Autenticação via JWT na API (Layer & StoreFront/WebAPI)

- 1. INTRODUÇÃO
- 2. AUTENTICAÇÃO JWT – LAYER (API)
- 3. AUTENTICAÇÃO JWT – STORE FRONT (WEBAPI)

1. INTRODUÇÃO

É isso mesmo que você viu! Temos uma novidade técnica importantíssima para você!

Aprimoramos a autenticação das nossas APIs! Onde, antes usávamos o tradicional **Basic Auth**, que sempre garantiu segurança nos acessos.
Mas agora estamos evoluindo para **JWT (JSON Web Token)**, um modelo ainda mais **eficiente**, **flexível** e alinhado com as melhores práticas do mercado.

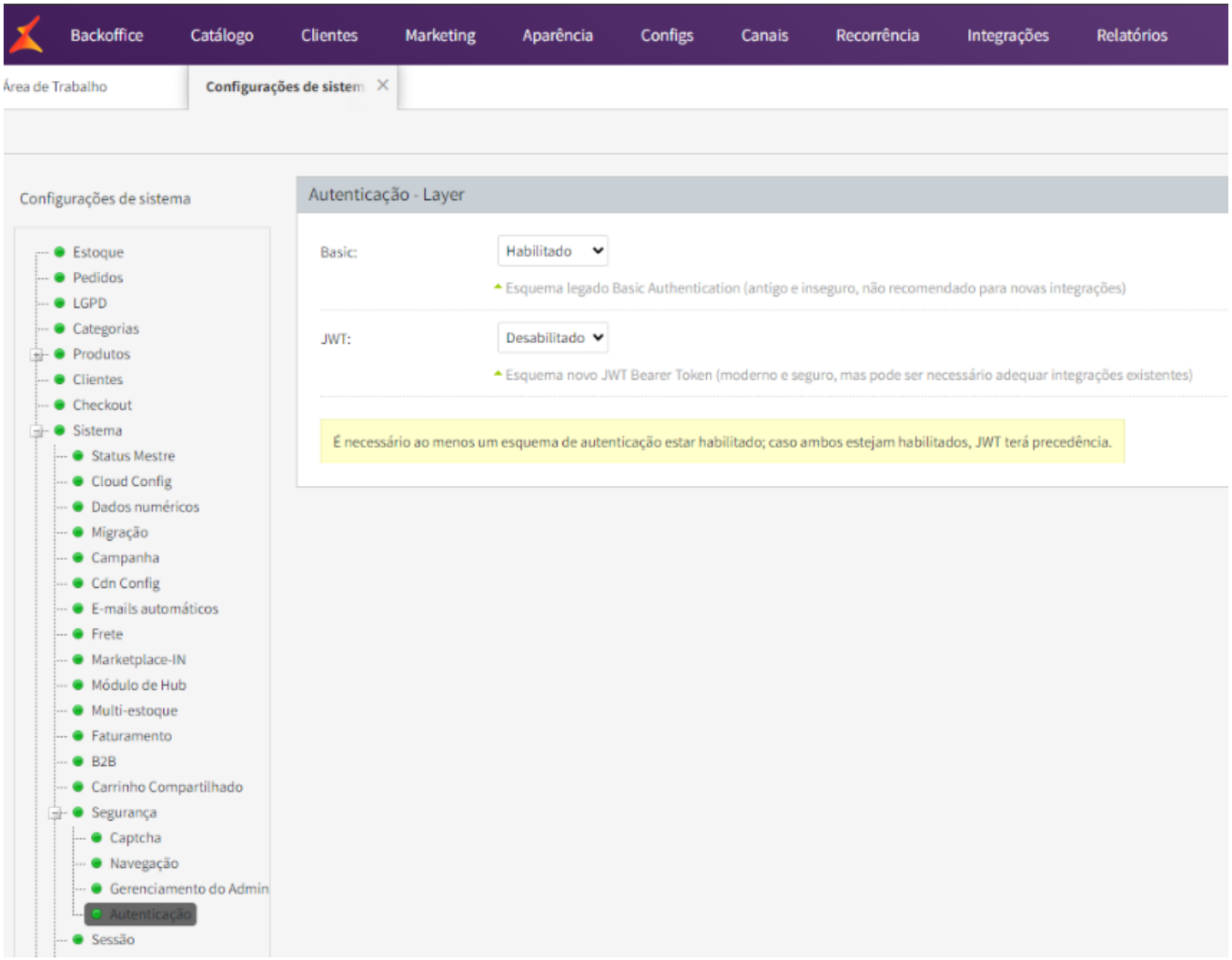
Por que essa mudança?

Mais performance: Menos chamadas ao servidor para validar acessos.
Mais flexibilidade: Melhor integração com diferentes serviços e arquiteturas modernas.
Mesmo nível de segurança (ou até melhor!): O Basic Auth sempre foi seguro, mas o JWT permite um controle mais refinado sobre autenticação e permissões.

E claro, este documento tem tudo que você precisa saber caso surjam dúvidas em meio ao processo!

2. AUTENTICAÇÃO JWT – LAYER (API)

Para habilitar a autenticação/autorização *JWT* na **Layer**, navegue até a seguinte tela do **Admin** (*Configs/Configurações gerais/Sistema/Segurança/Autenticação*):



Na área "*Autenticação - Layer*", selecione o valor "*Habilitado*" para o campo "*JWT*" e clique em "*Salvar*"; esta ação definirá a configuração "*Security.Authentication.EnableLayerJWTAuthentication*" com o valor "*True*"; aguarde até a atualização da configuração ser refletida na **Layer** para que o sistema passe a utilizar este tipo de autorização.

***Informação importante:** nesta tela, é necessário ao menos um esquema de autenticação estar habilitado; caso ambos estejam habilitados, *JWT* terá precedência, e será o esquema *default*.

ACESSO VIA INTERFACE GRÁFICA WEB DO SWAGGER

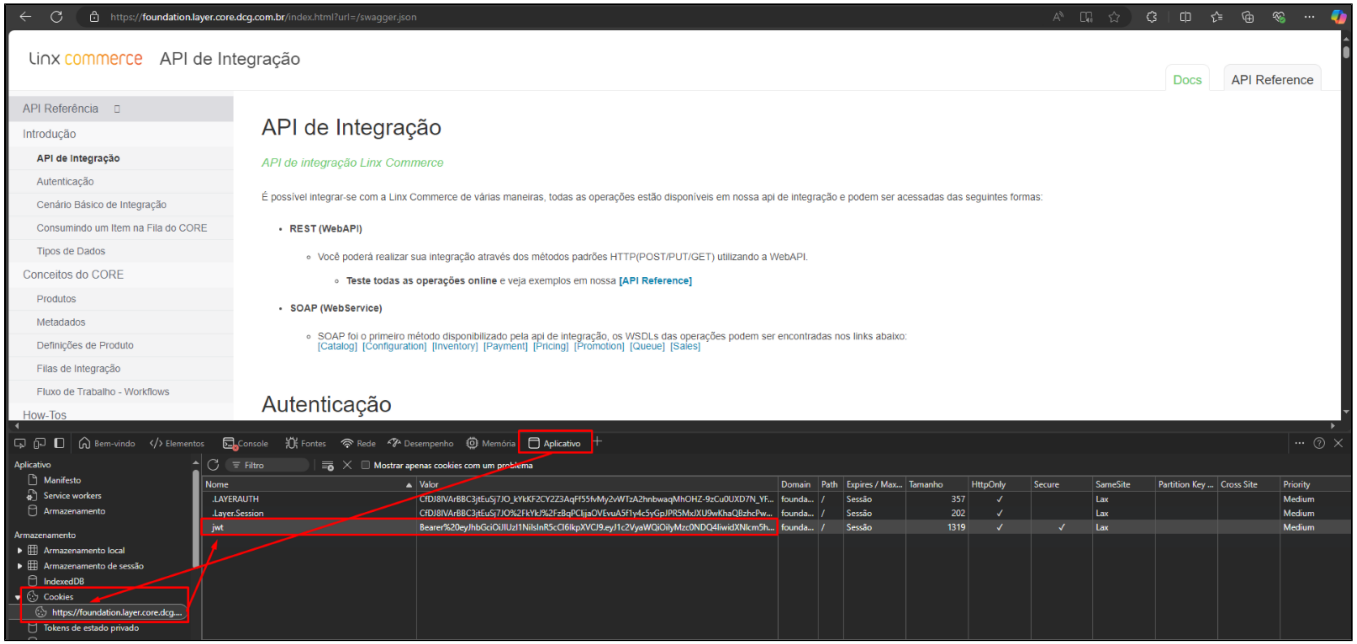
Acesse no navegador de sua preferência a *URL* no esquema *https://**<tenant_name>**.layer.core.dcg.com.br*.

Ex.: https://foundation.layer.core.dcg.com.br

Você será redirecionado para a rota "*/logon*" e verá então a seguinte tela, semelhante a tela de autenticação do site **Admin**:



Ao logar corretamente com suas credenciais, a interface do *Swagger* será carregada, por onde será possível realizar requisições aos *endpoints* da **Layer** de forma gráfica, e um *cookie* com o *token JWT* será configurado:



ACESSO VIA INTEGRAÇÃO - API

AUTENTICAÇÃO COM USUÁRIO E SENHA PARA OBTENÇÃO DO TOKEN JWT

Realizar uma requisição **POST** no *endpoint:*

*https://**<tenant_name>**.layer.core.dcg.com.br/v1/Security/API.svc/web/GenerateJWT*

...fornecendo o seguinte *payload* no *body*:

> *{ "Username": "<username>", "Password": "<password>" }*

...e os *headers*:

> *User-Agent: LayerIntegration (obrigatório)*
>
> *Accept: application/json*
>
> *Content-Type: application/json*

Se os dados de login estiverem incorretos, será devolvida uma *response* com o *HTTP Status Code* **401 – Unauthorized** e o seguinte *body*:


```
{
   "Errors": [
      {
         "ErrorCode": null,
         "ErrorMessage": "Usuário ou senha não confere",
         "HelpEntry": null,
         "Key": null,
         "TechnicalMessage": null
      }
   ],
   "IsValid": false,
   "Warnings": null,
   "ExpiresIn": "/Date(-62135585640000-0306)/",
   "Token": null
}
```


Caso as credenciais estejam corretas, será devolvida uma *response* com o *HTTP Status Code* **200 – OK** e um *body* com a estrutura parecida ao a seguir:

```
{
    "Errors": [],
    "IsValid": true,
    "Warnings": null,
    "ExpiresIn": "/Date(1736267445426)/",
    "Token": "xxxx"
}
```

O *body* da *response* possui duas informações relevantes: o campo “*ExpiresIn*”, com a data/hora exatas em que o *token* irá expirar (esta informação também se encontra dentro do token, na *claim* “*exp*”), e o campo “*T oken*”, que é o token *JWT* propriamente dito.

AUTORIZAÇÃO NOS ENDPOINTS COM O TOKEN JWT

Uma vez de posse do *token*, todas as demais requisições aos *endpoints* devem fornecer o *header* adicional:

Authorization: Bearer <token_jwt>

...para que a requisição possa ser devidamente autorizada, até que, ao expirar a validade do token, com a **Layer** devolvendo um *HTTP Status Code* **401 – Unauthorized**, o fluxo deve ser reiniciado, realizando outra requisição para o *endpoint* de autenticação descrito na seção anterior para obtenção de um novo *token* para ser utilizado nas requisições subsequentes.

Segue abaixo um exemplo de requisição ao *endpoint* que retorna os dados de um cliente fornecendo seu ID de usuário:

POST: *https://**<tenant_name>**.layer.core.dcg.com.br/v1/Profile/API.svc/web/GetCustomer*

Headers:

```
User-Agent: LayerIntegration
Accept: application/json
Content-Type: application/json
Authorization: Bearer  xxx
```

Body (raw):

> *1 (ID do usuário)*

Response HTTP Status Code* 200 – OK com os dados do cliente*:

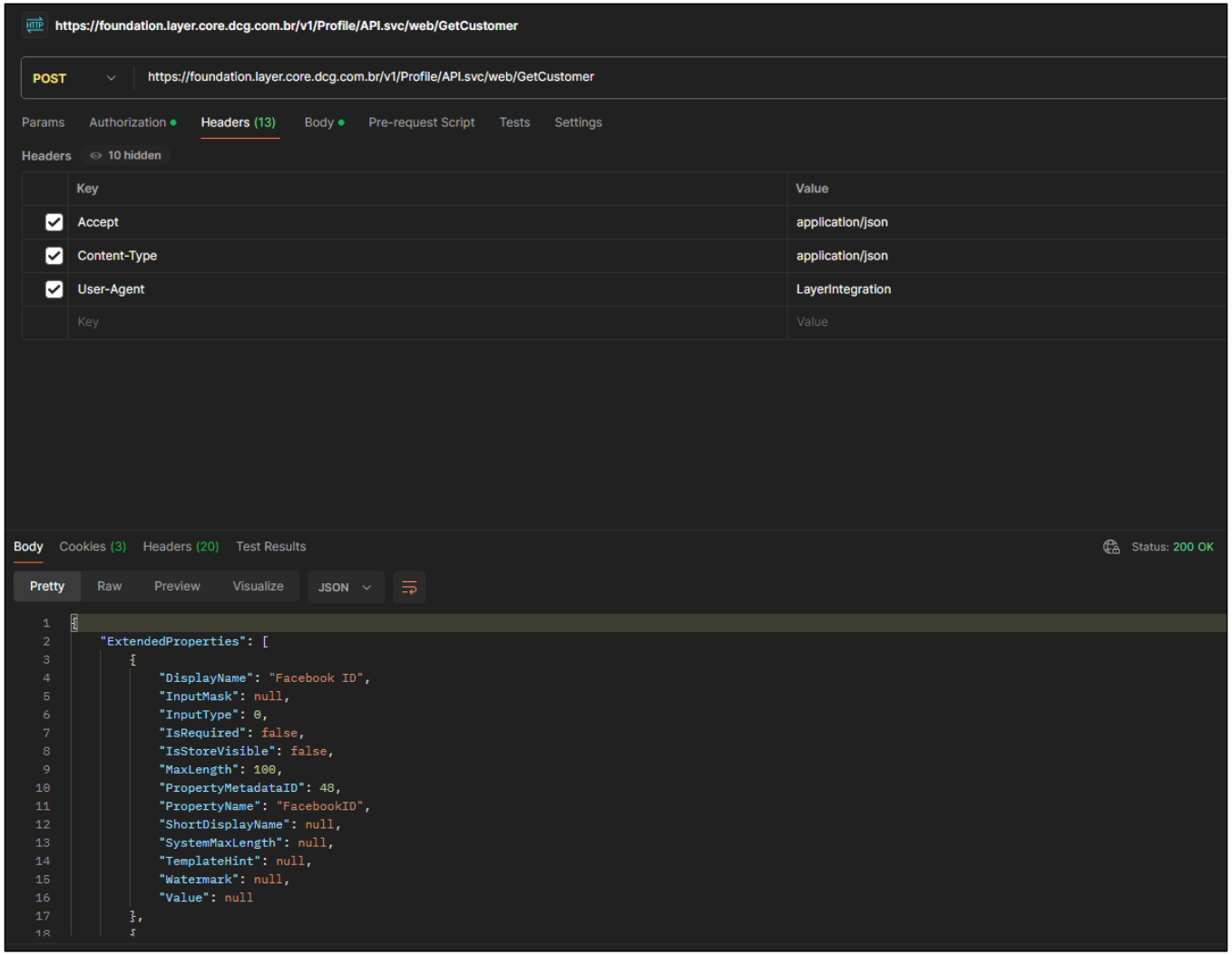
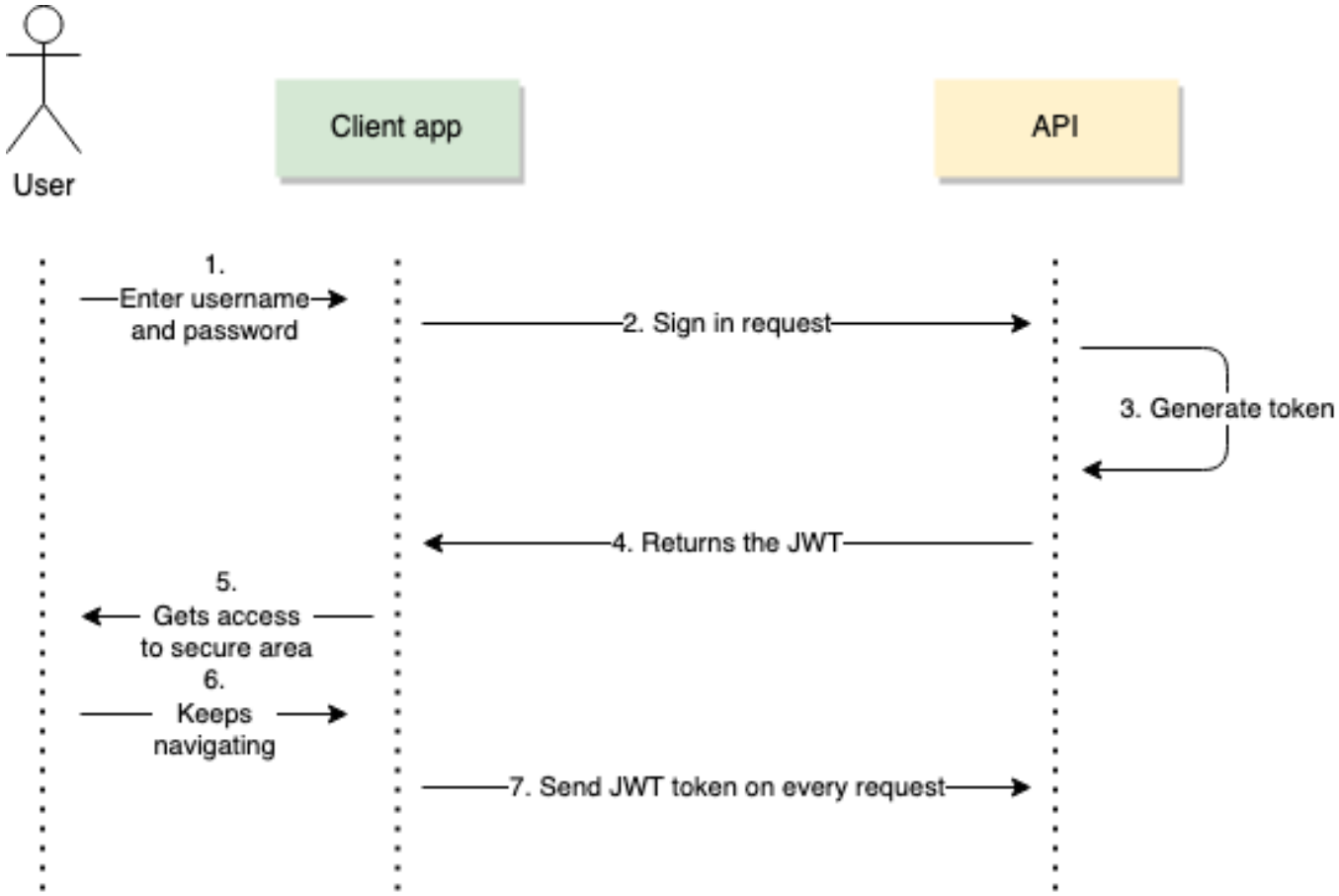
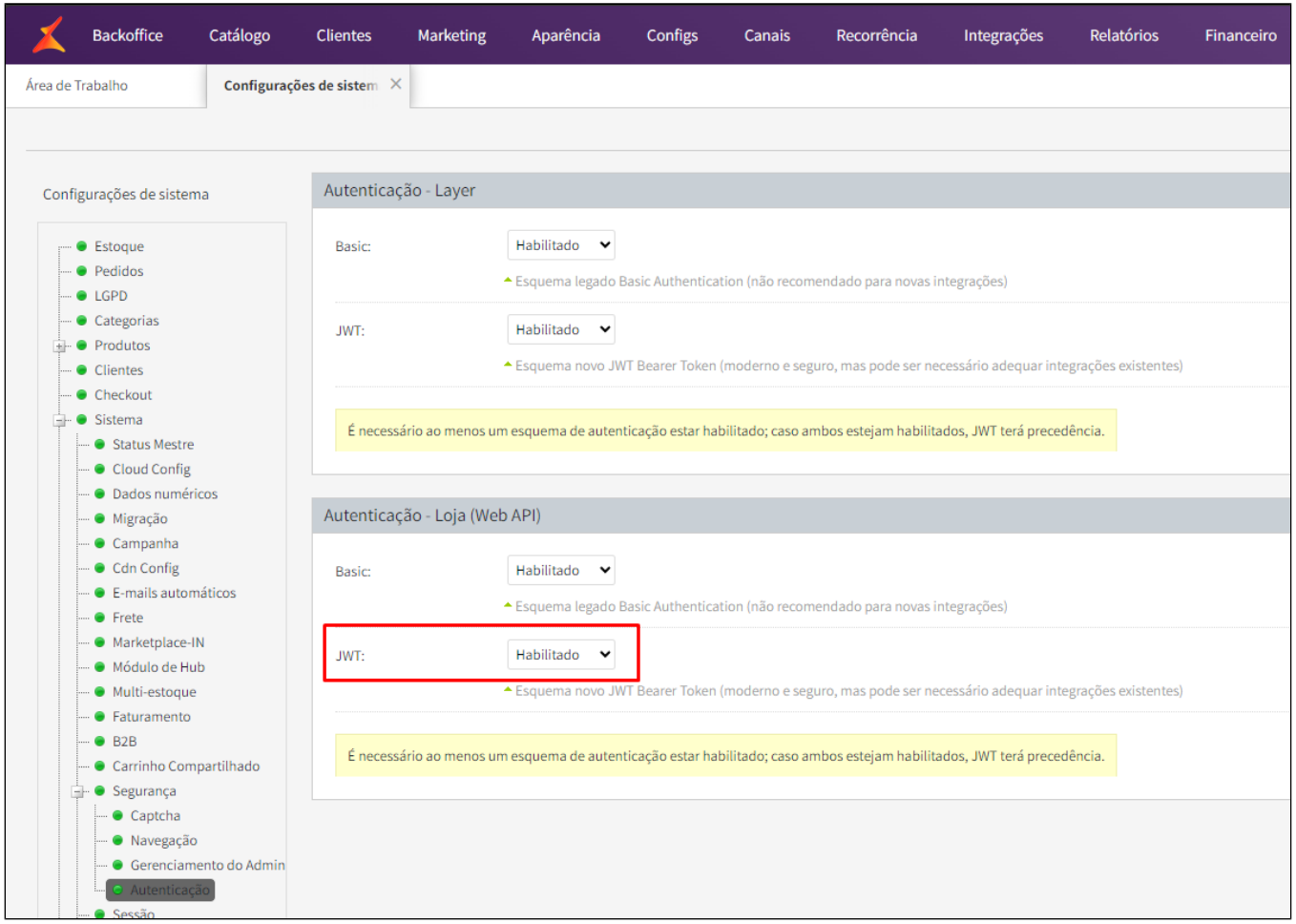


DIAGRAMA DO FLUXO



3. AUTENTICAÇÃO JWT – STORE FRONT (WEBAPI)

Para habilitar a autenticação/autorização *JWT* na Web-API da **Store**, navegue até a seguinte tela do **Admin** (*Configs/Configurações gerais/Sistema/Segurança/Autenticação*):



Na área "*Autenticação - Loja (Web API)*", selecione o valor "*Habilitado*" para o campo "*JWT*" e clique em "*Salvar*"; esta ação definirá a configuração "*Security.Authentication.EnableStoreWebApiJWTAuthentication*" com o valor "*True*"; aguarde até a atualização da configuração ser refletida na **Store** para que o sistema passe a utilizar este tipo de autorização.

***Informação importante:** nesta tela, é necessário ao menos um esquema de autenticação estar habilitado; caso ambos estejam habilitados, *JWT* terá precedência, e será o esquema *default*.

ACESSO VIA INTEGRAÇÃO - API

AUTENTICAÇÃO COM USUÁRIO E SENHA PARA OBTENÇÃO DO TOKEN JWT

Realizar uma requisição **POST** no *endpoint:*

https://<tenant_name>.core.dcg.com.br/web-api/v1/Security/Authentication/Authenticate

...ou:

https://<url_site_cliente>.com.br/web-api/v1/Security/Authentication/Authenticate

...fornecendo o seguinte *payload* no *body*:

> *{ "Username": "**<username>**", "Password": "**<password>**" }*

Se os dados de login estiverem incorretos, será devolvida uma *response* com o *HTTP Status Code* **401 – Unauthorized** e o seguinte *body*:

```
{
    "Token": null,
    "ExpiresIn": null
}
```

Caso as credenciais estejam corretas, será devolvida uma *response* com o *HTTP Status Code* **200 – OK** e um *body* com a estrutura parecida ao a seguir:

```
{
    "Token": "xxxx",
    "ExpiresIn": "2025-01-07T20:20:37.8517431Z"
}
```

O *body* da *response* possui duas informações relevantes: o campo “*ExpiresIn*”, com a data/hora exatas em que o *token* irá expirar (esta informação também se encontra dentro do token, na *claim* “*exp*”), e o campo “*T oken*”, que é o token *JWT* propriamente dito.

AUTORIZAÇÃO NOS ENDPOINTS COM O TOKEN JWT

Uma vez de posse do *token*, todas as demais requisições aos *endpoints* devem fornecer o *header* adicional:

Authorization: Bearer <token_jwt>

...para que a requisição possa ser devidamente autorizada, até que, ao expirar a validade do token, com a **Store** devolvendo um *HTTP Status Code* **401 – Unauthorized**, o fluxo deve ser reiniciado, realizando outra requisição para o *endpoint* de autenticação descrito na seção anterior para obtenção de um novo *token* para ser utilizado nas requisições subsequentes.

Segue abaixo um exemplo de requisição ao *endpoint* que retorna os dados de um cliente fornecendo seu documento (CPF ou CNPJ válido):

GET: *https://<tenant_name>.core.dcg.com.br/web-api/v1/Profile/Customer/GetCustomer/01234567890*

> **Headers**:
>
> *Authorization: Bearer Xxx*

Response HTTP Status Code* 200 – OK com os dados do cliente*:

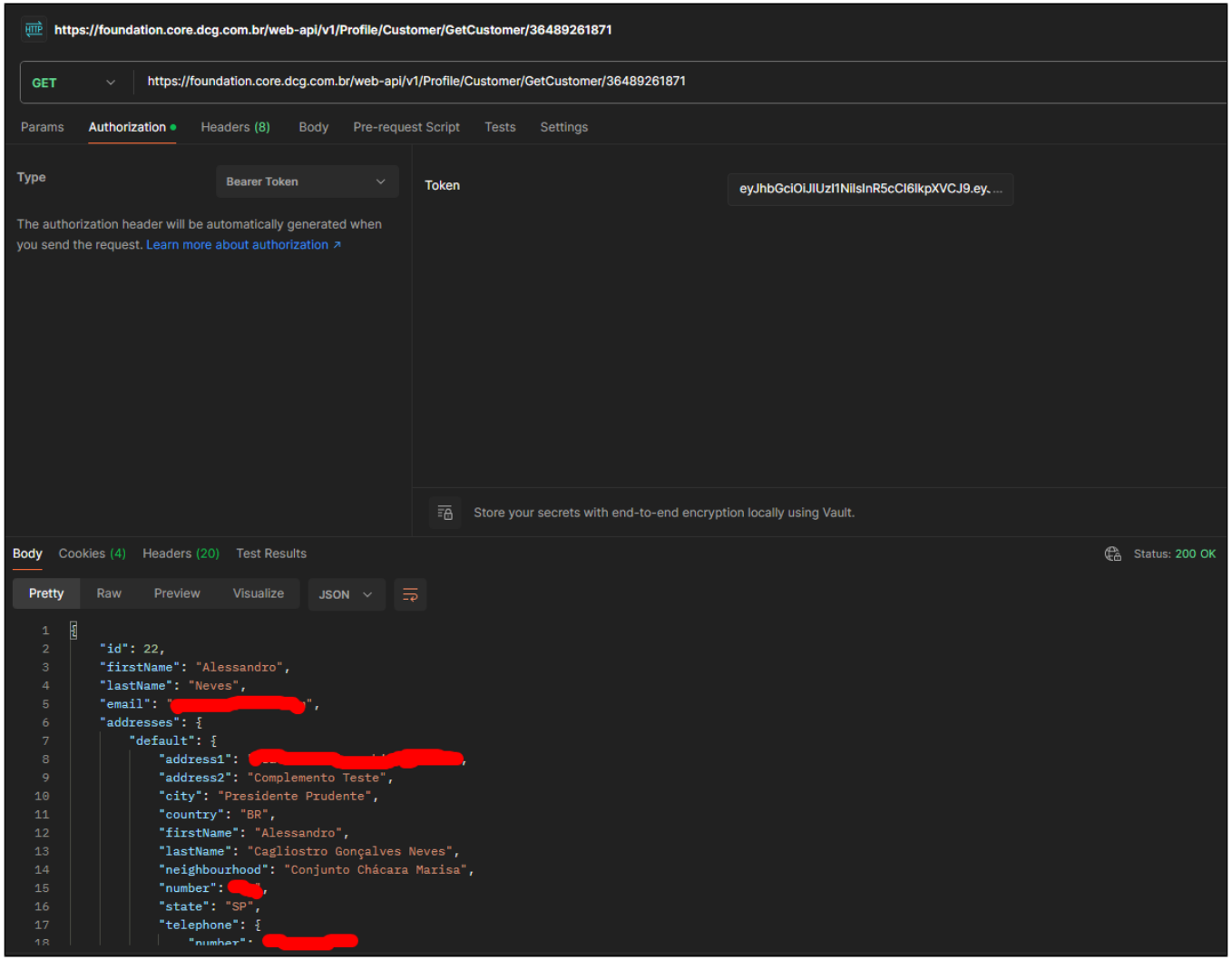
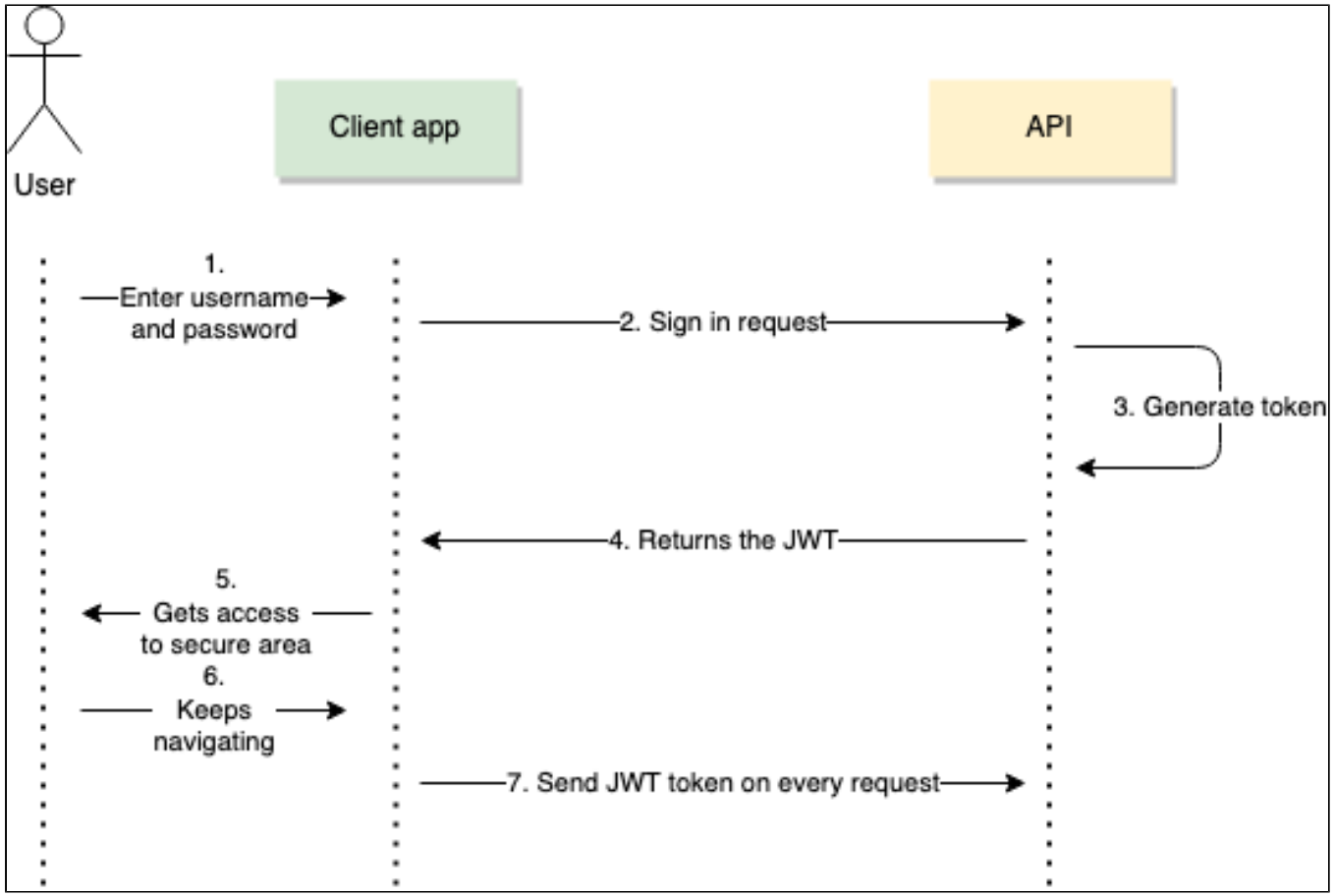


DIAGRAMA DO FLUXO



Em caso de dúvidas, acione o nosso suporte!