InlineChoice](#))

getFiles-methods	<i>Get file paths for attachment of test</i>
------------------	--

Description

Get file paths for attachment of test

Usage

```

getFiles(object)

## S4 method for signature 'AssessmentItem'
getFiles(object)

## S4 method for signature 'AssessmentSection'
getFiles(object)

## S4 method for signature 'character'
getFiles(object)

```

Arguments

object an instance of the S4 object ([SingleChoice](#), [MultipleChoice](#), [Essay](#), [Entry](#), [Order](#), [OneInRowTable](#), [OneInColTable](#), [MultipleChoiceTable](#), [DirectedPair](#), [TextGap](#), [NumericGap](#), [InlineChoice](#))

getIdentifier-methods *Get identifier*

Description

Get identifier

Usage

```
getIdentifier(object)

## S4 method for signature 'AssessmentItem'
getIdentifier(object)

## S4 method for signature 'AssessmentSection'
getIdentifier(object)

## S4 method for signature 'Gap'
getIdentifier(object)

## S4 method for signature 'character'
getIdentifier(object)
```

Arguments

object	an instance of the S4 object (SingleChoice , MultipleChoice , Essay , Entry , Order , OneInRowTable , OneInColTable , MultipleChoiceTable , DirectedPair , TextGap , NumericGap , InlineChoice)
--------	---

getObject-methods *Get object*

Description

Get object

Usage

```
getObject(object)

## S4 method for signature 'AssessmentItem'
getObject(object)

## S4 method for signature 'AssessmentSection'
getObject(object)
```

```
## S4 method for signature 'character'
getObject(object)
```

Arguments

object an instance of the S4 object ([SingleChoice](#), [MultipleChoice](#), [Essay](#), [Entry](#), [Order](#), [OneInRowTable](#), [OneInColTable](#), [MultipleChoiceTable](#), [DirectedPair](#), [TextGap](#), [NumericGap](#), [InlineChoice](#))

getPoints-methods *Get points from AssessmentItem object*

Description

Get points from AssessmentItem object

Usage

```
getPoints(object)

## S4 method for signature 'AssessmentItem'
getPoints(object)

## S4 method for signature 'AssessmentSection'
getPoints(object)

## S4 method for signature 'MultipleChoice'
getPoints(object)

## S4 method for signature 'character'
getPoints(object)
```

Arguments

object an instance of the S4 object ([SingleChoice](#), [MultipleChoice](#), [Essay](#), [Entry](#), [Order](#), [OneInRowTable](#), [OneInColTable](#), [MultipleChoiceTable](#), [DirectedPair](#), [TextGap](#), [NumericGap](#), [InlineChoice](#))

getResponse	<i>Get and process a piece of question content</i>
-------------	--

Description

Generic function to get and process a different types of question content (text with instances of gaps or dropdown lists) for XML document of specification the question following the QTI schema v2.1

Usage

```
getResponse(object)

## S4 method for signature 'InlineChoice'
getResponse(object)

## S4 method for signature 'NumericGap'
getResponse(object)

## S4 method for signature 'TextGap'
getResponse(object)

## S4 method for signature 'character'
getResponse(object)
```

Arguments

object	an instance of the S4 object (NumericGap, TextGap, InlineChoice, character)
--------	---

get_resources	<i>Get records of all current user's resources on LMS OPAL</i>
---------------	--

Description

Get records of all current user's resources on LMS OPAL

Usage

```
get_resources(api_user = NULL, api_password = NULL, endpoint = NULL)
```

Arguments

api_user	A character value of the username in the OPAL.
api_password	A character value of the password in the OPAL.
endpoint	A string of endpoint of LMS Opal; by default it is got from environment variable QTI_API_ENDPOINT. To set a global environment variable, you need to call <code>Sys.setenv(QTI_API_ENDPOINT='xxxxxxxxxxxxxxxx')</code> or you can put these command into <code>.Renviron</code> .

Value

A dataframe with attributes of user's resources.

Examples

```
df <- get_resources()
```

get_resource_url	<i>Create a URL using the resource's display name in LMS OPAL</i>
------------------	---

Description

Create a URL using the resource's display name in LMS OPAL

Usage

```
get_resource_url(  
  display_name,  
  endpoint = NULL,  
  api_user = NULL,  
  api_password = NULL  
)
```

Arguments

display_name	A length one character vector to entitle file in OPAL; file name without extension by default; optional.
endpoint	A string of endpoint of LMS Opal; by default it is got from environment variable QTI_API_ENDPOINT. To set a global environment variable, you need to call <code>Sys.setenv(QTI_API_ENDPOINT='xxxxxxxxxxxxxxxx')</code> or you can put these command into <code>.Renviron</code> .
api_user	A character value of the username in the OPAL.
api_password	A character value of the password in the OPAL.

Value

A string value of URL.

Examples

```
url <- get_resource_url("my test")
```

 InlineChoice-class *Class "InlineChoice"*

Description

Class `InlineChoice` is responsible for creating instances of dropdown lists as answer options in [Entry](#) type assessment tasks according to the QTI 2.1 standard.

Slots

`response_identifier` A character value representing an identifier for the answer. By default, it is generated as `'id_gap_dddd'`, where `dddd` represents random digits.

`points` A numeric value, optional, representing the number of points for this gap. Default is 1.

`placeholder` A character value, optional, responsible for placing helpful text in the text input field in the content delivery engine.

`expected_length` A numeric value, optional, responsible for setting the size of the text input field in the content delivery engine.

`choices` A character vector containing the answers shown in the dropdown list.

`solution_index` A numeric value, optional, representing the index of the correct answer in the options vector. Default is 1.

`choices_identifiers` A character vector, optional, containing a set of identifiers for answers. By default, identifiers are generated automatically according to the template "OptionD", where D is a letter representing the alphabetical order of the answer in the list.

`shuffle` A boolean value, optional, determining whether to randomize the order in which the choices are initially presented to the candidate. Default is `TRUE`.

Examples

```
dd <- new("InlineChoice",
  response_identifier = "id_gap_1234",
  points = 1,
  choices = c("answer1", "answer2", "answer3"),
  solution_index = 1,
  choices_identifiers = c("OptionA", "OptionB", "OptionC"),
  shuffle = TRUE)
```

 MatchTable-class *Class "MatchTable"*

Description

Abstract class `MatchTable` is not meant to be instantiated directly; instead, it serves as a base for derived classes such as [OneInRowTable](#), [OneInColTable](#), [MultipleChoiceTable](#), and [DirectedPair](#).

Slots

- identifier** A character representing the unique identifier of the assessment task. By default, it is generated as 'id_task_ddd', where dddd represents random digits.
- title** A character representing the title of the XML file associated with the task. By default, it takes the value of the identifier.
- content** A list of character content to form the text of the question, which can include HTML tags. For tasks of the [Entry](#) type, it must also contain at least one instance of Gap objects, such as [TextGap](#), [NumericGap](#), or [InlineChoice](#).
- prompt** An optional character representing a simple question text, consisting of one paragraph. This can supplement or replace content in the task. Default is "".
- points** A numeric value, optional, representing the number of points for the entire task. Default is 1, but pay attention:
- For tasks of the [Entry](#) type, it is calculated as the sum of the gap points by default.
 - For tasks of the [MatchTable](#) type, it can also be calculated as the sum of points for individual answers, when provided.
 - For tasks of the [MultipleChoice](#) type, points is numeric vector and required. Each number in this vector determines the number of points that will be awarded to a candidate if they select the corresponding answer. The order of the scores must match the order of the choices. It is possible to assign negative values to incorrect answers. All answers with a positive score are considered correct.
- feedback** A list containing feedback messages for candidates. Each element of the list should be an instance of either [ModalFeedback](#), [CorrectFeedback](#), or [WrongFeedback](#) class.
- calculator** A character, optional, determining whether to show a calculator to the candidate. Possible values:
- "simple-calculator"
 - "scientific-calculator"
- files** A character vector, optional, containing paths to files that will be accessible to the candidate during the test/exam.
- qti_version** A character representing the QTI (Question and Test Interoperability) information model version. Default is 'v2p1'.
- rows** A character vector specifying answer options as row names in the table or the first elements in couples in [DirectedPair](#).
- rows_identifiers** A character vector, optional, specifying identifiers for answer options defined in rows of the table or identifiers of the first elements in couples in [DirectedPair](#).
- cols** A character vector specifying answer options as column headers in the table or the second elements in couples in [DirectedPair](#).
- cols_identifiers** A character vector, optional, specifying identifiers for answer options defined in columns of the table or identifiers of the second elements in couples in [DirectedPair](#).
- answers_identifiers** A character vector specifying couples of identifiers that combine the correct answers.
- answers_scores** A numeric vector, optional, where each number determines the number of points awarded to a candidate if they select the corresponding answer. If not assigned, the individual values for correct answers are calculated from the task points and the number of correct options.

`shuffle` A boolean value, optional, determining whether to randomize the order in which the choices are initially presented to the candidate. Default is TRUE.

`shuffle_rows` A boolean value, optional, determining whether to randomize the order of the choices only in rows. Default is TRUE.

`shuffle_cols` A boolean value, optional, determining whether to randomize the order of the choices only in columns. Default is TRUE.

See Also

[OneInRowTable](#), [OneInColTable](#), [MultipleChoiceTable](#), [DirectedPair](#)

mdlist

Create a markdown list for answer options

Description

Create a markdown list for answer options

Usage

```
mdlist(vect, solutions = NULL, gaps = NULL)
```

Arguments

<code>vect</code>	A string or numeric vector of answer options for single/multiple choice task.
<code>solutions</code>	An integer value, optional; indexes of right answer options in <code>vect</code> .
<code>gaps</code>	numeric or string vector, optional; provides primitive gap description if there is a need to build a list of gaps.

Value

A markdown list.

See Also

[gap_text\(\)](#), [gap_numeric\(\)](#), [dropdown\(\)](#)

Examples

```
#list for multiple choice task
mdlist(c("A", "B", "C"), c(2, 3))
# it returns:
#- A
#- *B*
#- *C*
#list of gaps
mdlist(c("A", "B", "C"), c(2, 3), c(1, 2, 3))
# it returns:
```

```
#- A <gap>1</gap>
#- *B* <gap>2</gap>
#- *C* <gap>3</gap>
```

ModalFeedback-class *Class "ModalFeedback"*

Description

Class ModalFeedback is responsible for delivering feedback messages to the candidate, regardless of whether the answer was correct or incorrect.

Slots

`outcome_identifier` A character representing the unique identifier of the outcome declaration variable that relates to feedback. Default is "FEEDBACKMODAL".

`show` A boolean value, optional, determining whether to show (TRUE) or hide (FALSE) the modal feedback. Default is TRUE.

`title` A character value, optional, representing the title of the modal feedback window.

`content` A list of character content to form the text of the modal feedback, which can include HTML tags.

`identifier` A character value representing the identifier of the modal feedback item. Default is "modal_feedback".

Examples

```
fb <- new("ModalFeedback",
         title = "Possible solution",
         content = list("<b>Some explanation</b>"))
```

MultipleChoice-class *Class "MultipleChoice"*

Description

Class MultipleChoice is responsible for creating multiple choice assessment task according to QTI 2.1.

Slots

- identifier** A character representing the unique identifier of the assessment task. By default, it is generated as 'id_task_ddd', where dddd represents random digits.
- title** A character representing the title of the XML file associated with the task. By default, it takes the value of the identifier.
- content** A list of character content to form the text of the question, which can include HTML tags. For tasks of the [Entry](#) type, it must also contain at least one instance of Gap objects, such as [TextGap](#), [NumericGap](#), or [InlineChoice](#).
- prompt** An optional character representing a simple question text, consisting of one paragraph. This can supplement or replace content in the task. Default is "".
- points** A numeric value, optional, representing the number of points for the entire task. Default is 1, but pay attention:
- For tasks of the [Entry](#) type, it is calculated as the sum of the gap points by default.
 - For tasks of the [MatchTable](#) type, it can also be calculated as the sum of points for individual answers, when provided.
 - For tasks of the [MultipleChoice](#) type, points is numeric vector and required. Each number in this vector determines the number of points that will be awarded to a candidate if they select the corresponding answer. The order of the scores must match the order of the choices. It is possible to assign negative values to incorrect answers. All answers with a positive score are considered correct.
- feedback** A list containing feedback messages for candidates. Each element of the list should be an instance of either [ModalFeedback](#), [CorrectFeedback](#), or [WrongFeedback](#) class.
- calculator** A character, optional, determining whether to show a calculator to the candidate. Possible values:
- "simple-calculator"
 - "scientific-calculator"
- files** A character vector, optional, containing paths to files that will be accessible to the candidate during the test/exam.
- qti_version** A character representing the QTI (Question and Test Interoperability) information model version. Default is 'v2p1'.
- choices** A character vector defining a set of answer options in the question.
- choice_identifiers** A character vector, optional, containing a set of identifiers for answers. By default, identifiers are generated automatically according to the template "ChoiceD", where D is a letter representing the alphabetical order of the answer in the list.
- shuffle** A boolean value indicating whether to randomize the order in which the choices are initially presented to the candidate. Default is TRUE.
- orientation** A character, determining whether to place answers in vertical or horizontal mode. Possible values:
- "vertical" - Default.
 - "horizontal"

Examples

```
mc <- new("MultipleChoice",
  identifier = "id_task_1234",
  title = "Multiple Choice Task",
  content = list("<p>Pick up the right options</p>"),
  prompt = "Plain text, can be used instead of content",
  points = c(1, -1, 1, -1),
  feedback = list(new("WrongFeedback", content = list("Wrong answer")),
  qti_version = "v2p1"),
  calculator = "scientific-calculator",
  files = "text_book.pdf",
  qti_version = "v2p1",
  choices = c("option 1", "option 2", "option 3", "option 4"),
  choice_identifiers = c("ChoiceA", "ChoiceB", "ChoiceC", "ChoiceD"),
  shuffle = TRUE,
  orientation = "vertical")
```

MultipleChoiceTable-class

Class "MultipleChoiceTable"

Description

Class `MultipleChoiceTable` is responsible for creating assessment tasks according to the QTI 2.1 standard with a table of answer options, where many correct answers in each row and column are possible.

Slots

- `identifier` A character representing the unique identifier of the assessment task. By default, it is generated as `'id_task_ddd'`, where `ddd` represents random digits.
- `title` A character representing the title of the XML file associated with the task. By default, it takes the value of the identifier.
- `content` A list of character content to form the text of the question, which can include HTML tags. For tasks of the [Entry](#) type, it must also contain at least one instance of Gap objects, such as [TextGap](#), [NumericGap](#), or [InlineChoice](#).
- `prompt` An optional character representing a simple question text, consisting of one paragraph. This can supplement or replace content in the task. Default is `""`.
- `points` A numeric value, optional, representing the number of points for the entire task. Default is 1, but pay attention:
- For tasks of the [Entry](#) type, it is calculated as the sum of the gap points by default.
 - For tasks of the [MatchTable](#) type, it can also be calculated as the sum of points for individual answers, when provided.

- For tasks of the [MultipleChoice](#) type, `points` is numeric vector and required. Each number in this vector determines the number of points that will be awarded to a candidate if they select the corresponding answer. The order of the scores must match the order of the choices. It is possible to assign negative values to incorrect answers. All answers with a positive score are considered correct.

`feedback` A list containing feedback messages for candidates. Each element of the list should be an instance of either [ModalFeedback](#), [CorrectFeedback](#), or [WrongFeedback](#) class.

`calculator` A character, optional, determining whether to show a calculator to the candidate. Possible values:

- "simple-calculator"
- "scientific-calculator"

`files` A character vector, optional, containing paths to files that will be accessible to the candidate during the test/exam.

`qti_version` A character representing the QTI (Question and Test Interoperability) information model version. Default is 'v2p1'.

`rows` A character vector specifying answer options as row names in the table or the first elements in couples in [DirectedPair](#).

`rows_identifiers` A character vector, optional, specifying identifiers for answer options defined in rows of the table or identifiers of the first elements in couples in [DirectedPair](#).

`cols` A character vector specifying answer options as column headers in the table or the second elements in couples in [DirectedPair](#).

`cols_identifiers` A character vector, optional, specifying identifiers for answer options defined in columns of the table or identifiers of the second elements in couples in [DirectedPair](#).

`answers_identifiers` A character vector specifying couples of identifiers that combine the correct answers.

`answers_scores` A numeric vector, optional, where each number determines the number of points awarded to a candidate if they select the corresponding answer. If not assigned, the individual values for correct answers are calculated from the task points and the number of correct options.

`shuffle` A boolean value, optional, determining whether to randomize the order in which the choices are initially presented to the candidate. Default is TRUE.

`shuffle_rows` A boolean value, optional, determining whether to randomize the order of the choices only in rows. Default is TRUE.

`shuffle_cols` A boolean value, optional, determining whether to randomize the order of the choices only in columns. Default is TRUE.

`mapping` Do not use directly; values are initialized automatically. This slot contains a named numeric vector of points, where names correspond to all possible combinations of row and column identifiers.

Examples

```
mt <- new("MultipleChoiceTable",
  identifier = "id_task_1234",
  title = "Multiple choice table",
```

```

content = list("<p>Match table task</p>",
              "<i>table description</i>"),
points = 5,
rows = c("row1", "row2", "row3"),
rows_identifiers = c("a", "b", "c"),
cols = c("alfa", "beta", "gamma"),
cols_identifiers = c("a", "b", "c"),
answers_identifiers = c("a a", "b b", "b c"),
shuffle = TRUE)

```

NumericGap-class *Class "NumericGap"*

Description

Class `NumericGap` is responsible for creating instances of input fields with numeric type of answers in question [Entry](#) type assessment tasks according to the QTI 2.1 standard.

Slots

`response_identifier` A character value representing an identifier for the answer. By default, it is generated as `'id_gap_dddd'`, where `dddd` represents random digits.

`points` A numeric value, optional, representing the number of points for this gap. Default is 1.

`placeholder` A character value, optional, responsible for placing helpful text in the text input field in the content delivery engine.

`expected_length` A numeric value, optional, responsible for setting the size of the text input field in the content delivery engine.

`solution` A numeric value containing the correct answer for this numeric entry.

`tolerance` A numeric value, optional, specifying the value for the upper and lower boundaries of the tolerance rate for candidate answers. Default is 0.

`tolerance_type` A character value, optional, specifying the tolerance mode. Possible values:

- "exact"
- "absolute" - Default.
- "relative"

`include_lower_bound` A boolean value, optional, specifying whether the lower bound is included in the tolerance rate. Default is TRUE.

`include_upper_bound` A boolean value, optional, specifying whether the upper bound is included in the tolerance rate. Default is TRUE.

See Also

[Entry](#), [TextGap](#), [TextGapOpal](#) and [InlineChoice](#).

Examples

```
ng <- new("NumericGap",
  response_identifier = "id_gap_1234",
  points = 1,
  placeholder = "use this format xx.xxx",
  solution = 5,
  tolerance = 1,
  tolerance_type = "relative",
  include_lower_bound = TRUE,
  include_upper_bound = TRUE)
```

OneInColTable-class *Class "OneInColTable"*

Description

Class `OneInColTable` is responsible for creating assessment tasks according to the QTI 2.1 standard with a table of answer options, where only one correct answer in each column is possible.

Slots

`identifier` A character representing the unique identifier of the assessment task. By default, it is generated as `'id_task_ddd'`, where `ddd` represents random digits.

`title` A character representing the title of the XML file associated with the task. By default, it takes the value of the identifier.

`content` A list of character content to form the text of the question, which can include HTML tags. For tasks of the [Entry](#) type, it must also contain at least one instance of Gap objects, such as [TextGap](#), [NumericGap](#), or [InlineChoice](#).

`prompt` An optional character representing a simple question text, consisting of one paragraph. This can supplement or replace content in the task. Default is `""`.

`points` A numeric value, optional, representing the number of points for the entire task. Default is 1, but pay attention:

- For tasks of the [Entry](#) type, it is calculated as the sum of the gap points by default.
- For tasks of the [MatchTable](#) type, it can also be calculated as the sum of points for individual answers, when provided.
- For tasks of the [MultipleChoice](#) type, `points` is numeric vector and required. Each number in this vector determines the number of points that will be awarded to a candidate if they select the corresponding answer. The order of the scores must match the order of the choices. It is possible to assign negative values to incorrect answers. All answers with a positive score are considered correct.

`feedback` A list containing feedback messages for candidates. Each element of the list should be an instance of either [ModalFeedback](#), [CorrectFeedback](#), or [WrongFeedback](#) class.

`calculator` A character, optional, determining whether to show a calculator to the candidate. Possible values:

- "simple-calculator"
- "scientific-calculator"

`files` A character vector, optional, containing paths to files that will be accessible to the candidate during the test/exam.

`qti_version` A character representing the QTI (Question and Test Interoperability) information model version. Default is 'v2p1'.

`rows` A character vector specifying answer options as row names in the table or the first elements in couples in [DirectedPair](#).

`rows_identifiers` A character vector, optional, specifying identifiers for answer options defined in rows of the table or identifiers of the first elements in couples in [DirectedPair](#).

`cols` A character vector specifying answer options as column headers in the table or the second elements in couples in [DirectedPair](#).

`cols_identifiers` A character vector, optional, specifying identifiers for answer options defined in columns of the table or identifiers of the second elements in couples in [DirectedPair](#).

`answers_identifiers` A character vector specifying couples of identifiers that combine the correct answers.

`answers_scores` A numeric vector, optional, where each number determines the number of points awarded to a candidate if they select the corresponding answer. If not assigned, the individual values for correct answers are calculated from the task points and the number of correct options.

`shuffle` A boolean value, optional, determining whether to randomize the order in which the choices are initially presented to the candidate. Default is TRUE.

`shuffle_rows` A boolean value, optional, determining whether to randomize the order of the choices only in rows. Default is TRUE.

`shuffle_cols` A boolean value, optional, determining whether to randomize the order of the choices only in columns. Default is TRUE.

Examples

```
mt <- new("OneInColTable",
  identifier = "id_task_1234",
  title = "One in Col choice table",
  content = list("<p>\"One in col\" table task</p>",
    "<i>table description</i>"),
  points = 5,
  rows = c("row1", "row2", "row3", "row4"),
  rows_identifiers = c("a", "b", "c", "d"),
  cols = c("alfa", "beta", "gamma"),
  cols_identifiers = c("k", "l", "m"),
  answers_identifiers = c("a k", "d l", 'd m'),
  shuffle = TRUE)
```

OneInRowTable-class *Class "OneInRowTable"*

Description

Class OneInRowTable is responsible for creating assessment tasks according to the QTI 2.1 standard with a table of answer options, where only one correct answer in each row is possible.

Slots

identifier A character representing the unique identifier of the assessment task. By default, it is generated as 'id_task_ddd', where dddd represents random digits.

title A character representing the title of the XML file associated with the task. By default, it takes the value of the identifier.

content A list of character content to form the text of the question, which can include HTML tags. For tasks of the [Entry](#) type, it must also contain at least one instance of Gap objects, such as [TextGap](#), [NumericGap](#), or [InlineChoice](#).

prompt An optional character representing a simple question text, consisting of one paragraph. This can supplement or replace content in the task. Default is "".

points A numeric value, optional, representing the number of points for the entire task. Default is 1, but pay attention:

- For tasks of the [Entry](#) type, it is calculated as the sum of the gap points by default.
- For tasks of the [MatchTable](#) type, it can also be calculated as the sum of points for individual answers, when provided.
- For tasks of the [MultipleChoice](#) type, points is numeric vector and required. Each number in this vector determines the number of points that will be awarded to a candidate if they select the corresponding answer. The order of the scores must match the order of the choices. It is possible to assign negative values to incorrect answers. All answers with a positive score are considered correct.

feedback A list containing feedback messages for candidates. Each element of the list should be an instance of either [ModalFeedback](#), [CorrectFeedback](#), or [WrongFeedback](#) class.

calculator A character, optional, determining whether to show a calculator to the candidate. Possible values:

- "simple-calculator"
- "scientific-calculator"

files A character vector, optional, containing paths to files that will be accessible to the candidate during the test/exam.

qti_version A character representing the QTI (Question and Test Interoperability) information model version. Default is 'v2p1'.

rows A character vector specifying answer options as row names in the table or the first elements in couples in [DirectedPair](#).

rows_identifiers A character vector, optional, specifying identifiers for answer options defined in rows of the table or identifiers of the first elements in couples in [DirectedPair](#).

`cols` A character vector specifying answer options as column headers in the table or the second elements in couples in [DirectedPair](#).

`cols_identifiers` A character vector, optional, specifying identifiers for answer options defined in columns of the table or identifiers of the second elements in couples in [DirectedPair](#).

`answers_identifiers` A character vector specifying couples of identifiers that combine the correct answers.

`answers_scores` A numeric vector, optional, where each number determines the number of points awarded to a candidate if they select the corresponding answer. If not assigned, the individual values for correct answers are calculated from the task points and the number of correct options.

`shuffle` A boolean value, optional, determining whether to randomize the order in which the choices are initially presented to the candidate. Default is TRUE.

`shuffle_rows` A boolean value, optional, determining whether to randomize the order of the choices only in rows. Default is TRUE.

`shuffle_cols` A boolean value, optional, determining whether to randomize the order of the choices only in columns. Default is TRUE.

Examples

```
mt <- new("OneInRowTable",
  identifier = "id_task_1234",
  title = "One in Row choice table",
  content = list("<p>\\"One in row\\" table task</p>",
    "<i>table description</i>"),
  points = 5,
  rows = c("row1", "row2", "row3", "row4"),
  rows_identifiers = c("a", "b", "c", "d"),
  cols = c("alfa", "beta", "gamma"),
  cols_identifiers = c("k", "l", "m"),
  answers_identifiers = c("a k", "b l", "c l", "d m"),
  shuffle = TRUE)
```

Order-class

Class "Order"

Description

Class Order is responsible for creating assessment task according to QTI 2.1., where candidate has to place answers in a specific order

Slots

`identifier` A character representing the unique identifier of the assessment task. By default, it is generated as 'id_task_ddd', where dddd represents random digits.

`title` A character representing the title of the XML file associated with the task. By default, it takes the value of the identifier.

- content** A list of character content to form the text of the question, which can include HTML tags. For tasks of the [Entry](#) type, it must also contain at least one instance of Gap objects, such as [TextGap](#), [NumericGap](#), or [InlineChoice](#).
- prompt** An optional character representing a simple question text, consisting of one paragraph. This can supplement or replace content in the task. Default is "".
- points** A numeric value, optional, representing the number of points for the entire task. Default is 1, but pay attention:
- For tasks of the [Entry](#) type, it is calculated as the sum of the gap points by default.
 - For tasks of the [MatchTable](#) type, it can also be calculated as the sum of points for individual answers, when provided.
 - For tasks of the [MultipleChoice](#) type, points is numeric vector and required. Each number in this vector determines the number of points that will be awarded to a candidate if they select the corresponding answer. The order of the scores must match the order of the choices. It is possible to assign negative values to incorrect answers. All answers with a positive score are considered correct.
- feedback** A list containing feedback messages for candidates. Each element of the list should be an instance of either [ModalFeedback](#), [CorrectFeedback](#), or [WrongFeedback](#) class.
- calculator** A character, optional, determining whether to show a calculator to the candidate. Possible values:
- "simple-calculator"
 - "scientific-calculator"
- files** A character vector, optional, containing paths to files that will be accessible to the candidate during the test/exam.
- qti_version** A character representing the QTI (Question and Test Interoperability) information model version. Default is 'v2p1'.
- choices** A character vector containing the answers. The order of answers in the vector represents the correct response for the task.
- choices_identifiers** A character vector, optional, containing a set of identifiers for answers. By default, identifiers are generated automatically. By default, identifiers are generated automatically according to the template "ChoiceL", where L is a letter representing the alphabetical order of the answer in the list.
- shuffle** A boolean value indicating whether to randomize the order in which the choices are initially presented to the candidate. Default is TRUE.
- points_per_answer** A boolean value indicating the scoring method. If TRUE, each selected answer will be scored individually. If FALSE, only fully correct answers will be scored with the maximum score. Default is TRUE.

Examples

```
ord <- new("Order",
  identifier = "id_task_1234",
  title = "order",
  content = list("<p>Put these items in a right order</p>"),
  prompt = "",
  points = 2,
```

```

feedback = list(),
choices = c("first", "second", "third"),
choices_identifiers = c("ChoiceA", "ChoiceB", "ChoiceC"),
shuffle = TRUE,
points_per_answer = TRUE)

```

```
prepareQTIJSFiles-methods
```

Prepare files to render them with QTIJS

Description

Prepare files to render them with QTIJS

Usage

```

prepareQTIJSFiles(object, dir = NULL)

## S4 method for signature 'AssessmentItem'
prepareQTIJSFiles(object, dir = "")

## S4 method for signature 'AssessmentSection'
prepareQTIJSFiles(object, dir = NULL)

## S4 method for signature 'AssessmentTest'
prepareQTIJSFiles(object, dir = NULL)

## S4 method for signature 'character'
prepareQTIJSFiles(object, dir = NULL)

```

Arguments

object	an instance of AssessmentItem , AssessmentTest , AssessmentTestOpal , AssessmentSection , or string path to xml, rmd or md files
dir	QTIJS path

```
prepare_renderer
```

Prepare QTIJS renderer

Description

Starts server for QTIJS, returns path of QTIJS and the url of the server.

Usage

```
prepare_renderer()
```

qtijjs_path	<i>shortcut for the correct QTIJS path</i>
-------------	--

Description

shortcut for the correct QTIJS path

Usage

```
qtijjs_path()
```

render_opal	<i>Render Rmd directly in Opal via API</i>
-------------	--

Description

Render Rmd directly in Opal via API

Usage

```
render_opal(input, ...)
```

Arguments

input	(the path to the input Rmd document)
...	required for passing arguments when knitting

Details

Customize knit function in the Rmd file using the following YAML setting after the word `knit knit`:

```
rkti::render_opal.
```

Value

A list with the key, display name, and URL of the resource in Opal.

Examples

```
file <- system.file("exercises/sc1.Rmd", package='rkti')
render_opal(file)
```

render_qtjjs

Render an RMD/xml file or rqtj-object as qti xml with QTIJS

Description

Generates the qti xml file via rmd2xml. The xml is copied into the QTIJS folder of the package which transforms the xml into HTML. Finally, the HTML is displayed and the user can have a preview of the exercise or exam.

Usage

```
render_qtjjs(input, ...)
```

Arguments

input	(the path to the input Rmd/md/xml document or AssessmentItem , AssessmentTest , AssessmentTestOpal , AssessmentSection object)
...	required for passing arguments when knitting

Details

Requires a running QTIJS server, which can be started with `start_server()`. When loading the package `rqtj`, a server is started automatically.

The preview is automatically loaded into the RStudio viewer. Alternatively you can just open the browser in the corresponding local server which is displayed when rendering. Since the function is supposed to be called via the Knit-Button in RStudio, it defaults to the RStudio viewer pane.

Customize knit function in the Rmd file using the following YAML setting after the word `knit`:
`rqtj::render_qtjjs`.

Value

An URL of the corresponding local server to display the rendering result.

Examples

```
file <- system.file("exercises/sc1.Rmd", package='rqtj')
render_qtjjs(file)
```

render_xml	<i>Render a single xml file with QTIJS</i>
------------	--

Description

Uses QTIJS to render a single xml file in the RStudio viewer pane with a local server.

Usage

```
render_xml(input)
```

Arguments

input	input file
-------	------------

Value

nothing, has side effects

render_zip	<i>Render a zipped qti archive with QTIJS</i>
------------	---

Description

Uses QTIJS to render a zipped qti archive in the RStudio viewer pane with a local server.

Usage

```
render_zip(input)
```

Arguments

input	input file
-------	------------

Value

nothing, has side effects

rmd2xml *Create qti-XML task file from Rmd (md) description*

Description

Create XML file for question specification from Rmd (md) description according to qti 2.1 information model

Usage

```
rmd2xml(file, path = getwd(), verification = FALSE)
```

Arguments

file A string of path to file with markdown description of question.
path A string, optional; a folder to store xml file. Default is working directory.
verification A boolean value, optional; enable validation of the xml file. Default is FALSE.

Value

The path string to the xml file.

Examples

```
## Not run:  
# creates folder with xml (side effect)  
rmd2xml("task.Rmd", "target_folder", TRUE)  
  
## End(Not run)
```

rmd2zip *Create test zip file with one task xml file from Rmd (md) description*

Description

Create zip file with test, that contains one xml question specification generated from Rmd (md) description according to qti 2.1 information model

Usage

```
rmd2zip(file, path = getwd(), verification = FALSE)
```

Arguments

file A string of path to file with markdown description of question.
path A string, optional; a folder to store xml file. Default is working directory.
verification A boolean value, optional; enable validation of the xml file. Default is FALSE.

Value

The path string to the zip file.

Examples

```
## Not run:
# creates folder with zip (side effect)
rmd2zip("task.Rmd", "target_folder", TRUE)

## End(Not run)
```

section	<i>Create a section as part of a test content</i>
---------	---

Description

Create an AssessmentSection rqti-object as part of a test content

Usage

```
section(
  content,
  n_variants = 1L,
  seed_number = NULL,
  id = NULL,
  by = "variants",
  selection = NULL,
  title = character(0),
  time_limits = NA_integer_,
  visible = TRUE,
  shuffle = FALSE,
  max_attempts = NA_integer_,
  allow_comment = TRUE
)
```

Arguments

content	A character vector of Rmd, md, xml files, task- or section-objects.
n_variants	An integer value indicating the number of task variants to create from Rmd files. Default is 1.
seed_number	An integer vector, optional, specifying seed numbers to reproduce the result of calculations.
id	A character value, optional, serving as the identifier of the assessment section.
by	A character with two possible values: "variants" or "files", indicating the type of the test structure. Default is "variants".

selection	An integer value, optional, defining how many children of the section are delivered in the test. Default is NULL, meaning "no selection".
title	A character value, optional, representing the title of the section. If not provided, it defaults to identifier.
time_limits	An integer value, optional, controlling the amount of time a candidate is allowed for this part of the test.
visible	A boolean value, optional, indicating whether the title of this section is shown in the hierarchy of the test structure. Default is TRUE.
shuffle	A boolean value, optional, responsible for randomizing the order in which the assessment items and subsections are initially presented to the candidate. Default is FALSE.
max_attempts	An integer value, optional, enabling the maximum number of attempts allowed for a candidate to pass this section.
allow_comment	A boolean value, optional, enabling candidates to leave comments on each question of the section. Default is TRUE.

Value

An object of class [AssessmentSection](#).

See Also

[test\(\)](#), [test4opal\(\)](#)

Examples

```
sc <- new("SingleChoice", prompt = "Question", choices = c("A", "B", "C"))
es <- new("Essay", prompt = "Question")
# Since ready-made S4 "AssessmentItem" objects are taken, in this example a
#permanent section consisting of two tasks is created.
s <- section(c(sc, es), title = "Section with nonrandomized tasks")

# Since Rmd files with randomization of internal variables are taken,
#in this example 2 variants are created with a different seed number for each.
path <- system.file("rmarkdown/templates/", package='rqi')
file1 <- file.path(path, "singlechoice-simple/skeleton/skeleton.Rmd")
file2 <- file.path(path, "singlechoice-complex/skeleton/skeleton.Rmd")
s <- section(c(file1, file2), n_variants = 2,
title = "Section with two variants of tasks")
```

SingleChoice-class *Class "SingleChoice"*

Description

Class SingleChoice is responsible for creating single-choice assessment tasks according to the QTI 2.1 standard.

Slots

- identifier** A character representing the unique identifier of the assessment task. By default, it is generated as 'id_task_ddd', where dddd represents random digits.
- title** A character representing the title of the XML file associated with the task. By default, it takes the value of the identifier.
- content** A list of character content to form the text of the question, which can include HTML tags. For tasks of the [Entry](#) type, it must also contain at least one instance of Gap objects, such as [TextGap](#), [NumericGap](#), or [InlineChoice](#).
- prompt** An optional character representing a simple question text, consisting of one paragraph. This can supplement or replace content in the task. Default is "".
- points** A numeric value, optional, representing the number of points for the entire task. Default is 1, but pay attention:
- For tasks of the [Entry](#) type, it is calculated as the sum of the gap points by default.
 - For tasks of the [MatchTable](#) type, it can also be calculated as the sum of points for individual answers, when provided.
 - For tasks of the [MultipleChoice](#) type, points is numeric vector and required. Each number in this vector determines the number of points that will be awarded to a candidate if they select the corresponding answer. The order of the scores must match the order of the choices. It is possible to assign negative values to incorrect answers. All answers with a positive score are considered correct.
- feedback** A list containing feedback messages for candidates. Each element of the list should be an instance of either [ModalFeedback](#), [CorrectFeedback](#), or [WrongFeedback](#) class.
- calculator** A character, optional, determining whether to show a calculator to the candidate. Possible values:
- "simple-calculator"
 - "scientific-calculator"
- files** A character vector, optional, containing paths to files that will be accessible to the candidate during the test/exam.
- qti_version** A character representing the QTI (Question and Test Interoperability) information model version. Default is 'v2p1'.
- choices** A character vector defining a set of answer options in the question.
- choice_identifiers** A character vector, optional, containing a set of identifiers for answers. By default, identifiers are generated automatically according to the template "ChoiceD", where D is a letter representing the alphabetical order of the answer in the list.
- shuffle** A boolean value indicating whether to randomize the order in which the choices are initially presented to the candidate. Default is TRUE.
- orientation** A character, determining whether to place answers in vertical or horizontal mode. Possible values:
- "vertical" - Default.
 - "horizontal"
- solution** A numeric value, optional. Represents the index of the correct answer in the choices slot. By default, the first item in the choices slot is considered the correct answer. Default is 1.

Examples

```

sc <- new("SingleChoice",
  identifier = "id_task_1234",
  title = "Single Choice Task",
  content = list("<p>Pick up the right option</p>"),
  prompt = "Plain text, can be used instead of content",
  points = 2,
  feedback = list(new("WrongFeedback", content = list("Wrong answer")),
  qti_version = "v2p1"),
  calculator = "scientific-calculator",
  files = "text_book.pdf",
  qti_version = "v2p1",
  choices = c("option 1", "option 2", "option 3", "option 4"),
  choice_identifiers = c("ChoiceA", "ChoiceB", "ChoiceC", "ChoiceD"),
  shuffle = TRUE,
  orientation = "vertical",
  solution = 2)

```

start_server

Start QTIJS on a local server

Description

This function starts an http server with the QTIJS renderer. The renderer performs the conversion of qti.xml into HTML.

Usage

```
start_server()
```

Details

The server has to be started manually by the user, otherwise the Knit Button will not work. The Button starts a new session and invoking a server there does not make much sense.

Value

The URL string for QTIJS server.

Examples

```

## Not run:
# Initiated server in qtiViewer folder
start_server()
# Initiated server in a specific folder provided by the user. This folder
# contains the QTI renderer
start_server("/pathToTheQtiRenderer/")

## End(Not run)

```

stop_server	<i>Stop QTIJS local server</i>
-------------	--------------------------------

Description

Stop QTIJS local server

Usage

```
stop_server()
```

Value

nothing, has side effects

test	<i>Create a test</i>
------	----------------------

Description

Create an AssessmentTest rqti-object.

Usage

```
test(
  content,
  identifier = "test_identifier",
  title = "Test Title",
  time_limit = 90L,
  max_attempts = 1L,
  academic_grading = FALSE,
  grade_label = c(en = "Grade", de = "Note"),
  table_label = c(en = "Grade", de = "Note"),
  navigation_mode = "nonlinear",
  submission_mode = "individual",
  allow_comment = TRUE,
  rebuild_variables = TRUE
)
```

Arguments

content	A list containing AssessmentSection objects.
identifier	A character value indicating the identifier of the test file. Default is 'test_identifier'.
title	A character value, optional, representing the file title. Default is 'Test Title'.

<code>time_limit</code>	An integer value, optional, controlling the time given to a candidate for the test in minutes. Default is 90 minutes.
<code>max_attempts</code>	An integer value, optional, indicating the maximum number of attempts allowed for the candidate. Default is 1.
<code>academic_grading</code>	A boolean, optional; enables showing a grade to the candidate at the end of the testing according to the 5-point academic grade system as feedback. Default is FALSE.
<code>grade_label</code>	A character value, optional; a short message that shows with a grade in the final feedback; for multilingual use, it can be a named vector with two-letter ISO language codes as names (e.g., <code>c(en="Grade", de="Note")</code>); during test creation, it takes the value for the language of the operating system; <code>c(en="Grade", de="Note")</code> is default.
<code>table_label</code>	A character value, optional; a concise message to display as the column title of the grading table in the final feedback; for multilingual use, it can be a named vector with two-letter ISO language codes as names (e.g., <code>c(en="Grade", de="Note")</code>); during test creation, it takes the value for the language of the operating system; <code>c(en="Grade", de="Note")</code> is default.
<code>navigation_mode</code>	A character value, optional, determining the general paths that the candidate may have during the exam. Two mode options are possible: - 'linear': Candidate is not allowed to return to previous questions. - 'nonlinear': Candidate is free to navigate; used by default.
<code>submission_mode</code>	A character value, optional, determining when the candidate's responses are submitted for response processing. One of two mode options is possible: - 'individual': Submit candidates' responses on an item-by-item basis; used by default. - 'simultaneous': Candidates' responses are submitted all together by the end of the test.
<code>allow_comment</code>	A boolean, optional, enabling the candidate to leave comments in each question. Default is TRUE.
<code>rebuild_variables</code>	A boolean, optional, enabling the recalculation of variables and reshuffling the order of choices for each item-attempt. Default is TRUE.

Value

An [AssessmentTest](#) object.

See Also

[test4opal\(\)](#), [section\(\)](#), [AssessmentTest](#), [AssessmentSection](#)

Examples

```
sc <- new("SingleChoice", prompt = "Question", choices = c("A", "B", "C"))
es <- new("Essay", prompt = "Question")
s <- section(c(sc, es), title = "Section with nonrandomized tasks")
```

```
t <- test(s, title = "Example of the Exam", academic_grading = TRUE)
```

test4opal

Create a test for LMS OPAL

Description

Create an `AssessmentTestOpal` rqti-object.

Usage

```
test4opal(
  content,
  identifier = "test_identifier",
  title = "Test Title",
  time_limit = 90L,
  max_attempts = 1L,
  files = NULL,
  calculator = "scientific-calculator",
  academic_grading = FALSE,
  grade_label = c(en = "Grade", de = "Note"),
  table_label = c(en = "Grade", de = "Note"),
  navigation_mode = "nonlinear",
  submission_mode = "individual",
  allow_comment = TRUE,
  rebuild_variables = TRUE,
  show_test_time = TRUE,
  mark_items = TRUE,
  keep_responses = FALSE
)
```

Arguments

<code>content</code>	A list containing AssessmentSection objects.
<code>identifier</code>	A character value indicating the identifier of the test file. Default is 'test_identifier'.
<code>title</code>	A character value, optional, representing the file title. Default is 'Test Title'.
<code>time_limit</code>	An integer value, optional, controlling the time given to a candidate for the test in minutes. Default is 90 minutes.
<code>max_attempts</code>	An integer value, optional, indicating the maximum number of attempts allowed for the candidate. Default is 1.
<code>files</code>	A character vector, optional; paths to files that will be accessible to the candidate during the test/exam.
<code>calculator</code>	A character, optional; determines whether to show a calculator to the candidate. Possible values:

- 'simple-calculator'
- 'scientific-calculator' (assigned by default).

academic_grading	A boolean, optional; enables to show to candidate at the end of the testing a grade according to 5-point academic grade system as a feedback; Default is FALSE.
grade_label	A character value, optional; a short message that shows with a grade in the final feedback; for multilingual use, it can be a named vector with two-letter ISO language codes as names (e.g., c(en="Grade", de="Note")); during test creation, it takes the value for the language of the operating system; c(en="Grade", de="Note") is default.
table_label	A character value, optional; a concise message to display as the column title of the grading table in the final feedback; for multilingual use, it can be a named vector with two-letter ISO language codes as names (e.g., c(en="Grade", de="Note")); during test creation, it takes the value for the language of the operating system; c(en="Grade", de="Note") is default.
navigation_mode	A character value, optional, determining the general paths that the candidate may have during the exam. Two mode options are possible: - 'linear': Candidate is not allowed to return to previous questions. - 'nonlinear': Candidate is free to navigate; used by default.
submission_mode	A character value, optional, determining when the candidate's responses are submitted for response processing. One of two mode options is possible: - 'individual': Submit candidates' responses on an item-by-item basis; used by default. - 'simultaneous': Candidates' responses are submitted all together by the end of the test.
allow_comment	A boolean, optional, enabling the candidate to leave comments in each question. Default is TRUE.
rebuild_variables	A boolean, optional, enabling the recalculation of variables and reshuffling the order of choices for each item-attempt. Default is TRUE.
show_test_time	A boolean, optional, determining whether to show candidate elapsed processing time without a time limit. Default is TRUE.
mark_items	A boolean, optional, determining whether to allow candidate marking of questions. Default is TRUE.
keep_responses	A boolean, optional, determining whether to save the candidate's answers from the previous attempt. Default is FALSE.

Value

An [AssessmentTestOpal](#) object

See Also

[test\(\)](#), [section\(\)](#), [AssessmentTestOpal](#), [AssessmentSection](#)

Examples

```
sc <- new("SingleChoice", prompt = "Question", choices = c("A", "B", "C"))
es <- new("Essay", prompt = "Question")
s <- section(c(sc, es), title = "Section with nonrandomized tasks")
t <- test4opal(s, title = "Example of the Exam", academic_grading = TRUE,
show_test_time = FALSE)
```

TextGap-class

*Class "TextGap"***Description**

Class TextGap is responsible for creating instances of input fields with text type of answers in question [Entry](#) type assessment tasks according to the QTI 2.1 standard.

Slots

`response_identifier` A character value representing an identifier for the answer. By default, it is generated as 'id_gap_ddd', where dddd represents random digits.

`points` A numeric value, optional, representing the number of points for this gap. Default is 1.

`placeholder` A character value, optional, responsible for placing helpful text in the text input field in the content delivery engine.

`expected_length` A numeric value, optional, responsible for setting the size of the text input field in the content delivery engine.

`solution` A character vector containing the values considered as correct answers.

`case_sensitive` A boolean value, determining whether the evaluation of the correct answer is case sensitive. Default is FALSE.

See Also

[Entry](#), [NumericGap](#), [TextGapOpal](#) and [InlineChoice](#).

Examples

```
tg <- new("TextGap",
  response_identifier = "id_gap_1234",
  points = 2,
  placeholder = "do not put special characters",
  expected_length = 20,
  solution = c("answer", "answerr", "aanswer"),
  case_sensitive = FALSE)
```

TextGapOpal-class	Class "TextGapOpal"
-------------------	---------------------

Description

Class TextGapOpal is responsible for creating instances of input fields with text type of answers in question [Entry](#) type assessment tasks according to the QTI 2.1 standard for LMS Opal.

Slots

`response_identifier` A character value representing an identifier for the answer. By default, it is generated as 'id_gap_ddd', where dddd represents random digits.

`points` A numeric value, optional, representing the number of points for this gap. Default is 1.

`placeholder` A character value, optional, responsible for placing helpful text in the text input field in the content delivery engine.

`expected_length` A numeric value, optional, responsible for setting the size of the text input field in the content delivery engine.

`solution` A character vector containing the values considered as correct answers.

`case_sensitive` A boolean value, determining whether the evaluation of the correct answer is case sensitive. Default is FALSE.

`tolerance` A numeric value defining how many characters will be taken into account to tolerate spelling mistakes in the evaluation of candidate answers. Default is 0.

See Also

[Entry](#), [NumericGap](#), [TextGap](#) and [InlineChoice](#).

Examples

```
tgo <- new("TextGapOpal",
  response_identifier = "id_gap_1234",
  points = 2,
  placeholder = "do not put special characters",
  expected_length = 20,
  solution = "answer",
  case_sensitive = FALSE,
  tolerance = 1)
```

upload2opal	<i>Upload a resource on OPAL</i>
-------------	----------------------------------

Description

Function `upload2opal()` takes full prepared zip archive of QTI-test or QTI-task and uploads it to the OPAL. before calling `upload2opal()` authentication procedure has to be performed. See [auth_opal](#)

Usage

```
upload2opal(
  test,
  display_name = NULL,
  access = 4,
  overwrite = TRUE,
  endpoint = NULL,
  open_in_browser = TRUE,
  api_user = NULL,
  api_password = NULL
)
```

Arguments

<code>test</code>	A length one character vector of AssessmentTest , AssessmentTestOpal or AssessmentItem objects, Rmd or md files; required.
<code>display_name</code>	A length one character vector to entitle file in OPAL; file name without extension by default; optional.
<code>access</code>	An integer value, optional; it is responsible for publication status, where 1 - only those responsible for this learning resource; 2 - responsible and other authors; 3 - all registered users; 4 - registered users and guests. Default is 4.
<code>overwrite</code>	A boolean value; if the value is TRUE, if only one file with the specified display name is found, it will be overwritten. Default is TRUE.
<code>endpoint</code>	A string of endpoint of LMS Opal; by default it is got from environment variable <code>QTI_API_ENDPOINT</code> . To set a global environment variable, you need to call <code>Sys.setenv(QTI_API_ENDPOINT='xxxxxxxxxxxxxxxx')</code> or you can put these command into <code>.Renviron</code> .
<code>open_in_browser</code>	A boolean value; optional; it controls whether to open a URL in default browser. Default is TRUE.
<code>api_user</code>	A character value of the username in the OPAL.
<code>api_password</code>	A character value of the password in the OPAL.

Value

A list with the key, display name, and URL of the resource in Opal.

Authentication

To use OPAL API, you need to provide your OPAL-username and password. This package uses system credential store 'keyring' to store user's name and password. After the first successful access to the OPAL API, there is no need to specify the username and password again, they will be extracted from the system credential store

Examples

```
file <- system.file("exercises/sc1.Rmd", package='rqi')
upload2opal(file, "task 1", open_in_browser = FALSE)
```

WrongFeedback-class *Class "WrongFeedback"*

Description

Class WrongFeedback is responsible for delivering feedback messages to the candidate in case of an incorrect answer on the entire exercise.

Slots

`outcome_identifier` A character representing the unique identifier of the outcome declaration variable that relates to feedback. Default is "FEEDBACKMODAL".

`show` A boolean value, optional, determining whether to show (TRUE) or hide (FALSE) the modal feedback. Default is TRUE.

`title` A character value, optional, representing the title of the modal feedback window.

`content` A list of character content to form the text of the modal feedback, which can include HTML tags.

`identifier` A character value representing the identifier of the modal feedback item. Default is "incorrect".

Examples

```
wfb <- new("WrongFeedback",
          title = "Wrong answer",
          content = list("<b>Some demonstration</b>"))
```

Index

- AssessmentItem, [11](#), [15](#), [31](#), [50](#), [52](#), [65](#)
- AssessmentItem (AssessmentItem-class), [3](#)
- AssessmentItem-class, [3](#)
- AssessmentSection, [5–9](#), [11](#), [31](#), [50](#), [52](#), [56](#),
[59–62](#)
- AssessmentSection
(AssessmentSection-class), [4](#)
- AssessmentSection-class, [4](#)
- AssessmentTest, [5](#), [9](#), [12](#), [15](#), [19](#), [21](#), [50](#), [52](#),
[60](#), [65](#)
- AssessmentTest (AssessmentTest-class), [6](#)
- AssessmentTest-class, [6](#)
- AssessmentTestOpal, [5](#), [7](#), [12](#), [15](#), [19](#), [50](#), [52](#),
[62](#), [65](#)
- AssessmentTestOpal
(AssessmentTestOpal-class), [7](#)
- AssessmentTestOpal-class, [7](#)
- auth_opal, [9](#), [65](#)

- buildAssesmentSection, AssessmentItem
(buildAssessmentSection), [10](#)
- buildAssesmentSection, character
(buildAssessmentSection), [10](#)
- buildAssessmentSection, [10](#)
- buildAssessmentSection, AssessmentItem-method
(buildAssessmentSection), [10](#)
- buildAssessmentSection, AssessmentSection
(buildAssessmentSection), [10](#)
- buildAssessmentSection, AssessmentSection-method
(buildAssessmentSection), [10](#)
- buildAssessmentSection, character-method
(buildAssessmentSection), [10](#)

- Choice (Choice-class), [11](#)
- Choice-class, [11](#)
- CorrectFeedback, [4](#), [21](#), [24](#), [26](#), [38](#), [41](#), [43](#),
[45](#), [47](#), [49](#), [57](#)
- CorrectFeedback
(CorrectFeedback-class), [11](#)
- CorrectFeedback-class, [11](#)

- create_assessment_item, [19](#)
- create_qti_task, [20](#)
- create_qti_test, [20](#)
- createAssessmentTest, [12](#)
- createAssessmentTest, AssessmentTest
(createAssessmentTest), [12](#)
- createAssessmentTest, AssessmentTest-method
(createAssessmentTest), [12](#)
- createAssessmentTest, AssessmentTestOpal
(createAssessmentTest), [12](#)
- createAssessmentTest, AssessmentTestOpal-method
(createAssessmentTest), [12](#)
- createItemBody, [13](#)
- createItemBody, DirectedPair
(createItemBody), [13](#)
- createItemBody, DirectedPair-method
(createItemBody), [13](#)
- createItemBody, Entry (createItemBody),
[13](#)
- createItemBody, Entry-method
(createItemBody), [13](#)
- createItemBody, Essay (createItemBody),
[13](#)
- createItemBody, Essay-method
(createItemBody), [13](#)
- createItemBody, MultipleChoice
(createItemBody), [13](#)
- createItemBody, MultipleChoice-method
(createItemBody), [13](#)
- createItemBody, MultipleChoiceTable
(createItemBody), [13](#)
- createItemBody, MultipleChoiceTable-method
(createItemBody), [13](#)
- createItemBody, OneInColTable
(createItemBody), [13](#)
- createItemBody, OneInColTable-method
(createItemBody), [13](#)
- createItemBody, OneInRowTable
(createItemBody), [13](#)

- createItemBody, OneInRowTable-method
(createItemBody), 13
- createItemBody, Order (createItemBody),
13
- createItemBody, Order-method
(createItemBody), 13
- createItemBody, SingleChoice
(createItemBody), 13
- createItemBody, SingleChoice-method
(createItemBody), 13
- createOutcomeDeclaration, 14
- createOutcomeDeclaration, AssessmentItem
(createOutcomeDeclaration), 14
- createOutcomeDeclaration, AssessmentItem-method
(createOutcomeDeclaration), 14
- createOutcomeDeclaration, AssessmentTest
(createOutcomeDeclaration), 14
- createOutcomeDeclaration, AssessmentTest-method
(createOutcomeDeclaration), 14
- createOutcomeDeclaration, Entry
(createOutcomeDeclaration), 14
- createOutcomeDeclaration, Entry-method
(createOutcomeDeclaration), 14
- createOutcomeDeclaration, Gap-method
(createOutcomeDeclaration), 14
- createOutcomeDeclaration, TextGap
(createOutcomeDeclaration), 14
- createQtiTask (createQtiTask-methods),
14
- createQtiTask, AssessmentItem
(createQtiTask-methods), 14
- createQtiTask, AssessmentItem-method
(createQtiTask-methods), 14
- createQtiTask-methods, 14
- createQtiTest (createQtiTest-methods),
15
- createQtiTest, AssessmentItem
(createQtiTest-methods), 15
- createQtiTest, AssessmentItem-method
(createQtiTest-methods), 15
- createQtiTest, AssessmentTest
(createQtiTest-methods), 15
- createQtiTest, AssessmentTest-method
(createQtiTest-methods), 15
- createQtiTest, character
(createQtiTest-methods), 15
- createQtiTest, character-method
(createQtiTest-methods), 15
- createQtiTest-methods, 15
- createResponseDeclaration, 16
- createResponseDeclaration, AssessmentItem
(createResponseDeclaration), 16
- createResponseDeclaration, AssessmentItem-method
(createResponseDeclaration), 16
- createResponseDeclaration, Entry
(createResponseDeclaration), 16
- createResponseDeclaration, Entry-method
(createResponseDeclaration), 16
- createResponseDeclaration, Essay
(createResponseDeclaration), 16
- createResponseDeclaration, Essay-method
(createResponseDeclaration), 16
- createResponseDeclaration, InlineChoice
(createResponseDeclaration), 16
- createResponseDeclaration, InlineChoice-method
(createResponseDeclaration), 16
- createResponseDeclaration, MatchTable
(createResponseDeclaration), 16
- createResponseDeclaration, MatchTable-method
(createResponseDeclaration), 16
- createResponseDeclaration, MultipleChoice
(createResponseDeclaration), 16
- createResponseDeclaration, MultipleChoice-method
(createResponseDeclaration), 16
- createResponseDeclaration, MultipleChoiceTable
(createResponseDeclaration), 16
- createResponseDeclaration, MultipleChoiceTable-method
(createResponseDeclaration), 16
- createResponseDeclaration, NumericGap
(createResponseDeclaration), 16
- createResponseDeclaration, NumericGap-method
(createResponseDeclaration), 16
- createResponseDeclaration, Order
(createResponseDeclaration), 16
- createResponseDeclaration, Order-method
(createResponseDeclaration), 16
- createResponseDeclaration, SingleChoice
(createResponseDeclaration), 16
- createResponseDeclaration, SingleChoice-method
(createResponseDeclaration), 16
- createResponseDeclaration, TextGap
(createResponseDeclaration), 16
- createResponseDeclaration, TextGap-method
(createResponseDeclaration), 16
- createResponseProcessing, 17
- createResponseProcessing, AssessmentItem

- (createResponseProcessing), 17
- createResponseProcessing, AssessmentItem-method
 - (createResponseProcessing), 17
- createResponseProcessing, Entry
 - (createResponseProcessing), 17
- createResponseProcessing, Entry-method
 - (createResponseProcessing), 17
- createResponseProcessing, Essay
 - (createResponseProcessing), 17
- createResponseProcessing, Essay-method
 - (createResponseProcessing), 17
- createResponseProcessing, Gap
 - (createResponseProcessing), 17
- createResponseProcessing, Gap-method
 - (createResponseProcessing), 17
- createResponseProcessing, NumericGap
 - (createResponseProcessing), 17
- createResponseProcessing, NumericGap-method
 - (createResponseProcessing), 17
- createResponseProcessing, Order
 - (createResponseProcessing), 17
- createResponseProcessing, Order-method
 - (createResponseProcessing), 17
- createResponseProcessing, SingleChoice
 - (createResponseProcessing), 17
- createResponseProcessing, SingleChoice-method
 - (createResponseProcessing), 17
- createResponseProcessing, TextGapOpal
 - (createResponseProcessing), 17
- createResponseProcessing, TextGapOpal-method
 - (createResponseProcessing), 17
- createText, 18
- createText, character (createText), 18
- createText, character-method
 - (createText), 18
- createText, Gap (createText), 18
- createText, Gap-method (createText), 18
- createText, InlineChoice (createText), 18
- createText, InlineChoice-method
 - (createText), 18
- createZip, 19
- createZip, AssessmentTest (createZip), 19
- createZip, AssessmentTest-method
 - (createZip), 19
- createZip, AssessmentTestOpal
 - (createZip), 19
- createZip, AssessmentTestOpal-method
 - (createZip), 19
- DirectedPair, 5, 13–15, 17, 18, 20, 22, 32–34, 37–39, 43, 46–48
- DirectedPair (DirectedPair-class), 21
- DirectedPair-class, 21
- dropdown, 23
- dropdown(), 30, 31, 39
- Entry, 4, 5, 13–15, 17, 18, 20, 21, 24, 25, 32–34, 37, 38, 41, 42, 44, 45, 47, 49, 57, 63, 64
- Entry (Entry-class), 24
- Entry-class, 24
- Essay, 5, 13–15, 18, 20, 32–34
- Essay (Essay-class), 25
- Essay-class, 25
- extract_results, 27
- Gap (Gap-class), 28
- Gap-class, 28
- gap_numeric, 29
- gap_numeric(), 23, 31, 39
- gap_text, 30
- gap_text(), 23, 30, 39
- get_resource_url, 36
- get_resources, 35
- getAssessmentItems, 31
- getAssessmentItems, AssessmentItem
 - (getAssessmentItems), 31
- getAssessmentItems, AssessmentItem-method
 - (getAssessmentItems), 31
- getAssessmentItems, AssessmentSection
 - (getAssessmentItems), 31
- getAssessmentItems, AssessmentSection-method
 - (getAssessmentItems), 31
- getAssessmentItems, character
 - (getAssessmentItems), 31
- getAssessmentItems, character-method
 - (getAssessmentItems), 31
- getCalculator (getCalculator-methods), 31
- getCalculator, AssessmentItem
 - (getCalculator-methods), 31
- getCalculator, AssessmentItem-method
 - (getCalculator-methods), 31
- getCalculator, AssessmentSection
 - (getCalculator-methods), 31
- getCalculator, AssessmentSection-method
 - (getCalculator-methods), 31

- getCalculator, character
 - (getCalculator-methods), 31
- getCalculator, character-method
 - (getCalculator-methods), 31
- getCalculator-methods, 31
- getFiles (getFiles-methods), 32
- getFiles, AssessmentItem
 - (getFiles-methods), 32
- getFiles, AssessmentItem-method
 - (getFiles-methods), 32
- getFiles, AssessmentSection
 - (getFiles-methods), 32
- getFiles, AssessmentSection-method
 - (getFiles-methods), 32
- getFiles, character (getFiles-methods), 32
- getFiles, character-method
 - (getFiles-methods), 32
- getFiles-methods, 32
- getIdentifier (getIdentifier-methods), 33
- getIdentifier, AssessmentItem
 - (getIdentifier-methods), 33
- getIdentifier, AssessmentItem-method
 - (getIdentifier-methods), 33
- getIdentifier, AssessmentSection
 - (getIdentifier-methods), 33
- getIdentifier, AssessmentSection-method
 - (getIdentifier-methods), 33
- getIdentifier, character
 - (getIdentifier-methods), 33
- getIdentifier, character-method
 - (getIdentifier-methods), 33
- getIdentifier, Gap
 - (getIdentifier-methods), 33
- getIdentifier, Gap-method
 - (getIdentifier-methods), 33
- getIdentifier-methods, 33
- getObject (getObject-methods), 33
- getObject, AssessmentItem
 - (getObject-methods), 33
- getObject, AssessmentItem-method
 - (getObject-methods), 33
- getObject, AssessmentSection
 - (getObject-methods), 33
- getObject, AssessmentSection-method
 - (getObject-methods), 33
- getObject, character
 - (getObject-methods), 33
- getObject, character-method
 - (getObject-methods), 33
- getObject-methods, 33
- getPoints (getPoints-methods), 34
- getPoints, AssessmentItem
 - (getPoints-methods), 34
- getPoints, AssessmentItem-method
 - (getPoints-methods), 34
- getPoints, AssessmentSection
 - (getPoints-methods), 34
- getPoints, AssessmentSection-method
 - (getPoints-methods), 34
- getPoints, character
 - (getPoints-methods), 34
- getPoints, character-method
 - (getPoints-methods), 34
- getPoints, MultipleChoice
 - (getPoints-methods), 34
- getPoints, MultipleChoice-method
 - (getPoints-methods), 34
- getPoints-methods, 34
- getResponse, 35
- getResponse, character (getResponse), 35
- getResponse, character-method
 - (getResponse), 35
- getResponse, InlineChoice (getResponse), 35
- getResponse, InlineChoice-method
 - (getResponse), 35
- getResponse, NumericGap (getResponse), 35
- getResponse, NumericGap-method
 - (getResponse), 35
- getResponse, TextGap (getResponse), 35
- getResponse, TextGap-method
 - (getResponse), 35
- InlineChoice, 4, 14, 17, 18, 21, 24, 25, 28, 32–34, 38, 41, 42, 44, 45, 47, 49, 57, 63, 64
- InlineChoice (InlineChoice-class), 37
- InlineChoice-class, 37
- MatchTable-classs, 37
- MatchTable, 4, 21, 24, 25, 38, 41, 42, 45, 47, 49, 57
- MatchTable (MatchTable-class), 37
- mdlist, 39
- mdlist(), 23, 30, 31

- ModalFeedback, [4](#), [21](#), [24](#), [26](#), [38](#), [41](#), [43](#), [45](#), [47](#), [49](#), [57](#)
- ModalFeedback (ModalFeedback-class), [40](#)
- ModalFeedback-class, [40](#)
- MultipleChoice, [4](#), [5](#), [11](#), [13–15](#), [17](#), [18](#), [20](#), [21](#), [24](#), [26](#), [32–34](#), [38](#), [41](#), [43](#), [45](#), [47](#), [49](#), [57](#)
- MultipleChoice (MultipleChoice-class), [40](#)
- MultipleChoice-class, [40](#)
- MultipleChoiceTable, [5](#), [13–15](#), [17](#), [18](#), [20](#), [32–34](#), [37](#), [39](#)
- MultipleChoiceTable (MultipleChoiceTable-class), [42](#)
- MultipleChoiceTable-class, [42](#)
- NumericGap, [4](#), [14](#), [17](#), [18](#), [21](#), [24](#), [25](#), [28](#), [32–34](#), [38](#), [41](#), [42](#), [45](#), [47](#), [49](#), [57](#), [63](#), [64](#)
- NumericGap (NumericGap-class), [44](#)
- NumericGap-class, [44](#)
- OneInColTable, [5](#), [13–15](#), [17](#), [18](#), [20](#), [32–34](#), [37](#), [39](#)
- OneInColTable (OneInColTable-class), [45](#)
- OneInColTable-class, [45](#)
- OneInRowTable, [5](#), [13–15](#), [17](#), [18](#), [20](#), [32–34](#), [37](#), [39](#)
- OneInRowTable (OneInRowTable-class), [47](#)
- OneInRowTable-class, [47](#)
- Order, [5](#), [13–15](#), [17](#), [18](#), [20](#), [32–34](#)
- Order (Order-class), [48](#)
- Order-class, [48](#)
- prepare_renderer, [50](#)
- prepareQTIJSFiles (prepareQTIJSFiles-methods), [50](#)
- prepareQTIJSFiles,AssessmentItem (prepareQTIJSFiles-methods), [50](#)
- prepareQTIJSFiles,AssessmentItem-method (prepareQTIJSFiles-methods), [50](#)
- prepareQTIJSFiles,AssessmentSection (prepareQTIJSFiles-methods), [50](#)
- prepareQTIJSFiles,AssessmentSection-method (prepareQTIJSFiles-methods), [50](#)
- prepareQTIJSFiles,AssessmentTest (prepareQTIJSFiles-methods), [50](#)
- prepareQTIJSFiles,AssessmentTest-method (prepareQTIJSFiles-methods), [50](#)
- prepareQTIJSFiles,character-method (prepareQTIJSFiles-methods), [50](#)
- prepareQTIJSFiles-methods, [50](#)
- qtijs_path, [51](#)
- render_opal, [51](#)
- render_qtijs, [52](#)
- render_xml, [53](#)
- render_zip, [53](#)
- rmd2xml, [54](#)
- rmd2zip, [54](#)
- section, [55](#)
- section(), [5](#), [7](#), [9](#), [60](#), [62](#)
- SingleChoice, [5](#), [11](#), [13–15](#), [17](#), [18](#), [20](#), [32–34](#)
- SingleChoice (SingleChoice-class), [56](#)
- SingleChoice-class, [56](#)
- start_server, [58](#)
- stop_server, [59](#)
- test, [59](#)
- test(), [5](#), [7](#), [9](#), [56](#), [62](#)
- test4opal, [61](#)
- test4opal(), [5](#), [7](#), [9](#), [56](#), [60](#)
- TextGap, [4](#), [14](#), [17](#), [18](#), [21](#), [24](#), [25](#), [28](#), [32–34](#), [38](#), [41](#), [42](#), [44](#), [45](#), [47](#), [49](#), [57](#), [64](#)
- TextGap (TextGap-class), [63](#)
- TextGap-class, [63](#)
- TextGapOpal, [28](#), [44](#), [63](#)
- TextGapOpal (TextGapOpal-class), [64](#)
- TextGapOpal-class, [64](#)
- upload2opal, [65](#)
- WrongFeedback, [4](#), [21](#), [24](#), [26](#), [38](#), [41](#), [43](#), [45](#), [47](#), [49](#), [57](#)
- WrongFeedback (WrongFeedback-class), [66](#)
- WrongFeedback-class, [66](#)