

API de integração com o *Marketplace* do KuntoKusta

Diogo Rolo and F. Jorge F. Duarte

Instituto Superior de Engenharia do Porto
1161199@isep.ipp.pt; fjd@isep.ipp.pt

Resumo. Este artigo evidencia todo o trabalho desenvolvido na implementação de uma REST API [1] de integração com o *Marketplace* do KuntoKusta. A API tem como finalidade, possibilitar o acesso e a alteração de informação relativa às vendas, aos produtos e às encomendas das lojas aderentes ao *Marketplace* do KuntoKusta, de forma segura, utilizando um sistema de API Keys.

Palavras-chave: *Marketplace*, Gestão de acessos, REST API, Segurança.

1 Introdução

Os Marketplaces [2] são plataformas de e-commerce de venda de produtos de diversas lojas aderentes e que têm vindo cada vez mais a ganhar protagonismo no mercado tecnológico. As lojas aderentes dos Marketplaces, manifestam frequentemente, a preocupação pela perda do relacionamento entre o cliente e a loja aderente e a perda dos dados comportamentais e de compra subsequentes. Pelo facto, das lojas aderentes terem um papel importantíssimo nestas plataformas, é necessário possibilitar uma melhor utilização das mesmas, dando-lhes acesso à informação sobre os produtos e encomendas efetuadas. No caso específico do Kuntokusta, foi desenvolvida uma API focada nos interesses das lojas aderentes ao Marketplace do KuntoKusta, dando-lhes a possibilidade

2 Utilização de *Marketplaces* por parte das lojas e consumidores

Os consumidores digitais aprovam a utilização de Marketplaces, pois estes permitem a compra de vários produtos de diferentes marcas com apenas uma única transação. Muitos Marketplaces permitem ainda a comparação do preço de um produto em diferentes lojas, possibilitando o consumidor fazer a melhor compra possível. O comércio eletrónico já vinha em crescendo durante os últimos anos e o aparecimento do COVID-19 e o consequente confinamento, veio fazer com que esses números aumentassem ainda mais. As pessoas viram-se forçadas a recorrer cada vez mais ao ‘e-commerce’ e foram registados crescimentos sem precedentes nestas plataformas.

Como é possível verificar no gráfico de barras da Fig. 1, é notório o aumento de compras em plataformas ‘e-commerce’ durante os últimos anos. Pode-se esperar que esse crescimento se venha a constatar nos próximos anos [3].

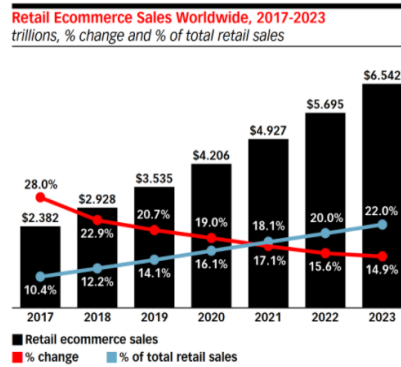


Fig. 1. Vendas do comércio eletrônico em todo o mundo (Fonte: eMarketer, Maio 2019)

Com este crescimento e com a necessidade da compra online cada vez mais acentuada e banalizada, as lojas veem-se forçadas a entrar neste mercado. Apesar da entrada em Marketplaces por parte das lojas, parecer algo atrativo, ainda existem muitas que não olham para essa oportunidade, como algo acertado.

Segundo um estudo realizado pela Jumpshot, em 2018 [4], é constatado que apesar dos benefícios, 83% das lojas associadas e presentes em Marketplaces, se preocupam com os mercados de comércio eletrônico ou e-commerce. Razões como a perda do relacionamento entre o cliente e a loja e dos dados comportamentais e de compra subsequentes, são algumas das razões que ainda afastam algumas lojas dos Marketplaces.

Para ajudar a resolução deste problema, foi desenvolvida uma API que permitisse às lojas aderentes, perder alguns destes receios e continuar a ter controle da informação da encomenda, assim como do cliente, apesar desta ser tratada pelo Marketplace do Kuantokusta diretamente. Esta API visa proporcionar às lojas aderentes um uso otimizado da plataforma, assim como fornecer uma rápida e segura resposta às lojas clientes que pretendam saber ou alterar informação relativa às mesmas.

3 Tecnologias de desenvolvimento utilizadas

Tendo em conta que o principal objetivo consistiu no desenvolvimento de uma API de integração com o Marketplace do Kuantokusta, foram utilizadas tecnologias compatíveis com os serviços do mesmo. Para tal e por decisão da empresa, foram utilizadas tecnologias já usadas anteriormente. Foi utilizada a framework NestJS [5], indicada para consumir aplicações Node.js eficientes e escaláveis, do lado do servidor. Esta framework suporta totalmente TypeScript, linguagem que foi escolhida para o desenvolvimento, de modo a manter uma certa coerência com os serviços já existentes. Para o armazenamento e pesquisa de informação, foi utilizado o MongoDB. Este tipo de base dados é NoSQL e é orientado a documentos para armazenamento de dados de alto volume. Em vez de serem utilizadas tabelas e linhas como nas bases de dados relacionais, o MongoDB utiliza documentos do tipo JSON e coleções.

4 Desenvolvimento

Como se pretendia, possibilitar às lojas aderentes, a gestão dos seus produtos e encomendas, foi necessário dar prioridade à automatização do processo para proporcionar uma maior comodidade e garantir que todos os pedidos efetuados à API estivessem protegidos. Para garantir a segurança da API, todos os pedidos efetuados à mesma, estão protegidos por duas validações. É verificada a autenticidade do token privado inserido (API Key), bem como o “role” do utilizador que efetuou o pedido. Se o “role” do utilizador coincidir com o configurado e se se confirmar a compatibilidade entre os dois roles, o acesso é validado e o pedido é processado.

Todas as funcionalidades desenvolvidas e que são apresentadas na secção seguinte, estão protegidas com os sistemas de proteção referidos anteriormente e ainda com alguns mecanismos adicionais como a limitação da taxa [6] e validation pipes [7].

4.1 Funcionalidades implementadas

As funcionalidades apresentadas nas secções seguintes, foram implementadas com o objetivo de permitir ao utilizador consultar e alterar informação sobre as suas encomendas e tornar a utilização da plataforma mais convidativa e útil. Todas as funcionalidades sofrem uma verificação do token privado, para confirmar a autenticação e uma verificação de permissões para validar a entrada, antes do pedido ser processado.

Criar chave identificadora privada

Através desta proteção é assegurada a autenticidade de um utilizador quando este faz um pedido. Cada utilizador possui uma chave que o acompanhará e que será utilizada sempre que este efetuar um pedido, permitindo assim a confidencialidade da sua informação pessoal. Quando a loja faz o pedido, é efetuada uma verificação para validar se o pedido é feito por um utilizador autorizado. De seguida, e depois de validada e autorizada a entrada, é recolhida informação para se prosseguir com a criação da chave. A informação é recolhida a partir de um payload que contém informação sobre a loja. O payload foi extraído a partir de um JWT [8]. Após a recolha de toda a informação, é criada uma hash MD5 [9] que será usada como chave de acesso a outros pedidos/endpoints por parte da loja.

O algoritmo MD5 é uma função hash que permite transformar informação num valor hash de 128 bits. Este algoritmo foi o eleito entre um leque de diferentes algoritmos de encriptação, tendo em conta, fatores como, o comprimento da token gerado (128 bits), a velocidade em gerar o token e a taxa de colisão extremamente baixa.

Listar todas as encomendas feitas na sua loja

Esta funcionalidade permite ao utilizador, listar todas as encomendas efetuadas anteriormente na sua loja, por intermédio do Marketplace do KuntoKusta.

Envio, aprovação e cancelamento de uma encomenda

Esta funcionalidade permite ao utilizador, alterar o estado de uma encomenda efetuada na sua loja, permitindo que esta seja enviada, aprovada para ser posteriormente enviada ou cancelada caso exista algum problema na encomenda.

Visualização de todas as transportadoras existentes

Esta funcionalidade permite ao utilizador obter todas as empresas de transporte existentes, de forma a proceder ao envio da encomenda e definir qual a transportadora que irá ser responsável pela entrega.

Visualização de todas as razões de cancelamento

Esta funcionalidade permite ao utilizador obter todas as razões de cancelamento existentes, para proceder ao cancelamento de uma encomenda.

Consulta e alteração de informação dos seus produtos

Esta funcionalidade permite ao utilizador consultar e alterar informação dos seus produtos que se encontram à venda no Marketplace do KuntoKusta.

4.2 Implementações adicionais

Foram ainda implementadas algumas técnicas tanto para proteger a aplicação, como o utilizador no uso da mesma.

Limitação de taxa

A limitação de taxa, é uma técnica comum para proteger as aplicações de ataques de força bruta (Brute-force attack) [11]. De uma forma geral, permite controlar a taxa em que as solicitações dos utilizadores são processadas pelo servidor.

Uma vez que a API está em produção e é utilizada por um vasto número de utilizadores, foi necessário recorrer ao controlo dos fluxos existentes. Para tal, criaram-se limitadores de taxa diferenciados, que permitam evitar que um sistema seja sobrecarregado em situações benignas ou maléficas. Assim, pode-se bloquear um utilizador, durante um tempo previamente definido, se a sua atividade for suspeita, isto é, se este efetuar excessivos pedidos à API e a determinadas rotas.

Filtros de exceção

Uma exceção é um evento, que ocorre durante a execução de um programa e que interrompe o normal fluxo do programa. Os filtros de exceção são utilizados para capturar erros que não são tratados pela aplicação, enviando uma mensagem automática com uma resposta apropriada. O NestJS possui uma camada de exceções embutida, que é responsável pelo tratamento desses erros, assim como pelo envio das mensagens.

Embora exista um filtro de exceção embutido, capaz de lidar com muitos erros, foi desenvolvido um filtro de exceção personalizado, para ser possível obter o controlo total sobre a camada de exceções, e permitir ao programador controlar o fluxo exato e o conteúdo da resposta enviada, de volta para o cliente.

O filtro de exceção personalizado criado, é responsável por capturar exceções (que são instâncias da classe `HttpException`) e por enviar uma resposta de erro personalizada.

Loggers

Os Loggers permitem imprimir no standard output, vários tipos de mensagens em runtime. Podem ser utilizados localmente num controlador ou globalmente por todo o programa e servir para fins estatísticos, pois é possível saber a quantidade de vezes que um utilizador acede a um servidor, quantas vezes falha ao aceder e quantas vezes é bem sucedido.

Validation Pipes

Validation Pipe é uma prática fornecida pelo NestJS, recomendada para validar a exatidão de todos os dados enviados para uma aplicação web. O Validation Pipe faz uso do package `class-validator` e dos seus decoradores de validação declarativa. Fornece uma abordagem padrão para impor regras de validação para todos os dados de entrada.

Desta forma, a API valida se os dados recebidos são válidos e se existem e só depois de passar em todas as validações, é que processa o pedido.

5 Conclusões

Todas as funcionalidades inicialmente definidas para serem desenvolvidas neste trabalho, foram implementadas com sucesso, encontrando-se a API em produção. As novas funcionalidades trouxeram outro tipo de confiança ao utilizador (loja) na utilização do Marketplace do KuntoKusta. O receio em perder a relação cliente-loja deixou de existir, pois, todos os dados de encomenda estão disponíveis e visíveis à loja, tornando o Marketplace do KuntoKusta um mero intermediário da transação. As implementações efetuadas relativas à segurança da API, como a gestão de acessos com a utilização de tokens privados, acessos restritos a determinadas informações a partir de uma verificação de permissões, ou até, o uso de limitadores de taxa que permitem bloquear diversos tipos de ataques, tornaram mais convidativa a utilização da mesma. O utilizador sente-se mais seguro já que a sua informação está mais protegida.

Relativamente ao feedback manifestado por parte das lojas aderentes, foi possível constatar, que a API veio simplificar bastante o contacto entre o cliente e a loja, tornando todo o processo de venda mais otimizado. Foi possível verificar também, um aumento no número de adesões ao Marketplace do KuntoKusta, visto a API ter contribuído com a eliminação de alguns receios que existiam por parte das lojas, em aderir a este tipo de negócio.

Referências

1. “What is REST - REST API Tutorial.” <https://restfulapi.net/>, último acesso em 2020/10/02.
2. “What Is A Marketplace? Our understanding of multi-seller businesses.” <https://insights.shoperly.com/what-is-a-marketplace>, último acesso em 2020/10/02.
3. “eMarketer. 2020. Global Ecommerce 2019.” <https://www.emarketer.com/content/global-ecommerce-2019>, último acesso em 2020/10/02.
4. “Prnewswire.com. 2020. New Survey Reveals 84 Percent Of Marketers Lack Insights Into Purchase Behaviors Across Ecommerce Marketplaces.” <https://www.prnewswire.com/news-releases/new-survey-reveals-84-percent-of-marketers-lack-insights-into-purchase-behaviors-across-ecommerce-marketplaces-300669106.html>, último acesso em 2020/10/02.
5. “NestJS - A progressive Node.js framework.” <https://nestjs.com/>, último acesso em 2020/10/03.
6. “Security | NestJS - A progressive Node.js framework.” <https://docs.nestjs.com/techniques/security#rate-limiting>, último acesso em 2020/10/04.
7. “Pipes | NestJS - A progressive Node.js framework” <https://docs.nestjs.com/pipes>, último acesso em 2020/10/03.
8. “JSON Web Algorithms (JWA) “ <https://tools.ietf.org/html/rfc7519>, último acesso em 2020/10/03.
9. “MD5 – What is MD5? - Definition from WhatIs.com“ <https://searchsecurity.techtarget.com/definition/MD5>, último acesso em 2020/10/03.
10. “Back Office Definition“ <https://www.investopedia.com/terms/b/backoffice.asp>, último acesso em 2020/10/04.
11. “What is a Brute Force Attack? Definition | Varonis “<https://www.varonis.com/blog/brute-force-attack/>, último acesso em 2020/10/04.