Especificação de API

Portal EC

TecBan	Especificação de API
Portal do EC	Áreas Responsáveis: **Demandas e Soluções Corporativas**
Data: **19/07/2022**	Class. Informação: **Uso Interno e Externo**
Versão do Documento: **2.2**	

ÍNDICE

Importante: As informações deste documento são confidenciais, devendo circular somente entre os responsáveis pelo processo.

1 IDENTIFICAÇÃO DO DOCUMENTO

TABELA – IDENTIFICAÇÃO

NOME DO DOCUMENTO:	**Especificação de API – Portal EC**
BJETIVO:	Esta especificação permite que os Estabelecimentos Comerciais implementem em seus sistemas a integração para consulta aos extratos de depósitos e saques feitos nos equipamentos da Rede Banco24Horas®®.

2 DEFINIÇÕES E ABREVIATURAS

TABELA – TERMOS

TERMO	DESCRIÇÃO
API	Application Programming Interface: conjunto de definições de métodos e funções que fornece uma biblioteca para ser utilizada por outros programas (no caso desse projeto, para páginas Web). De modo geral, a API é composta por uma série de funções acessíveis somente por programação, e que permitem utilizar características do software menos evidentes ao usuário tradicional.
ATM	Automated Teller Machine (Terminal de Autoatendimento)
Banco24Horas®®	O Banco24Horas®® é o principal canal de autoatendimento externo em locais de acesso público. Mais de 40 Bancos disponibilizam operações financeiras a seus clientes através do Banco24Horas®®, levando serviços para os grandes centros, periferias e municípios de menor porte.

TERMO	DESCRIÇÃO
Certificado Digital	Arquivo eletrônico que serve como identidade virtual para uma pessoa física ou jurídica, e por ele pode se fazer transações online com garantia de autenticidade e com toda proteção das informações trocadas.
Sandbox	Ambiente para testes da aplicação.
Swagger	Traduzido do inglês – O Swagger é um framework open source que facilita os desenvolvedores desenhar, especificar e documentar suas APIs. Ele segue a iniciativa Open API que busca a padronização de APIs REST. Esse termo, descreve os recursos de suas APIs, como endpoints, parâmetros de entrada, objetos de retorno, códigos HTTP, métodos de autenticação, entre outros.
UUID	(Universally Unique Identifier)

3 VISÃO GERAL

O Portal do EC é disponibilizado para os Estabelecimentos Comerciais consultarem transações realizadas nos equipamentos do Banco24Horas instalados em uma ou mais lojas.

4 IMPLEMENTAÇÃO DA API

Para consultar as transações deverá ser cadastrado um usuário com Perfil API no Portal do EC pelo Administrador Cliente do EC. Para isso, será necessário gerar um certificado digital para viabilizar o cadastro do usuário e obter a identificação do usuário que serão utilizados no processo de segurança das requisições.

Obs.: Para mais detalhes sobre o processo de segurança, verifique a seção "5 Segurança" deste documento.

4.1 Integração

A comunicação entre o EC e a API do Portal EC é feita através de chamada de URL.

- **Consulta Transações de Depósitos realizados em PCs do Banco24Horas pelos Estabelecimentos Comerciais conveniados.**

Para consultar as transações de depósitos, o EC deverá fazer uma requisição ao Rest Service da TecBan através da **URL de Consulta de Transações de Depósito.**

Essa consulta disponibiliza as transações de depósito concluídas com sucesso e também as transações de depósito regularizadas posterioremente, em casos que ocorreu alguma falha no equipamento durante o depósito. É possível distinguir ambos os registros pela Situação:

- ✓ **Concluída – transação de depósito concluída com sucesso.**
- ✓ **Regularizada – transação de depósito regularizada posteriormente. A data da regularização indicadará a data que a mesma foi creditada ao EC.**

- **Consulta Transações de Saques realizados em PCs do Banco24Horas pelos Estabelecimentos Comerciais conveniados.**

Para consultar os saques, o EC deverá fazer uma requisição ao Rest Service da TecBan através da **URL de Consulta de Saque.**

Essa consulta disponibiliza todos as transações de saques concluídas com sucesso em equipamentos Totems e ATMO, e a origem pode ser distinguida pelo Código de Solicitação:

- ✓ **452 – ATMO**
- ✓ **458 - TOTEM**

Obs.: Para mais detalhes sobre as URLs citadas, verifique a seção "6 Conectividade" deste documento.

4.2 Swagger

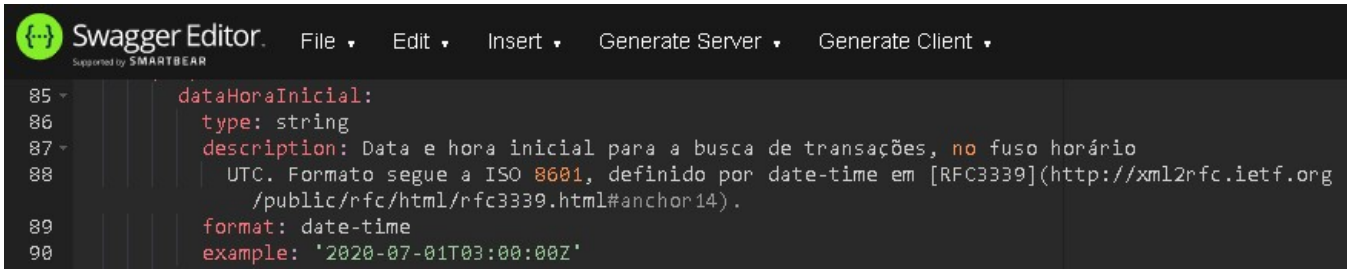
O layout das mensagens e o detalhamento dos campos podem ser consultados através no link: **https://editor.swagger.io/**

Para visualizar os campos, selecione “File > Import File” e importe o arquivo .yml disponibilizado juntamente com esta Especificação Técnica Funcional.

Campos “date-time”

Os swaggers da TecBan utilizam o padrão **GMT + 0** nos campos com o formato “date-time”. Dessa forma, ao consumir as APIs da TecBan, o EC deverá utilizar esse esquema definido no swagger.

Exemplo:



4.2.1 API de integração para os Estabelecimentos Comerciais

Swagger: banco24horasvarejo-partner-openapi_v2.2.0.yaml

Esse Swagger contempla a API que será disponibilizada pela TecBan para que o EC possa realizar a consulta de saques e transações de depósitos.

- **Consulta de transações de depósito**

  O EC deve usar o método POST para enviar a requisição de consulta ao Host TecBan. No método POST, deverão ser informados os dados abaixo:

POST /transacao/consulta-depositos

- **Consulta de transações de saque**

  O EC deve usar o método POST para enviar a requisição de consulta ao Host TecBan. No método POST, deverão ser informados os dados abaixo:

POST /transacao/consulta-saques

4.3 Retornos e Respostas

4.3.1 HTTPS

Ao receber as requisições, a plataforma devolve retornos imediatos referentes à interpretação da requisição. Esses códigos indicam se uma requisitação foi corretamente concluída ou se houve algum problema de comunicação entre as APIs. Seguem alguns exemplos:

TABELA – CÓDIGOS DE RETORNO HTTPS

RETORNO	DESCRIÇÃO
200	Requisição processada com sucesso
400	Bad Request: A requisição possui parâmetro(s) inválido(s)
403	Unauthorized: Identificação ou assinatura inválidas
502	Bad Gateway: O servidor atuava como um gateway ou proxy e recebeu uma resposta inválida do servidor a montante
504	Timeout da operação

A lista com todos os códigos de resposta HTTPS podem ser consultados no site da W3C: https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html

5 SEGURANÇA

A segurança é um item primordial para garantir a viabilidade da solução. A partir disso, os requisitos abaixo devem tratar as possíveis ameaças relacionadas à:

1. **Especificação de utilização de comunicação cifrada via HTTPs TLS1.2 com cifras fortes** (TLS_DHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 ou TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ou superiores).

2. **Acesso não autorizado –** a segurança deve garantir que o acesso só seja concedido se o usuário (EC) possuir credenciais válidas e permissão de acordo com as politicas de segurança definidas pela TecBan.

3. **Garantia de origem –** as aplicações deverão garantir a origem através de validação de assinatura digital.

5.1 Certificado Digital

O Certificado Digital é utilizado para autenticar o EC na TecBan. Essa autenticação garante que as mensagens estão sendo trocadas entre os hosts corretos, evitando que um host não autorizado tenha acesso às transações.

O EC deve gerar dois certificados, sendo um para o ambiente de homologação e outro para o ambiente de produção. O certificado deve ser compartilhado com a TecBan da seguinte forma: O certificado de homologação pode ser enviado para a equipe do projeto, e o certificado de produção deve ser enviado ao Administrador Cliente do estabelecimento comercial, que fara o upload na tela de configuração de usuário com perfil API.

Nossa recomendação para geração do certificado digital, é utilizar certificados com data de validade máxima de 1 ano e padrão RSA 2048. A renovação do certificado é de responsabilidade do EC e deve ser encaminhado ao Administrador Cliente do estabelecimento comercial para atualização no sistema Portal EC. Requisições assinadas com certificados vencidos, serão recusadas pela TecBan.

A criação do certificado é de responsabilidade do Estabelecimento Comercial e podem optar por um dos cenários a seguir:

- ✓ EC pode gerar um certificado autoassinado usando um aplicativo chamado openssl (não tem custo); ou
- ✓ EC pode adquirir um certificado do tipo A1 junto a uma empresa e usar.

Para gerar a chave assimétrica e certificado digital, utilizar o comando abaixo como exemplo:

```
openssl req -nodes -x509 -newkey rsa:2048 -out www_cliente_com_br.cer -keyout
www_cliente_com_br.key -subj "/C=BR/ST=SP/L=Sao
Paulo/O=Cliente/CN=www.cliente.com.br" -days 365
```

5.2 Cadastro de Usuário API

Após a geração do certificado digital deverá ser solicitado que Administrador Cliente do estabelecimento comercial realize o cadastro do Usuário com Perfil API, seguindo os passos à seguir:

- ✓ Acessar a funcionalidade “Cadastro Usuário API”;
  - Acessar a opção “+ Novo Usuário API”:
    - Realizar o upload do certificado gerado; e
    - Definir o nível de acesso aos dados que o usuário API terá;
    - Clicar em “Cadastrar” para finalizar o cadastro do Usuário API.

5.3 Assinatura Digital (Token JWT)

Segurança é fundamental para integração com nossas APIs, utilizamos elementos de segurança como autenticação e autorização.

Para autorizar a requisição, é necessário que o EC gere um token no padrão JWT (JSON Web Tokens).

O Token JWT (JSON Web Tokens) deverá ser gerado e enviado no header da mensagem HTTP como “**Authorization**”.

O JWT é composto por três blocos: Header, Payload e Assinatura.

- O primeiro bloco contém a mensagem Header no formato Json, convertido para Encode Base64
- O segundo bloco é composto pelo Payload no formato Json, convertido para Encode Base64
- O terceiro bloco é composto pela Assinatura do Header e do Payload.

Exemplo de Token JWT enviado no Header HTTP:

Authorization:Bearer

eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsInVzZXJJZCI6IjdiOTc2OWRiLWI3MDEtNDkxMS1hMGM5LTFlYTJiMzBmZTdkMiJ9.eyJkYXRhSG9yYUluaWNpYWwiOiIyMDIwLTA3LTAxVDAzOjAwOjAwWiIsImRhdGFIb3JhRmluYWwiOiIyMDIwLTA3LTAyVDE2OjM5OjU3WiIsImNucGpSYWl6TGlzdCI6WyIxMjM0NTY3OCIsIjAwNTQzMjEwIl0sImxvamFMaXN0IjpbMTIzNDU2Nzg5MDEsMV0sImF0bUxpc3QiOlsxMDk4NzY1NDMyMSwyXSwiYmFuY29MaXN0IjpbMTIzLDddfQ.aI-RxXpQ_fuBChDpIzQJo9sPzqa5gaGWi5PVIfp78hwm5LZUIZS4tGVbzXIedAIdU9YAvxeROs648l1MQPPZw-3n50xjAYQQmOjeG1ojxqII-FIsY-IS_f8wHcRbUp8QRXsrlaiFVOOPF47XXH0v2Yg1LO7YxM7K3i2PkwDbJokyz3QOFzTlO6f3KTrmqNPg0YGQMU0lcfT6yJ2f5jutf17j6zAXT-2kTQxNoHyDJRAJPMX4orbSNGtdG9IhhyVGyPhSaZLsiBZi_IIFuLsiatuVUq_O8tap74htT7BL62oBt18CLPgUVnwMAuMpy7Hsx95LPZLmdokp8R8SLa16dg

- **Passo a passo para geração do Token JWT**

Primeiro Bloco: Header JWT

O header é composto pelo objeto JSON que define informações sobre o tipo do token (**typ**), o algoritmo de criptografia usado em sua assinatura (**alg**) e a identificação do usuário de perfil API (**userId**).

Convertido no formato Encode Base64.

Campos do JSON	Definição	Valor
alg	Algoritmo de criptografia usado na assinatura	**RS256**
typ	Tipo do Token	JWT
userId	Identificação do usuário de perfil API no formato UUID (gerado pelo Portal do EC)	7b9769db-b701-4911-a0c9-1ea2b30fe7d2

ATENÇÃO: Utilizar o algoritmo 'RS256'.

Exemplo do objeto Header no formato JSON e convertido em Base64:

Header JSON	ENCODE Base64
{ "alg": "RS256", "typ": "JWT", "userId": "7b9769db-b701-4911-a0c9-1ea2b30fe7d2" }	eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsInVzZXJJZCI6IjdiOTc2OWRiLWI3MDEtNDkxMS1hMGM5LTFlYTJiMzBmZTdkMiJ9

Segundo Bloco: Payload JWT

O Payload JWT é a cópia da mensagem no formato JSON. Convertido no formato Encode Base64.

Exemplo do objeto Payload no formato JSON e convertido em Base64:

Payload JSON	ENCODE para Base64

{
"dataHoraInicial": "2020-07-01T03:00:00Z",
"dataHoraFinal": "2020-07-02T16:39:57Z",
"cnpjRaizList": [
"12345678",
"00543210"
],
"lojaList": [
12345678901,
1
],
"atmList": [
10987654321,
2
],
"bancoList": [
123,
7
]
}
	eyJkYXRhSG9yYUluaWNpYWwiOiIyMDIwLTA3LTAxVDAzOjAwOjAwWiIsImRhdGFIb3JhRmluYWwiOiIyMDIwLTA3LTAyVDE2OjM5OjU3WiIsImNucGpSYWl6TGlzdCI6WyIxMjM0NTY3OCIsIjAwNTQzMjEwIl0sImxvamFMaXN0IjpbMTIzNDU2Nzg5MDEsMV0sImF0bUxpc3QiOlsxMDk4NzY1NDMyMSwyXSwiYmFuY29MaXN0IjpbMTIzLDddfQ

Terceiro Bloco: Assinatura

Assinatura do header e do payload, usando a chave privada do EC.

Exemplo de assinatura:

al-RxXpQ_fuBChDpIzQJo9sPzqa5gaGWi5PVIfp78hwm5LZUIZS4tGVbzXIedAIdU9YAvxeROs648l1MQPPZw-3n50xjAYQQmOjeG1ojxqlI-FIsY-IS_f8wHcRbUp8QRXsrlaiFVOOPF47XXH0v2Yg1LO7YxM7K3i2PkwDbJokyz3QOFzTlO6f3KTrmqNPg0YGQMU0lcfT6yJ2f5jutf17j6zAXT-2kTQxNoHyDJRAJPMX4orbSNGtdG9IhhyVGyPhSaZLsiBZi_IIFuLsiatuVUq_O8tap74htT7BL62oBt18CLPgUVnwMAuMpy7Hsx95LPZLmdokp8R8SLa16dg

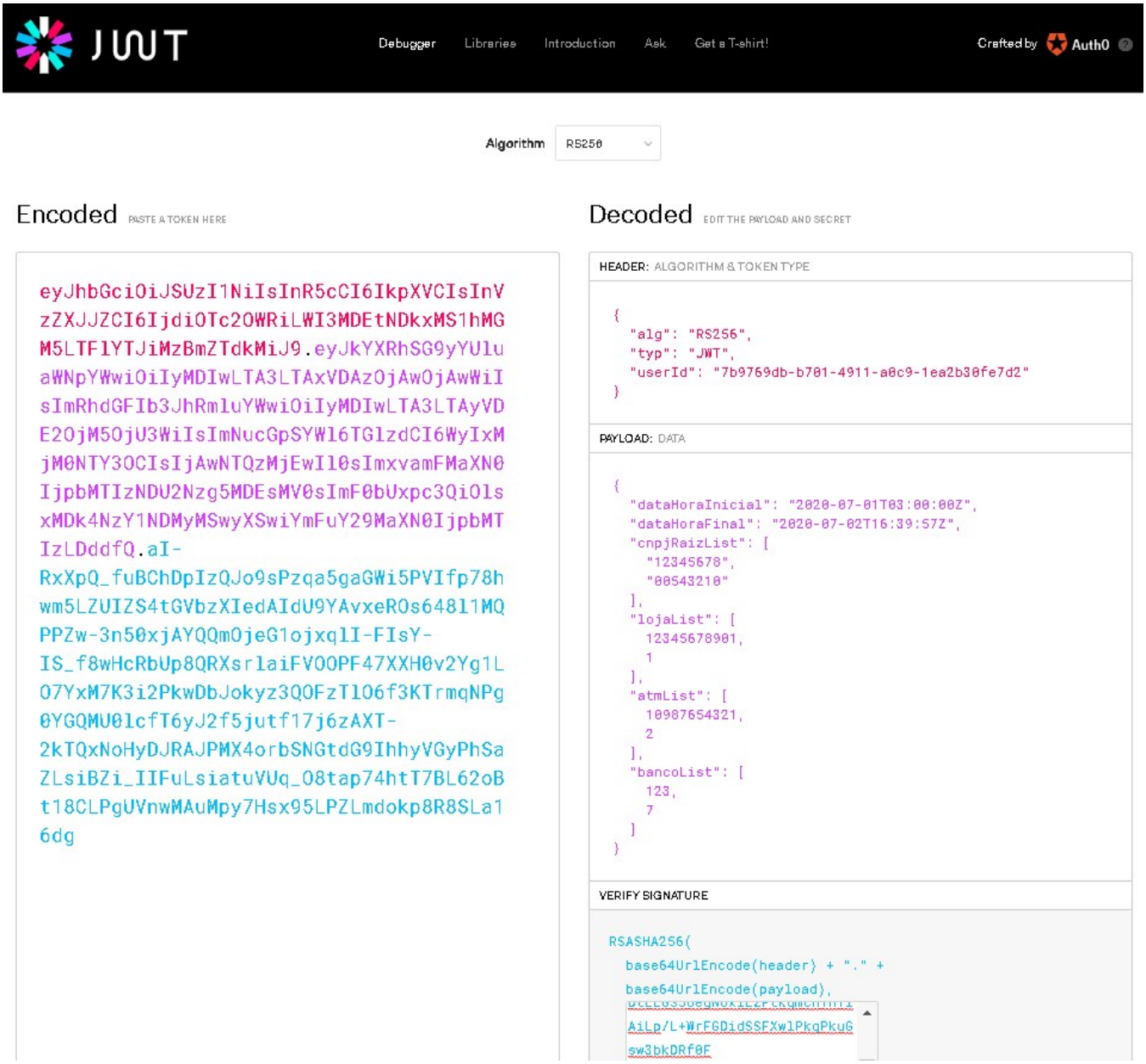
- **Formato do token JWT:**

O Token JWT é formado pelos três blocos (header no formato base64, Payload no formato base64 e Assinatura). Seguindo essa respectiva ordem, separando os blocos por ponto ".", conforme o exemplo a seguir.

Exemplo de Token JWT:

TOKEN JWT	INFORMAÇÕES DO FORMATO
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsInVzZXJJZCI6IjdiOTc2OWRiLWI3MDEtNDkxMS1hMGM5LTFlYTJiMzBmZTdkMiJ9.eyJkYXRhSG9yYUluaWNpYWwiOiIyMDIwLTA3LTAxVDAzOjAwOjAwWiIsImRhdGFIb3JhRmluYWwiOiIyMDIwLTA3LTAyVDE2OjM5OjU3WiIsImNucGpSYWI6TGlzdCI6WyIxMjM0NTY3OCIsIjAwNTQzMjEwIl0sImxvamFMaXN0IjpbMTIzNDU2Nzg5MDEsMV0sImF0bUxpc3QiOlsxMDk4NzY1NDMyMSwyXSwiYmFuY29MaXN0IjpbMTIzLDddfQ.aI-RxXpQ_fuBChDpIzQJo9sPzqa5gaGWi5PVIfp78hwm5LZUIZS4tGVbzXIedAIdU9YAvxeROs648I1MQPPZw-3n50xjAYQQmOjeG1ojxqII-FIsY-IS_f8wHcRbUp8QRXsrlaiFVOOPF47XXH0v2Yg1LO7YxM7K3i2PkwDbJokyz3QOFzTlO6f3KTrmqNPg0YGQMU0lcfT6yJ2f5jutf17j6zAXT-2kTQxNoHyDJRAJPMX4orbSNGtdG9IhhyVGyPhSaZLsiBZi_IIFuLsiatuVUq_O8tap74htT7BL62oBt18CLPgUVnwMAuMpy7Hsx95LPZLmdokp8R8SLa16dg	Formato BASE64_ENCODED(**Header**). Formato BASE64_ENCODED(**Payload**). ChavePrivada_EC(**Assinatura**)

DICA: No site JWT **https://jwt.io/** é possível fazer o debugger e gerar o token JWT para teste e validação do formato. Conforme exemplo a seguir:



6 CONECTIVIDADE

6.1 Comunicação via internet

TABELA – URLS PARA INTEGRAÇÃO VIA INTERNET

HOMOLOGAÇÃO	
CONSULTA DE TRANSAÇÕES	https://api.homol.partner.banco24horasvarejo.com.br/v2/transacao/consulta-depositos https://api.homol.partner.banco24horasvarejo.com.br/v2/transacao/consulta-saques
PRODUÇÃO	
CONSULTA DE TRANSAÇÕES	https://api.partner.banco24horasvarejo.com.br/v2/transacao/consulta-depositos https://api.partner.banco24horasvarejo.com.br/v2/transacao/consulta-saques

6.2 Ambiente Sandbox

Para auxiliar nos fluxos iniciais de testes, disponibilizamos um ambiente de Sandbox para que os Ecs façam seus testes individuais.

Este ambiente é completamente separado do ambiente de Produção e nenhum dado é compartilhado entre os ambientes.

O ambiente Sandbox permite exercitar as chamadas sem as validações de segurança como certificado e assinatura, sendo um contato inicial com as APIs. Para acessa-lo é necessário fazer a requisição na URL abaixo:

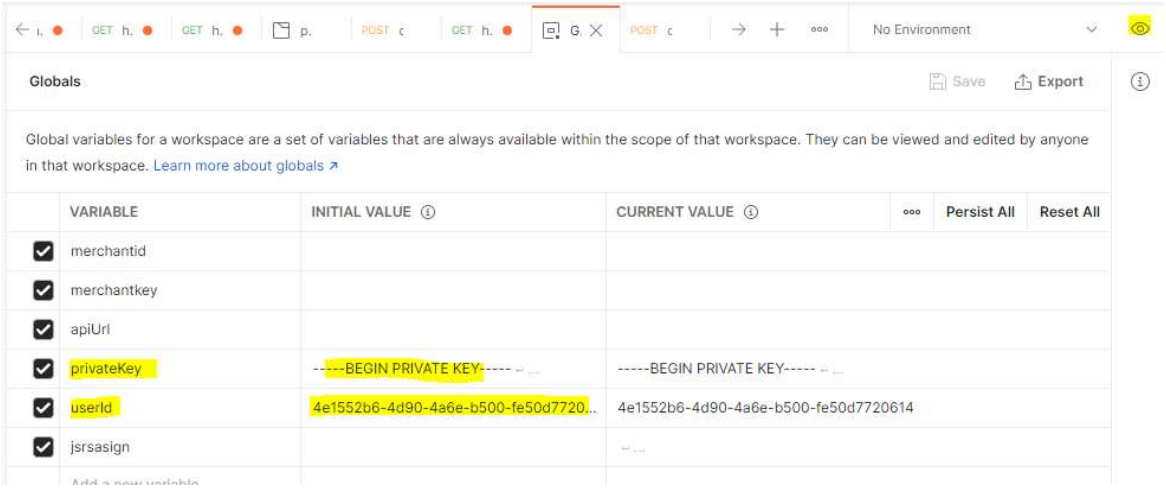
https://api.sandbox.partner.banco24horasvarejo.com.br/v2/transacao/consulta-depositos

https://api.sandbox.partner.banco24horasvarejo.com.br/v2/transacao/consulta-saques

6.3 Ferramenta Postman

Para realizar os testes de acesso a API, de forma manual antes de ter uma implementação finalizada, pode ser utilizada a ferramenta Postman. Após a instalação, importar a collection (portalec-partner.postman_collection.json) em anexo.

Configurar as variáveis globais privateKey e userId, conforme abaixo:



Obs.: A varivável privateKey deve ser preenchida com o conteúdo do arquivo da chave privada e a variável userId com o código do usuário de perfil API que foi criado no Portal do EC.

7 DOCUMENTOS DE APOIO

TABELA – DOCUMENTOS DE APOIO

DOCUMENTO
banco24horasvarejo-partner-openapi_v2.2.0.yaml
portalec-partner.postman_collection.json


TecBan