0.1 At JSONPath

Performs a lookup at *path* within *source* similar to atKey

$$atJsonPath(source, path)$$

such that the fundamental functionality of JSONPath is covered in this definition.

- A more complete definition will come at a future date if/as necessary

0.1.1 Arguments

- *source* is an object Scalar, KV, Statement or an Algorithm State
- *path* is a JSONPath string which adheres to the additional requirements, clarifications, and additions placed on JSONPath by the xAPI Profile Specification

0.1.2 Relevant Operations

The primitive *atJsonPath* uses the operations

- atKey
- atIndex
- append
- count

0.1.3 Summary

atJsonPath will return a v found within *source* after converting

$$path \rightarrow < path_{i+1}..path_j >$$

such that if

$$path = \$.a.b$$

then

$$path \rightarrow < a, b >$$

so that

$$atJsonPath(< a \mapsto b \mapsto 123 >, \$.a.b) = 123$$

0.1.4 Usage of Operations

In order to convert

$$path \rightarrow < path_{i+2}..path_j >$$

an empty Collection $keyState$ is introduced

$$keyState = <>$$

so that the relevant k'(s) can be stored in $keyState$ during iteration over $path$

$$\forall n : i..j \ \bullet i = 0 \ \wedge j = count(path) - 1$$

and the number of stored keys can be tracked using $curKeyStateIndex$

$$curKeyStateIndex = count(keyState) - 1$$

such that the current $path_n$ can be retrieved

$$curKey = atIndex(path, n)$$

and $keepKey?$ can indicate the relevance of $path_n$

$$keepKey? = true \iff curKey \neq \$ \ \wedge curKey \neq .$$

such that during each iteration n, $keyState$ will be updated if necessary

$$keyState = append(keyState,\ curKey,\ curKeyStateIndex) \iff keepKey? = true$$

so at the end of the loop

$$keyState = < path_{i+2}..path_n..path_j >$$

which provides the Collection of Key(s) necessary for calling $atKey$

$$valueInSource = atKey(source, keyState)$$

such that

$$atJsonPath(source, path) \equiv atKey(source, keyState)$$

0.1.5 Example output

Given an example $source$

$$source = < a \mapsto < b \mapsto 123, c \mapsto 456 >, d \mapsto foo >$$

then

$$atJsonPath(source, \$.a) = < b \mapsto 123, c \mapsto 456 >$$

and

$$atJsonPath(source, \$.a.b) = 123$$

and

$$atJsonPath(source, \$.a.c) = 456$$

and

$$atJsonPath(source, \$.d) = foo$$