

XAPI Deep Dive Result

So far throughout the Deep Dive series that I've been writing, there has been one thing notably lacking, e-learning. It turns out that the framers of the Experience API specification hit on something big enough that it need not be boxed in by just the e-learning world, and so far we've seen a lot of adopters thinking outside the typical e-learning sphere. But there is a history there, and the work on the Experience API specification was born out of a real desire to advance the specific space around e-learning. The Result object is there to capture some of what is not already catered for and make sure the Experience API can first, and maybe foremost, flourish in the e-learning space, though even it doesn't have to be used only for e-learning.

Statements have an optional 'result' property that can be used to capture a "measurable outcome" from an experience. The value of the 'result' property is specifically a Result object which has a number of properties designed to capture the types of results that have historically been generated from learning activities. The properties of the Result object are 'success', 'completion', 'response', 'duration', 'score' and 'extensions.' For readers used to SCORM, these properties have direct analogs to the data you are already capturing, but in xAPI all properties are optional and the specification doesn't indicate how they must be used together.

success

The 'success' property takes a boolean value (true/false) and provides for a pass/fail categorization of an Activity. It is an important distinction that the specification says Activity in its description of 'success' because it is possible for an experience to be made up of more than one Activity, particularly when the experience is easily sub-divided. It is entirely possible for an overall experience to be a success or failure with individual sub-activities having the opposite value. For example, the 'success' property of a statement capturing the answering of a question might be 'false' (or failure), but within a statement capturing a summary of the overall quiz the Result object's 'success' property might be 'true' (or pass).

completion

The 'completion' property is a boolean value (true/false) as well and captures whether the Activity was completed. This is somewhat confusing in light of the very commonly used "completed" (<http://adlnet.gov/expapi/verbs/completed>) verb. The key here is that the 'completion' property is a part of the specification and indicates that based solely on this statement it can be determined that the actor has done what is required to consider the Activity completed. The commonly used "completed" verb may indicate the same thing for a particular profile, but that is not captured by the specification itself, it is profile specific. The xAPI specification provides for flexibility in how systems will react to the data stored, the 'completion' property is very much intended to capture the concept of "completion" as it is in the LMS AICC/SCORM world.

duration

Continuing the tradition of well named properties, the 'duration' property takes a value indicating the length of time taken for the experience captured by the statement. The value is a string but must be formatted using the [ISO 8601](#) standard and is specified to have a maximum precision of 0.01 seconds. Activity Providers need to pay particular attention to the notion of precision with the 'duration' property. For example, two values that are both acceptable based on the formatting requirements are "P1M" and "P30D", but in the first case we can only read this as one month which may be any of 28, 29, 30, or 31 days whereas in the latter we know this to be explicitly 30 days. Because of the requirement to use an ISO 8601 formatted value, arbitrary units can't be included as a duration easily, that is, the duration can't be something like "12 slides" or "3 courses" in that case separate statements or an extension is probably the best practice.

As I mentioned in "[Deep Dive: Extras](#)", the 'duration' property can work together with the 'timestamp' property to give an indication of precisely when and for how long an experience took place allowing reporting systems to do interesting things with time overlaps. One interesting point brought up by [Andrew Downes via GitHub](#) is that the specification leaves open when in relation to the life of the experience the 'timestamp'

occurs. The bottom line seems to be that if the duration is going to be used in relation to the timestamp that there is an agreement between the two and reporting systems need to take this into account. Consensus seems to be favoring that the 'timestamp' represents the ending point of the duration when it is included.

response

The 'response' property takes a string as its value which has a format that is Activity specific which is to say it can store virtually anything. This value could correspond to text entered by a user to answer a question, or a serialized true/false answer, or the identifier for an answer in a multiple choice question, etc. When using the predefined "cmi.interaction" (<http://adlnet.gov/expapi/activities/cmi.interaction>) Activity Type the value should correspond to one of the entries in the 'correctResponsesPattern' property.

score

Unlike the others in the Result object, the 'score' property takes an object itself, the Score object which has its own set of properties. The Score object's properties are all optional and each is a decimal number. These properties, 'scaled', 'raw', 'min', and 'max,' correspond directly to the CMI properties from the SCORM 2004 specification. The 'raw' value corresponds to a nominal value and when they are provided it must be between the values of 'min' and 'max.' Those two properties, 'min' and 'max,' correspond to nominal values marking the starting and ending point of a range of values. If the 'raw' value can be calculated as a percentage, then it is expected that the 'scaled' value is populated with a value between -1 and 1. For example, on a 25 question quiz a user may get 20 questions correct, corresponding to a 'raw' value of '20' and a scaled value of '0.8' (or 80%), 'min' could be included as '0' and 'max' as '25.'

One important note, the specification indicates that the Score object and associated properties should not be included for determining progress or completion. Presumably

the 'completion' boolean property in the outer Result object should be used for the latter, and an extension is recommended for the former.

extensions

As we saw in “[Deep Dive: Context](#)” and “[Deep Dive: Activity](#)”, the 'extensions' property holds an object of catch all data that is necessary to capture the meaning of the result. An example of an extension that might be useful in a Result object that we've already added to [The Registry](#) is “Ending Position,” <http://id.tincanapi.com/extension/ending-position>. For example, it can be used to indicate the final place of a runner in a race. (More about extensions is coming in a future post.)

Leveraging the elements of a Result object in a xAPI statement facilitates conveying information about what happened during an experience and directly lends itself to tracking objective outcomes. It gives us a way to measure the performance of an Agent when interacting with an Activity, particularly those that are prescribed to the task of assessing capability associated with a learning process. The Result object captures the essence of the Experience API being forged from past experience in the learning standards world and provides an easier path forward for existing content created for the formal training model.

Go now, make statements!