

# Assessment plugin for Adapt

---

The assessment plugin enables a quiz to be configured based on settings in the course.json file. A quiz must always be set at Article level.

## Table of Contents

### Contents

Table of Contents .....	1
Introduction .....	2
The purpose of this document.....	2
Who this document is for .....	2
Prerequisites .....	2
What <i>isn't</i> covered .....	2
Getting started .....	2
Installing the plugin.....	2
Setting an Article as an Assessment .....	2
Important to Note.....	2
Quiz Settings .....	3
Banks .....	3
Randomise .....	3
Random Count .....	3
Start Blocks .....	3
End Blocks .....	3
Scoring and Feedbacks .....	4
Scoring.....	4
Passing score .....	4
Submit score .....	4
Feedback .....	4
SCORM .....	4

# Introduction

## The purpose of this document

This document will familiarise you with the assessment plugin, how to use it in your course, and how to configure to the requirements of the quiz.

## Who this document is for

This document is for anyone wishing to enable the assessment plugin for your own course.

## Prerequisites

This document assumes knowledge of the Adapt framework and its structure, as well as the ability to understand and edit JSON data. Knowledge of SCORM is helpful, but not essential.

## What *isn't* covered

This document covers neither the tools nor the skills required to edit JSON data. It doesn't cover the structure of the Adapt framework, or the screen ID convention used therein. As such, if you are unsure of any of the above, you should familiarise yourself elsewhere before attempting to install and use the assessment plugin.

# Getting started

## Installing the plugin

In order to use the plugin, it first has to be present in your course. To do this, simply copy the assessment folder into the root of your course. Once done, paste the following references into index.html and index\_lms.html, just before the line in which `core/js/app.js` is loaded:

```
<script src="plugins/assessment/templates/templates.js" type="text/javascript"></script>
<script src="plugins/assessment/js/QuizArticle.js" type="text/javascript"></script>
<script src="plugins/assessment/js/QuestionBank.js" type="text/javascript"></script>
<script src="plugins/assessment/js/result.js" type="text/javascript"></script>
```

## Setting an Article as an Assessment

At the article level in course.json, set:

```
"assessment":true
```

## Important to Note

All filtering of banks and randomisation of elements is done at block level

# Quiz Settings

## Banks

Syntax: "banks": "x,y,z" – where we want x number of question blocks in bank 1, y number of questions in bank 2, and z number of questions in bank z. Ex. "banks": "4,5,3"

Set at: Article level

When banks are used then the “bankID” property should be added to all question blocks to be added to the banks. The ids should start at 1 i.e. "bankID": 1 (set at block level).

To have no banks set "banks": "", or remove attribute completely from json.

## Randomise

Syntax: "randomise": true/false

Set at: Article level

This applies to both quizzes with and without banks i.e. if "randomise": false and using banks, the blocks will be displayed in order according to bankID. Set "randomise": true if you wish to mix up the order of the banks.

## Random Count

Syntax: "randomCount": n – where n is the number of question blocks required in the quiz

Set at: Article level

Random count applies when banks are switched off and not all questions are to be displayed. It is effectively a single bank where x from a possible y are displayed.

## Start Blocks

Syntax: "startBlocks": n – where n is the number of blocks that will be displayed before the randomised question blocks.

Set at: Article level

Start blocks are typically presentation components e.g. Text, but don't have to be. Default value is 0 if not set.

## End Blocks

Syntax: "endBlocks": n – where n is the number of blocks that will be displayed after the randomised question blocks.

Set at: Article level

End blocks usually consist of a Results block and possibly other presentation block to reinforce the learning. Default value is 1 if not set (for the results block).

## Scoring and Feedbacks

### Scoring

#### Passing score

Syntax: "passingScore":x – where x is the score required to pass the quiz. The default scoring mechanism is 1 point per question component answered correctly

Set at: On the result component

#### Submit score

Syntax: "submitScore":true/false – whether or not to submit the status to the LMS

Set at: On the result component

### Feedback

Feedback in the Results component is banded based on overall quiz score.

Syntax: "bands":[a,b,c,d] – where a is the score required to get the top banded feedback, b the 2<sup>nd</sup> top, and so on.

Set at: On the result component

Example: "bands":[100,80,50,0] – in this example a score of 100% would be required to get the top feedback, 80% for the 2<sup>nd</sup> feedback, and 50% for the 3<sup>rd</sup> feedback. A score below 50% would mean the 4<sup>th</sup> (bottom) feedback being shown.

These bands must correspond with the number of items in the feedback array, also set on the result component e.g. "feedback": ["feedback 1", "feedback 2", "feedback 3", "feedback 4"]

There can be any number of banded feedbacks and the required scores can be adjusted in the bands array.

### SCORM

To set lesson status based on the result of the quiz, 2 things are required:

- In the result component set: "assessment": true
- In spoor/scormData.json set "submitStatusQuizComplete":true and set the desired values for "onQuizFailure" and "onQuizPassed"