

Trabalho Final AWS

Objetivos

- ✓ Praticar os conceitos vistos em sala de aula;
- ✓ Entender os conceitos principais sobre a criação de APIs;
- ✓ Entrar em contato com os ambientes em nuvem de infraestrutura como serviço;
- ✓ Criar e Implantar APIs;
- ✓ Garantir a robustez de cada API;

Enunciado

Uma empresa possui um **sistema de drop shipping**¹, consistindo em um **catálogo de produtos** (dos mais variados segmentos) disponíveis em plataforma web e também em aplicativo mobile. Neste tipo de negócio, **o revendedor não mantém os produtos em estoque**, mas comercializa produtos que estão no estoque do fornecedor.

O sistema apresenta os 5 principais contextos de negócio:

- **Montagem e exibição do catálogo**: robô (crawling)² automático que visita vários fornecedores para **construir** e **atualizar** o **catálogo de produtos** (as suas propriedades, como as **imagens, descrição, preço e o estoque**) disponíveis nas lojas;
- **Compra de um produto**: consulta a montagem e exibição do catálogo para verificar a disponibilidade do produto. Em caso afirmativo, inicia o processo de compra do produto junto ao fornecedor.
- **Pagamento**: realiza o pagamento do produto através das operações de aprovação de crédito e eventuais análises de fraude;
- **Acompanhamento**: uma vez comprado o produto e dado baixa no estoque do fornecedor, este contexto é responsável por acompanhar o estado de envio do produto (despachado, enviado para a distribuidora, etc.) e reportar o estado para a plataforma web e para o aplicativo mobile.

¹ https://pt.wikipedia.org/wiki/Drop_shipping

² https://pt.wikipedia.org/wiki/Rastreador_web

- CRM: acompanha o processo de informações de cada cliente: as compras realizadas até o momento, os emails enviados, potenciais leads, etc.

Você foi contratado para coordenar um dos 5 grupos (montagem do catálogo, compra de um produto, pagamento, acompanhamento e CRM) responsáveis pelo modelo de negócio da empresa. Sua missão é desenhar as APIs envolvidas em seu grupo, desenvolvendo uma **camada de mock para cada uma para a consequente implantação do sistema**.

Cada conjunto de API pode estar implantado em uma cloud qualquer, como (por exemplo):

- AWS
- Microsoft Azure
- Google Cloud
- IBM Cloud
- Alibaba Cloud
- ...
- Postman

Para a construção da API, pode-se utilizar qualquer uma das seguintes estratégias:

- Execução direta na máquina alvo
- Utilização de uma arquitetura FAAS
- Utilização de virtualização, como o Docker/Kubernetes

Critérios de avaliação

- Arquitetura do conjunto de APIs escolhido (RestFul, SOA, ...)
- **Documentação**
- Implantação
- Interação com as demais APIs e contextos de negócio
- Cloud escolhida

- Comunicação com as demais APIs
- Cache
- Monitoramento
- Atributos de resiliência, como o throttling, circuit breaker, etc.

Exemplo de solução

Caso um grupo seja do CRM, espera-se inicialmente uma documentação semelhante à seguir:

Especificação

openapi: "3.0.0"

info:

version: 1.0.0

title: CRM API

license:

name: MIT

paths:

/leads:

get:

summary: List all leads

operationId: listLeads

tags:

- leads

parameters:

- name: limit

in: query

description: How many leads to return at one time (max 100)

required: false

schema:

type: integer

format: int32

responses:

'200':

description: A paged array of leads

headers:

x-next:

description: A link to the next page of responses

schema:

type: string

content:

application/json:

schema:

\$ref: "#/components/schemas/Leads"

default:

description: unexpected error

content:

application/json:

schema:

\$ref: "#/components/schemas/Error"

post:

summary: Create a lead

operationId: createLeads

tags:

- leads

responses:

'201':

description: Null response

default:

description: unexpected error

content:

application/json:

schema:

\$ref: "#/components/schemas/Error"

/leads/{leadId}:

get:

summary: Info for a specific lead

operationId: showLeadById

tags:

- leads

parameters:

- name: leadId

in: path

required: true

description: The id of the lead to retrieve

schema:

type: string

responses:

'200':

description: Expected response to a valid request

content:

application/json:

schema:

\$ref: "#/components/schemas/Lead"

default:

description: unexpected error

content:

application/json:

schema:

\$ref: "#/components/schemas/Error"

components:

schemas:

Lead:

type: object

required:

- id
- name
- email

properties:

id:

type: integer

format: int64

name:

type: string

email:

type: string

Leads:

type: array

items:

\$ref: "#/components/schemas/Lead"

Error:

type: object

required:

- code
- message

properties:

code:

type: integer

format: int32

message:

type: string

Implantação

