Jornada AppSec

Esboçando o desenho da API

Autor: Felipe Pinheiro

GitHub: https://github.com/pinheiro-felipe/jornada-appsec-criando-api-de-postagem-de-status-do-pedido-para-delivery

5º Dia: Quinta, 11 de julho de 2024_____
Dia de sol e bom para caminhar na orla

Bom dia pessoal, após as explicações anteriores e modelagem do nosso mapa de contexto, vamos começar a esboçar o desenho da nossa API. Gostaria de ressaltar que mudanças ainda podem acontecer.

Vamos nos basear no template abaixo para esboçar nossa API.

Template:

Ator:
Comando:
Evento:

Contexto delimitado:

Agregado:

Recurso(s) envolvido(s):

Descrição do relacionamento:

Representação do relacionamento entre o recurso e o item contido, em formato de path:

Path:

Método HTTP:

Autenticação:

Explicando o template:

Contexto delimitado: Relembrando um pouco do que já foi dito, contexto delimitado seria a divisão de um domínio em partes menores que possuem suas próprias regras, modelos de negócio e interagem entre si para no todo formar o domínio. Cada parte dessas seria um contexto delimitado, ou seja, uma área com uma fronteira que define suas limitações, linguagem ubíqua, as intenções de suas entidades e o que cabe somente a ela e a nenhum outro contexto.

Ator: É aquele que executa a ação.

Agregado: Define em cima de quem um conjunto de comandos e eventos relacionados acontece, ou seja, em cima de quem é realizada a ação.

Os agregados podem-se dizer que são entidades que possuem um relacionamento forte e geralmente devem ser persistidas numa única transação, além de possuir apenas um repositório por agregação.

Por exemplo, imagine que de forma geral um cliente faça um pedido em nossa plataforma de delivery seguindo os seguintes passos:

1º- Cliente se cadastra

2º- Cliente realiza pedido

Pode-se dizer que cliente é um agregado e pedido com seus produtos é outro agregado. Pois primeiro eu cadastrei cliente e posteriormente quando o pedido foi realizado eu cadastrei o pedido juntamente com seus produtos.

Num outro exemplo, num vídeo do Eduardo Pires que eu assisti, ele faz o seguinte questionamento: cliente e endereço são entidades distintas ou devem compor uma agregação?

Segundo ele, os limites de uma agregação são determinados pela regra de negócio. Então, para entender o limite, devemos nos perguntar se esse endereço serve pra outra coisa além de pertencer a um cliente. Caso não, deveria compor uma agregação que tem cliente como raiz.

Outra forma de pensar em agregado é pensar se faz sentido acessar um item filho, sem antes acessar o pai.

Dessa forma deixo uma pergunta: Faz sentido acessar um item de pedido, sem antes acessar o pedido? Se a resposta for não, temos um agregado de pedido e item de pedido.

Comando: São ações que representam as operações do software e são descritas usando o verbo no infinitivo. Assim que invocadas podem desencadear um ou mais eventos.

Evento: É construído da seguinte forma [sujeito] + [verbo] e sempre no passado. Considera-se na sua descrição que uma ação importante para o negócio já aconteceu. A descrição deve ser de fácil compreensão por pessoas não técnicas.

Método HTTP: O protocolo HTTP possui alguns métodos de requisição, ou, como também são chamados, verbos. Os verbos HTTP definem qual ação deve ser realizada em determinado recurso e, dependendo do verbo, o servidor pode retornar uma mensagem diferente.

Path: Parte da URL que representa o caminho para acessar o recurso desejado.

Recurso(s) envolvido(s): É qualquer coisa que seja o alvo de uma requisição HTTP. Pode ser um documento, foto, áudio, vídeo ou qualquer outra coisa.

Funcionalidade start, foco do nosso esboço.

Neste momento nós não vamos esboçar a API da nossa plataforma de delivery de forma completa, vamos ser mais direto ao ponto e focar apenas na funcionalidade start, que à funcionalidade que deu origem a tudo isso que fizemos até aqui. Num outro momento podemos expandir um pouco mais a nossa API.

Nosso foco agora é na funcionalidade start, pois tivemos todo um trabalho para colocar a nossa oportunidade fictícia no papel e destrinchar tudo à sua volta.

Procuramos entender como criar uma API completa que possibilitasse a funcionalidade start existir e fazer parte de uma plataforma de delivery nova no mercado. Com isso entendemos como a funcionalidade start se encaixava na nova plataforma de delivery e para quem ela deveria prover recursos.

A funcionalidade start faz parte do subdomínio principal (core subdomain) da nossa plataforma e traz uma vantagem competitiva para a empresa, além de a diferenciar no mercado. Ela pode, quem sabe, até ser extraída da plataforma de delivery e ser transformada num novo negócio.

Com base nessa percepção, podemos focar somente na funcionalidade start e o subdomínio ao qual ela pertence, pois ela pode ser tratada como uma aplicação que independe das outras funcionalidades para existir, já que tudo gira à sua volta.

Ela sozinha pode ser transformada num novo negócio que permite que a nossa plataforma de delivery fictícia ou outras plataformas de delivery antes concorrentes e agora parceiras, possam desfrutar da nossa funcionalidade de enviar o status do pedido em forma de posts de mídias sociais aos seus clientes.

Dessa forma monetizamos a nossa inteligência de negócio para além da nossa própria plataforma, pois conseguimos criar novas formas de receita vendendo aquilo que somos especialistas e nos diferenciamos no mercado.

15- Esboçando nossa API baseado na funcionalidade start

Abaixo vamos esboçar os paths e relacionamentos na nossa API de postagem de status do pedido para delivery, tomando como base os comandos e eventos definidos anteriormente no “Arquivo1” listado mais abaixo.

Os comandos e eventos que serão utilizados na nossa funcionalidade start estão dentro da nossa jornada background de adição de postagem de mídia referente ao status do pedido.

Arquivo1:

1-Colocando-a-ideia-da-api-de-postagem-de-status-do-pedido-para-delivery-no-papel.pdf

Dentro da nossa jornada background abaixo, pegamos apenas a parte que nos interessa e representa a funcionalidade start.

Jornada background de adição de postagem de mídia referente ao status do pedido:

Plataforma de delivery
Comando: Adicionar postagem de mídia referente ao status do pedido
Evento: Mídia adicionada
Evento: Descrição da mídia adicionada
Evento: Postagem de mídia referente ao status do pedido adicionada

Contexto delimitado: Postagem

Agregado: Postagem

Recurso(s) envolvido(s): postagem

Descrição do relacionamento:

Recurso postagem > Possui uma coleção do tipo > postagem

Representação do relacionamento entre o recurso e o item contido, em formato de path:

/postagens/{idPostagem}

Path: /postagens

Método HTTP: POST (Adiciona nova postagem)

Autenticação: Obrigatória

Ao adicionar postagem de mídia referente ao status do pedido, como dependência plataforma de delivery precisa poder: Gerenciar postagem de mídia referente ao status do pedido

Gerenciar postagem de mídia referente ao status do pedido é composta por alguns comandos básicos, que podem ser executados sempre que uma postagem for adicionada.

Para a empresa gerenciar postagem de mídia referente ao status do pedido, a plataforma de delivery precisa poder:

Plataforma de delivery

Comando: Pesquisar postagens de mídia referente ao status do pedido

Evento: Postagens de mídia referente ao status do pedido pesquisadas

Contexto delimitado: Postagem

Agregado: Postagem

Recurso(s) envolvido(s): postagem

Descrição do relacionamento:

Recurso postagem > Possui uma coleção do tipo > postagem

Representação do relacionamento entre o recurso e o item contido, em formato de path:

/postagens/{idPostagem}

Path: /postagens?pedido={codPedido}

Método HTTP: GET (Retorna uma coleção de

postagens do pedido)

Autenticação: Obrigatória

Plataforma de delivery

Comando: Remover postagem de mídia referente ao status do pedido

Evento: Postagem de mídia referente ao status do pedido removida

Contexto delimitado: Postagem

Agregado: Postagem

Recurso(s) envolvido(s): postagem

Descrição do relacionamento:

Recurso postagem > Possui uma coleção do tipo > postagem

Representação do relacionamento entre o recurso e o item contido, em formato de path:

/postagens/{idPostagem}

Path: /postagens/{idPostagem}

Método HTTP: DELETE (Remove um item da coleção de postagens)

Autenticação: Obrigatória

Obs: No miro nos dois comandos acima, o ator que foi definido é o ator: Empresa (Responsável pela fase em que o pedido se encontra), mas eu mudei aqui neste documento para Plataforma de delivery, para deixar mais genérico. Pois pegamos essa parte do desenho da nossa própria plataforma de delivery e separamos num novo produto de API, que serve tanto a nossa própria plataforma, quanto para outras plataformas externas que contratam nosso serviço de postagem de mídia referente ao status do pedido.