

## ACamoPRJCT.com

### Project Overview

ACamoPRJCT, LLC is an up and coming game and entertainment business. The purpose of [acamoprjct.com](http://acamoprjct.com) is to represent The Company's culture online, promote its Brand, market and sell The Company's products, and connect with its customers.

The website is composed of these primary features to accomplish the project's stated goal:

- Static web pages for:
  - Introduction / Home Page
  - PawnWars Mobile Game and Chess Set (Flagship Product)
  - ACamoPRJCT Team Members
  - About Page
- E-Commerce Store
  - Independent e-commerce store platform containing The Company's products
  - Stripe payments, future integrations with Apple, Google Pay
  - Shippo service for shipping rates and labels
- Newswire
  - News feed to push news, updates, promotions, etc... to web page
- User Accounts
  - Site member option
  - Members-only promotions
  - Advanced user metrics
  - Connect to mobile-game accounts
- Admin Portal
  - Update store items
  - Process open orders, purchase shipping labels
  - Publish newswire updates
  - View site analytics data
  - View internal business data
    - Transaction log
    - Shipping log

These core features are designed to scale and be expanded upon as the business grows and website traffic increases.

### Technical Overview

- Front end web pages are built with standard HTML, CSS, and JavaScript, and also include Bootstrap components.
- Backend is built in Go implementing the AWS Serverless Application Model architecture (AWS CloudFormation, Lambda, API Gateway, SNS, DynamoDB, etc...).

- Hosting and routing is accomplished with S3, CloudFront, and Route53.
- Data storage is comprised of S3 for static files and logs, and DynamoDB for data processed by the application (customers, orders, products, etc...).

## Architecture Overview

Note: service still in development; final architecture and function names are subject to change

### Service

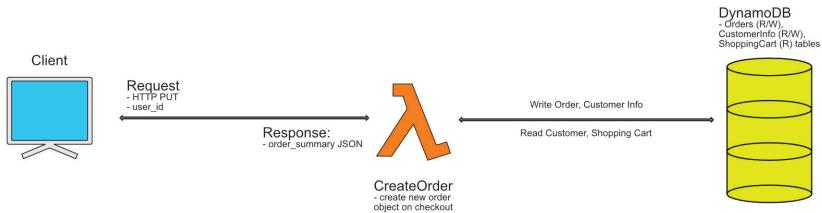
- store
  - viewItems
  - addToCart
  - checkout
    - createOrder
    - getShippingMethods
    - payment
  - fulfillment
    - closeOpenOrder
    - getOpenOrder
    - purchaseLabel
    - queueOrder
    - scanOpenOrders
    - sendEmail
    - sendShippingNotification
    - updateOrder
    - updateShipment
    - viewOpenOrders
  - orders
    - stageOrder
    - processOrder
  - store (contains models/base data types)
- newswire
- user
- admin

## Architecture Examples

### Checkout - createOrder

Action: User clicks “Checkout” from Cart.

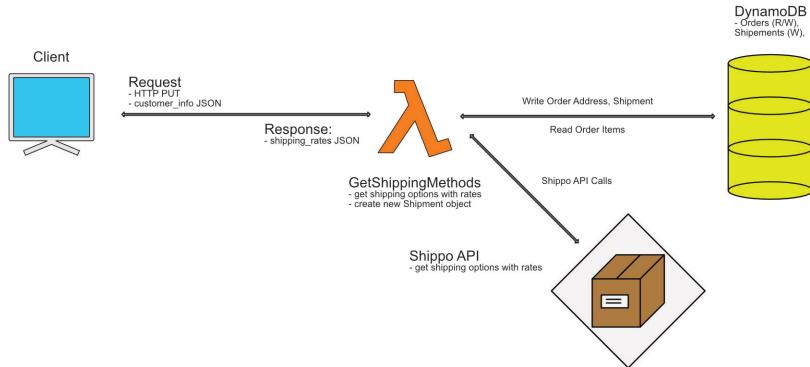
## Checkout Service CreateOrder



### Checkout - getShippingMethods

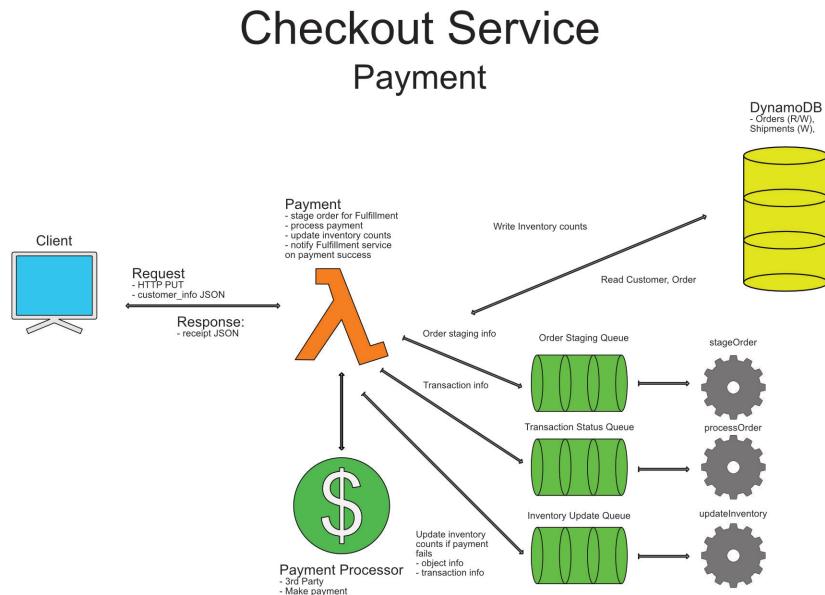
Action: Shipping options and rates are returned after user inputs shipping address.

## Checkout Service GetShippingMethods



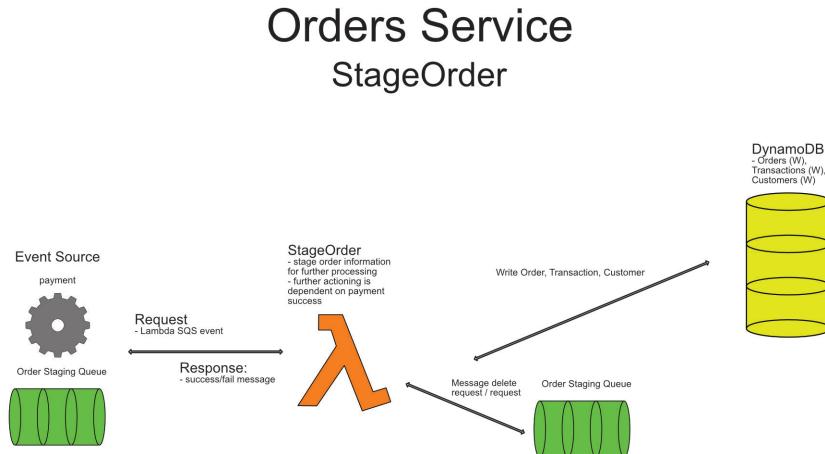
## Checkout - payment

Action: User submits payment info to be processed by 3rd party service.



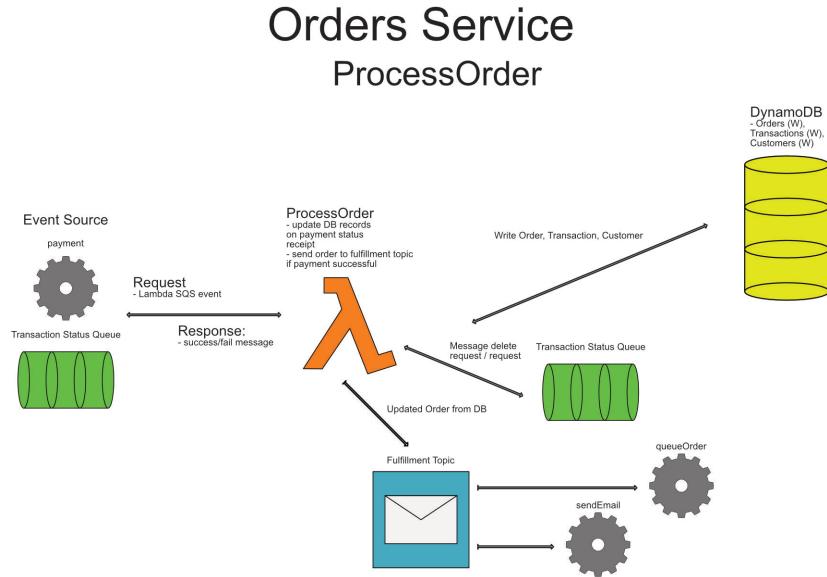
## Orders - stageOrder

Action: In-progress order data is persisted for further actioning upon payment receipt.



## Orders - processOrder

Action: Order is actioned further depending on payment success.



## Current and Future Development

This project is currently under development and will be updated as new features are developed, tested, and implemented. This document and the code files will be updated accordingly.

## Private Repository

Technical recruiters and others in Talent Acquisition can request access to the private repository for this project by contacting Gilberto Garcia.

## Licensing

Unless noted otherwise in the file's licensing, all files contained herein are the sole property of Gilberto Garcia and are not to be copied, shared, or distributed without the written consent of Gilberto Garcia.