Lemonade Partner API

Implementation Guide for Power Mortgage

Offload Insurance Process

Enhance your mortgage offer without building any infrastructure. Developers make simple requests to our service, and we handle the rest. No complicated setup required.

1. Features

One stop shop. Full support for everything related to Home Insurance.

- ➔ **Quotes**
  Create and store quotes for clients
- ➔ **Payment**
  Process payment immediately or let us handle it.
- ➔ **Policies**
  Create and store policies based on quotes

Why work with Lemonade instead our competitors?

USAA®

Allstate®
You're in good hands.

Lemonade Ranks #1

US News ranks Lemonade #1 in their list of "Best Homeowners Insurance Rating"

Ease of use and Documentation.

Technical

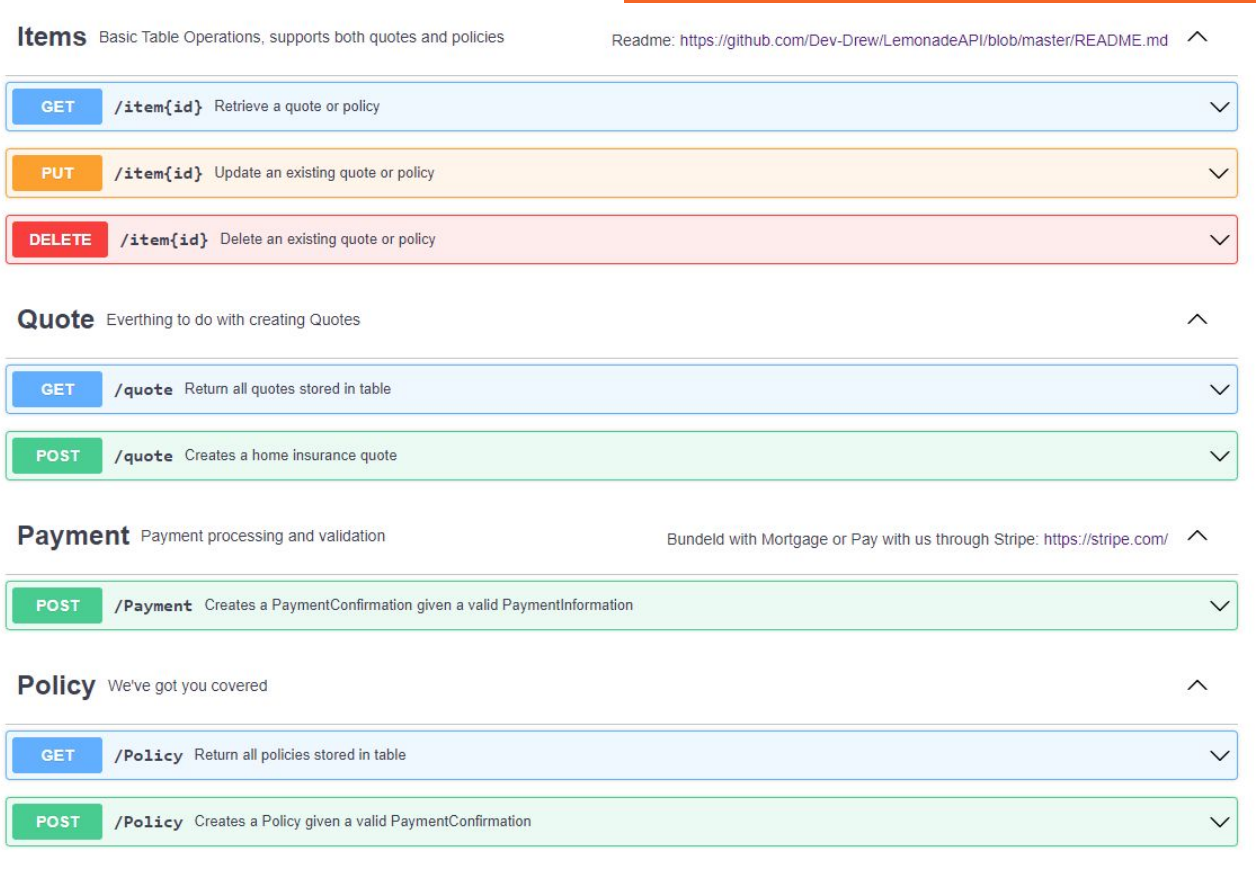
Code is on Github

Postman Collection

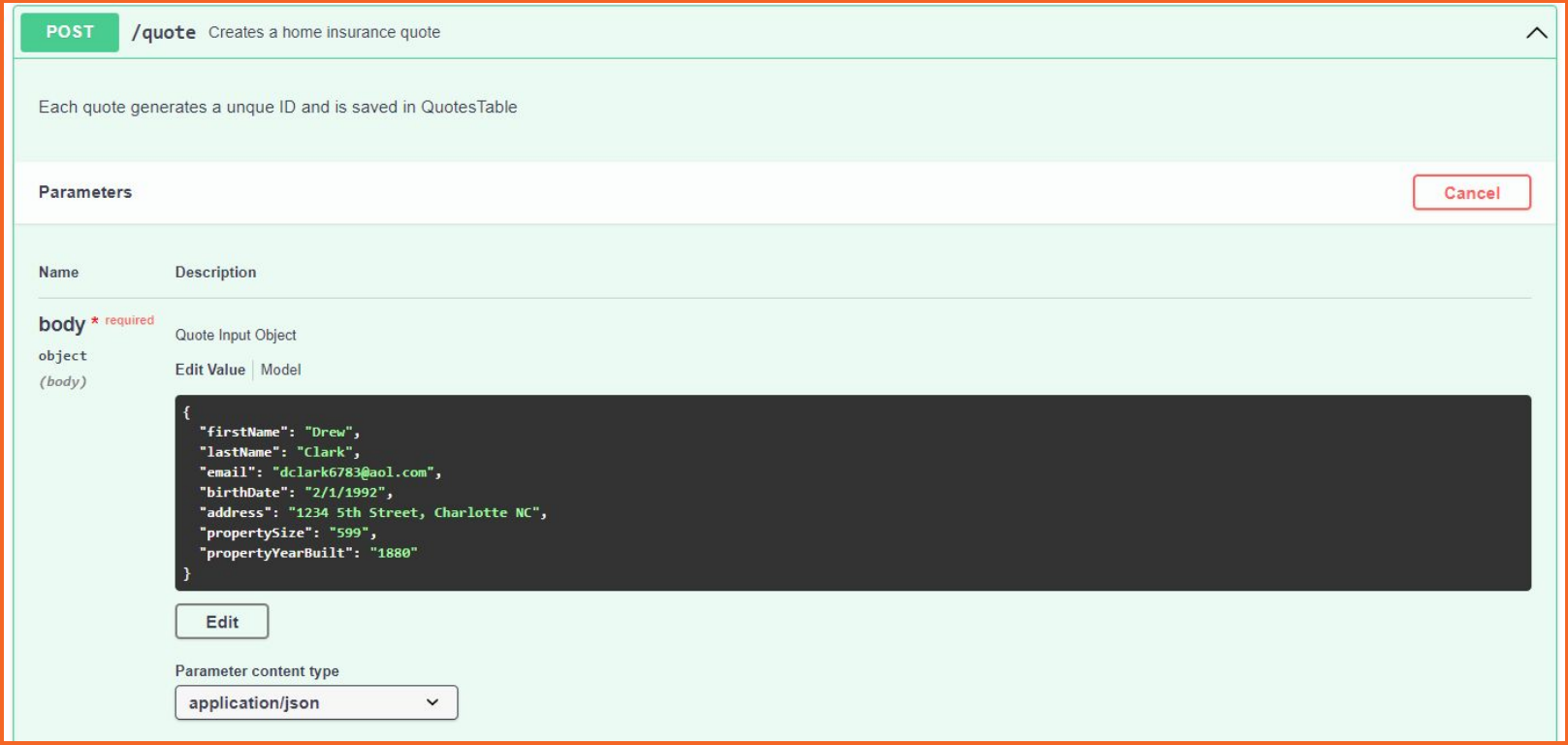
API Spec

Detailed Setup Guide

Endpoints available right now

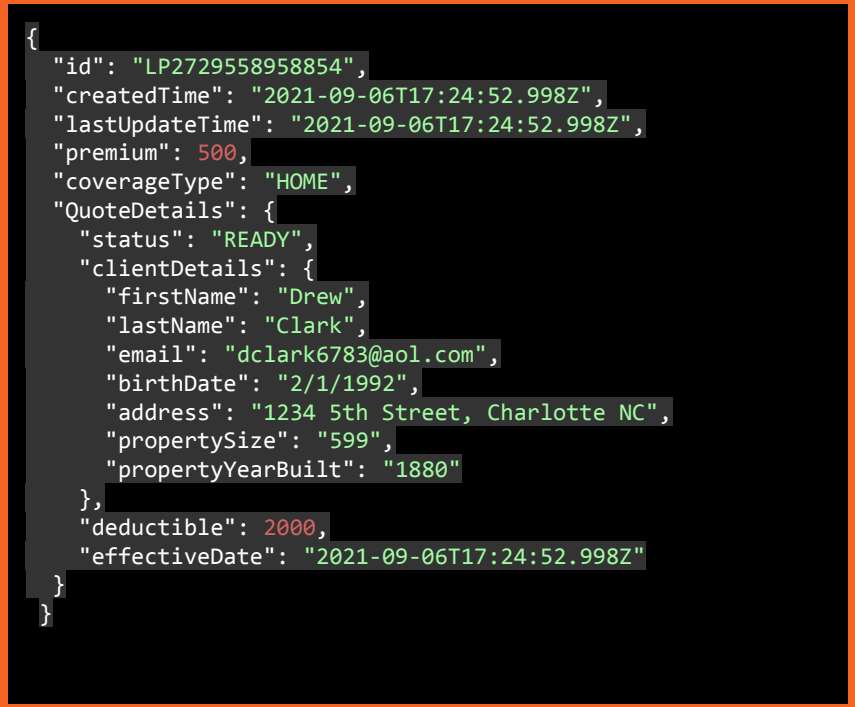


Create a Quote /quote



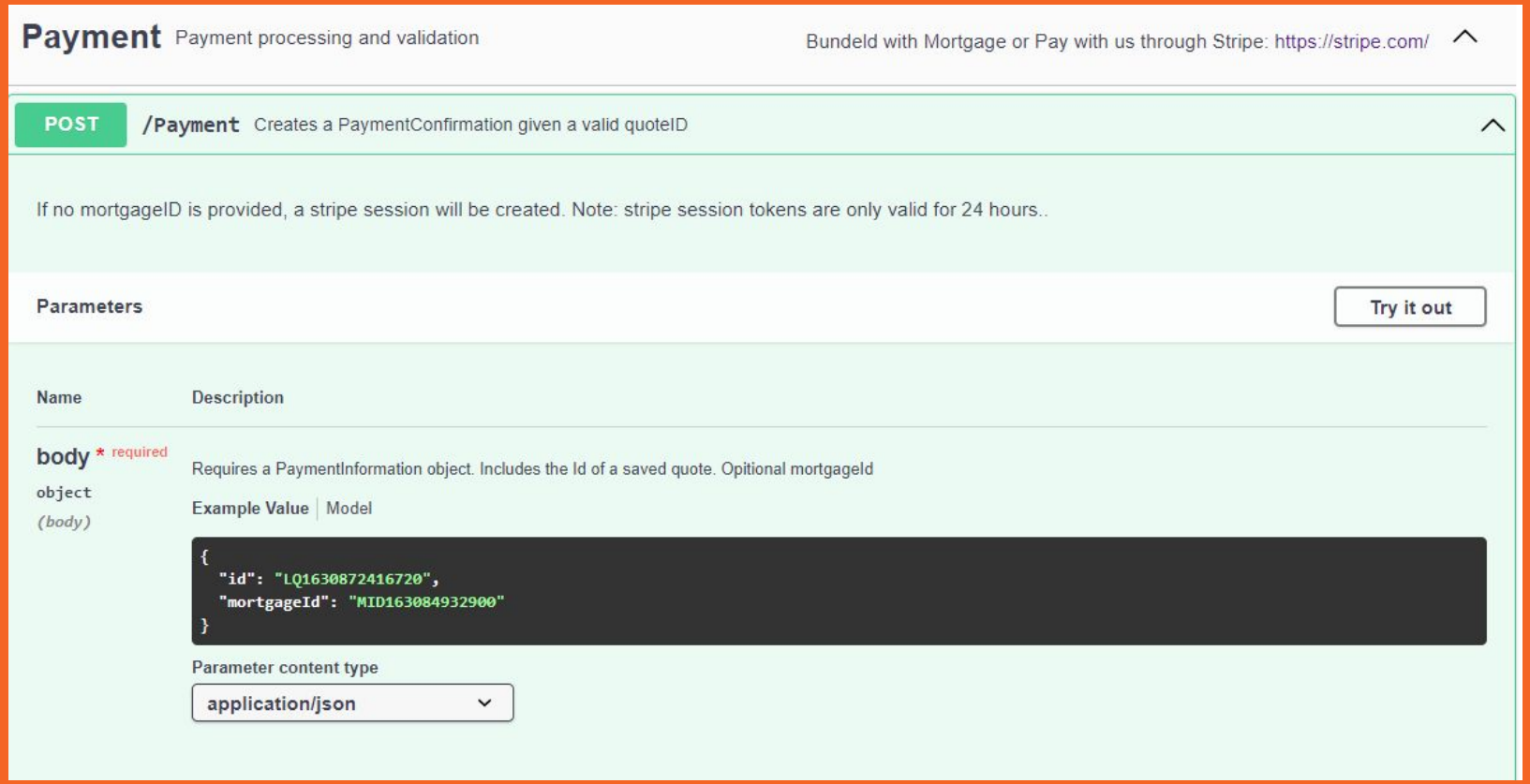
So, what's a quote?

Object that contains a unique id, premium / deductible, and client information.



Request sample in poststman. Checkout documents folder on the repo

Process Payment /payment



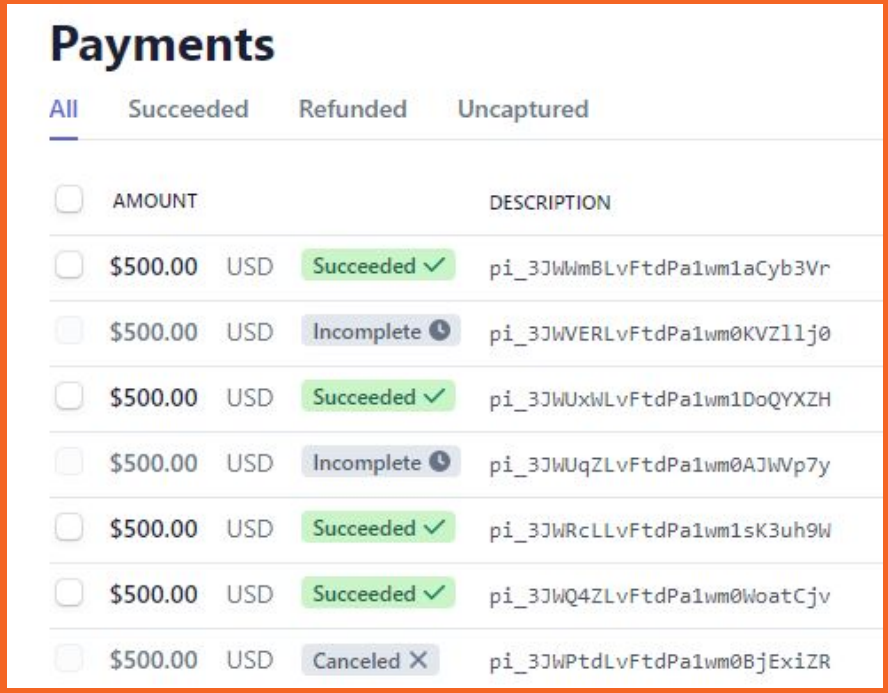
How do you handle payments?

Stripe handles payment through credit card or Google Pay

Request sample in Postman. Checkout documents folder on the repo

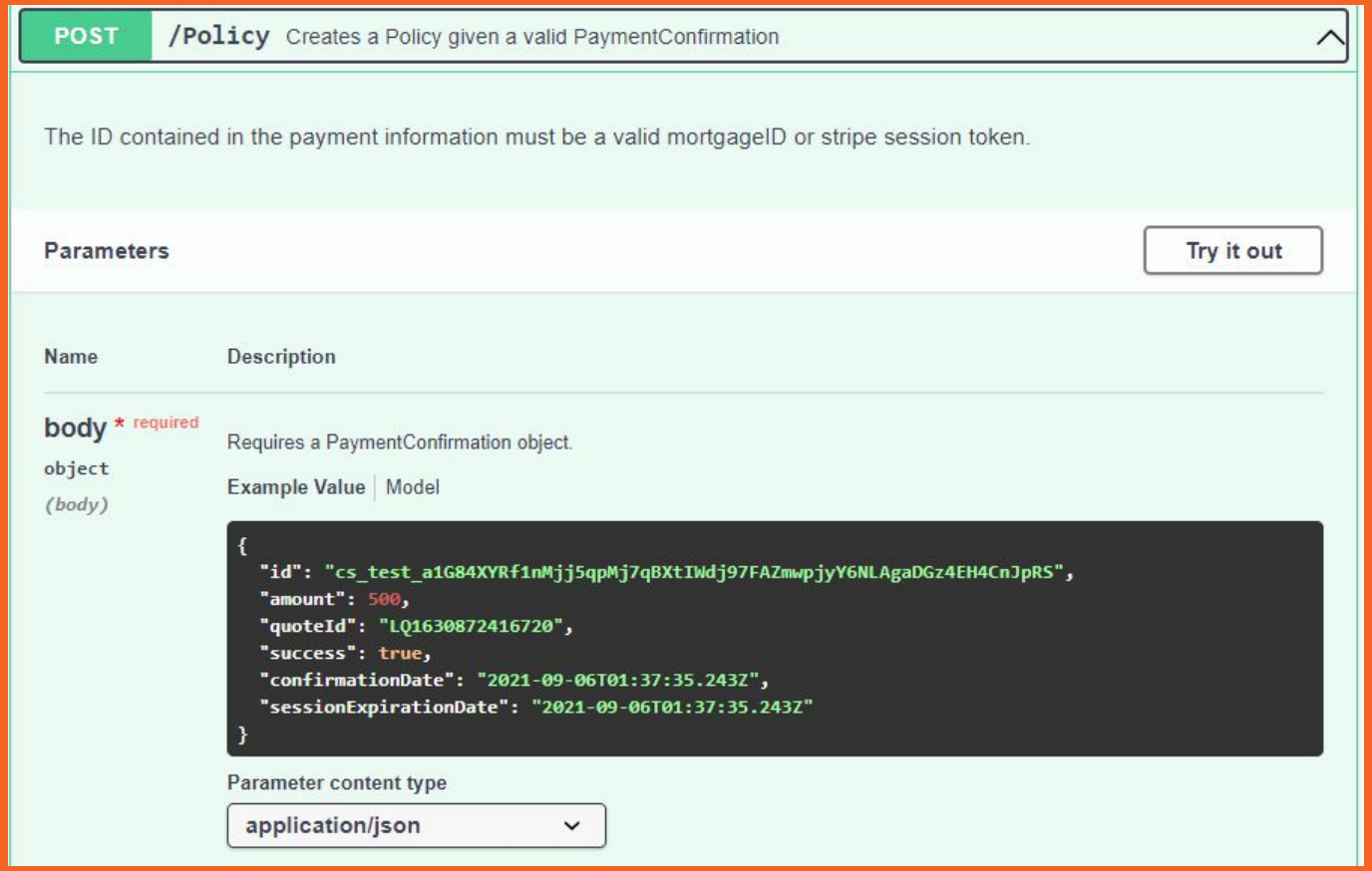
How to validate if payment was successful?

Stripe UI gives detailed information



Note to run on your local, you'll need a stripe token.

Create a Policy



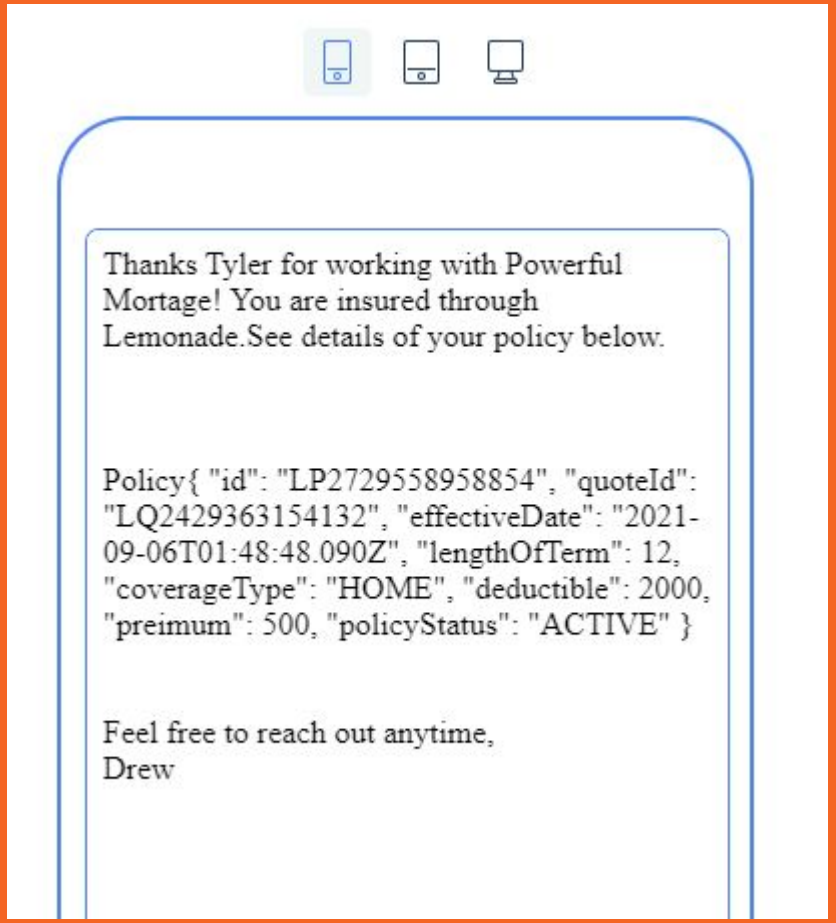
What's a policy look like?

Stripe UI gives detailed information

Note: You may only create a Policy for a Quote that is "READY"

How does a client know their policy was created?

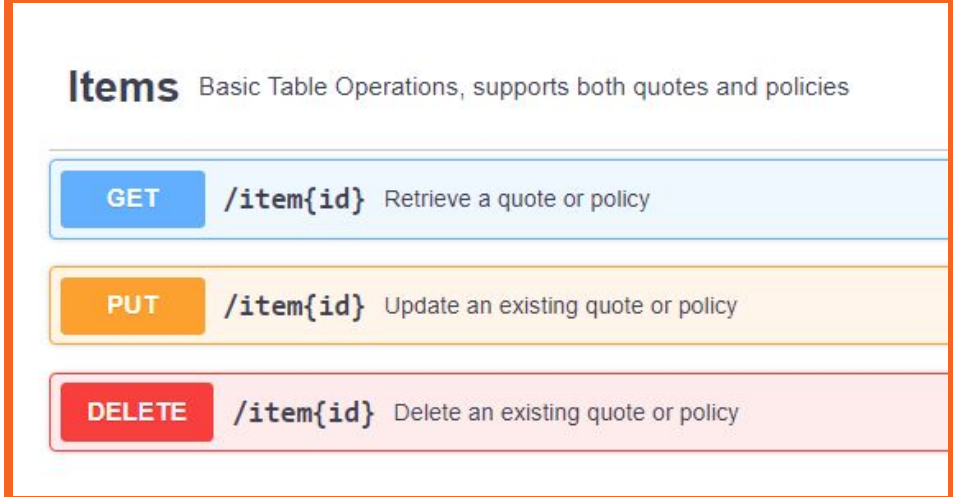
Email automatically sent.



Partner API Flow

What if I just want to do basic operations?

We’ve got you covered



Note: All QuoteIDs start with “LQ”, PolicyIDs start with “LP

Need all the data?

- **GET /policy**
  Returns all policy data stored in the Policy Dynamo Table.
- **GET /quote**
  Returns all quote data stored in the Quotes Dynamo Table.

Tech Stack

- **TypeScript**: Primary Language
- **NodeJS**: Runtime
- **NestJS**: NodeJS Framework
- **NestJS-stripe**: Stripe Wrapper
- **DyanmoDB**: NoSQL DB
- **AWS SDK**: Access Tables
- **Nodemailer**: Send Emails
- **Open**: Open Links in Browser
- **Postman**: Send HTTP
- **DB Browser**: GUI for DB

Note: See full list in the README

Demo