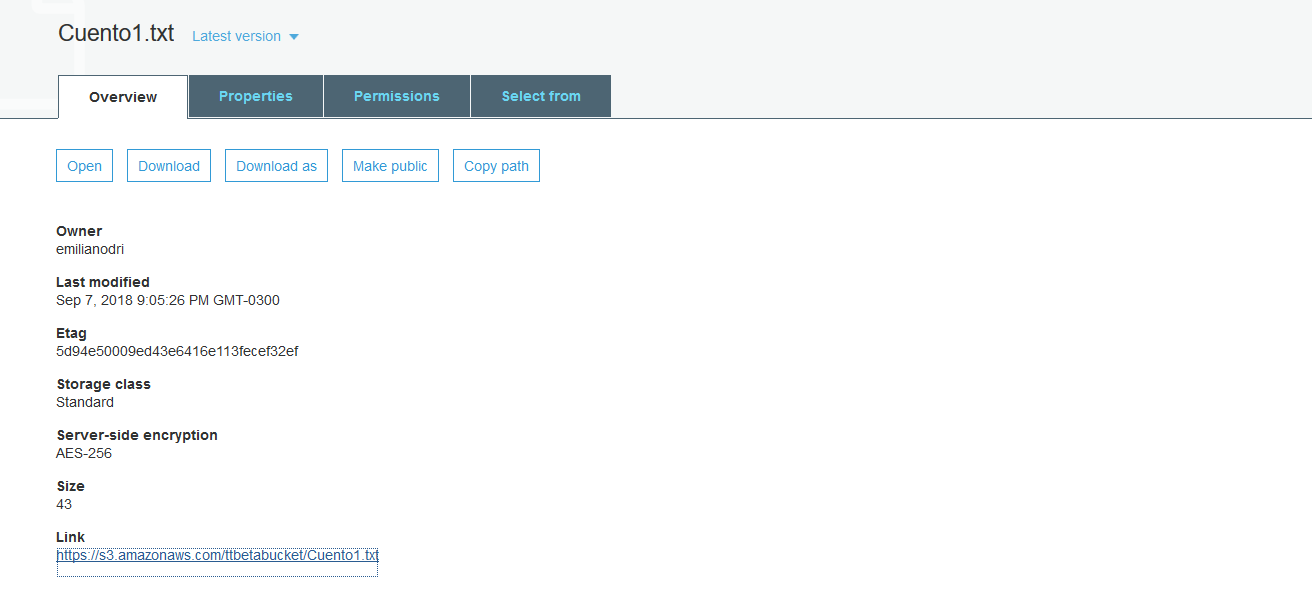


Skill Alexa

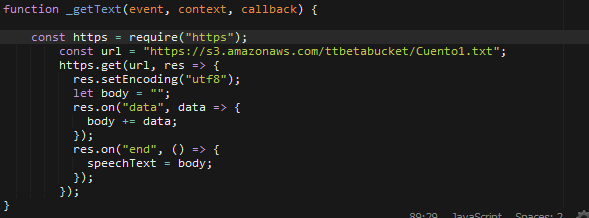
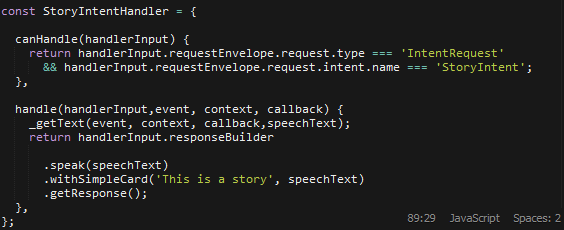
Contador de cuentos/Taleteller

Prueba 1

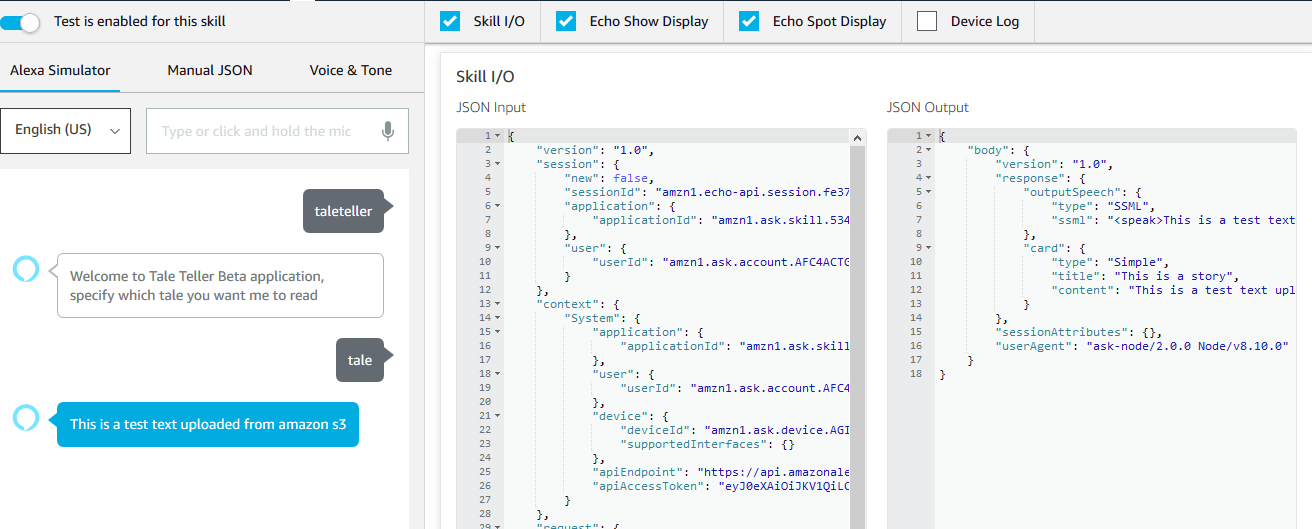
En base al repositorio Alexa Skill Facts (https://github.com/alexa/skill-sample-nodejs-fact) se creó una función lambda que solicita información a un servidor AwS S3 por medio de un GET Https, el archivo es un txt en texto plano accesible de forma pública, se encuentra alojado en <https://s3.amazonaws.com/ttbetabucket/Cuento1.txt> siendo ttbetabucket el nombre del bucket S3 y Cuento1.txt el archivo de texto que contiene la información



El Intent StoryIntentHandler que es llamado por el JSON del Alexa con el término “tale” llamara a la función \_getText que recibe el texto y lo almacena en speechText para usarlo

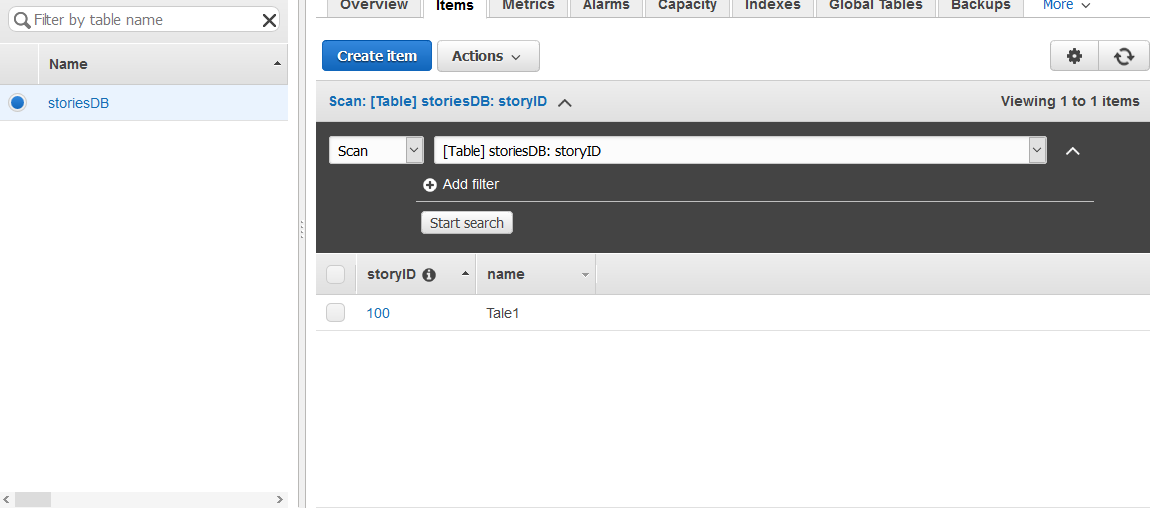


El contenido del archivo de texto es “This is a test text uploaded from amazon s3” y es mostrado por Alexa una vez que se llama el intent correspondiente.



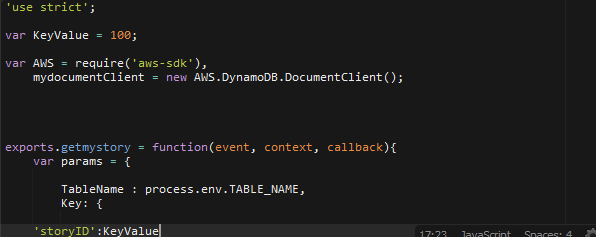
DynamoDB + Lambda NODE.JS

En la función lambda se incluye un .js llamado GetDynamo que ejecuta una simple función GetItem utilizando el AwsSDK+DynamoDB, la estructura de la base de datos es la siguiente:



Siendo storiesDB el nombre de la tabla y storyID La Primary Key

Al hacer el get se obtiene la confirmación de que existe un storyID (Que es una clave única) con el valor 100, además nos proporciona el valor del string name



El nombre de la tabla que se va a llamar tiene que ser editado en las variables de entorno globales, además se debe proporcionar acceso al Lambda para que acceda a la base de datos DynamoDB, para esto se le asigna un rol IAM que de Full Access al DynamoDB (Tanto para escritura como para lectura). Este script no se ejecuta de ninguna forma si Alexa llama al index.handler, se tiene que ejecutar manualmente desde la consola lambda con el controlador GetDynamo.getmystory, o invocar el script desde el index.js, los valores devueltos por el get se muestran en la consola por medio de un Callback.