



Jennifer Gupta

xAPI

A tutorial on how to use Storyline 360's inbuilt xAPI features



What data is collected automatically?



Examples



Links to test LRS/LMS for you to play with



Youtube video tutorials to help you



Publish Settings



Troubleshooting issues

What data is collected automatically?



Jennifer Gupta

Storyline will automatically record every time a learner begins and leaves a slide - it will use the experienced and left verbs, respectively. It also automatically records how long a user was on each slide.

2022-02-04T20:33:14.540	Jennifer Gupta left 'More Terms and Concepts'
2022-02-04T20:33:10.635	Jennifer Gupta experienced 'More Terms and Concepts'
2022-02-04T20:33:10.318	Jennifer Gupta left 'Using the Video Player'
2022-02-04T20:33:07.077	Jennifer Gupta experienced 'Using the Video Player'
2022-02-04T20:33:07.015	Jennifer Gupta left 'Welcome'
2022-02-04T20:33:01.484	Jennifer Gupta experienced 'Welcome'

Storyline will automatically record (correctly answered) or (incorrectly answered) for any graded question that exists in your course. It will also provide the score points associated with that question.

2022-02-04T20:26:54.126 Jennifer Gupta **correctly answered** 'Drag and drop each term in the right column next to its corresponding description in the left column.' with response 'statement_6RwVSc3GMNv[.jcho...' with score 1

Clicking on the bold words (correctly answered) in the statement above, opens a drop down with further information.

```
"version": "1.0.0",
"object": {
  "id": "http://64S8ajJ1dDM_course_id/6nc6MBxMu85/6SurgljuKwX",
  "definition": {
    "name": {
      "und": "Drag and drop each term in the right column next to its corresponding d
    },
    "description": {
      "und": "Drag and drop each term in the right column next to its corresponding d
    },
    "type": "http://adlnet.gov/expapi/activities/cmi.interaction",
    "target": [
      {
        "id": "choice_60PVicFQMio",
        "description": {
          "und": "Case"
        }
      },
      {
        "id": "choice_6ZMqKIENpir",
        "description": {
          "und": "Work Order"
        }
      },
      {
        "id": "choice_6RzE8W3m9o7",
        "description": {
          "und": "BPF (Business Process Flow)"
        }
      },
      {
        "id": "choice_6nYoctrL30l",
        "description": {
          "und": "Identify"
        }
      },
      {
        "id": "choice_6LPqBlmR86b",
```

This information contains the full details of the xAPI statement sent, which INCLUDES the answers the learner selected. This is where you will tell what a learner

answered, if they answered questions incorrectly. This information is available on the LMS as well as in xAPI form.

Examples

JG Jennifer Gupta

Trigger to alert a user has pressed a button

The screenshot displays the 'xAPI Statement Editor' and 'Trigger Wizard' windows. The Trigger Wizard is configured with the following settings:

- Action:** Send xAPI statement
- Statement:** Custom xAPI Statement
- When:** When the user clicks
- Object:** Retry Button
- Conditions:** + if

The xAPI Statement Editor shows the following JSON statement:

```
1 {  
2   "verb": {  
3     "id": "http://adlnet.gov/expapi/verbs/interacted",  
4     "display": {  
5       "en-US": "Interacted"  
6     }  
7   },  
8   "object": {  
9     "objectType": "Activity",  
10    "definition": {  
11      "name": {  
12        "en-US": "%Project.SlideTitle%"  
13      }  
14    },  
15    "id": "%Project.ActivityId%/%Slide.Id%"  
16  },  
17  "result": {  
18    "duration": "%Slide.ElapsedTime%",  
19    "response": "The user retried the assessment"  
20  }  
21 }
```

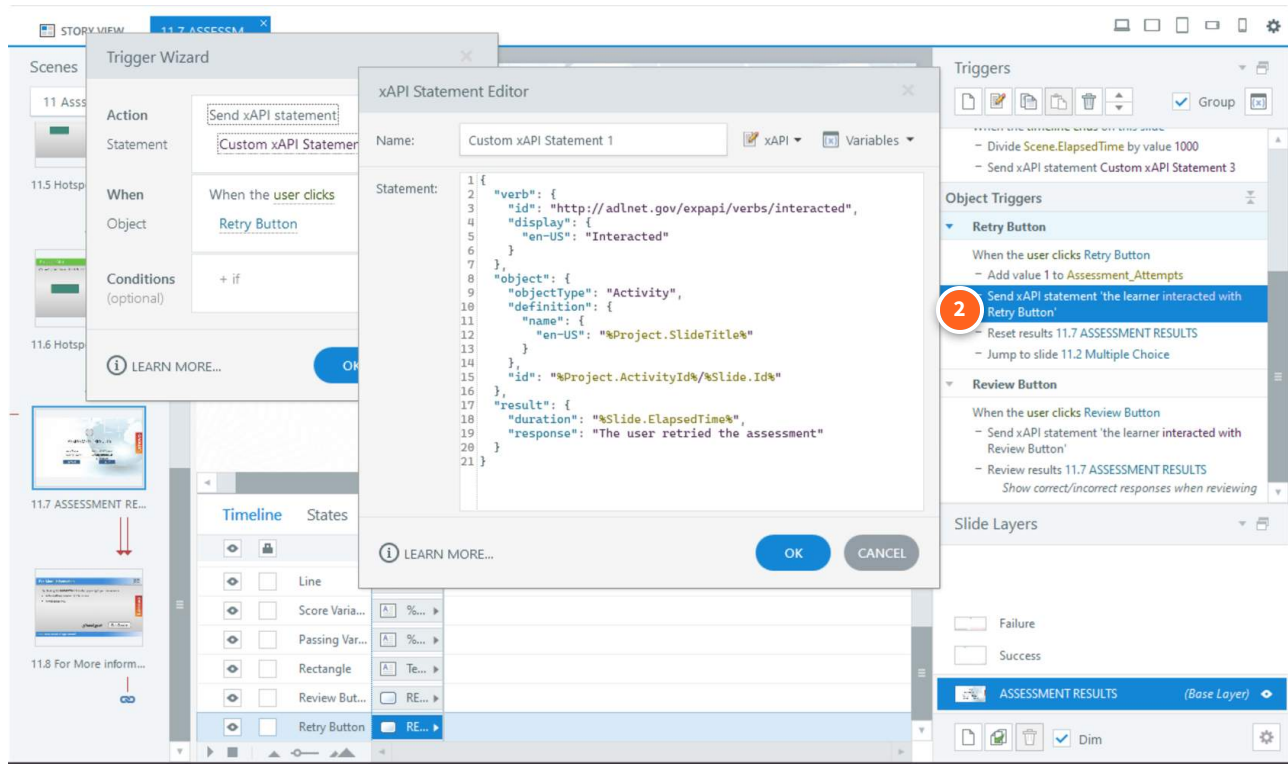
The Triggers panel on the right shows the following configuration for the 'Retry Button' trigger:

- When the user clicks Retry Button
- Add value 1 to Assessment_Attempts
- Send xAPI statement 'the learner interacted with Retry Button'
- Reset results 11.7 ASSESSMENT RESULTS
- Jump to slide 11.2 Multiple Choice

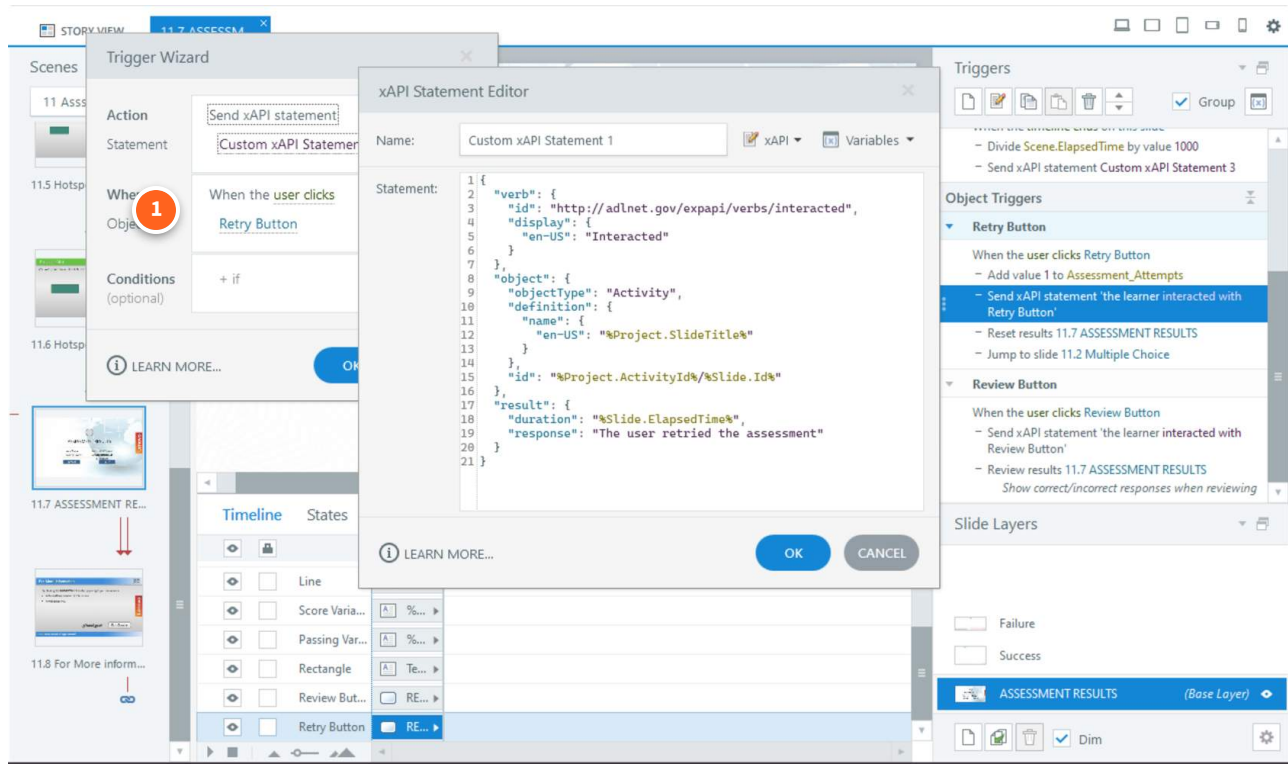
The 'Review Button' trigger is also visible, configured with:

- When the user clicks Review Button
- Send xAPI statement 'the learner interacted with Review Button'
- Review results 11.7 ASSESSMENT RESULTS
- Show correct/incorrect responses when reviewing

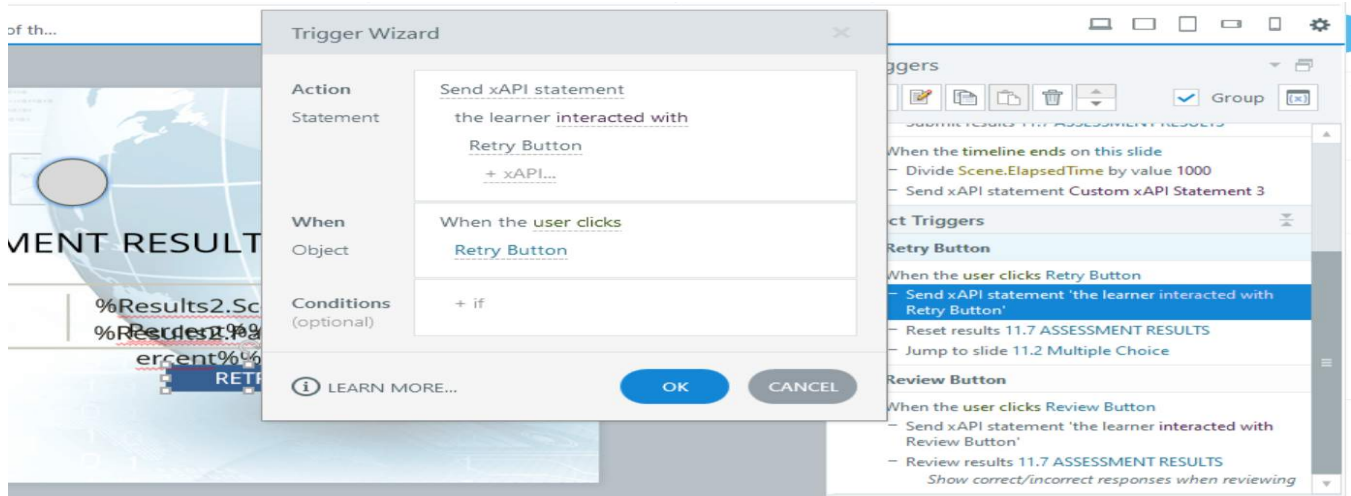
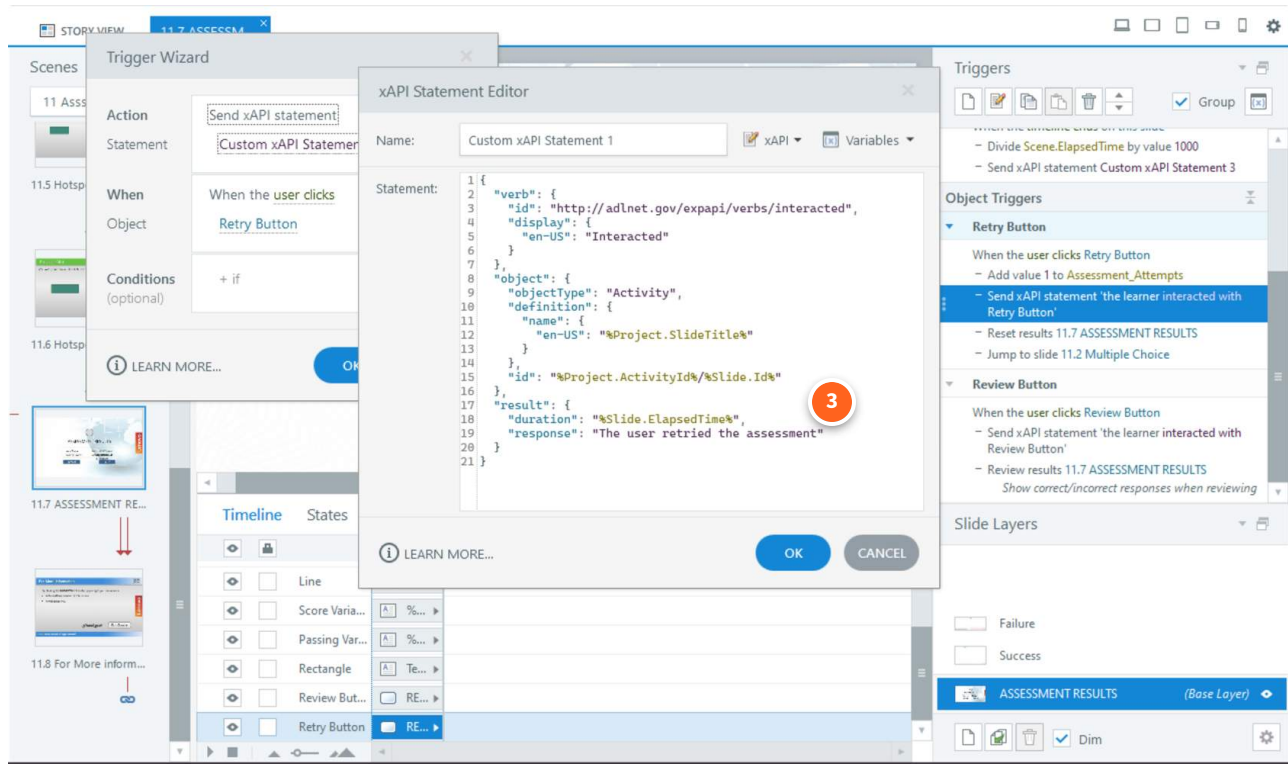
Numbered callouts 1, 2, and 3 highlight the 'When' section, the 'Send xAPI statement' action, and the JSON statement respectively.



To begin our statement, we send a statement to the LRS that says that the learner has interacted with the slide –and we send this WHEN they click the Retry button



Set the statement to occur before you submit any results or jump to any other slides, but after you have updated any variables that need to be set.



We have to ADD in a part - the RESULT. In this, the duration is set automatically, but we are adding in the "response" key:text pairing. We can write anything in the text field (as long as it's in "quotes"). Here, I used plain English to tell me what happened.

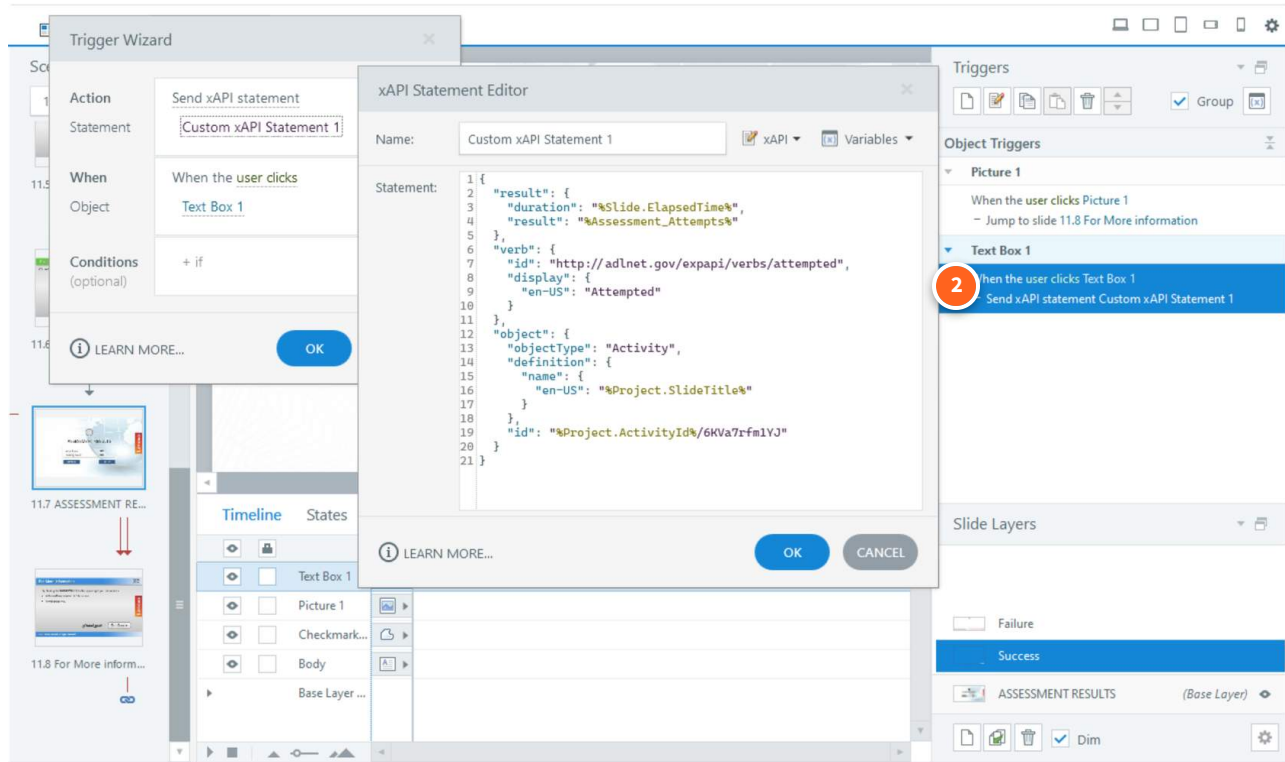
In the above image, I kept the statement set to Send xAPI statement when the learner interacted with (SLIDE NAME) for consistency. In this way, when I query the LRS to see what was clicked and when, I can query by the slide name itself. It is my recommendation that we adopt this as standard procedure.

Trigger to send variable information

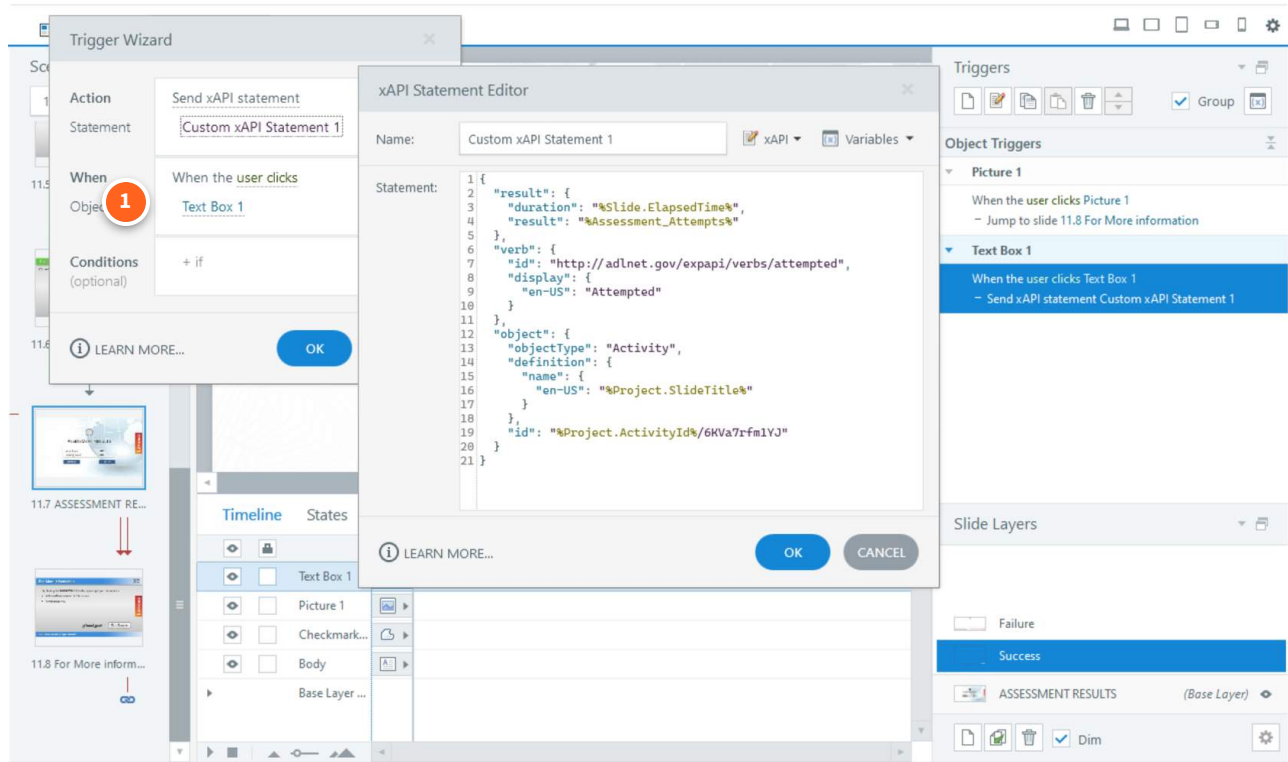
The image shows a software interface with three main components:

- Trigger Wizard:** Located at the top left, it has a tab labeled "Trigger Wizard". The "Action" is "Send xAPI statement" and the "Statement" is "Custom xAPI Statement 1". The "When" section is "When the user clicks" and the "Object" is "Text Box 1" (marked with a red circle 1). The "Conditions" section is "+ if". There is an "OK" button and a "LEARN MORE..." link.
- xAPI Statement Editor:** Located in the center, it has a tab labeled "xAPI Statement Editor". The "Name" is "Custom xAPI Statement 1". The "Statement" is a JSON object (marked with a red circle 3):

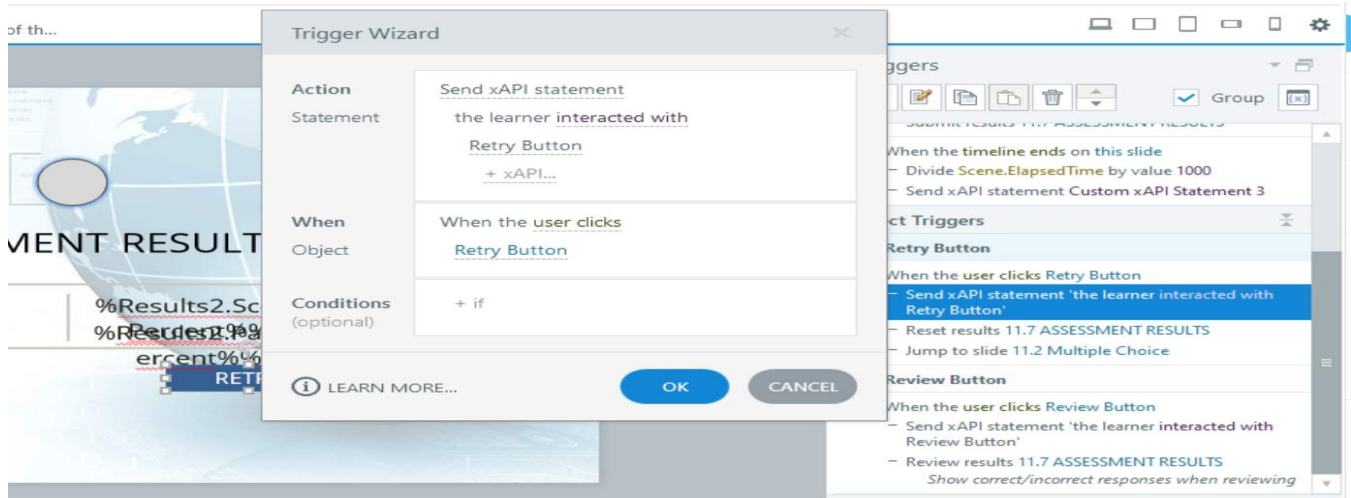
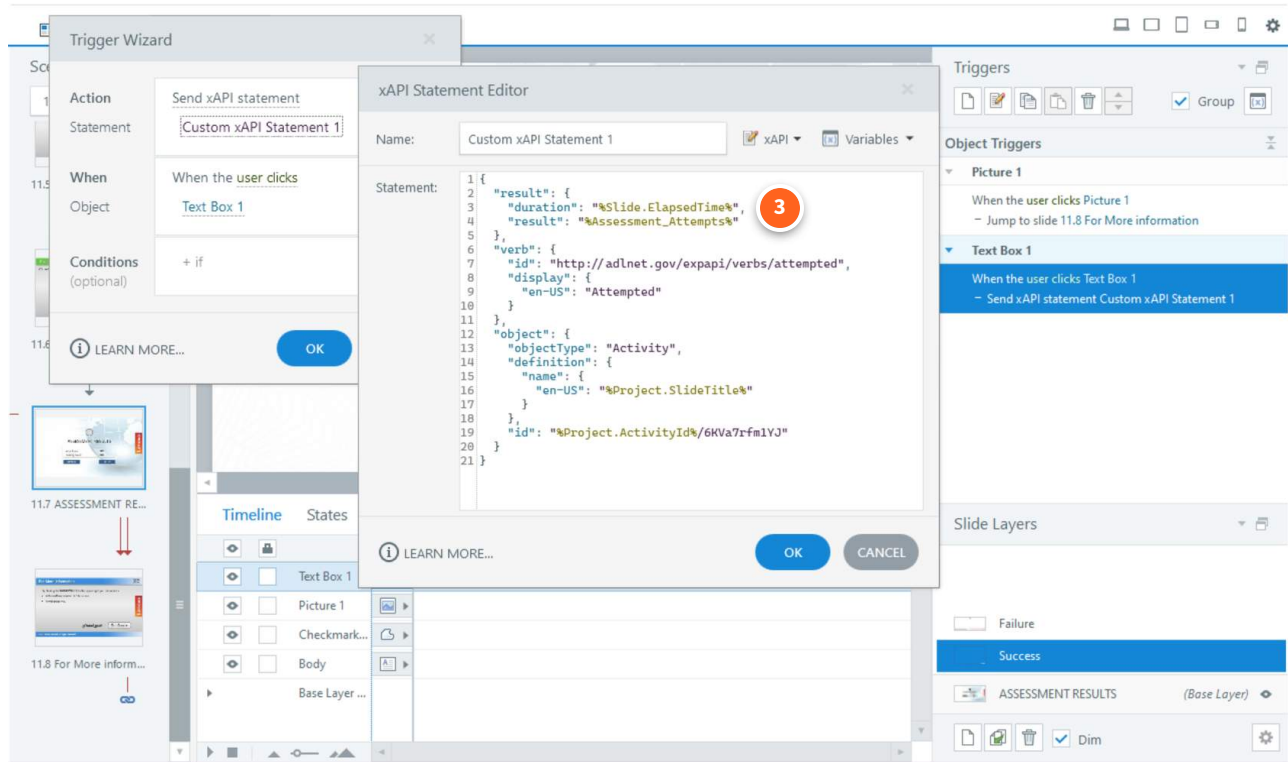
```
1 {
2   "result": {
3     "duration": "%Slide.ElapsedTime%",
4     "result": "%Assessment_Attempts%"
5   },
6   "verb": {
7     "id": "http://adlnet.gov/expapi/verbs/attempted",
8     "display": {
9       "en-US": "Attempted"
10    }
11  },
12  "object": {
13    "objectType": "Activity",
14    "definition": {
15      "name": {
16        "en-US": "%Project.SlideTitle%"
17      }
18    },
19    "id": "%Project.ActivityId%/6KVa7rFm1YJ"
20  }
21 }
```
- Triggers:** Located on the right, it has a tab labeled "Triggers". The "Object Triggers" section shows a trigger for "Text Box 1" (marked with a red circle 2):
 - When the user clicks Picture 1
 - Jump to slide 11.8 For More information
 - When the user clicks Text Box 1
 - Send xAPI statement Custom xAPI Statement 1



To begin our statement, we send a statement to the LRS that says that the learner has interacted with the slide -and we send this WHEN they click the Continue button -labelled here as Text Box 1 (bad name - this should be more specific!)

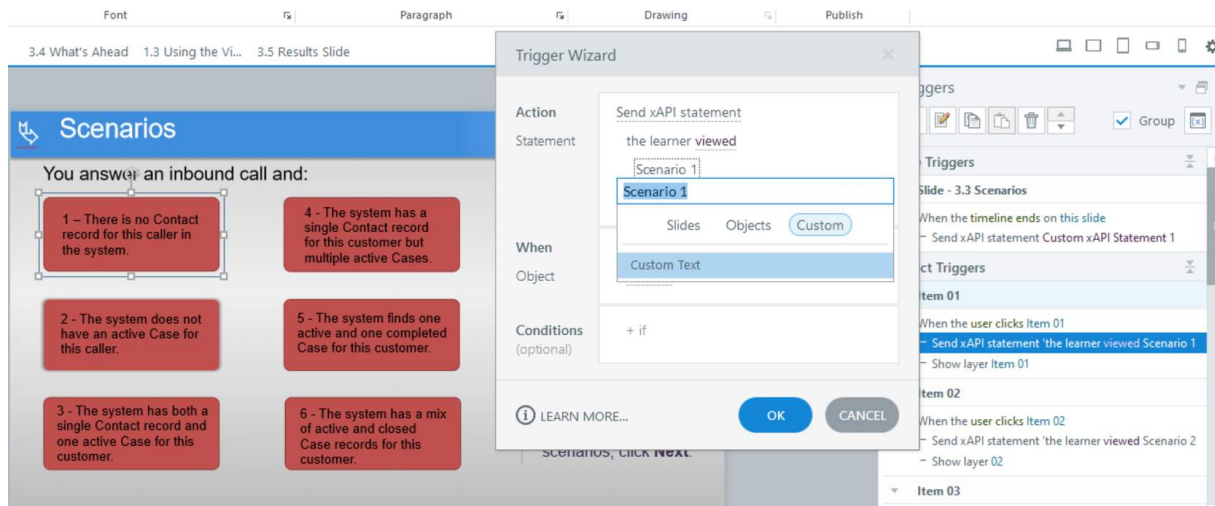


Set the statement to occur before you submit any results or jump to any other slides, but after you have updated any variables that need to be set.



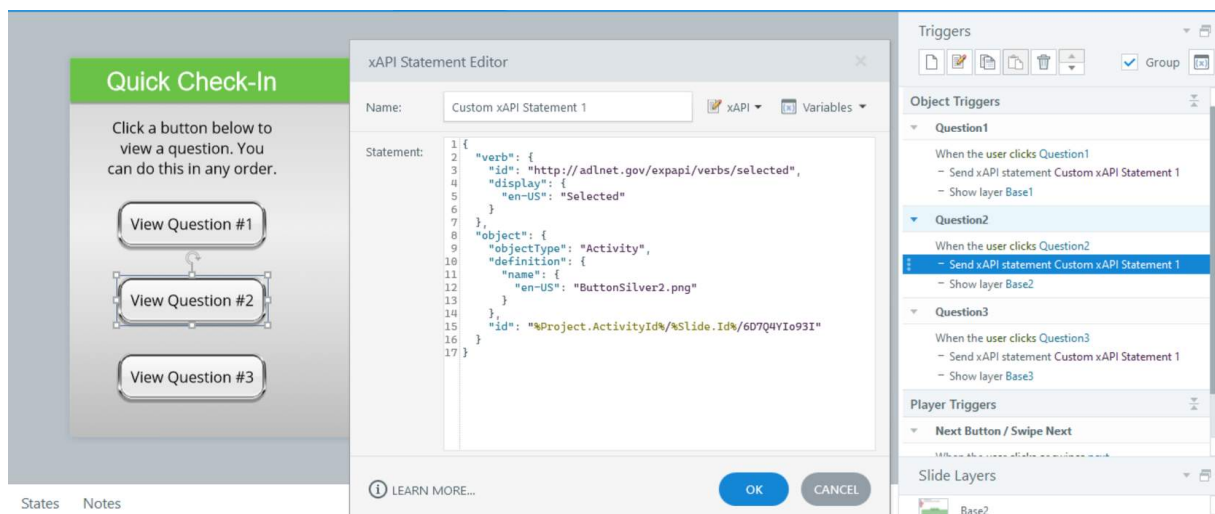
We have to ADD in a part - the RESULT. In this, the duration is set automatically, but we are adding in the "response" key:text pairing. We can write anything in the text field (as long as it's in "quotes"). Here, I used the Variables pull down to select the Assessment_Attempts variable that I created. It increments a number every time the user presses the retry button from the base slide.

There may be some instances when we want to trigger an xAPI statement based on the button name, group name or something other than the slide name. Here are a few tips if you need/want to do this.



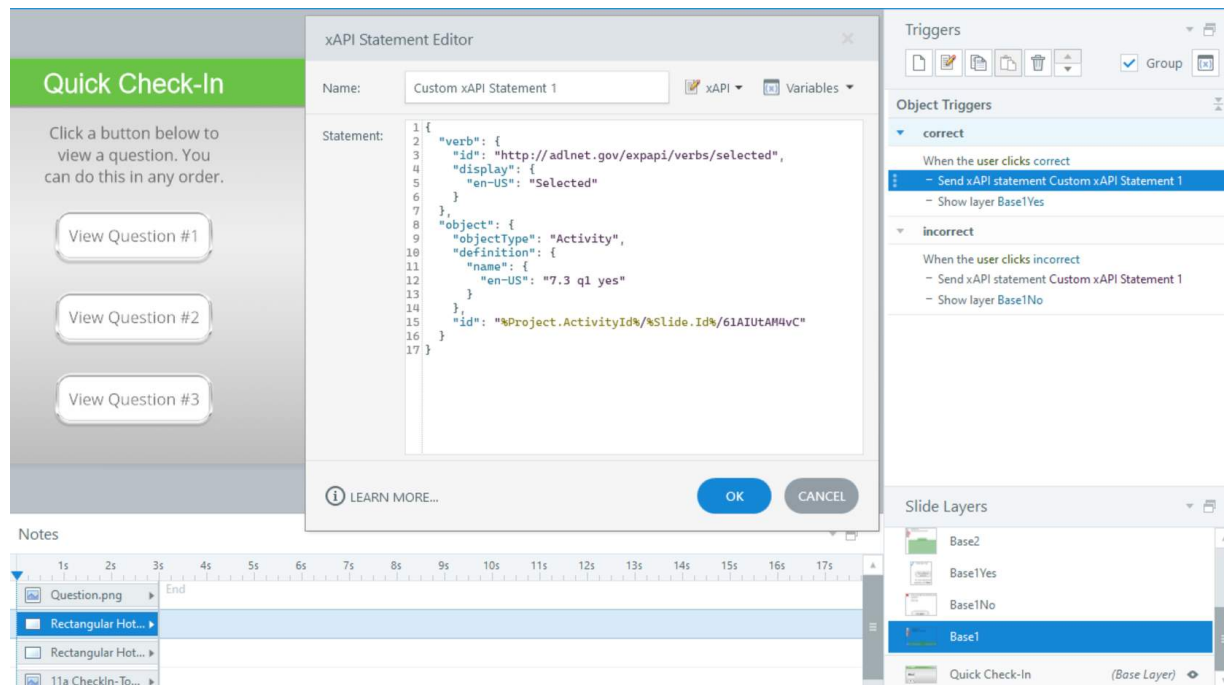
When your button is actually a group

Storyline will send a trigger on a group, and name it group, which is confusing. You can get around that by using the Custom tag (and typing in what you want to name the button) rather than clicking on the object itself.



When your button is named something funny

Storyline will send the statement with the name of the object itself. So, if your object is named "buttonsilver" then you will read it as "learner clicked button silver" --Do be unique and descriptive with your names in your timeline. If you cannot, then you can still change them in the xAPI statement, just change "buttonsilver.png" to something like "Check In Question 1".



When you have hotspots on multiple layers

In this example, we have a few questions that lead to slide layers. Each set of slide layers has a base and a correct and incorrect layer. To create the xapi statment that tells us whether the learner clicked on the correct hotspot, we only need to add 2 triggers onto each of the bottom layer (in this case called base) layer.

1 for when they click the correct hotspot, and one for when they do not. (do this for each 'base' layer)

Create each trigger with a viewed or Selected verb. in the Ojbject's "en-US" language add something like "slide.number q.number yes" or "7.3 q1 no" this should give you enough information to know which selection the learner chose.

Ensure that the xAPI statement is sent before you show any additional layers (such as the correct/incorrect layers)

Triggers to notify of elapsed time (project, scene, slide)

ELAPSED TIME ON A SLIDE

PROJECT ELAPSED TIME

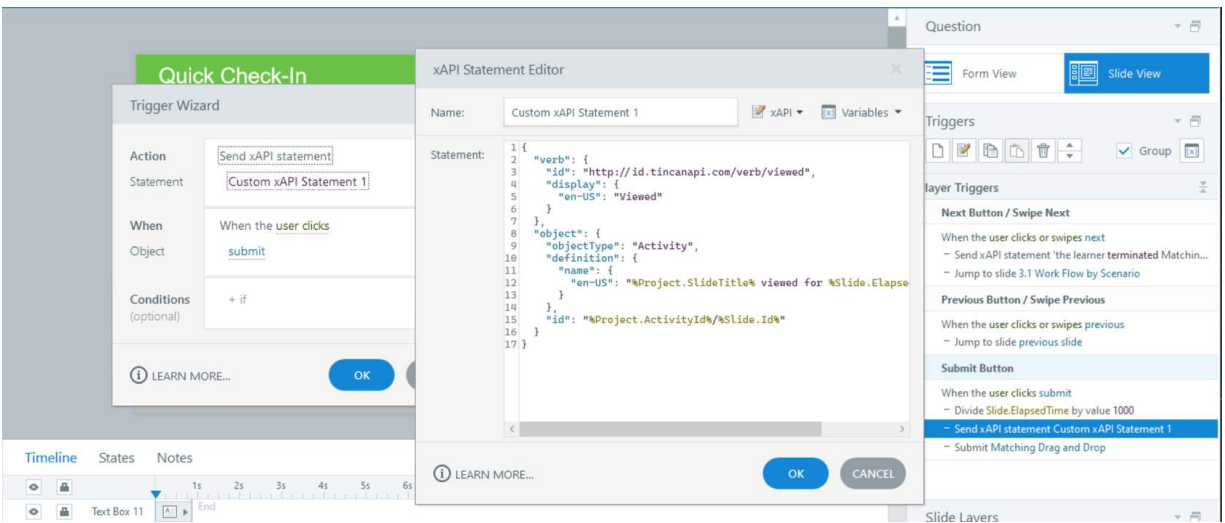
SCENE ELAPSED TIME

The trigger consists of 2 parts - 1 that divides the system variable Slide.elapstedTime by 1000 (to convert milliseconds to seconds) and a second one that sends the xAPI to the LRS.

For the xAPI trigger, start with the viewed verb - object (this slide) then edit the statement and add into the object "en-US" tag viewed for and then the slide.ElapstedTime variable seconds. Keep it all inside " ".

Note: The order of the triggers matters! Divide first, send xAPI second, THEN submit the interaction.

In this example, I chose to add this xAPI call onto the very last slide, so I set the call to run when the slide began, rather than on a button click or slide ending. You can choose what's best for your course.



ELAPSED TIME ON A SLIDE

PROJECT ELAPSED TIME

SCENE ELAPSED TIME

Add the trigger to the last slide of your entire project.

The trigger consists of 2 parts - 1 that divides the system variable Scene.ElapsedTime by 1000 (to convert milliseconds to seconds) and a second one that sends the xAPI to the LRS.

For the xAPI trigger, start with the viewed verb - object (this slide) then edit the statement and add into the object "en-US" tag *Completed something in Scene.ElapstedTime variable seconds*.

Keep it all inside " ".

Change *something* to what the learner completed.

Note: The order of the triggers matters! Divide first, send xAPI second, THEN

The trigger consists of 2 parts - 1 that divides the system variable Project.ElapstedTime by 1000 (to convert milliseconds to seconds) and a second one that sends the xAPI to the LRS.

The screenshot shows the 'xAPI Statement Editor' window with the following JSON statement:

```

1 {
2   "verb": {
3     "id": "http://id.tincanapi.com/verb/viewed",
4     "display": {
5       "en-US": "Viewed"
6     }
7   },
8   "object": {
9     "objectType": "Activity",
10    "definition": {
11      "name": {
12        "en-US": "Completed Cases 1 in %Project.ElapsedTime% se
13      }
14    },
15    "id": "%Project.ActivityId%/assessment"
16  }
17 }

```

The 'Triggers' panel on the right shows 'Slide Triggers' for 'Slide - 3.4 What's Ahead'. The selected trigger is:

- When the timeline starts on this slide
 - Divide `Project.ElapsedTime` by value 1000
 - Send xAPI statement Custom xAPI Statement 1

Other triggers include 'Object Triggers' for 'Left Arrow 1' and 'Picture 1', and 'Player Triggers' for 'Previous Button / Swipe Previous'.

ELAPSED TIME ON A SLIDE

PROJECT ELAPSED TIME

SCENE ELAPSED TIME

Add the trigger to the last slide of a scene that the learner can access.

The trigger consists of 2 parts - 1 that divides the system variable `Scene.ElapsedTime` by 1000 (to convert milliseconds to seconds) and a second one that sends the xAPI to the LRS.

For the xAPI trigger, start with the viewed verb - object (this slide) then edit the statement and add into the object "en-US" tag *Completed something in Scene.ElapsedTime variable seconds*. Keep it all inside " ".

Change *something* to what the learner completed.

Note: The order of the triggers matters! Divide first, send xAPI second, THEN submit the interaction.

xAPI Statement Editor

Name: Custom xAPI Statement 3

Statement:

```
1 {
2   "verb": {
3     "id": "http://id.tincanapi.com/verb/viewed",
4     "display": {
5       "en-US": "Viewed"
6     }
7   },
8   "object": {
9     "objectType": "Activity",
10    "definition": {
11      "name": {
12        "en-US": "Completed assessment in %Scene.ElapsedTime% seconds"
13      }
14    },
15    "id": "%Project.ActivityId%/assessment"
16  }
17 }
```

Triggers

- When the timeline starts on this slide
 - Show layer Success
 - If Results2.ScorePoints ≥ variable Results2.Pass...
 - Show layer Failure
 - If Results2.ScorePoints < variable Results2.Pass...
 - Send xAPI statement Custom xAPI Statement 3
 - Submit results 11.7 ASSESSMENT RESULTS
- When the timeline ends on this slide
 - Divide Scene.ElapsedTime by value 1000
 - Send xAPI statement Custom xAPI Statement 3

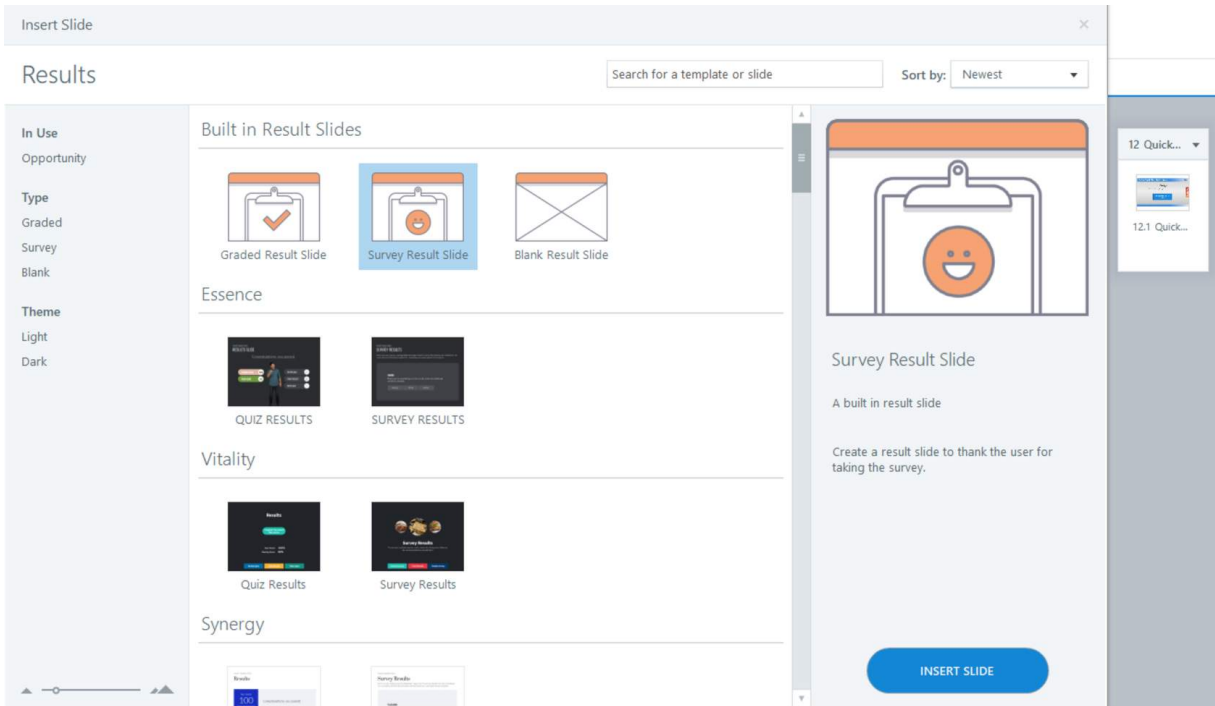
Object Triggers

- Retry Button
 - When the user clicks Retry Button
 - Reset results 11.7 ASSESSMENT RESULTS
 - Jump to slide 11.2 Multiple Choice
 - Send xAPI statement 'the learner interacted with Retry Button'

Slide Layers

LEARN MORE... OK CANCEL

Tracking Quick Check in's (Non-graded assessments)



Storyline will not perform any automatic reporting on any questions slides that are not graded. You will need to have a point system for all of them, and point them to a hidden results slide.

Begin by creating a survey (blank / default) results slide and adding it somewhere where it won't be seen, like a new Scene.

Step 1

Add some points

Enter the question

Quick Check-in: Matching

Enter the choices

	Choice	Match
A	<u>Yyy</u>	<u>Y1</u>
B	<u>Yyy</u>	<u>Y2</u>
C	<u>Yyy</u>	<u>Y3</u>
D	<u>Yyy</u>	<u>Y4</u>
E	Click to enter a choice	
F		

Set feedback and branching

	Feedback	Points
Correct	Good! You matched all three correctly. Please click Continue .	1
Incorrect	Sorry, not all your selections are correct.	0
Try again	That is incorrect. Please try again.	-
Post-quiz review		-

The first thing you need to do is add a grade into the Form View. Even if it's just in 1 point. Don't worry, the learner will not see this, so as long as there is a number in there, it will work. In your design tab, point the results to the new survey results slide you just created.

Step 2

All Done

Storyline will now inform you (automatically) if the learner passes your question or fails it.

Links to test LRS/LMS for you to play with



Veracity LRS

LRS for use with Talent LMS or other hosting site such as web publishing, AWS, etc.

VERACITY LRS

Scorm Cloud

You can upload your file here (LMS)AND record/ read your xAPI data (LRS)

SCORM CLOUD

Talent LMS

LMS (Free for up to 10 courses / 5 users)

TALENT LMS

Watershed LRS

Get your free account and test other features that some other LRS's don't give you.

WATERSHED

Create New Version [Learn more about versioning here](#)

Choose file

Browse

Course Versions

Version 0	Updated on February 3, 2022 at 2:20 PM	Review
Version 1	Updated on February 3, 2022 at 2:46 PM	Review
Version 2	Updated on February 4, 2022 at 8:34 AM	Review

Move Existing Registrations To Newest Version When...

Always

save

One note about Scorm Cloud - when you upload many different versions - do click the Always button in the version history section or when you go to test it, you may not be reviewing the latest version of your course... (Don't be me....)

Lesson 4 of 6

Youtube video tutorials to help you

 Jennifer Gupta

How to Send xAPI Statements from Storyline

Devlin Peck



Watch on

YOUTUBE



Creating xAPI Statements in Articulate Storyline [Part 1]

[VIEW ON YOUTUBE >](#)

 **YOUTUBE**

How to Send xAPI Statements from Articulate Storyline

Learning Dojo



Watch on

Creating xAPI Statements in Articulate Storyline [Part 2]

In my last video, I talked about the basics of Articulate Storyline 360's new xAPI trigger feature. This new update allows you to trigger custom xAPI stateme...

VIEW ON YOUTUBE >

 **YOUTUBE**

Tracking Time Spent on Page in xAPI Statements

Learning Dojo



Watch on

Tracking Time Spent on Page in xAPI Statements [Using Storyline]

Ever wonder how much time people are spending within an eLearning course? The new elapsed time variable allows us to see how much time someone spends on a pa...

VIEW ON YOUTUBE >

Lesson 5 of 6

Publish Settings



Jennifer Gupta



Publish



Review 360

Web

Video

LMS / LRS

CD

Word

Title and Location

Title:

Description:

Folder:

Properties

Player: Classic - Storyline Player
Quality: Optimized for standard delivery
Publish: Entire Project
Tracking: Slides viewed (12 of 12)

Reporting and Tracking

- Report to an LMS
- Report to an external LRS

REPORTING AND TRACKING...

[LEARN MORE ABOUT PUBLISHING](#)

PUBLISH

CANCEL

Reporting and Tracking Options

LMS

LRS

Tracking

Learning Record Store (LRS) Settings

Report to an external LRS

Activity ID:

LRS Configuration (i)

Supplied at launch Manual

LRS Endpoint:

Key:

Secret:

TEST SETTINGS

(i) LEARN MORE... OK CANCEL

Settings

*/*Removed for safety*/*

Troubleshooting issues

JG Jennifer Gupta

The screenshot shows the 'xAPI Statement Editor' window. The title bar reads 'xAPI Statement Editor' with a close button on the right. Below the title bar, there is a 'Name:' field containing 'Custom xAPI Statement 1'. To the right of the name field are two dropdown menus: 'xAPI' and 'Variables'. The main area is labeled 'Statement:' and contains a JSON statement with line numbers 1 through 21. The JSON is as follows:

```
1 {
2   "result": {
3     "duration": "%Slide.ElapsedTime%",
4     "response": "%Assessment_Attempts%"
5   },
6   "verb": {
7     "id": "http://adlnet.gov/expapi/verbs/attempted",
8     "display": {
9       "en-US": "Attempted"
10    }
11  },
12  "object": {
13    "objectType": "Activity",
14    "definition": {
15      "name": {
16        "en-US": "%Project.SlideTitle%"
17      }
18    },
19    "id": "%Project.ActivityId%/6KVa7rfm1YJ"
20  }
21 }
```

At the bottom of the window, there is an information icon followed by the text 'LEARN MORE...', a blue 'OK' button, and a grey 'CANCEL' button.

Things to watch out for in your xAPI statements

(Not all instances of the below are highlighted in the picture)

- 1 Every open curly bracket MUST have a closing curly bracket
- 2 Every line or section that has something after it MUST end in a comma
- 3 Every key word pair must have a : in between them. For example
"response": "%Assessment_Attempts%"
- 4 Every Key and Word must have " around them. "response" :
"answer"