Enhancing the AWS Lambda Function

Richard Monson-Haefel
SR. SOFTWARE ENGINEER

@rmonson www.monsonhaefel.com

Overview

Leverage Java inheritance in AWS Lambda

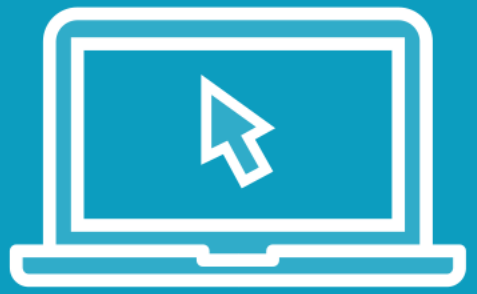
Create two new Lambda functions

- InventoryInsertFunction
- InventoryDeleteFunction

Web APIs for POST and DELETE

An exercise deploying a new function on your own

Demo



Import two new classes

- InventoryS3Client
- HttpRequest

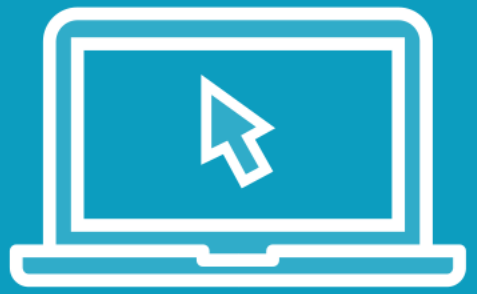
Abstract S3 interactions into a reusable supertype for functions

Demo

Refactor InventoryFindFunction to extend InventoryS3Client

Leverage the InventoryS3Client super-class to return all of the products in inventory

Demo



Create a new function

- InventoryInsertFunction
- Extend InventoryS3 Client
- Read and *write* data to S3

Demo

Create a new Web API HTTP POST

- Send JSON from client to function
- Marshal JSON into a Java bean

Demo

Create a new function

- InventoryDeleteFunction
- Extend InventoryS3 Client
- Read and *write* data to S3

Create a new Web API HTTP DELETE

- Use HTTP path parameter

Demo

Independent exercise

- Import InventoryUpdateFunction to project
- Upload function to AWS Lambda
- Create a new HTTP PUT Web API
- Send an update for a Product using the new Web API and JSON

Summary

Learned how powerful Java inheritance can be in AWS Lambda functions

Learned how to create two new functions

- InventoryInsertFunction
- InventoryDeleteFunction

Learned to create and deploy Web APIs for

- HTTP POST with JSON data
- HTTP DELETE with path parameters

Independent exercise

- Upload InventoryUpdatefunction
- Create an HTTP PUT Web API
- Tested using Postman