

<https://ploomesteste.azurewebsites.net>

Logando na API

POST

/api/v1/Auth/Login

⌵🔒

Parameters

Cancel

Reset

No parameters

Request body

application/json

```
{
  "login": "Ploomes",
  "password": "PloomesAPI"
}
```

Execute

Após o sucesso no Login devemos copiar o `accessToken` gerado na response e cola-lo no botão do `authorize` como: `Bearer {accessToken}`

Autorizando com o accessToken

200

Response body

```
{
  "authenticated": true,
  "created": "2023-04-25 12:22:47",
  "expiration": "2023-04-25 13:22:47",
  "accessToken": "eyJ0eXN0IHR5cGU6ImN5cy1yYXV0eG8iLCJ1ZD06N2Y3YU40MzQ6bWVjIj50bW10T1tsNm9u3JUZ2Y3S1tsbmVwXzI3Zm9uY2V1IiwiaWF0Ijoi2023-04-25T12:22:47.4456608-03:00",
  "refreshToken": "VglsqABg9fw7S1ABHOURCv5d0H14U8irpawWm0Iedw",
  "id": "00000000-0000-0000-0000-000000000000",
  "codeGetExternal": null,
  "inclusao": "2023-04-25T12:22:47.4456608-03:00",
  "alteracao": null,
  "excluido": false
}
```

Response headers

Available authorizations

Bearer (apiKey)

JWT Authorization header using the Bearer scheme. \n\n Enter 'Bearer' [space] and then your token in the text input below. \n\nExample: 'Bearer 12345abcdef'

Name: Authorization

In: header

Value:

Bearer eyJhbGciOiJIUzI1NiIs

Authorize

Close

Com isso as funções do Cliente e de File estarão habilitadas

Cliente:

GetAll -> Retorna todos clientes

GET

/api/v1/Cliente/GetAll

GetClienteById -> Passamos o Id do cliente que pegamos no GetAll por exemplo e colamos nessa função para retornar o cliente com aquele Id

GET

/api/v1/Cliente/GetClienteById/{Id}

Parameters

Name	Description
Id * required	
string(\$uuid)	Id
(path)	

GetClienteByNome -> Passamos o nome do cliente e se encontrarmos um cliente com aquele nome, retornamos ele

GET	
/api/v1/Cliente/GetClienteByNome/{Nome}	
Parameters	
Name	Description
Nome * required string (path)	<input type="text" value="Nome"/>

Insert -> Preenchemos os campos do json e então ele gera um cliente para nós com Id e Data de inclusão já criados automaticamente

POST	
/api/v1/Cliente/Insert	
Parameters	
No parameters	
Request body	
Example Value Schema	
<pre>{ "nome": "string", "cpf": "string", "nascimento": "string", "endereco": "string", "cidade": "string", "estado": "string", "email": "string" }</pre>	

Update-> Passamos o Id de um cliente já existente e preenchemos o json, se os dados forem diferentes nós atualizamos o cliente com aquele Id

PUT **/api/v1/Cliente/Update**

Parameters

Name	Description
Id string(\$uuid) (header)	<input type="text" value="Id"/>

Request body

Example Value | Schema

```
{
  "nome": "string",
  "cpf": "string",
  "nascimento": "string",
  "endereco": "string",
  "cidade": "string",
  "estado": "string",
  "email": "string"
}
```

Delete -> O cliente com o Id passado é deletado do banco de dados

DELETE **/api/v1/Cliente/Delete/{Id}**

Parameters

Name	Description
Id * required string(\$uuid) (path)	<input type="text" value="Id"/>

File:

UploadArquivo -> Nós escolhemos um arquivo para dar upload, no caso é uma implementação simples onde o arquivo fica apenas salvo no código.

Os arquivos permitidos são: jpg, jpeg, pdf e png

POST `/api/v1/File/UploadArquivo`

Parameters

No parameters

Request body

file
string(\$binary)

Escolher Arquivo

Nenhum arquivo escolhido

☒ Send empty value

UploadArquivosMultiplos -> Semelhante ao UploadArquivo, no entanto podemos fazer uma seleção múltipla de arquivos para incluir no sistema

POST `/api/v1/File/UploadArquivosMultiplos`

Parameters

No parameters

Request body

files array

Escolher Arquivo

Nenhum arquivo escolhido

-

Escolher Arquivo

Nenhum arquivo escolhido

-

Escolher Arquivo

Nenhum arquivo escolhido

-

Add string item

☐ Send empty value

downloadArquivo -> Passamos o nome do arquivo que queremos pegar do sistema e ele retorna um conjunto de bytes que ao salvarmos no formato correto ele gera a imagem. Por exemplo se baixarmos a imagem.jpg, salvamos a resposta como response.jpg e a imagem será gerada.

GET		/api/v1/File/downloadArquivo/{Nome}	
Parameters			
Name	Description		
Nome * required			
string			
(path)			
		Nome	

Caso decida usar o PostMan para consultas segue o link para import:
<https://ploomesteste.azurewebsites.net/swagger/v1/swagger.json>