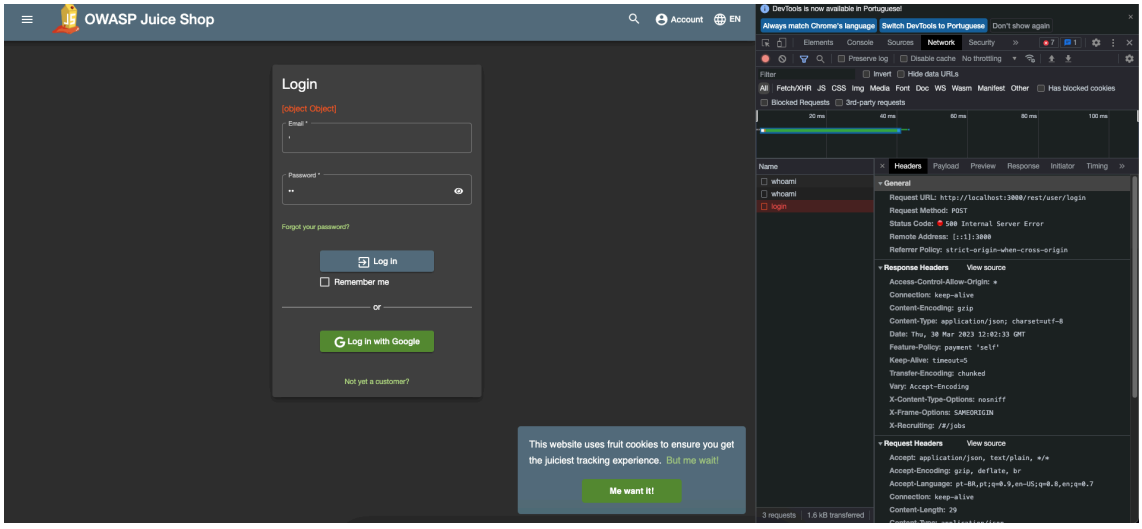
Faça login com a conta de usuário do administrador

Usando testes semelhantes ao que realizamos, Burp foi capaz de determinar que o endpoint da API rest/user/login é vulnerável à injeção de SQL.
Agora, vamos tentar explorar isso, assumindo que a consulta executada no back-end sempre que uma tentativa de login ocorre é algo como isto:

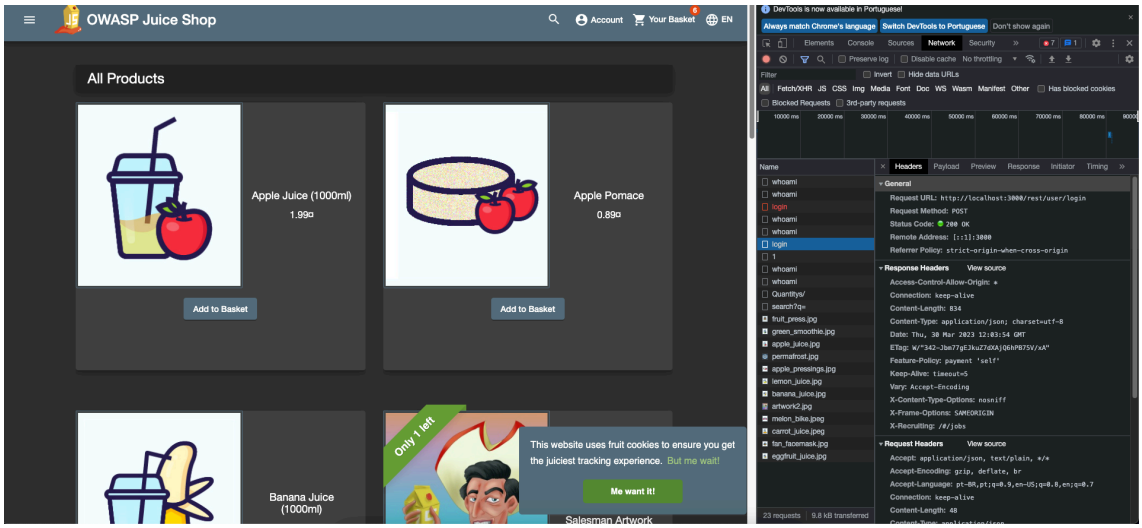
SELECT ... FROM users WHERE email = '<input>' and password = '<hashed input>';
Podemos tentar outro payload simples: ' or 1=1– . Assim a consulta fica:

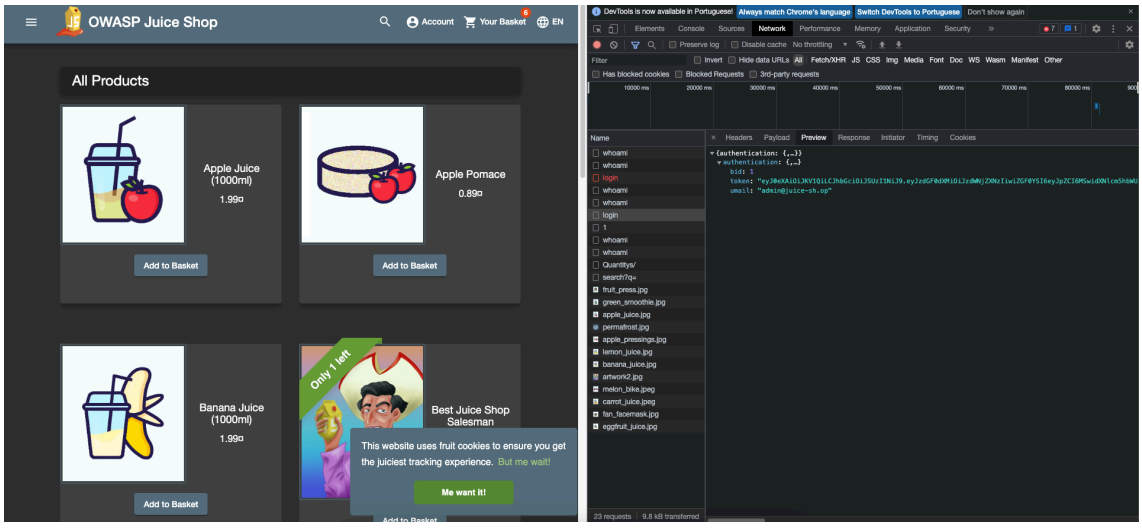
SELECT ... FROM users WHERE email = '' or 1=1--' and password = '...';
Observe que – é o símbolo de comentário no SQL, portanto, a instrução SQL que será executada é:

SELECT ... FROM users WHERE email = '' or 1=1;
Isso nos dará acesso à conta de administrador , que é a primeira entrada na tabela de usuários, resolvendo assim esse desafio.

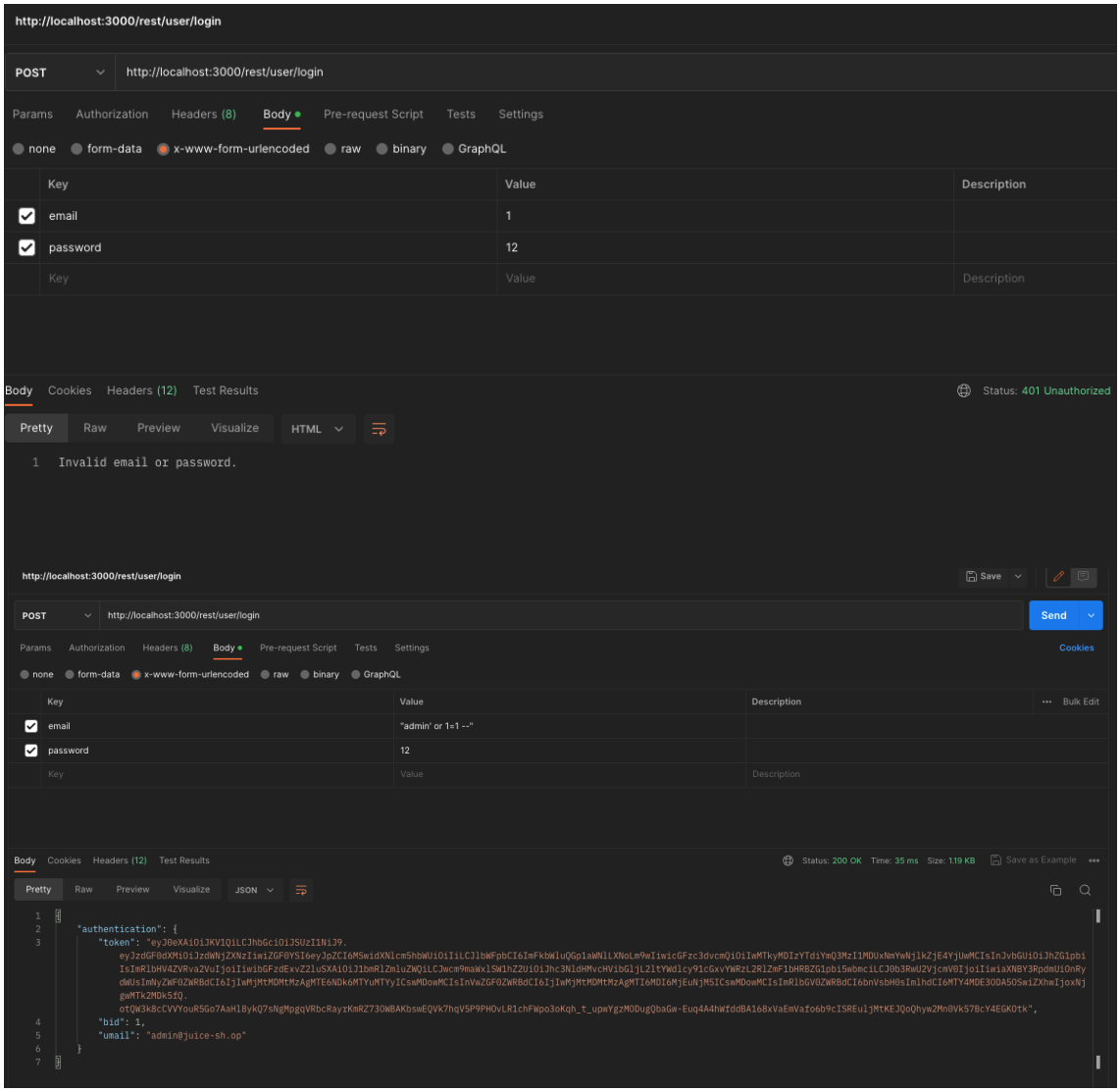


```
describe('User Login', () => {
    it('SQL Injection', async () => {
        await req(API_URL)
        .post('/sql/login')
        .send({
            "email": "admin' or 1=1 --",
            "password": "admin"
        })
        .set("Accept", "application/json")
        .then(response =>{
            expect(response.statusCode).toEqual(401)
        })
    });
});
```

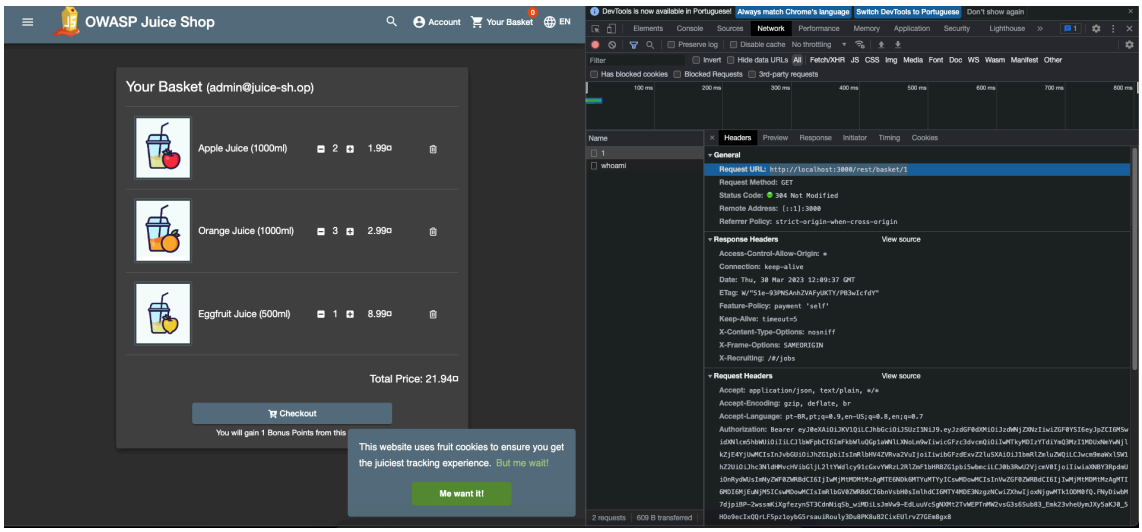




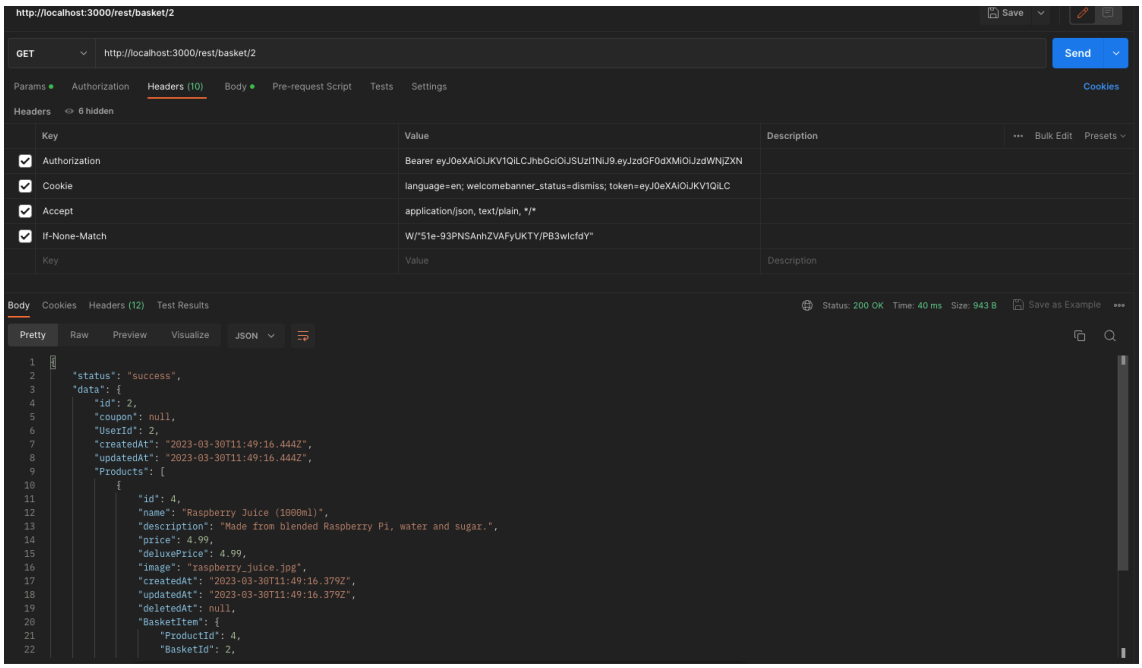
POSTMAN



Exibir a cesta de compras de outro usuário

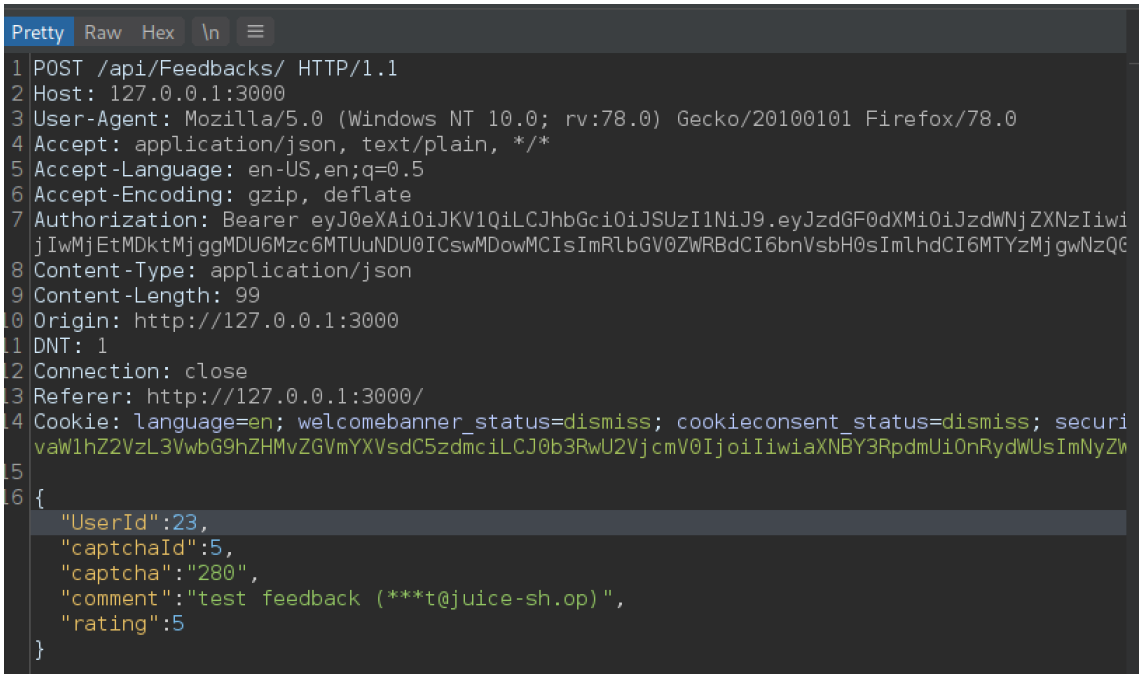


Podemos ver que o formato é rest/basket/BasketId, nosso BasketId é 1, realizei o aumento em 1 para ver se conseguimos acessar a cesta 2 que pertence a outro usuário.



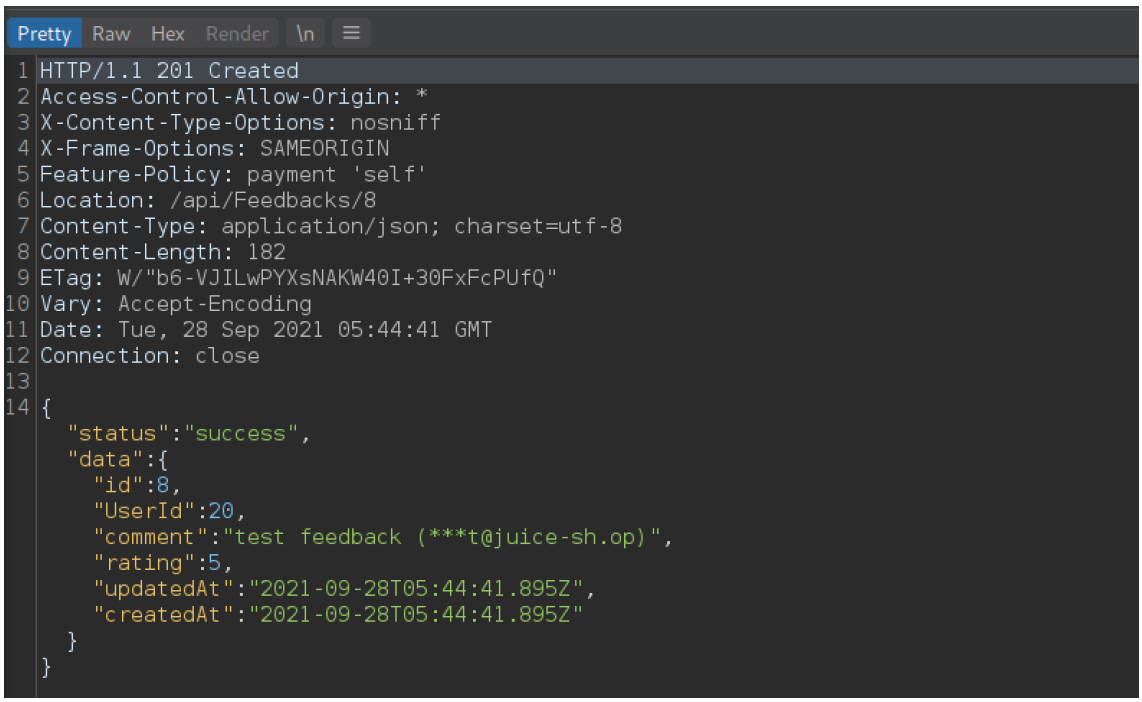
Poste algum feedback em nome de outro usuário

De /contact, um usuário pode enviar seu feedback, que será exibido em /administration . Depois de navegar para /contact , enviar um feedback e interceptar a solicitação com o proxy Burp, obtemos o seguinte:

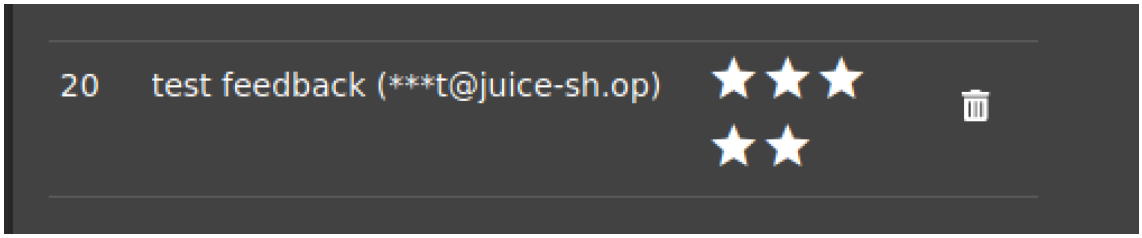


É visto que meu usuário é UserId é 23 (que é um campo oculto), mas o aplicativo realiza as verificações necessárias para detectar se um usuário tenta se passar por

outro usuário e enviou um feedback usando o id desse usuário? Realizie a mudança para o UserId para 20 para ver o que acontece.



O servidor responde com o código de status 201 criado e podemos ver no JSON retornado que o UserId foi de fato alterado para 20 para esse feedback.
Na página /administração, podemos ver nossos comentários enviados sob UserId 20.



Portanto, o aplicativo não realiza as verificações necessárias para verificar se o usuário que está enviando o feedback (sessão atual) é quem afirma ser (valor do campo oculto UserId).