Programme Workflow

Program Entry Point:
Consumer.py - argument handling and processing, sets up logger. Calls process in processor.py

Processor.py:
programme loop - loops until it finds no more objects in the s3 bucket. getWidget is called in the programme loop uses the WidgetList object to query the read s3 bucket for widgets and stores them in a sorted list, the top key is returned and handed to getWidgetJson which pulls the data out of the s3 bucket and stores it as a dictionary object. This object is validated and then passed back to the loop where it is handed to the create/update/delete methods in Action.py

Action.py:
Action.py contains the endpoints for the workflow based on the type specified in the validated json. createWidget stores the widget in the write s3 bucket, then uses the formatWidget to prepare the widget to be stored in the DynamoDB. The formatted json is then put in the DynamoDB

Consumer.py

Main():Null

Processor.py

+processor(str: read, str: write): Null
-getBucket(str:name): s3 Bucket Obj
-getDB(): Table object
-getWidget(s3Obj: readBucket, widgetList):dict
-getWidgetJson(s3Obj: readBucket, str:fileName):dict
-validateJson(dict: input): boolean

WidgetList

+keys[]

+init(readBucket): self
-getKeys(readBucket):Null
+getNextKey():String
+checkKeys():Boolean

Actions.py

+createWidget(dict: widget, s3Obj: writeBucket, Table: db)
+updateWidget(dict: widget, s3Obj: writeBucket, Table: db)
+deleteWidget(dict:widget, s3Obj: writeBucket, Table: db)
-formatWidget(dict:widget):dict