OCSIM

Um framework open source para autenticação e autorização

Jéssica Bonson
Principal Engineer no Olist

TDC Floripa 2020

olist

Jéssica Pauli de C Bonson

- +-8 anos de exp em pesquisa/desenvolvimento
- graduação/mestrado em Ciências da Computação
- foco em dev backend, machine learning e big data

Hobbies:

Jogos
RPG
Canto

olist

Maior loja nos principais marketplaces do Brasil.

Arquitetura em microsserviços e serverless.
Python. Go. PostgreSQL. AWS. Heroku.

olist
empowering commerce

O que é autenticação e autorização?

Tipos de Autenticação de APIs

- Basic

  curl "https://example.com/" -H "Authorization: **Basic** dXNlcm5hbWU6cGFzc3dvcmQ="

Tipos de Autenticação de APIs

- Basic

  curl "https://example.com/" -H "Authorization: **Basic** username:password"

Tipos de Autenticação de APIs

- API Key

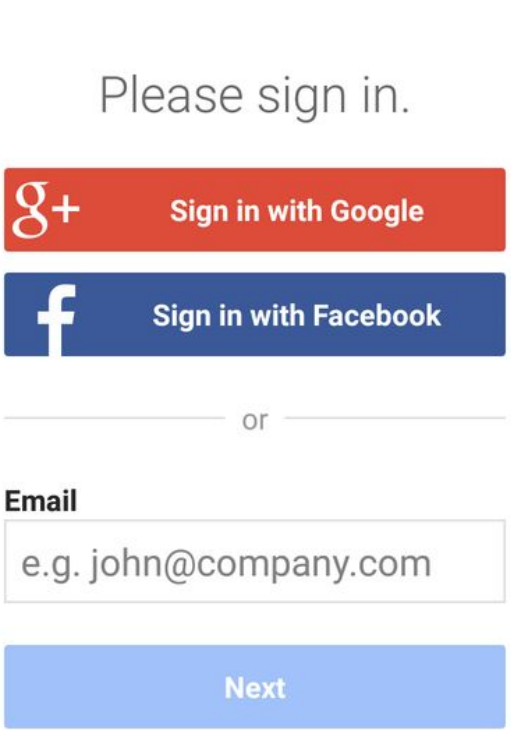
  curl "https://example.com/" -H "Authorization: **Token** 7h2aa9d9172f329b1k67113a9f5cc9ns2f”

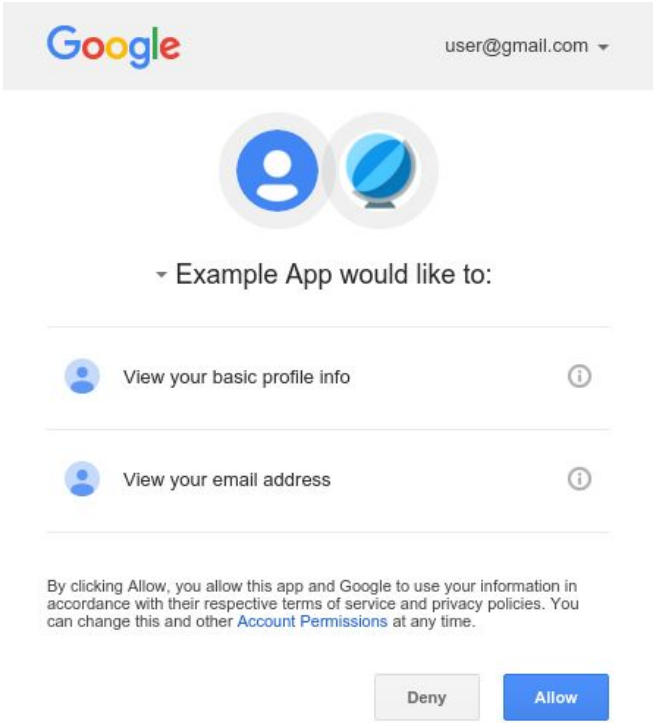
Tipos de Autenticação de APIs

- OAuth 2.0

  curl "https://example.com/" -H "Authorization: **Bearer** eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJtZXNzYWdlIjoiSldUIFJ1bGVzISIsImlhdCI6MTQ1OTQ0ODExOSwiZXhwIjoxNDU5NDU0NTE5fQ.-yIVBD5b73C75osbmwwshQNRC7frWUYrqaTjTpza2y4"

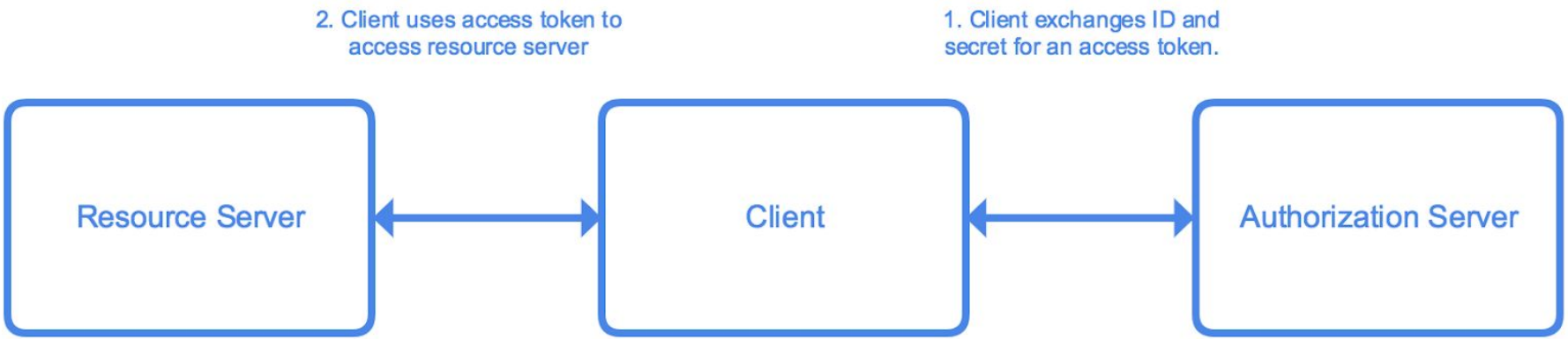
OAuth 2.0: Authorization Code





olist

OAuth 2.0: Client Credentials



Tipos de Autenticação de APIs

- OAuth 2.0

  curl "https://example.com/" -H "Authorization: **Bearer** eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJtZXNzYWdlIjoiSldUIFJ1bGVzISIsImlhdCI6MTQ1OTQ0ODExOSwiZXhwIjoxNDU5NDU0NTE5fQ.-yIVBD5b73C75osbmwwshQNRC7frWUYrqaTjTpza2y4"

Tipos de Autenticação de APIs

- OAuth 2.0

  curl "https://example.com/" -H "Authorization: **Bearer** eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJtZXNzYWdlIjoiSldUIFJ1bGVzISIsImlhdCI6MTQ1OTQ0ODExOSwiZXhwIjoxNDU5NDU0NTE5fQ.-ylVBD5b73C75osbmwwshQNRC7frWUYrqaTjTpza2y4"

Formato: header.payload.signature

Tokens JWT (JSON Web Token)

- Header

  {"typ": "JWT", "alg": "HS256"}

- Payload

  {"message": "JWT Rules!", "iat": 1459448119, "exp": 1459454519}

- Signature

  Valida que o token não foi alterado e é de origem confiável

OAuth 2.0

+

Token JWT

olist

OpenID Connect (OIDC)

OAuth 2.0

+

ID Token (Token JWT)

+

Endpoints (API HTTP RESTful)

OpenID Connect (OIDC): ID Token


```
{
  "iss": "https://server.example.com",
  "sub": "NzbLsXh8uDCcd-6MNwXF4W_7noWXFZAfHkxZsRGC9Xs",
  "aud": "https://api.example.org",
  "exp": 1311281970,
  "iat": 1311280970
 }
```

Arquitetura do Olist

olist

Tipos de Autenticação de APIs

- API Key

  curl "https://example.com/" -H "Authorization: **Token** 7h2aa9d9172f329b1k67113a9f5cc9ns2f”

OpenID Connect (OIDC)

OAuth 2.0

+

ID Token (Token JWT)

+

Endpoints (API HTTP RESTful)

Vantagens

- Melhor **performance** das APIs
- Maior **segurança** das APIs
- **Rastreabilidade**
- **Permissionamento**

Características da Arquitetura

- APIs usam o Django Rest Framework (DRF)

- Usamos microserviços
  - 20+ APIs
  - 100+ serviços

Requisitos

- Facilidade de instalação
- Retrocompatibilidade
- Token auto-contido
- Não reinventar a roda

OCSIM

- Foi construído em cima do django-oauth-toolkit, a biblioteca recomendada pelo Django Rest Framework (DRF), que implementa o OAuth 2.0
- Adicionamos suporte ao padrão OIDC
- Solução pronta para uso

O que é o OCSIM?

- API (ocsim)
- Biblioteca (ocsim-drf)
- Autenticador

Usado há quase 1 ano em produção, mas ainda está em desenvolvimento.

API do OCSIM

Endpoints, Banco de Dados, Comandos

olist

Comandos: Criar Aplicação

heroku run python ocsim/manage.py **applications create --name** "dummy-service" **--client-type**="confidential" **--auth-grant-type**="client-credentials" **--organization-name** "Olist"

Application: dummy-service
Client ID: 93pKbKujRiO8hAkdTNyf8yA2A4OEliDfbQo6kIxE
Client Secret: mOiUxPoyNfFure5cZwsL8dFkJjg2lBiW18TC08bo1vLuGjA2e48XuAV4LEedPBVzH7Hw30MtoyjJqvEJylw025lvPtxFpsM7ZOCMHTbZVez2yxz81ndqVlJDAZWbdKoZ

Comandos: Criar Permissão

heroku run python ocsim/manage.py **permissions create --name**="Dummy API: Get product" **--orn**="orn:dummy-api:products:retrieve"

Permission created.
NAME: Dummy API: Get product
ORN: orn:dummy-api:products:retrieve

Comandos: Aplicação com Permissões

heroku run python ocsim/manage.py **applications add-permission**
--client-id 93pKbKujRiO8hAkdTNyf8yA2A4OEliDfbQo6kIxE
--orn orn:dummy-api:products:retrieve

heroku run python ocsim/manage.py **applications remove-permission**
--client-id 93pKbKujRiO8hAkdTNyf8yA2A4OEliDfbQo6kIxE
--orn orn:dummy-api:products:retrieve

Como usar?

https://example.com/o/token/

curl -X POST -d "**client_id**=<client_id>" -d "**client_secret**=<client_secret>"

-d "**grant_type**=client_credentials" https://example.com/o/token/

{"**id_token**": "<token>", "**token_type**": "Bearer", "**expires_in**": 3600}

Token do OCSIM


```
{
    "exp": 1582607027,
    "iat": 1582603427,
    "profile": {
      "client_id": "93pKbKujRiO8hAkdTNyf8yA2A4OEliDfbQo6kIxE",
      "name": "dummy-service",
      "organization": {"name": "Olist"},
      "permissions": [
        {"orn": "orn:dummy-api:products:retrieve"}
      ]
    }
}
```

Setup

Adicionar a variável de ambiente:

- OIDC_RSA_PRIVATE_KEY (ou obtê-la do AWS Secrets)

- Instalação de dependências
- Criação do banco de dados
- Configuração no Heroku

Biblioteca do OCSIM

Validação de tokens na API

olist

Setup

Adicionar as variáveis de ambiente:

OIDC_RSA_PUBLIC_KEY e API_NAME

Alterar no REST_FRAMEWORK do settings.py:

```
"DEFAULT_AUTHENTICATION_CLASSES": (
    "ocsim_drf.authentication.OCSIMAuthentication",
),
"DEFAULT_PERMISSION_CLASSES": (
    "ocsim_drf.permissions.ORNPermission",
),
```

Debates

olist

- Tokens criptografados? (JWE)
- O que fazer com a autenticação por API Key?
- Como invalidar um token?
- Impacto na performance de um aplicativo

Valeu!

@jessica.bonson

@jpbonson

olist

Referências

https://auth0.com/learn/token-based-authentication-made-easy/

https://thiagolima.blog.br/parte-3-seguran%C3%A7a-em-apis-restful-a780bd9f186a

https://oauth.net/2/

https://developer.okta.com/blog/2017/06/21/what-the-heck-is-oauth

https://medium.com/google-cloud/understanding-oauth2-and-building-a-basic-authorization-server-of-your-own-a-beginners-guide-cf7451a16f66

https://connect2id.com/learn/openid-connect

https://openid.net/specs/openid-connect-core-1_0.html

https://auth0.com/docs/protocols/oidc

olist

Referências

https://blog.bearer.sh/the-three-most-common-api-authentication-methods/

https://nordicapis.com/3-common-methods-api-authentication-explained/

https://nordicapis.com/why-api-keys-are-not-enough/

https://nordicapis.com/api-security-oauth-openid-connect-depth/

https://nordicapis.com/how-to-control-user-identity-within-microservices/

https://swagger.io/docs/specification/authentication/bearer-authentication/

https://medium.com/@darutk/understanding-id-token-5f83f50fa02e