

# ***Manual API associado para Recebimento de eventos Webhook e Contratação Webhook***

*Cobrança*

28/02/2023 – Versão 1.0

## Sumário

API associado para Recebimentos de eventos Webhook.....	3
1. Apresentação API Recebimento de eventos .....	3
2. Objetivo .....	3
3. Orientações Iniciais.....	4
4. Quais as funcionalidades? .....	5
5. Parâmetros API de Recebimentos de eventos .....	5
API de Contratação Webhook.....	10
6. Apresentação API de Contratação Webhook .....	10
7. Para quem?.....	10
8. Objetivo .....	10
9. Orientações Iniciais.....	11
10. Quais as funcionalidades? .....	11
11. Iniciando a integração .....	13
12. Recursos disponíveis.....	16
12.1. Autenticação.....	16
URL Produção: .....	19
Execução (via ferramenta Postman) .....	20
12.2. Realizar Contratação Webhook.....	25
URL Produção: .....	30
Execução (via ferramenta Postman) .....	31
12.3. Consulta Contratos .....	33
URL Produção: .....	36
Execução (via ferramenta Postman) .....	36
12.1. Altera Contrato Webhook .....	40
URL Produção: .....	45
Execução (via ferramenta Postman) .....	46
13. Glossário .....	49



*A integração com o produto o Webhook Cobrança é realizado em duas etapas distintas que serão descritas neste manual.*

*1ª Etapa: Desenvolvimento de uma API para Recebimento de Eventos Webhook:  
É uma API desenvolvida pelo associado, responsável por receber os eventos  
(movimentações) dos títulos via webhook.*

*2ª Etapa: Contratação do produto via API de contratação informando a URL  
desenvolvida na etapa acima.*

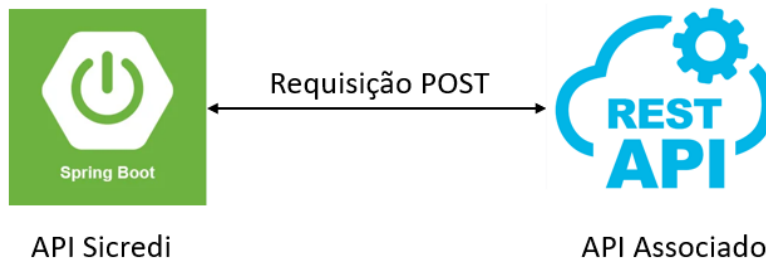
## API associado para Recebimentos de eventos Webhook

# 1. Apresentação API Recebimento de eventos

A API de Recebimento de eventos Webhook é uma API que deve ser desenvolvida pelo associado com o objetivo de se comunicar e realizar uma integração com a API Webhook do Sicredi, a fim de receber os eventos sobre as movimentações que ocorrem em seus títulos de cobrança.

## 2. Objetivo

Este manual **destina-se às pessoas responsáveis pelo desenvolvimento** e adaptação dos sistemas das empresas associadas ao Sicredi que desejam utilizar o Webhook para receber eventos de movimentações refere aos seus títulos de cobrança. O objetivo desse manual é orientar os desenvolvedores na construção de uma api para o Recebimentos de eventos que deve estar preparada para receber uma requisição POST, realizada por uma API do Sicredi, que irá conter os dados de uma determinada movimentação do título (evento). Segue abaixo uma ilustração desse processo:



### 3. Orientações Iniciais

Para iniciar o processo desenvolvimento de sua **API de Recebimentos de eventos Webhook** e promover a integração sistêmica, você deve seguir as orientações abaixo:

- O Protocolo de comunicação deve ser HTTPS;
- Sua API deve utilizar protocolo TLS 1.2 nas requisições;
- O certificado de sua API não pode ser auto assinado;
- A API Sicredi possui um timeout padrão de 10 segundos por requisição. Dessa forma, sua API de Recebimentos de eventos deverá responder o request em até 10 segundos. Em caso de ausência de resposta no tempo estipulado, o evento será marcado como "NÃO ENTREGUE";
- O código de retorno HTTP de esperado pela API de Recebimentos de eventos é 200;
- Nesse primeiro momento, sua API de Recebimentos de eventos não deve possuir qualquer tipo de autenticação. Estamos no processo de discussão e implementação de uma autenticação via certificado de segurança que ainda não foi finalizado;
- Ao fim do processo de desenvolvimento da API de Recebimentos de eventos, o associado deverá informar ao Sicredi o endpoint (URL) de sua API para que a API Sicredi consiga iniciar o processo de integração;

## 4. Quais as funcionalidades?

A API de Recebimentos de eventos deverá conter as seguintes funcionalidades:

- **POST - Recebimento de Eventos:** Método que irá ser chamado via API Sicredi para o recebimento dos eventos (movimentações). Neste primeiro os eventos disponibilizados são:
  - **LIQUIDACAO\_PIX:** Liquidações efetuadas via PIX.
  - **LIQUIDACAO\_REDE \*\*:** Liquidações efetuadas nos canais Sicredi.
  - **LIQUIDACAO\_COMPE\_H5:** Liquidações efetuadas em outras instituições (Fora da Rede SICREDI)
  - **LIQUIDACAO\_COMPE\_H6:** Liquidações com valor superior a 250 mil efetuadas em outras instituições (Fora da Rede SICREDI)
  - **LIQUIDACAO\_COMPE\_H8:** Recebimento de liquidação fora da Rede SICREDI-Contingência (Alguma inconsistência de sistema).
  - **LIQUIDACAO\_CARTORIO:** Liquidações efetuadas em cartórios.
  - **AVISO\_PAGAMENTO\_COMPE:** Para associados que possuem esse produto contratado, trata-se de um aviso de que a liquidação de um determinado título já ocorreu.
  - **ESTORNO\_LIQUIDACAO\_REDE:** Liquidações que foram estornadas.

💡 **\*\* Observação LIQUIDACÃO\_REDE:** Essa liquidação é a única que pode ter estorno no mesmo dia. Havendo estorno será enviado o evento "ESTORNO\_LIQUIDACAO\_REDE". Fica a critério do associado acatar de imediato a liquidação ou aguardar o fim do dia para garantir que não ocorreu um estorno.

## 5. Parâmetros API de Recebimentos de eventos

Para iniciar a integração, a API Sicredi irá encapsular os dados do evento descritos abaixo e irá realizar uma chamada, utilizando a URL disponibilizada pelo associado, em sua API de Recebimentos de eventos Webhook, que deve estar preparada para receber via POST uma requisição com os seguintes parâmetros:

**1. Exemplo JSON dos parâmetros:**

```
{
  "agencia": "9999",
  "posto": "99",
  "beneficiario": "12345",
  "nossoNumero": "221000144",
  "dataEvento": "2022-12-31T00:00:00.000Z",
  "movimento": "LIQUIDACAO_PIX",
  "valorLiquidacao": "101.01",
  "valorDesconto": "0",
  "valorJuros": "0",
  "valorMulta": "0",
  "valorAbatimento": "0",
  "idEventoWebhook": "N00000000000000000000000000000000LIQUIDACAO_PIX"
}
```

ENTRADA:

Ord	Nome	Tipo	Tamanho	Formatação	Descrição	Obrigatório
1	agencia	String	4	Sem formatação	Código da Cooperativa	Obrigatório
2	posto	String	2	Sem formatação	Código do Posto	Obrigatório
3	beneficiario	String	5	Sem formatação	Código do Convênio de Cobrança	Obrigatório
4	nossoNumero	String	9	Sem formatação	Nosso número do título	Obrigatório
5	dataEvento	Date	40	YYYY-MM-DDT00:00:00.000Z	Data e hora que ocorreu o evento	Obrigatório
6	movimento	String	753	Domínios: "LIQUIDACAO_PIX", "LIQUIDACAO_COMPE_H5", "LIQUIDACAO_COMPE_H6", "LIQUIDACAO_COMPE_H8", * "LIQUIDACAO_REDE",	Lista de eventos que o associado receberá informações em caso de movimentação.  <b>"LIQUIDACAO_REDE"</b> <b>*Obs: Essa liquidação é a única que pode ter estorno no mesmo dia. Havendo estorno</b>	Obrigatório

				"LIQUIDACAO_CARTORIO" "AVISO_PAGAMENTO_COMPE", "ESTORNO_LIQUIDACAO_REDE"	<b>será enviado o evento "ESTORNO_LIQUIDACAO_REDE".</b> <b>Fica a critério do associado acatar de imediato a liquidação ou aguardar o fim do dia para garantir que não ocorreu um estorno.</b>	
7	valorLiquidacao	String	14,2	Sem formatação	Valor da liquidação	Obrigatório
8	valorDesconto	String	14,2	Sem formatação	Valor de desconto aplicado na liquidação.	Obrigatório
9	valorJuros	String	14,2	Sem formatação	Valor de juros aplicado na liquidação	Obrigatório
10	valorMulta	String	5,2	Sem formatação	Valor de multa aplicado na liquidação	Obrigatório
11	valorAbatimento	String	14,2	Sem formatação	Valor de abatimento aplicado na liquidação	Obrigatório



12	idEventoWebhook	String	50	Sem formatação	Código de identificação único de cada evento. Cada evento possui um identificador único, representado por esse parâmetro.	Obrigatório
----	-----------------	--------	----	----------------	---	-------------

**RETORNO (Response):**

Código	Descrição
HTTP_OK (200)	Operação realizada com sucesso

## API de Contratação Webhook

# 6. Apresentação API de Contratação Webhook

A API de Contrato Webhook realiza a criação de um contrato para os beneficiários que desejam utilizar este produto, que tem como finalidade o recebimento de eventos sobre as movimentações que ocorrem em seus títulos de cobrança. O Webhook é um recurso que possibilita o envio de dados em tempo real entre dois sistemas. Através do Webhook, diferentes sistemas podem ser integrados para que os eventos entre produtor e cliente permaneçam sincronizados.

# 7. Para quem?

São **variados os tipos de negócios** que podem empregar este tipo de recebimento. Negócios representados por uma Pessoa Jurídica, seja do ramo da indústria, comércio, prestação de serviço, agronegócio, Governo\*, não importando esferas municipais, estaduais ou federal, da administração direta ou indireta, ou Pessoa Física, profissional liberal, produtor rural ou autônomo, que precisem receber recursos oriundos de uma oferta, proposta, venda ou serviço prestado.

# 8. Objetivo

Este manual **destina-se às pessoas responsáveis pelo desenvolvimento** e adaptação dos sistemas das empresas associadas ao Sicredi que desejam realizar contratação do Webhook e receber os eventos de movimentações dos títulos através da integração via API.

## 9. Orientações Iniciais

Para iniciar o processo de contratação do Webhook, o **associado deve possuir um convênio ATIVO com o produto Cobrança do Sicredi e possuir a modalidade Cobrança Online.**

Logo, para promover a integração sistêmica de registro on-line de Boletos a empresa ou pessoa associada inicialmente precisa:


- Ter contratado o produto Cobrança através da sua cooperativa
- Ter optado pela modalidade API (Cobrança Online);
- Ter o termo de contratação/adesão assinado;

## 10. Quais as funcionalidades?


A API de Contratação Webhook contém as seguintes funcionalidades:

- **Autenticação:** Operação responsável pela autenticação do código de acesso e criação do token de autenticação, que permitirá acessar a **API** de Contratação Webhook.
- **Cadastro de Contrato Webhook:** Operação responsável pela criação de um contrato para utilização do Webhook. Cada associado poderá possuir apenas 1 contrato por código de beneficiário e cooperativa. Nesse primeiro momento, está disponível a contratação de apenas 1 evento:
  - **LIQUIDAÇÃO:** Onde o associado receberá informações de qualquer movimentação relacionada as liquidações de seus boletos, sejam elas:
    - **LIQUIDACAO\_PIX:** Liquidações efetuadas via PIX.
    - **LIQUIDACAO\_REDE \*\*:** Liquidações efetuadas nos canais Sicredi.
    - **LIQUIDACAO\_COMPE\_H5:** Liquidações efetuadas em outras instituições (Fora da Rede SICREDI)
    - **LIQUIDACAO\_COMPE\_H6:** Liquidações com valor superior a 250 mil efetuadas em outras instituições (Fora da Rede SICREDI)
    - **LIQUIDACAO\_COMPE\_H8:** Recebimento de liquidação fora da Rede SICREDI-Contingência (Alguma inconsistência de sistema).
    - **LIQUIDACAO\_CARTORIO:** Liquidações efetuadas em cartórios.

- **AVISO\_PAGAMENTO\_COMPE:** Para associados que possuem esse produto contratado, trata-se de um aviso de que a liquidação de um determinado título já ocorreu.
- **ESTORNO\_LIQUIDACAO\_REDE:** Liquidações que foram estornadas.

 **\*\* Observação LIQUIDACÃO\_REDE:** Essa liquidação é a única que pode ter estorno no mesmo dia. Havendo estorno será enviado o evento "ESTORNO\_LIQUIDACAO\_REDE". Fica a critério do associado acatar de imediato a liquidação ou aguardar o fim do dia para garantir que não ocorreu um estorno.

- **Consulta de Contrato Webhook:** Operação responsável pela consulta dos dados de um contrato através do código de beneficiário, cooperativa e posto.
- **Altera Contrato Webhook:** Operação responsável por alterar dados de um contrato através de um IdContrato.

 **Observação:** O tópico abaixo (11. Iniciando a Integração) é destinado para os usuários que não conhecem o processo de integração da nova API Cobrança Online. Caso já tenha conhecimento neste processo, o tópico abaixo pode ser ignorado, visto que a autenticação na API de Contratação Webhook é realizada com as mesmas credenciais da API Cobrança Online.

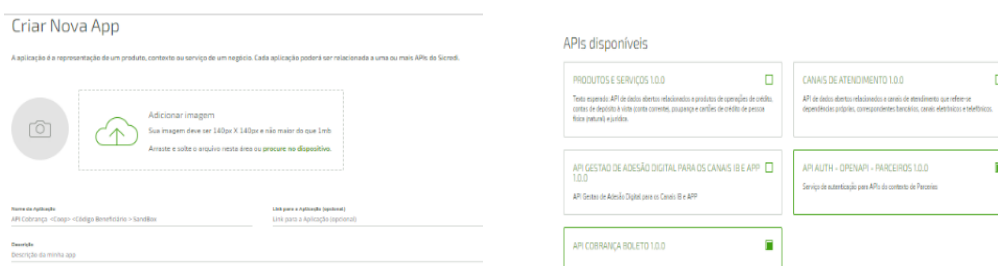
## 11. Iniciando a integração

Para iniciar a integração há um processo constituído de etapas sequenciais para Validação e Produção:

### 2. Validação/Teste

Acessar o Portal do Desenvolvedor:

- a. Criação de **APP para Homologação**
  - b. Solicitação do **Access Token de Homologação (Sandbox)**
  - c. **Testes nas URLS**
- a. Criação de **APP para Homologação:** para criar a APP de homologação, o desenvolvedor deve acessar o Portal do Desenvolvedor Sicredi ([clique aqui](#)) e criar uma conta ou login. Após criar uma conta ou login, o desenvolvedor deve clicar em 'criar uma nova aplicação'. Ao criar uma nova aplicação deve nomeá-la com o prefixo 'API Cobrança <Coop> <Código Beneficiário> SandBox' para identificar que é a APP de homologação e selecionar a API da Cobrança [ API COBRANÇA BOLETO 1.0.0 ] e a API de Autenticação Parceiros na lista de APIs disponíveis [API AUTH - OPENAPI - PARCEIROS 1.0.0]. Ao criar uma app você já terá um Client ID gerado automaticamente para essa APP.



1. Criando a APP no portal

2. Selecionando as APIs na lista

- b. **Geração do Access Token de Homologação (Sandbox):** após criar a APP de Homologação, você deve, ainda no portal do desenvolvedor, clicar no menu suporte > abrir chamado. Selecione a opção “COLOCAR AQUI QUAL OPÇÃO”, e preencha o nome da APP com aquele nome que você criou no passo anterior. Pronto, o seu Access token será gerado e dentro de X dias você poderá visualizar o seu token de homologação no menu ‘minhas apps’, selecionando a app criada e clicando em ‘ver detalhes’.

## Minhas Apps

Criar nova aplicação

3. Visualizando o access token

API Cobrança - Sandbox	API Cobrança - Sandbox	XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX	Aprovada	Ver Detalhes
------------------------	------------------------	--------------------------------------	----------	--------------

a teste dos

recursos disponíveis em ambiente de homologação.

## 2. Produção

Acessar o Portal do Desenvolvedor

- Criação de **APP para Produção**
- Solicitação do **Access Token de Produção**

Acessar o Internet Banking

- Geração da **Código de acesso (password) no internet banking;**

- a. **Criação de APP para produção:** Para criar a APP de produção, você pode repetir os mesmos passos da criação da APP de homologação, no entanto, **lembre-se de nomear essa nova APP de uma forma que você identifique que ela se trata da APP de produção**, nossa sugestão é: ‘API Cobrança <Coop> <Código Beneficiário > Produção’. Após criar a APP você terá acesso imediato ao Client ID de produção.

- b. **Geração do Access Token de Produção:** Repita os mesmos passos já realizados para a geração de token de homologação, mas **dessa vez ao abrir o chamado, preencha o campo Client ID ou Nome da APP com os dados da APP de produção**. O seu token será disponibilizado no mesmo prazo anteriormente citado e poderá ser visualizado no menu minhas apps.

## Minhas Apps

Criar nova aplicação

Nome	Descrição	Client ID / Client Secret	Status	Access Tokens
API Cobrança - Sandbox	API Cobrança - Sandbox	XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX	Aprovada	Ver Detalhes
API Cobrança - Produção	API Cobrança - Produção	XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX	Aprovada	Ver Detalhes

28/02/2023 Vers

Página 14

4. Visualizando as APPs no menu


Classificação da informação: Uso Interno

É importante lembrar que em todas as operações será necessário informar no cabeçalho da requisição no campo x-api-key o token de autenticação recebido. Esse token é diferente para o ambiente de Homologação (Sandbox) e Produção, deve ser informado o token correspondente ao ambiente que está sendo utilizado.

- c. **Geração do Código de Acesso (password):** a empresa ou pessoa associada deve acessar o Internet Banking, pela conta vinculada ao seu convênio de Cobrança, e selecionar a opção de menu no: **Cobrança >> Código de Acesso >> Gerar** - Esse menu é apresentado exclusivamente para os beneficiários que possuem a **modalidade API(Cobrança Online) habilitada no convênio**.



5. Geração de Código de Acesso (password)

 ***Se você ou a sua empresa associada não possui a modalidade habilitada e deseja habilitá-la, converse com a pessoa gerente de conta.***

A chave será gerada após informado o dispositivo de segurança. **Ela deve ser armazenada e utilizada na configuração da integração da API**, pois esse código gerado é utilizado no recurso de autenticação para permitir as operações. **Caso o beneficiário possua mais de uma conta com o produto Cobrança contratado e deseje utilizá-las, precisará gerar o token de autenticação para cada conta específica.**

## 12. Recursos disponíveis

### 12.1. Autenticação

Em todas as operações será necessário informar no cabeçalho da requisição no campo x-api-key o token de autenticação recebida do Sicredi. Esse token é diferente para o ambiente de Sandbox e Produção, deve ser informado o token correspondente ao ambiente que está sendo utilizado.

A operação **POST** “token” é responsável por criar uma chave criptografada, denominada “access\_token”, utilizando o Beneficiário + Cooperativa como usuário (username) e o Código de Acesso (password) que foi gerada no Internet Banking. O padrão de autenticação é o OAuth2 e o token utiliza o padrão JWT.

O formato de entrada é “x-www-form-urlencoded” e saída é JSON, considerando que a saída sempre é composta da entidade a ser retornada e um código HTTP dentro de um *Response*. O tipo de codificação utilizado no *Response* é Unicode UTF-8.

**Observação:** deve ser respeitada a orientação da segurança da informação referente a expiração do access\_token e do refresh\_token, observando os parâmetros “expires\_in” (exemplo: 300 segundos) e “refresh\_expires\_in” (exemplo: 900 segundos) de retorno do serviço respectivamente. Os parâmetros de expiração (“expires\_in” e “refresh\_expires\_in”) podem ser diferentes em Teste e Produção.

Para verificar se o access\_token está expirado, deve-se observar o campo “expires\_in” que retorna a duração do token em segundos.

O access\_token deve ser enviado no cabeçalho das chamadas das próximas operações até que ele expire, não deve ser realizado autenticação a cada chamada. Quando o access\_token expirar, deve ser utilizado o refresh\_token para gerar um novo token.

**OBS:** Quando da utilização do refresh\_token, não será necessário informar os campos “username” e “password”.

Quando o refresh\_token estiver expirado, deverá ser realizado o fluxo de autenticação normal, com usuário e senha (username e password).



**ENTRADA:**

Ord	Nome	Tipo	Tipo de Parâmetro	Tamanho	Formatação	Descrição	Obrigatório
1	x-api-key	String	HeaderParam	36	UUID 38 bits	Access token gerado no portal do desenvolvedor	Obrigatório
2	context	String	HeaderParam	8	<u>COBRANCA</u>	Contexto, passar fixo "COBRANCA"	Obrigatório
3	Content-Type		HeaderParam		application/x-www-form-urlencoded	Indica o tipo de arquivo do recurso	Obrigatório
4	grant_type	String	Body	Até 13	password ou refresh_token	Tipo de autenticação. Ver diagrama, pág 113	Obrigatório
5	username	String	Body	9	Sem formatação	Código do beneficiário + Código da Cooperativa	Obrigatório para Grant Type igual a password
6	password	String	Body	64	Sem formatação	Código de Acesso gerado no Internet Banking	Obrigatório para Grant Type igual password

7	refresh_token	String	Body	753	Sem formatação dígitos	Refresh token obtido anteriormente na autenticação	Obrigatório para Grant Type igual a refresh_token
8	scope	String	Body	8	cobranca	Escopo de acesso. Passar fixo "cobranca".	Obrigatório para Grant Type password

**SAÍDA:**

Ord	Nome	Tipo	Tipo de Parâmetro	Formatação	Descrição
1	access_token	String	JSON	576 bits	Token de autenticação para ser utilizado
2	token_type	String	JSON	2017-08-09T08:53:11.679-03:00	Tipo do Token
3	refresh_token	String	JSON	636 dígitos	Será utilizado para gerar um novo access_token, sem a necessidade de uma nova autenticação com usuário e senha
4	expires_in	String	JSON	3 dígitos	Tempo de expiração (em segundos) do Access Token
5	refresh_expires_in	String	JSON	6 dígitos	Tempo de expiração (em segundos) do Refresh Token
6	scope	String	JSON	cobranca profile email	Escopo que o usuário possui acesso.

**RETORNO (Response):**

Código	Descrição
HTTP_OK (200)	Operação realizada com sucesso

**FALHAS:**

Status	Mensagem	Parâmetros de entrada sujeitos a crítica	Descrição
HTTP_NOT_FOUND (404)	Ocorreu um erro ao realizar operação de autenticação.	Não possui	Erro quando o endereço (URL) inválido. Deve-se verificar a url utilizada.
HTTP_UNAUTHORIZED (401)	Invalid user credentials	username, password, context	Erros ocasionados por username ou password não informado ou inválidos.
HTTP_UNAUTHORIZED (401)	Missing parameter: username	username	Erro ocasionado por username não informado.
HTTP_UNAUTHORIZED (401)	Could not find a required Access Token in the request, identified by HEADER x-api-key	x-api-key	Erro ocasionado quando o parâmetro x-api-key não foi informado ou está inválido.
HTTP_BAD_REQUEST (400)	Missing form parameter: grant_type	grant_type	Erro ocasionado quando o parâmetro grant_type não é informado ou inválido.
HTTP_GATEWAY_TIMEOUT (504)	Gateway Timeout	Não se aplica	Erro ocasionado quando o serviço não conseguiu responder no tempo esperado.

## URL Produção:

<https://api-parceiro.sicredi.com.br/auth/openapi/token>

**RESPONSE:**

```
{  
  "access_token":  
    "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXLTJlbiA6ICJxMIRFLXliYWxnNjlxSmRCMU9IX0w2Sms5QllxODJtNkFSZ0NGaDZtQnZBIn0.eyJleHAiOiE2NTY0NTA5MzIsImhhbmhlcGlMTY1NjQ1MDYzMiwianRpIjoziNiYjg5YzQtOGRmM
```



POST

https://api-parceiro.sicredi.com.br/auth/openapi/token

Send

Params

Authorization

Headers (3)

Body

Pre-request Script

Tests

☐ none

☐ form-data

☒ x-www-form-urlencoded

☐ raw

☐ binary

	KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/>	username	123451234	
<input checked="" type="checkbox"/>	password	4DC57770EFFA0AEB7D21A9954DE980C5E3488BAACB148F90B...	
<input checked="" type="checkbox"/>	scope	cobranca	
<input checked="" type="checkbox"/>	grant_type	password	

**OBS:** Sempre verificar o tempo de expiração do `access_token`, através do campo `"expires_in"` (Exemplo: 300 segundos). Caso esteja expirado, deverá usar o `refresh_token` para gerar um novo token.

- **Cenário Positivo** – Utilizando o `refresh_token` para gerar um novo token

[illegible]

**OBS:** Sempre verificar o tempo de expiração do refresh\_token, através do campo “refresh\_expires\_in” (Exemplo: 1800 segundos). Caso esteja expirado, deverá gerar um novo access\_token. Quando da utilização do refresh token, não será necessário informar os campos “username” e “password”.

■ Cenário de Erro <400 - BAD\_REQUEST>

POST ▼ https://api-parceiro.sicredi.com.br/auth/openapi/token Send ▼

	KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/>	username	123450718	
<input checked="" type="checkbox"/>	password	4DC57770EFA0AEB7D21A9954DE980C5E3488B...	
<input checked="" type="checkbox"/>	scope	cobranca	
<input type="checkbox"/>	grant_type	password	
<input type="checkbox"/>	refresh_token	eyJhbGciOiJIUzI1NiIsInR5cCIgOiAiSldUiwiwia2lkliA...	
	Key	Value	Description

Body Cookies Headers (17) Test Results Status: 400 Bad Request Time: 142 ms Size: 1020 B

Pretty Raw Preview JSON ↺

```
1 {  
2   "error": "invalid_request",  
3   "error_description": "Missing form parameter: grant_type"  
4 }
```

■ Cenários de Erro <401 - UNAUTHORIZED>

POST ▼ https://api-parceiro.sicredi.com.br/auth/openapi/token Send ▼ Save

<input checked="" type="checkbox"/>	username	123450718	
<input checked="" type="checkbox"/>	password	4DC57770EFA0AEB7D21A9954DE980C5E3488B...	
<input checked="" type="checkbox"/>	scope	cobranca	
<input checked="" type="checkbox"/>	grant_type	password	
<input type="checkbox"/>	refresh_token	eyJhbGciOiJIUzI1NiIsInR5cCIgOiAiSldUiwiwia2lkliA...	
	Key	Value	Description

Body Cookies Headers (17) Test Results Status: 401 Unauthorized Time: 430 ms Size: 1009 B Download

Pretty Raw Preview JSON ↺ 📄

```
1 {  
2   "error": "invalid_grant",  
3   "error_description": "Invalid user credentials"  
4 }
```

POST ▼ https://api-parceiro.sicredi.com.br/auth/openapi/token Send ▼

	KEY	VALUE	DESCRIPTION	...
<input type="checkbox"/>	username	123450718		
<input checked="" type="checkbox"/>	password	4DC57770EFA0AEB7D21A9954DE980C5E3488B...		
<input checked="" type="checkbox"/>	scope	cobranca		
<input checked="" type="checkbox"/>	grant_type	password		
<input type="checkbox"/>	refresh_token	eyJhbGciOiJIUzI1NiIsInR5cCIgOiAiSldUiiwia2lkliA...		
	Key	Value	Description	

Body Cookies Headers (17) Test Results Status: 401 Unauthorized Time: 135 ms Size: 1014 B

Pretty Raw Preview JSON ≡

```
1 {  
2   "error": "invalid_request",  
3   "error_description": "Missing parameter: username"  
4 }
```

POST ▼ https://api-parceiro.sicredi.com.br/auth/openapi/token Send ▼

Params Authorization Headers (3) Body ● Pre-request Script Tests Cookies Code

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	Content-Type	application/x-www-form-urlencoded			
<input type="checkbox"/>	x-api-key	eb095b70-cec7-349a-86ac-7947e1fe8037			
<input checked="" type="checkbox"/>	context	COBRANCA			
	Key	Value	Description		

Body Cookies Headers (8) Test Results Status: 401 Unauthorized Time: 48 ms Size: 678 B

Pretty Raw Preview JSON ≡

```
1 {  
2   "error": "Could not find a required Access Token in the request, identified by HEADER x-api-key"  
3 }
```



## 12.2. Realizar Contratação Webhook

A operação POST “criar” é responsável por gerar um novo contrato Webhook. O formato de entrada e saída é JSON, considerando que a saída sempre é composta da entidade a ser retornada e um código HTTP dentro de um *Response*. O tipo de codificação utilizado no *Response* é Unicode UTF-8. Por padrão o recurso cadastro responde em milissegundos.

### ENTRADA:

Ord	Nome	Tipo	Parametro	Tamanho	Formatação	Descrição	Obrigatório
1	x-api-key	String	Header Param	38	UUID	Access token gerado no portal do desenvolvedor	Obrigatório
1	Authorization	String	Header Param	1413	Bearer + Token de Autenticação	Campo access_token obtido na saída da autenticação.	Obrigatório
2	Content-Type	Application /json	Header param		Sem formatação	Estabelece o tipo de conteúdo da requisição	
3	cooperativa	String	Body	4	Sem formatação	Código da Cooperativa -	Obrigatório
4	posto	String	Body	2	Sem formatação	Código da Agência	Obrigatório
5	codigoBeneficiario	String	Body	5	Sem formatação	Código do Convênio de Cobrança	Obrigatório
7	eventos	List<String>	Body		Domínios: LIQUIDACAO	Eventos enviados via Webhook	Obrigatório
8	url	String	Body	256	Sem formatação	URL de integração entre Sicredi e associado. Obs: A url deve possuir, obrigatoriamente, o protocolo (https)	Obrigatório

9	urlStatus	String	Body	10	Domínios: ATIVO, INATIVO, BLOQUEADO	Status da URL que está sendo utilizada na comunicação.	Obrigatório
10	contratoStatus	String	Body	10	Domínios: ATIVO, INATIVO, BLOQUEADO	Status do contrato gerado	Obrigatório
11	nomeResponsavel	String	Body	50	Sem formatação	Nome do responsável pelo contrato	Opcional
12	email	String	Body	40	Sem formatação	Email do responsável	Opcional
13	telefone	String	Body	20	Sem formatação	Telefone do Responsável	Opcional

**SAÍDA:**

Ord	Nome	Tipo	Tamanho	Formatação	Descrição
1	idContrato	String	25	Sem formatação	Identificador único do contrato.
2	cooperativa	String	4	Sem formatação	Código da Cooperativa do Beneficiário.
3	posto	String	2	Sem formatação	Código da Posto do Beneficiário.
4	codBeneficiario	String	5	Sem formatação	Código do Beneficiário de Cobrança.
5	eventos	List<String>		Domínios: "LIQUIDACAO_PIX", "LIQUIDACAO_COMPE_H5", "LIQUIDACAO_COMPE_H6", "LIQUIDACAO_COMPE_H8", * "LIQUIDACAO_REDE", "LIQUIDACAO_CARTORIO", "AVISO_PAGAMENTO_COMPE", "ESTORNO_LIQUIDACAO_REDE"	Lista de eventos que o associado receberá informações em caso de movimentação.  <b>"LIQUIDACAO_REDE"</b> *Obs: Essa liquidação é a única que pode ter estorno no mesmo dia. Havendo estorno será

					<p>enviado o evento "ESTORNO_LIQUIDACAO_REDE".</p> <p>Fica a critério do associado acatar de imediato a liquidação ou aguardar o fim do dia para garantir que não ocorreu um estorno.</p>
6	url	String	256	Sem formatação	URL de integração entre Sicredi e associado
7	urlStatus	String	10	Domínios: ATIVO, INATIVO, BLOQUEADO	Status da URL que está sendo utilizada na comunicação.
8	contratoStatus	String	10	Domínios: ATIVO, INATIVO, BLOQUEADO	Status do contrato gerado
9	nomeResponsavel	String	50	Sem formatação	Nome do responsável pelo contrato
10	Email	String	40	Sem formatação	Email do responsável
11	telefone	String	20	Sem formatação	Telefone do Responsável

**RETORNO (Response):**

Status	Descrição
HTTP_ACCEPTED (201)	Operação aceita com sucesso

**FALHAS:**

Status	Mensagem	Parâmetros sujeitos à crítica	Descrição
HTTP_NOT_FOUND (404)	Ocorreu um erro ao realizar operação de cadastro	Não possui	Erro ocasionado por URL inválida.
HTTP_UNSUPPORTED_MEDIA_TYPE (415)	Tipo de conteúdo não suportado	Parâmetro: Content-Type	Erro ocasionado por parâmetro obrigatório não informado ou inválido.
HTTP_UNAUTHORIZED (401)	Unauthorized	Parâmetro: Authorization	Erro ocasionado por parâmetro obrigatório não informado ou inválido.
HTTP_UNAUTHORIZED (401)	Cooperativa diferente da cooperativa do usuário	Parâmetro: cooperativa	Erro ocasionado quando o código de cooperativa informado no cadastro for diferente do contido na autenticação.
HTTP_UNAUTHORIZED (401)	Código de beneficiário diferente do beneficiário do usuário	Parâmetro: codigoBeneficiario	Erro ocasionado quando o código de beneficiário informado no cadastro for diferente do contido na autenticação.
HTTP_BAD_REQUEST (400)	Campo obrigatório em branco	Qualquer parâmetro que consta como obrigatório na listagem de parâmetros de entrada pode ser criticado	Erros ocasionados por parâmetros obrigatórios não informados ou inválidos.
HTTP_BAD_REQUEST (400)	“Cooperativa inválida”	Parâmetro: cooperativa	Erros ocasionados por parâmetros obrigatórios não preenchidos ou inválidos.
HTTP_BAD_REQUEST (400)	“Posto inválido”	Parâmetro: posto	Erros ocasionados por parâmetros obrigatórios não preenchidos ou inválidos

HTTP_BAD_REQUEST (400)	“Evento deve conter LIQUIDACAO”	Parâmetro: evento	Erro ocasionado por parâmetro obrigatório não informado ou inválido.
HTTP_BAD_REQUEST (400)	“URL é obrigatória”	Parâmetro: url	Erro ocasionado por parâmetro obrigatório não informado ou inválido.
HTTP_BAD_REQUEST (400)	“URL inválida”	Parâmetro: url	O campo foi preenchido em branco ("" ) ou o tipo de protocolo informado foi diferente de "http" ou "https".
HTTP_BAD_REQUEST (400)	“URL Status não foi informado ou está inválida”	Parâmetro: urlStatus	O parâmetro não foi informado ou foi preenchido com valor inválido.
HTTP_BAD_REQUEST (400)	“O campo URL Status deve ser ATIVO, INATIVO, BLOQUEADO”	Parâmetro: urlStatus	O campo “urlStatus” foi preenchido com um valor diferente dos domínios: ATIVO, INATIVO, BLOQUEADO
HTTP_BAD_REQUEST (400)	“Contrato Status é obrigatório”	Parâmetro: contratoStatus	O campo "contratoStatus" não foi preenchido.
HTTP_BAD_REQUEST (400)	O campo Contrato Status deve ser ATIVO, INATIVO ou BLOQUEADO.	Parâmetro: contratoStatus	O campo “contratoStatus” foi preenchido com um valor diferente dos domínios: ATIVO, INATIVO, BLOQUEADO
HTTP_BAD_REQUEST (400)	“Email inválido”	Parâmetro: email	O campo “email” foi preenchido com um endereço de email inválido. Exemplo: Sem o @.
UNPROCESSABLE_ENTITY (422)	“Contrato já existente.”		Já existe um contrato cadastrado com os mesmos dados (coop e bnf)

## URL Produção:

<https://api-parceiro.sicredi.com.br/cobranca/boleto/v1/webhook/contrato/>

### HEADERS:

Authorization  
x-api-key  
Content-Type: application/json

### REQUEST BODY (JSON):

```
{
  "cooperativa": "0123",
  "posto": "12",
  "codBeneficiario": "12345",
  "eventos": [
    "LIQUIDACAO"
  ],
  "url": "https://teste.instituicao.cloud/v1/contratos",
  "urlStatus": "ATIVO",
  "contratoStatus": "ATIVO",
  "nomeResponsavel": "NOME RESPONSAVEL",
  "email": "TESTE@EMAIL.COM.BR",
  "telefone": "51 999999999"
}
```

### RESPONSE BODY (JSON):

```
{
  "idContrato": "63a241a590364b703bbe6aac",
  "cooperativa": "0123",
  "posto": "12",
  "codBeneficiario": "12345",
  "eventos": [
    "LIQUIDACAO_PIX",
    "LIQUIDACAO_COMPE_H5",
    "LIQUIDACAO_COMPE_H6",
    "LIQUIDACAO_COMPE_H8",
    "LIQUIDACAO_REDE",
    "LIQUIDACAO_CARTORIO",
    "AVISO_PAGAMENTO_COMPE",
    "ESTORNO_LIQUIDACAO_REDE"
  ],
  "url": "https://teste.instituicao.cloud/v1/contratos",
  "urlStatus": "ATIVO",
}
```

```
"contratoStatus": "ATIVO",  
"nomeResponsavel": "NOME RESPONSAVEL",  
"email": "TESTE@EMAIL.COM.BR",  
"telefone": "51 999999999"  
}
```

## Execução (via ferramenta Postman)

### ■ Cenário Positivo:

The screenshot displays the Postman interface for a POST request. The 'Body' tab is selected, showing a JSON payload. The request is sent to the URL `https://teste.instituicao.cloud/v1/contratos`. The response status is 201 Created, with a time of 306 ms and a size of 1.02 KB. The response body is shown in the 'Pretty' view, displaying a JSON object with contract details and a list of events.

**Request Body (JSON):**

```
{  
  "cooperativa": "0101",  
  "posto": "19",  
  "codBeneficiario": "79129",  
  "eventos": [  
    "LIQUIDACAO"  
  ]  
}
```

**Response Body (JSON):**

```
{  
  "idContrato": "63a241a590364b703bbe6aac",  
  "cooperativa": "0101",  
  "posto": "19",  
  "codBeneficiario": "79129",  
  "eventos": [  
    "LIQUIDACAO_PIX",  
    "LIQUIDACAO_COMPE_H5",  
    "LIQUIDACAO_COMPE_H6",  
    "LIQUIDACAO_COMPE_H8",  
    "LIQUIDACAO_REDE",  
    "LIQUIDACAO_CARTORIO",  
    "AVISO_PAGAMENTO_COMPE",  
    "ESTADO_LIQUIDACAO_REDE"  
  ]  
}
```

■ Cenário de erro <Contrato já existente>

Params Authorization Headers (13) Body ● Pre-request Script Tests Settings

Headers 9 hidden

	Key	Value	Description
<input checked="" type="checkbox"/>	accept	application/json	
<input checked="" type="checkbox"/>	Content-Type	application/json	
<input checked="" type="checkbox"/>	Authorization	Bearer eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUiiwia2IklIA6ICJBU0ZSSm1V	

Body Cookies Headers (16) Test Results

Status: 422 Unprocessable Entity (WebDAV) (RFC 4918)

Pretty Raw Preview Visualize JSON

```
1  {
2    "timestamp": "2023-03-21T18:25:37.024+00:00",
3    "error": "UNPROCESSABLE_ENTITY",
4    "code": "422",
5    "message": "Contrato já existente"
6  }
```



## 12.3. Consulta Contratos

A operação GET “Consulta Contratos” é responsável pela é responsável por realizar a consulta de contratos de um determinado beneficiário. Nesse método você verificar seus contratos ATIVOS, INATIVOS ou BLOQUEADOS, bem como os dados do seu contrato. O formato de entrada é sempre JSON, e a saída sempre é composta da entidade a ser retornada e um código HTTP dentro de um *Response* em caso de sucesso, e JSON, em caso de falha.

### Entrada:

Ord	Nome	Tipo	Tipo de Parâmetro	Tamanho	Formatação	Descrição	Obrigatório
1	x-api-key	String	HeaderParam	36	UUID	Token de acesso fornecido pelo Sicredi	Obrigatório
2	Authorization	String	HeaderParam	1413	Bearer + Token de Autenticação	Acess token obtido na autenticação	Obrigatório
3	cooperativa	String	QueryParam	4	Sem formatação	Código da cooperativa.	Obrigatório
	posto	String	QueryParam	4	Sem formatação	Código do Posto	Obrigatório
	Beneficiário	String	QueryParam	5	Sem formatação	Código do Beneficiário	Obrigatório

### Saída:

Ord	Nome	Tipo	Tamanho	Formatação	Descrição
1	idContrato	String	25	Sem formatação	Identificador único do contrato.
2	cooperativa	String	4	Sem formatação	Código da Cooperativa do Beneficiário.
3	posto	String	2	Sem formatação	Código da Posto do Beneficiário.

4	codBeneficiario	String	5	Sem formatação	Código do Beneficiário de Cobrança.
5	eventos	List<String>		Domínios: "LIQUIDACAO_PIX", "LIQUIDACAO_COMPE_H5", "LIQUIDACAO_COMPE_H6", "LIQUIDACAO_COMPE_H8", "LIQUIDACAO_REDE", "LIQUIDACAO_CARTORIO" "AVISO_PAGAMENTO_COMPE", "ESTORNO_LIQUIDACAO_REDE"	Lista de eventos que o associado receberá informações em caso de movimentação.
6	url	String	256	Sem formatação	URL de integração entre Sicredi e associado
7	urlStatus	String	10	Domínios: ATIVO, INATIVO, BLOQUEADO	Status da URL que está sendo utilizada na comunicação.
8	contratoStatus	String	10	Domínios: ATIVO, INATIVO, BLOQUEADO	Status do contrato gerado
9	nomeResponsavel	String	50	Sem formatação	Nome do responsável pelo contrato
10	Email	String	40	Sem formatação	Email do responsável
11	telefone	String	20	Sem formatação	Telefone do Responsável

**FALHAS:**

Status	Mensagem	Parâmetros de entrada sujeitos a crítica	Descrição
HTTP_UNAUTHORIZED (401)	Could not find a required Access Token in the request, identified by HEADER x-api-key	Parâmetro: x-api-key	Erros ocasionados por parâmetros obrigatórios não informados ou inválidos.
HTTP_UNAUTHORIZED (401)	UNAUTHORIZED	Parâmetro: Authorization	Erros ocasionados por parâmetros obrigatórios não informados ou inválidos.

<b>HTTP_UNAUTHORIZED (401)</b>	Código de beneficiário diferente do beneficiário do usuário	Parâmetro: linhaDigitavel	Erro ocasionado quando a linha digitável informada é de beneficiário e/ou agência diferente.
<b>HTTP_BAD_REQUEST (400)</b>	“Cooperativa inválida”.	Parâmetro: cooperativa	Erros ocasionados por parâmetros obrigatórios não informados ou inválidos.
<b>HTTP_BAD_REQUEST (400)</b>	“Posto Inválido”	Parâmetro: posto	Erros ocasionados por parâmetros obrigatórios não informados ou inválidos.
<b>HTTP_BAD_REQUEST (400)</b>	“Código do Beneficiário inválido”	Parâmetro: beneficiario	Erros ocasionados por parâmetros obrigatórios não informados ou inválidos.
<b>UNPROCESSABLE_ENTITY (422)</b>	“Nenhum contrato encontrado”		Erros ocasionados caso o beneficiário pesquisado não possua contrato.

REQUEST – GET:

## URL Produção:

<https://api-parceiro.sicredi.com.br/cobranca/boleto/v1/webhook/contratos/?cooperativa=0100&posto=99&beneficiario=12345>

HEADERS:

Authorization  
x-api-key

PARAM:


cooperativa  
posto  
beneficiario

## Execução (via ferramenta Postman)

■ Parâmetros:

Params ● Authorization Headers (8) Body Pre-request Script Tests Settings		
KEY	VALUE	
<input checked="" type="checkbox"/> cooperativa	0101	
<input checked="" type="checkbox"/> posto	19	
<input checked="" type="checkbox"/> beneficiario	12345	
Key	Value	

■ Headers:

Params ● Authorization Headers (10) Body Pre-request Script Tests Settings		
Headers  7 hidden		
Key	Value	
<input checked="" type="checkbox"/> accept	*/*	
<input checked="" type="checkbox"/> Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXZW4iLCJ0eXAiOiJKV1QiLCJkaXIiOiJ1bWVhbnQifQ.eyJ1bWVhbnQifQ.eyJ1bWVhbnQifQ...	
<input checked="" type="checkbox"/> x-api-key	99999999-9999-0000-0000-9999999999	

■ Cenário Positivo:

Params • Authorization Headers (8) Body Pre-request Script Tests Settings

	KEY	VALUE
<input checked="" type="checkbox"/>	cooperativa	0101
<input checked="" type="checkbox"/>	posto	19
<input checked="" type="checkbox"/>	beneficiario	79129

Body Cookies Headers (18) Test Results


Pretty Raw Preview Visualize JSON ↕


```
1 {  
2   "idContrato": "63a35acc90364b703bbe6aad",  
3   "cooperativa": "0101",  
4   "posto": "19",  
5   "codBeneficiario": "79129",  
6   "eventos": [  
7     "LIQUIDACAO"  
8   ],  
9   "url": "https://teste.instituicao.cloud/v1/contratos",  
10  "urlStatus": "ATIVO",  
11  "contratoStatus": "ATIVO",  
12  "nomeResponsavel": "NOME RESPONSAVEL",  
13  "email": "TESTE@EMAIL.COM.BR",  
14  "telefone": "51 999999999"  
15 }
```

■ Cenário Negativo: 400 – BAD\_REQUEST (Não informou código beneficiário)

Params • Authorization Headers (8) Body Pre-request Script Tests Settings

	KEY	VALUE
<input checked="" type="checkbox"/>	cooperativa	0101
<input checked="" type="checkbox"/>	posto	19
<input checked="" type="checkbox"/>	beneficiario	

Body Cookies Headers (16) Test Results  Status: 4


Pretty Raw Preview Visualize JSON 


```
1
2   "status": 400,
3   "error": "Bad Request",
4   "message": "Validation error",
5   "detailedMessages": [
6     "'beneficiario': Código do Beneficiário inválido"
7   ]
8
```


■ Cenário Negativo: 400 – Bad Request (Cooperativa inválida)

Params • Authorization Headers (8) Body Pre-request Script Tests Settings

Query Params

	Key	Value	Description
<input checked="" type="checkbox"/>	cooperativa	85488 	
<input checked="" type="checkbox"/>	posto	19	
<input checked="" type="checkbox"/>	beneficiario	79129	

Body Cookies Headers (16) Test Results  Status: 400 Bad Request T

Pretty Raw Preview Visualize JSON 

```
1
2   "status": 400,
3   "error": "Bad Request",
4   "message": "Validation error",
5   "detailedMessages": [
6     "'cooperativa': Cooperativa inválida"
7   ]
8
```

■ **Cenário Negativo: 401 – UNAUTHORIZED**

Params ● Authorization Headers (8) Body Pre-request Script Tests Settings

Query Params

	KEY	VALUE
<input checked="" type="checkbox"/>	cooperativa	0101
<input checked="" type="checkbox"/>	posto	19
<input checked="" type="checkbox"/>	beneficiario	01015

Body Cookies Headers (16) Test Results

Pretty Raw Preview Visualize JSON ▾ 

```
1  {
2    "timestamp": "2022-12-23T19:39:23.319+00:00",
3    "error": "UNAUTHORIZED",
4    "code": "401",
5    "message": "Código de beneficiário diferente do beneficiário do usuário"
6  }
```

## 12.1. Altera Contrato Webhook

A operação PUT “criar/alterar” é responsável por alterar dados de um contrato Webhook já existente. O formato de entrada e saída é JSON, considerando que a saída sempre é composta da entidade a ser retornada e um código HTTP dentro de um *Response*. O tipo de codificação utilizado no *Response* é Unicode UTF-8. Por padrão o recurso de alterar responde em milissegundos.

### ENTRADA:

Ord	Nome	Tipo	Parametro	Tamanho	Formatação	Descrição	Obrigatório
1	x-api-key	String	Header Param	38	UUID	Access token gerado no portal do desenvolvedor	Obrigatório
2	Authorization	String	Header Param	1413	Bearer + Token de Autenticação	Campo access_token obtido na saída da autenticação.	Obrigatório
3	Content-Type	Application /json	Header param		Sem formatação	Estabelece o tipo de conteúdo da requisição	
4	idContrato	String	Patch	25	Sem formatação	Identificador único do contrato	Obrigatório
5	cooperativa	String	Body	4	Sem formatação	Código da Cooperativa -	Obrigatório
6	posto	String	Body	2	Sem formatação	Código da Agência	Obrigatório
7	codigoBeneficiario	String	Body	5	Sem formatação	Código do Convênio de Cobrança	Obrigatório
8	eventos	List<String>	Body		Domínios: LIQUIDACAO	Eventos enviados via Webhook	Obrigatório
9	url	String	Body	256	Sem formatação	URL de integração entre Sicredi e associado.	Obrigatório



						Obs: A url deve possuir, obrigatoriamente, o protocolo (https)	
10	urlStatus	String	Body	10	Domínios: ATIVO, INATIVO, BLOQUEADO	Status da URL que está sendo utilizada na comunicação.	Obrigatório
11	contratoStatus	String	Body	10	Domínios: ATIVO, INATIVO, BLOQUEADO	Status do contrato gerado	Obrigatório
12	nomeResponsavel	String	Body	50	Sem formatação	Nome do responsável pelo contrato	Opcional
13	email	String	Body	40	Sem formatação	Email do responsável	Opcional
14	telefone	String	Body	20	Sem formatação	Telefone do Responsável	Opcional

**SAÍDA:**

Ord	Nome	Tipo	Tamanho	Formatação	Descrição
1	idContrato	String	25	Sem formatação	Identificador único do contrato.
2	cooperativa	String	4	Sem formatação	Código da Cooperativa do Beneficiário.
3	posto	String	2	Sem formatação	Código da Posto do Beneficiário.
4	codBeneficiario	String	5	Sem formatação	Código do Beneficiário de Cobrança.
5	eventos	List<String>		Domínios: "LIQUIDACAO_PIX", "LIQUIDACAO_COMPE_H5", "LIQUIDACAO_COMPE_H6", "LIQUIDACAO_COMPE_H8", * "LIQUIDACAO_REDE",	Lista de eventos que o associado receberá informações em caso de movimentação.  <b>"LIQUIDACAO_REDE"</b>

				"LIQUIDACAO_CARTORIO" "AVISO_PAGAMENTO_COMPE", "ESTORNO_LIQUIDACAO_REDE"	<b>*Obs:</b> Essa liquidação é a única que pode ter estorno no mesmo dia. Havendo estorno será enviado o evento "ESTORNO_LIQUIDACAO_REDE". Fica a critério do associado acatar de imediato a liquidação ou aguardar o fim do dia para garantir que não ocorreu um estorno.
6	url	String	256	Sem formatação	URL de integração entre Sicredi e associado
7	urlStatus	String	10	Domínios: ATIVO, INATIVO, BLOQUEADO	Status da URL que está sendo utilizada na comunicação.
8	contratoStatus	String	10	Domínios: ATIVO, INATIVO, BLOQUEADO	Status do contrato gerado
9	nomeResponsavel	String	50	Sem formatação	Nome do responsável pelo contrato
10	Email	String	40	Sem formatação	Email do responsável
11	telefone	String	20	Sem formatação	Telefone do Responsável

#### RETORNO (Response):

Status	Descrição
HTTP_ACCEPTED (200)	Operação aceita com sucesso

**FALHAS:**

Status	Mensagem	Parâmetros sujeitos à crítica	Descrição
HTTP_NOT_FOUND (404)	Ocorreu um erro ao realizar operação de cadastro	Não possui	Erro ocasionado por URL inválida.
HTTP_UNSUPPORTED_MEDIA_TYPE (415)	Tipo de conteúdo não suportado	Parâmetro: Content-Type	Erro ocasionado por parâmetro obrigatório não informado ou inválido.
HTTP_UNAUTHORIZED (401)	Unauthorized	Parâmetro: Authorization	Erro ocasionado por parâmetro obrigatório não informado ou inválido.
HTTP_UNAUTHORIZED (401)	Cooperativa diferente da cooperativa do usuário	Parâmetro: cooperativa	Erro ocasionado quando o código de cooperativa informado no cadastro for diferente do contido na autenticação.
HTTP_UNAUTHORIZED (401)	Código de beneficiário diferente do beneficiário do usuário	Parâmetro: codigoBeneficiario	Erro ocasionado quando o código de beneficiário informado no cadastro for diferente do contido na autenticação.
HTTP_UNAUTHORIZED (422)	“Erro interno ao procurar contrato.”	Parâmetro: Pacht: id	Erro ocasionado quando o id de contrato informado corresponde a um id que não existe na base de dados.
HTTP_BAD_REQUEST (400)	Campo obrigatório em branco	Qualquer parâmetro que consta como obrigatório na listagem de parâmetros de entrada pode ser criticado	Erros ocasionados por parâmetros obrigatórios não informados ou inválidos.

HTTP_BAD_REQUEST (400)	“Cooperativa inválida”	Parâmetro: cooperativa	Erros ocasionados por parâmetros obrigatórios não preenchidos ou inválidos.
HTTP_BAD_REQUEST (400)	“Posto inválido”	Parâmetro: posto	Erros ocasionados por parâmetros obrigatórios não preenchidos ou inválidos
HTTP_BAD_REQUEST (400)	“Evento deve conter LIQUIDACAO”	Parâmetro: evento	Erro ocasionado por parâmetro obrigatório não informado ou inválido.
HTTP_BAD_REQUEST (400)	“URL é obrigatória”	Parâmetro: url	Erro ocasionado por parâmetro obrigatório não informado ou inválido.
HTTP_BAD_REQUEST (400)	“URL inválida”	Parâmetro: url	O campo foi preenchido em branco ("" ) ou o tipo de protocolo informado foi diferente de "http" ou "https".
HTTP_BAD_REQUEST (400)	“URL Status não foi informado ou está inválida”	Parâmetro: urlStatus	O parâmetro não foi informado ou foi preenchido com valor inválido.
HTTP_BAD_REQUEST (400)	“O campo URL Status deve ser ATIVO, INATIVO, BLOQUEADO”	Parâmetro: urlStatus	O campo “urlStatus” foi preenchido com um valor diferente dos domínios: ATIVO, INATIVO, BLOQUEADO
HTTP_BAD_REQUEST (400)	“Contrato Status é obrigatório”	Parâmetro: contratoStatus	O campo "contratoStatus" não foi preenchido.
HTTP_BAD_REQUEST (400)	O campo Contrato Status deve ser ATIVO, INATIVO ou BLOQUEADO.	Parâmetro: contratoStatus	O campo “contratoStatus” foi preenchido com um valor diferente dos domínios: ATIVO, INATIVO, BLOQUEADO
HTTP_BAD_REQUEST (400)	“Email inválido”	Parâmetro: email	O campo “email” foi preenchido com um endereço de email inválido. Exemplo: Sem o @.

**REQUEST – GET:**

## URL Produção:

<https://api-parceiro.sicredi.com.br/cobranca/boleto/v1/webhook/contrato/{id}>

### **HEADERS:**

Authorization  
x-api-key  
Content-Type: application/json

### **PATH:**

id

### **REQUEST BODY (JSON):**

```
{
  "cooperativa": "0123",
  "posto": "12",
  "codBeneficiario": "12345",
  "eventos": [
    "LIQUIDACAO"
  ],
  "url": "https://teste.instituicao.cloud/v1/contratos",
  "urlStatus": "ATIVO",
  "contratoStatus": "ATIVO",
  "nomeResponsavel": "NOME RESPONSAVEL",
  "email": "TESTE@EMAIL.COM.BR",
  "telefone": "51 999999999"
}
```

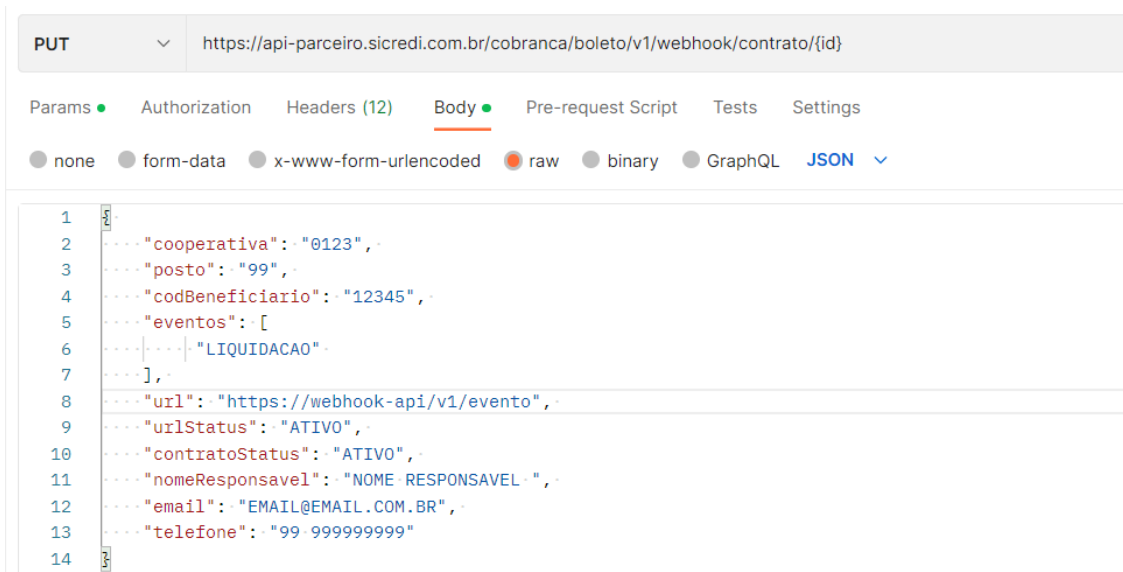
### **RESPONSE BODY (JSON):**

```
{
  "idContrato": "63a241a590364b703bbe6aac",
  "cooperativa": "0123",
  "posto": "12",
  "codBeneficiario": "12345",
  "eventos": [
    "LIQUIDACAO_PIX",
    "LIQUIDACAO_COMPE_H5",
    "LIQUIDACAO_COMPE_H6",
    "LIQUIDACAO_COMPE_H8",
  ]
}
```

```
"LIQUIDACAO_REDE",  
"LIQUIDACAO_CARTORIO",  
"AVISO_PAGAMENTO_COMPE",  
"ESTORNO_LIQUIDACAO_REDE"  
],  
"url": "https://teste.instituicao.cloud/v1/contratos",  
"urlStatus": "ATIVO",  
"contratoStatus": "ATIVO",  
"nomeResponsavel": "NOME RESPONSAVEL",  
"email": "TESTE@EMAIL.COM.BR",  
"telefone": "51 999999999"  
}
```

## Execução (via ferramenta Postman)

### ■ Body Request:



### ■ Headers/Path Param:

PUT ▼ https://api-parceiro.sicredi.com.br/cobranca/boleto/v1/webhook/contrato/{id}

Params ● Authorization Headers (12) ● Body ● Pre-request Script Tests Settings

Headers 👁 9 hidden

	Key	Value	D
<input checked="" type="checkbox"/>	Content-Type	application/json	
<input checked="" type="checkbox"/>	x-api-key	99999999-aaaa-9999-aaaa-999999aa99	
<input checked="" type="checkbox"/>	Authorization	Bearer eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUliwia2lkliA...	

### ■ Cenário Positivo:

Params Authorization Headers (12) **Body** ● Pre-request Script Tests Settings

● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL **JSON** ▼

```
1 {
2   "cooperativa": "0101",
3   "posto": "19",
4   "codBeneficiario": "79129",
5   "eventos": [
6     "LIQUIDACAO"
7   ],
8   "url": "https://teste.instituicao.cloud/v1/contratos",
9   "urlStatus": "ATIVO",
10  "contratoStatus": "ATIVO",
11  "nomeResponsavel": "NOME_RESPONSAVEL",
12  "email": "TESTE@EMAIL.COM.BR",
13  "telefone": "51 999999999"
14 }
```

Body Cookies Headers (16) Test Results 🌐 Status: 201 Created Time: 306 ms Size: 1.02 KB

Pretty Raw Preview Visualize **JSON** ≡

```
1 {
2   "idContrato": "63a241a590364b703bbe6aac",
3   "cooperativa": "0101",
4   "posto": "19",
5   "codBeneficiario": "79129",
6   "eventos": [
7     "LIQUIDACAO_PIX",
8     "LIQUIDACAO_COMPE_H5",
9     "LIQUIDACAO_COMPE_H6",
10    "LIQUIDACAO_COMPE_H8",
11    "LIQUIDACAO_REDE",
12    "LIQUIDACAO_CARTORIO",
13    "AVISO_PAGAMENTO_COMPE",
14    "AVISO_PAGAMENTO_REDE"
15  ]
16 }
```

■ Cenário de erro <Beneficiário não autenticado> :

PUT https://api-parceiro.sicredi.com.br/cobranca/boleto/v1/webhook/contrato/ [redacted]

Params Authorization Headers (12) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded **raw** binary GraphQL JSON

1

Body Cookies Headers (17) Test Results Status: 401 Unauthorized Time: 114 ms

Pretty Raw Preview Visualize JSON

```
1 {
2   "timestamp": "2023-04-11T02:03:26.561+0000",
3   "error": "UNAUTHORIZED",
4   "code": "401",
5   "message": "Código de beneficiário diferente do beneficiário do usuário"
6 }
```



## 13. Glossário

- **Beneficiário:** Empresa ou pessoa associada do Sicredi que contratou o produto Cobrança com a sua cooperativa.
- **Código de Beneficiário / Código de convênio:** Código gerado no momento da contratação do produto Cobrança na sua agência.
- **Número da cooperativa:** É o número da cooperativa a qual pertence a conta vinculada ao convênio de cobrança da pessoa associada. Esse número é o qual junto com o número de conta o associado pode logar no seu internet banking.
- **Número de posto:** É o número da agência a qual pertence a conta vinculada ao convênio de Cobrança da pessoa associada. Você pode encontrar essa informação acessando (ou solicitando à empresa associada que acesse) o internet banking, no menu “Cobrança”, na opção ‘Cadastrar títulos’. Ela encontrará a seguinte informação:

