

API Crédito com Garantia em Saque Aniversário do FGTS

Essa API tem como objetivo possibilitar a criação de operações de crédito com garantia em Saque Aniversário do FGTS, e compreende as operações de simulação e criação de proposta.

URL base

JavaScript

```
baseUrl: "https://appvctex.com.br/api/"
```

Como testar

- Postman

Para requisições não autenticadas (ex.: login):

1. Defina o “método HTTP” a ser usado;
2. Insira a “URL” (ex.: <https://www.appvctex.com.br/api/authentication/login>);
3. Na guia “Body”, insira o corpo da requisição, se houver:
 - a. Selecione a opção “raw” e a linguagem “JSON”.
4. Na guia “Params”, insira os parâmetros da requisição, se houver;
5. Clique em “Send”.

Para requisições autenticadas (ex.: simulação):

1. Defina o “método HTTP” a ser usado;
2. Insira a “URL” (ex.: <https://www.appvctex.com.br/api/service/simulation>);
3. Na guia “Authorization”:
 - a. Selecione o “type” “Bearer Token”;
 - b. Insira o “access token” retornado pelo login no campo “token”.
4. Na guia “Body”, insira o corpo da requisição, se houver:
 - a. Selecione a opção “raw” e a linguagem “JSON”.
5. Na guia “Params”, insira os parâmetros da requisição, se houver;
6. Clique em “Send”.

- cURL

Para requisições não autenticadas (ex.: login):

```
curl -X POST -H "Accept: application/json" -d '{  
  "cpf": "12345678900",  
  "password": "12345678"  
}' https://www.appvctex.com.br/api/authentication/login
```

Para requisições autenticadas (ex.: simulação):

```
curl -X [VERBO HTTP]  
  -H "Accept: application/json"  
  -H "Authorization: Bearer {token}"  
-d '{corpo da requisição}' https://www.appvctex.com.br/api/service/simulation
```

Roteiro Operacional

Para realizar as operações de crédito com garantia no Saque Aniversário do FGTS é necessário realizar o seguinte passo-a-passo:

1. Solicitar ao nosso time comercial/produto a criação do usuário de API;
2. Fornecer os IPs que deverão ser liberados para acessar as APIs;
3. Concluídas essas etapas, deve ser efetuar o login por meio do endpoint de login, para receber o Access Token, cuja validade é de 120 minutos;
4. Recebido o Access Token, as requisições seguintes devem conter o cabeçalho Authorization, cujo valor será o Access Token no formato de um Bearer Token;
5. Deve-se, então, realizar as simulações com o CPF do cliente no endpoint de simulação;
6. Escolhida uma simulação, deve ser criada a operação no endpoint de criação da operação;
7. Feito isso, a operação estará criada e pronta para ser acompanhada em nossa plataforma;
8. Para logar em nossa plataforma, basta inserir as mesmas credenciais usadas para logar com a API.

Autenticação

Todas as requisições para a API devem ser autenticadas. O processo se dá por meio de um login, que fornece um **JWT**. Esse, por sua vez, deve ser adicionado ao cabeçalho de todas as requisições como um **Bearer Token**, no **header Authorization**.

O token terá a validade de 120 minutos e, após expirado, será necessário fazer login novamente.

Endpoint de Login

- Requisição

Endereço da requisição: `{baseUrl}/authentication/login`

Método: **POST**

Body:

JavaScript

```
{
  cpf: "string", (cpf sem máscara)
  password: "string" (mínimo 8 caracteres)
}
```

- Resposta (sucesso)

StatusCode: 200

Body:

JavaScript

```
{
  statusCode: 200,
  message: "Login realizado com sucesso."
  token: {
    access-token: "string", (Token JWT)
    expires: "number", (vida útil do token em segundos)
    type: "string"
  }
  data: {
    id: "string",
    name: "string",
    role: "string", (nível hierárquico do usuário)
    permissions: ["string"], (lista de permissões do usuário)
    organization: {
      id: "string",
      fantasyName: "string", (nome fantasia da empresa do usuário)
      organizationType: "string" (tipo da empresa do usuário)
    }
  }
}
```

```
}
```

- Resposta (falha)

StatusCode: 400

Body:

JavaScript

```
{
  message: "{campo} é obrigatório"
  statusCode: 400
}
```

JavaScript

```
{
  message: "Credenciais inválidas."
  statusCode: 400
}
```

StatusCode: 404

Body:

JavaScript

```
{
  message: "Usuário não encontrado."
  statusCode: 404
}
```

Todos os demais endpoints podem retornar as seguintes mensagens em casos de problemas de autenticação ou autorização:

Obs.: Todos os erros de `statusCode` 401 dizem respeito a problemas com o `access token`. Para esses casos é indicado realizar um novo login.

StatusCode: 401

Body:

JavaScript

```
{  
  message: "Access Token inválido ou expirado."  
  statusCode: 401  
}
```

JavaScript

```
{  
  message: 'Campo "user-id" é obrigatório no header da requisição.'  
  statusCode: 401  
}
```

JavaScript

```
{  
  message: 'O campo "user-id" é obrigatório.'  
  statusCode: 401  
}
```

StatusCode: 403

Body:

JavaScript

```
{  
  message: "Usuário não tem permissão para realizar a operação."  
  statusCode: 403  
}
```

Tabelas

Para realizar uma simulação de crédito, será necessário informar a tabela que será utilizada na operação. Cada tabela possui suas taxas e valores mínimo e máximo para a operação.

Endpoint de Listagem de Tabelas

- Requisição

Endereço da requisição: `{baseUrl}/service/fee-schedule`

Método: `GET`

- Resposta (sucesso)

StatusCode: 200

Body:

JavaScript

```
{
  statusCode: 200,
  message: 'feeSchedule recuperado com sucesso',
  data: [
    {
      id: "number",
      name: "string",
      minDisbursedAmount: "number", (valor liberado mínimo)
      maxDisbursedAmount: "number", (valor liberado máximo)
      minNumberOfYearsAntecipated: "number", (mínimo de anos para antecipar)
      maxNumberOfYearsAntecipated: "number", (mpaximo de anos para antecipar)
      monthlyFee: "number", (taxa mensal)
      annualFee: "number" (taxa anual)
    },
    { ... }
  ]
}
```

- Resposta (falha)

StatusCode: 400

Body:

JavaScript

```
{
  message: "Nenhuma organização foi encontrada para o usuário dessa requisição."
  statusCode: 400
}
```

JavaScript

```
{
  message: 'O campo "{{nome_do_campo}}" precisa ser "{{formato_necessário}}".'
  statusCode: 400
}
```

StatusCode: 404

Body:

JavaScript

```
{
  message: "Nenhuma tabela encontrada para essa empresa."
  statusCode: 404
}
```

Simular Operação de Crédito

A simulação de operação de crédito consiste na consulta do saldo do cliente, seguida da simulação, considerando o saldo do cliente e a tabela selecionada. Esses dois passos (consulta de saldo do cliente e simulação) acontecem em sequência, sendo necessária somente uma requisição ao endpoint de simulação para realizar todo o processo.

A simulação pode ser realizada considerando o **saldo total do cliente** (comportamento padrão), ou pode ser enviado um **valor solicitado** (quando envia-se, no corpo da requisição, a chave `targetDisbursedAmount`). Esse valor deve estar entre os limites mínimo e máximo daquele cliente, e para obter esses limites, basta realizar uma simulação por valor total primeiro. O valor a ser considerado como limite máximo é o retornado na chave `totalReleasedAmount`, que corresponde ao valor liberado para aquele cliente considerando todo o seu saldo liberado do Saque Aniversário do FGTS. Já o valor a ser considerado como limite mínimo, é o retornado na chave `minDisbursedAmount`, que calcula o valor mínimo para aquele cliente, de acordo com as regras operacionais internas.

É importante destacar que, ainda que as chaves `totalReleasedAmount` e `minDisbursedAmount` sejam retornadas em simulações por **valor solicitado**, os valores retornados nessas chaves, nesse tipo de simulação, não correspondem aos limites mínimo e máximo reais do cliente e, portanto, se usados como tal, implicará em erros na criação das operações. **Para consultar os limites mínimo e máximo do cliente, é obrigatório realizar, previamente, uma simulação por valor total.**

Para realizar as simulações e, posteriormente, criar as operações, é necessário que o cliente tenha aderido ao Saque Aniversário do FGTS e autorizado as instituições:

- Q.I. SOCIEDADE DE CREDITO
- CDC SOCIEDADE DE CREDITO AO MICROEMPREENDEDOR E A EMPRESA DE PEQUENO PORTE LTDA

Ambas as operações devem ser realizadas diretamente pelo cliente, no aplicativo do FGTS.

Endpoint de Simulação

- Requisição

Endereço da requisição: `{baseUrl}/service/simulation`

Método: `POST`

Body:

JavaScript

```
{
  clientCpf: "string", (cpf do cliente, sem máscara)
  feeScheduleId: "number", (int) (ID da tabela que será utilizada na simulação)
  targetDisbursedAmount: "number" (float) (valor solicitado: valor liberado alvo da
simulação - essa chave somente deve ser enviada em simulações por valor solicitado)
}
```

- Resposta (sucesso)

StatusCode: 200

Body:

JavaScript

```
{
  statusCode: 200,
  message: "Simulação realizada com sucesso."
  data: {
    simulationData: {
      installments: [ (parcelas a serem antecipadas)
        {
          dueDate: "string", (formato AAAA-MM-DD)
          amount: "number" (float)
        },
        { ... }
      ],
      iofAmount: "number", (float)(valor do IOF)
      totalReleasedAmount: "number", (float)(valor liberado para o cliente)
      totalAmount: "number", (float)(valor de emissão)
      contractTACAmount: "number", (float) (valor da TAC)
      contractCETRate: "number", (float) (CET anual)
      contractRate: "number", (float) (taxa anual)
      minDisbursedAmount: "number" (float) | "null", (valor mínimo liberado)
    } | "null",
    isExponentialFeeScheduleAvailable: "boolean",
    financialId: "string" ()
  }
}
```

A chave data da resposta de sucesso é um objeto que possui 3 chaves:

- **simulationData**: dados da resposta da simulação do cliente;
- **isExponentialFeeScheduleAvailable**: indica se a tabela exponencial está disponível para esse cliente
- **financialId**: id referente à dados necessários para realizar a simulação. Esse id deve ser também enviado na etapa de criação da operação.

Obs.: Vale destacar que durante toda simulação é realizado uma verificação se os valores liberados se encaixam para uso da Tabela Exponencial, o qual é retornado como valor booleano na chave **isExponentialFeeScheduleAvailable**. Logo, não se faz necessário simular com esta tabela, caso seja retornado **False**.

- Resposta (falha)

StatusCode: 400

Body:

JavaScript

```
{
  message: "Cliente não possui saldo suficiente para realizar a operação (mínimo de R$10,00 para cada parcela).",
  statusCode: 400
}
```

JavaScript

```
{
  message: "A operação não pode ser concluída por política interna: a data da primeira parcela é superior a 24 meses."
  statusCode: 400
}
```

JavaScript

```
{
  message: "A operação não pode ser concluída por política interna: o cliente não pode antecipar a quantidade mínima de 03 parcelas."
  statusCode: 400
}
```

JavaScript

```
{  
  message: "Saldo insuficiente para realizar a operação."  
  statusCode: 400  
}
```

JavaScript

```
{  
  message: "A operação não pode ser concluída por política interna: o valor liberado  
mínimo é maior que o máximo permitido pela tabela."  
  statusCode: 400  
}
```

JavaScript

```
{  
  message: "Saldo insuficiente para realizar a operação."  
  statusCode: 400  
}
```

JavaScript

```
{  
  message: 'O campo "{{nome_do_campo}}" precisa ser "{{formato_necessário}}".'  
  statusCode: 400  
}
```

JavaScript

```
{  
  message: 'O campo "{{nomeDoCampo}}" é obrigatório.'  
  statusCode: 400  
}
```

JavaScript

```
{  
  message: "Consulta de saldo ainda em andamento, acompanhe na tela de consultas."  
  statusCode: 400  
}
```

StatusCode: 404

Body:

JavaScript

```
{  
  message: "Nenhuma tabela de taxas encontrada para o ID informado."  
  statusCode: 404  
}
```

StatusCode: 500

Body:

JavaScript

```
{  
  message: "Falha ao realizar consulta do saldo disponível."  
  statusCode: 500  
}
```

```
JavaScript
{
  message: "Saldo insuficiente para realizar uma operação."
  statusCode: 500
}
```

```
JavaScript
{
  message: "Já existe uma operação de Saque Aniversário em andamento para esse CPF, tente novamente mais tarde."
  statusCode: 500
}
```

```
JavaScript
{
  message: "Cliente ainda não autorizou para visualizar seu saldo no aplicativo do FGTS."
  statusCode: 500
}
```

```
JavaScript
{
  message: "Cliente não optante da modalidade saque aniversário."
  statusCode: 500
}
```

```
JavaScript
{
  message: "A operação não é permitida para a data atual."
  statusCode: 500
}
```

```
JavaScript
{
  message: "Cliente possui solicitação de retorno para Saque Rescisão em andamento, o que impede operações de Saque Aniversário."
  statusCode: 500
}
```

JavaScript

```
{  
  message: "Foram feitas alterações no cadastro da conta FGTS, impedindo a contratação.  
  Entre em contato com o departamento FGTS da Caixa."  
  statusCode: 500  
}
```

JavaScript

```
{  
  message: "Instabilidade momentânea na consulta com a Caixa, tente novamente mais  
  tarde."  
  statusCode: 500  
}
```

JavaScript

```
{  
  message: "Erro inesperado ao consultar saldo. Por favor, tente novamente mais tarde  
  e, caso o erro persista, entre em contato com o suporte."  
  statusCode: 500  
}
```

JavaScript

```
{  
  message: "A consulta de saldo não retornou resultado dentro do tempo limite (5  
  minutos)."  
  statusCode: 500  
}
```

JavaScript

```
{  
  message: "A operação não é permitida para a data atual."  
  statusCode: 500  
}
```

JavaScript

```
{
```

```
    message: "Há uma operação de confiança em andamento. Por favor, tente novamente mais tarde."
    statusCode: 500
  }
```

JavaScript

```
{
  message: "A quantidade de períodos deve ser maior que zero."
  statusCode: 500
}
```

JavaScript

```
{
  message: "Número de CPF inválido"
  statusCode: 500
}
```

JavaScript

```
{
  message: "O funcionário não está inscrito no saque aniversário da data atual."
  statusCode: 500
}
```

JavaScript

```
{
  message: "Operação não permitida antes de <data>."
  statusCode: 500
}
```

JavaScript

```
{
  message: "Não é possível realizar a operação para o CPF fornecido."
  statusCode: 500
}
```

JavaScript

```
{
  message: "Operação não permitida devido a problemas pendentes no processo de pagamento do saque aniversário"
  statusCode: 500
}
```

JavaScript

```
{
  message: "Trabalhador com um pedido de retorno referente ao saque por rescisão, que deve ser cancelado pelo próprio trabalhador para habilitar o pedido de garantia."
  statusCode: 500
}
```

JavaScript

```
{
  message: "Alterações no cadastro da conta do FGTS foram feitas, impedindo a contratação. Por favor, entre em contato com o departamento do FGTS da Caixa."
  statusCode: 500
}
```

JavaScript

```
{
  message: "Consulta executada com sucesso. O trabalhador NÃO possui saldo disponível para realizar Operações Fiduciárias."
  statusCode: 500
}
```

JavaScript

```
{
  message: "O trabalhador informado não possui contas no FGTS."
  statusCode: 500
}
```


JavaScript

```
{  
  message: "Ocorreu um erro inesperado. Por favor, tente novamente mais tarde."  
  statusCode: 500  
}
```

JavaScript

```
{  
  message: "Instituição fiduciária sem um acordo ativo para a modalidade de operação de  
confiança."  
  statusCode: 500  
}
```

JavaScript

```
{  
  message: "A instituição fiduciária não possui a autorização do trabalhador para a  
operação de confiança."  
  statusCode: 500  
}
```

Criar operação de crédito

Este processo cria a operação de Crédito com Garantia no Saque Aniversário do FGTS.

Para tal, são necessários dados do cliente, bem como os dados da simulação que foi executada anteriormente.

O **payload** é composto de 6 itens:

- **feeScheduleId**: ID da tabela utilizada na simulação;
- **financialId**: id retornado pela simulação;
- **borrower**: dados pessoais do cliente;
- **document**: dados do documento do cliente;
- **address**: dados do endereço do cliente;
- **disbursementBankAccount**: dados da conta bancária do cliente, na qual o valor será desembolsado.

Endpoint de Criação da Operação

- Requisição

Endereço da requisição: `{baseUrl}/service/proposal`

Método: `POST`

Body: (a seguir na próxima página)

Obs.: A seguir body para requisição de Criação de Proposta

JavaScript

```
{
  feeScheduleId: "number", (int)
  financialId: "string",
  borrower: {
    name: "string",
    cpf: "string", (sem máscara)
```

```
    birthdate: "string", (formato AAAA-MM-DD)
    gender: "string",
    phoneNumber: "string", (número móvel, sem máscara e com ddd - "XX9XXXXXXXX")
    email: "string",
    maritalStatus: "string",
    nationality: "string",
    naturalness: "string" | "null",
    motherName: "string",
    fatherName: "string" | "null",
    pep: "boolean" (pessoa exposta politicamente)
  },
  document: {
    type: "string", ("cnh" ou "rg")
    number: "string", (sem máscara)
    issuingState: "string", (Ex: "CE", "SP", "RS" ...),
    issuingAuthority: "string",
    issueDate: "string" (formato AAAA-MM-DD)
  },
  address: {
    zipCode: "string", (sem máscara)
    street: "string",
    number: "string",
    complement: "string" | "null",
    neighborhood: "string",
    city: "string",
    state: "string", (Ex: "CE", "SP", "RS" ...)
  },
  disbursementBankAccount: {
    bankCode: "string",
    accountType: "string", ("corrente" ou "poupanca"),
    accountNumber: "string",
    accountDigit: "string",
    branchNumber: "string"
  }
}
```

- Resposta (sucesso)

StatusCode: 200

Body:

```
JavaScript
{
  statusCode: 200,
  message: "Proposta criada com sucesso!",
  data: {
    proposalId: "string",
    proposalContractNumber: "string"
  }
}
```

- Resposta (falha)

StatusCode: 400

Body:

```
JavaScript
{
  message: 'O campo "{{nome_do_campo}}" precisa ser "{{formato_necessário}}".',
  statusCode: 400
}
```

```
JavaScript
{
  message: 'O campo "{{nomeDoCampo}}" é obrigatório.',
  statusCode: 400
}
```

StatusCode: 404

Body:

```
JavaScript
{
  message: "FeeSchedule não encontrado."
  statusCode: 404
}
```

StatusCode: 500

Body:

```
JavaScript
{
  message: "Falha na criação de Proposta"
  statusCode: 500
}
```

```
JavaScript
{
  message: "Falha na criação da proposta, tente novamente mais tarde."
  statusCode: 500
}
```

```
JavaScript
{
  message: "Falha na criação da proposta, favor verificar o número de telefone."
  statusCode: 500
}
```