



Manual do Integração via WebService

SOFTWARE DESENVOLVIDO POR:

- *PELAINTERNET Sistemas de Informação Ltda.*

MANUAL DESENVOLVIDO POR:

- *Jorge Henrique da Silva Naspolini*

VERSÃO:

- *1.0*



Sumário

1. Introdução	3
2. Informações Importantes	4
3. Fluxo de Dados	5
4. Detalhamento da Etapas	6
5. Parâmetros para criar a coleta	7
6. Parâmetros para consulta de tipo de documento.....	9
7. Parâmetro para consultar proposito de assinatura	11
8. Parâmetro para consultar coletas	13
9. Processo de homologação.....	15
Anexo 1 – Mensagem de erro	16



1. Introdução

Para realizar transações de dados entre sistemas, a PELAINETNET Sistemas de Informação LTDA utiliza o modelo de integração WebService (REST/JSON).

A integração via WebService (REST/JSON - HTTPS/POST) traz algumas vantagens que são: facilidade de integração, independência de plataforma, redução de custos em transporte de dados e um formato de transação universal.

Esse documento tem como objetivo especificar e orientar quanto as especificações para transação de dados entre as aplicações, através de solução de integração da PELAINETNET Tecnologia LTDA.

Para iniciar o processo de integração, o desenvolvedor da organização contratante deverá solicitar a PELAINETNET o token de acesso pelo e-mail suporte@pelainetnet.com.br.



2. Informações Importantes

Para que seja realizada a integração é exigido que o sistema utilizado pela contratante seja capaz de interpretar dados em formato JSON, padrão esse enviado pelo servidor da API. O servidor e sistema da contratante também deverá ser capaz de realizar chamadas pelo WebService (REST/JSON) e enviar requisições HTTPS/POST para o servidor da API.

Toda a comunicação realizada pela integração deverá utilizar o protocolo HTTPS/POST para comunicação com o servidor da contratada, sendo a resposta da mesma sempre no padrão JSON.

Para utilizar essa forma de integração é necessário um token que será gerado pela PELAINETNET.

3. Fluxo de Dados

O processo de comunicação e transação de dados é composto por

- Solicitar o tipo de documento;
- Solicitar o propósito de assinatura;
- Criar nova coleta;

Todas as fases em que a API responde à contratada os dados virão no formato JSON, assim sempre serão enviados os mesmos parâmetros em todas as etapas. Esses parâmetros serão descritos em cada especificação.

Seguindo o **FLUXO DE PROCESSO** entenderemos melhor cada etapa do processo:

1. Solicitar o tipo de documento, se faz necessário para poder informa o tipo de documento que será criado na hora da criação da coleta.
2. Solicitar o propósito de assinatura se faz necessário para poder informa o propósito do signatário na hora de criar a coleta.
3. Criar nova coleta, na primeira etapa a aplicação poderá ser acessada através da solicitação “Criar nova coleta” onde será realizado o início da integração.
4. **Integração, requisição e resposta:** Na segunda etapa serão encaminhados os dados mais token via Webservice para que ocorra a integração e respostas entre as aplicações.

4. Detalhamento da Etapas

Abaixo, segue a legenda para as tabelas de parâmetros utilizados na integração:

N: Numérico

O: Obrigatório

AN: Alfanumérico

OP: Opcional

T: Texto

ND: Não determinado

\$intNome Variável utilizada em PHP, muda para cada linguagem de programação escolhida para seu projeto.

ESCOPO Cria Coleta	
Protocolo / Método	HTTPS/POST
Endereço Homologação	https://wshomolog.assinepelainternet.com.br/rest.php?class=WSColetas2&method=novaColeta
Endereço Produção	https://ws.assinepelainternet.com.br/rest.php?class=WSColetas2&method=novaColeta
Formato de resposta	JSON

ESCOPO Consulta Tipo de Documentos	
Protocolo / Método	HTTPS/POST
Endereço Homologação	https://wshomolog.assinepelainternet.com.br/rest.php?class=WSTipoDocumento2&method=listar
Endereço Produção	https://ws.assinepelainternet.com.br/rest.php?class=WSTipoDocumento2&method=listar
Formato de resposta	JSON

Escopo Consulta Propósito Assinatura	
Protocolo / Método	HTTPS/POST
Endereço Homologação	https://ws.assinepelainternet.com.br/rest.php?class=WSPropositos2&method=listar
Endereço Produção	https://ws.assinepelainternet.com.br/rest.php?class=WSPropositos2&method=listar
Formato de resposta	JSON

Escopo Consulta Documento Assinado	
Protocolo / Método	HTTPS/POST
Endereço Homologação	https://wshomolog.assinapelainternet.com.br/rest.php?class=WSDocumentoAssinado2&method=documentoAssinado
Endereço Produção	https://ws.assinapelainternet.com.br/rest.php?class=WSDocumentoAssinado2&method=documentoAssinado
Formato de resposta	JSON

5. Parâmetros para criar a coleta

1. Esse Webservice cria a coleta tanto digital quanto eletrônica no portal de assinatura, os parâmetros deve seguir a tabela a seguir e enviado via JSON. Exemplo solicitar Collection do Postman.

2. Retorno Webservice API

Primeiro: parâmetro atributo "sucesso" que retorna TRUE caso tudo estiver correto, ou FALSE (ANEXO 2 - MENSAGENS DE ERRO) caso algo dê errado:

```
$sucesso = $objeto->sucesso;
```

```
$mensagem = $objeto->mensagem;
```

Segundo: parâmetro de atributo "mensagem" que retorna o que foi solicitado

```
$mensagem = $objeto->mensagem;
```

Caso o \$sucesso seja FALSE, retornará uma mensagem de acordo com o erro

Exemplo:

```
if($sucesso){
    echo $mensagem; // mensagem de sucesso
}else{ echo $mensagem; } // mensagem de erro
```

PARÂMETROS PARA CRIAR NOVA COLETA					
Parâmetro	Descrição	Obrigatoriedade	Formato	Tamanho	Exemplo
class	Classe para criação da coleta pelo WS	O	A	18	"WSColetaEletronica" (Fixo)
method	Método que vai criar a coleta pelos WS	O	A	5	"criar" (Fixo)
versao	versão do webservice	O	N	1	"1" (Fixo)
token	string com o token de acesso da empresa cliente fornecida pela API	O	AN	ND	\$token = "20384sT%3065";
cnpj	String com o cnpj da empresa contratante do portal	O	N	14	\$cnpj = " 12345678000190"
cpf	string com o cpf do usuário (sem máscara), já deve estar cadastrado no portal	O	N	11	\$cpf = "85641682004";
tipoAssinatura	Tipo de assinatura que vai ser feita nessa coleta	O	N	1	Para coleta Digital = 0 Para coleta Eletronica = 1
nomeColeta	nome da coleta	O	AN	50	\$nomeColeta = "Coleta 01";
dataVigencia	Data da vigência da coleta	O	AN (dd/mm/aaaa)	10	\$DataVigencia = '01/10/2018';
dataLimite	Data limite de assinatura da coleta	O	AN (dd/mm/aaaa)	10	\$ DataLimite = '20/10/2018';
documentos	array com os dados dos documentos que serão assinados contendo 3 parâmetros: 1- nomeDocumento = string com o nome do documento 2- conteudoDocumento = conteúdo / binário do documento em base64_encode 3- tipoDocumento = int com o tipo de documento	O	AN	ND	\$ListaDocumentos = array(); \$ListaDocumentos [] = array("nomeDocumento" => \$nomeDocumento, "conteudoDocumento" => base64_encode(\$conteudoDocumento), "tipoDocumento" => \$tipoDocumento); Verificar WebService para consultar tipo de documento
listaCPFSignatarios	array com os dados dos signatários contendo 8 parâmetros: 1 – versão: versão do ws de cadastro de signatários fixo 2 2 – cpf: CPF do signatário que vai assinar a coleta 3 – nome: nome do Signatários que vai assinar a coleta 4 – email: e-mail do Signatário que vai assinar a coleta 5 – fone: Telefone do Signatário que vai assinar a coleta 6 – dtasc: data de nascimento do Signatário que vai assinar a coleta. 7 – Ordemassinatura: Ordem que o Signatário vai assinar a coleta. 8 – proposito : Propósito que o Signatário vai assinar a coleta	O	AN	ND	listaCPFSignatarios": [{ "versao": "2", "cpf": "85641682004", "nome": "Nome Completo", "email": "email@email.com.br", "fone": "48987654321", "dtnasc": "00/00/0000", "ordemAssinatura": "", "proposito": "1" }]



6. Parâmetros para consulta de tipo de documento

1. Esse Webservice consulta os tipos de documentos ativos no portal de assinatura, os parâmetros deve seguir a tabela e enviado via JSON. Exemplo solicitar Collection do Postman.

2. Retorno WebService API:

Primeiro: parâmetro atributo "sucesso" que retorna TRUE caso tudo estiver correto, ou FALSE (ANEXO 1 - MENSAGENS DE ERRO) caso algo dê errado:

```
$sucesso = $objeto->sucesso;
```

```
$mensagem = $objeto->mensagem;
```

Segundo: parâmetro de atributo "mensagem" que retorna o que foi solicitado

```
$mensagem = $objeto->mensagem;
```

```
$mensagem = Será dado retorno em JSON conforme exemplo.
```

Caso \$sucesso seja FALSE, retornará uma mensagem de acordo com o erro.

PARÂMETROS PARA CONSULTAR TIPO DE DOCUMENTO

Parâmetro	Descrição	Obrigatoriedade	Formato	Tamanho	Exemplo
versao	versão da consulta a ser realizada	O	N	1	Esse valor deverá ser '1' pois é a versão correspondente a essa consulta
token	string com o token de acesso da empresa cliente fornecida pela API	O	AN	ND	\$token = "20384sT%3065";



7. Parâmetro para consultar propósito de assinatura

1. Esse Webservice consulta os propósitos de assinatura no portal de assinatura, os parâmetros deve seguir a tabela e enviado via JSON. Exemplo solicitar Collection do Postman.

2. Retorno WebService API:

Primeiro: parâmetro atributo "sucesso" que retorna TRUE caso tudo estiver correto, ou FALSE (ANEXO 1 - MENSAGENS DE ERRO) caso algo dê errado:

```
$sucesso = $objeto->sucesso;
```

```
$mensagem = $objeto->mensagem;
```

Segundo: parâmetro de atributo "mensagem" que retorna o que foi solicitado

```
$mensagem = $objeto->mensagem;
```

\$mensagem = será dado retorno em JSON conforme exemplo.

Caso \$sucesso seja FALSE, retornará uma mensagem de acordo com o erro

PARÂMETROS PARA CONSULTAR PROPÓSITO DE ASSINATURA					
Parâmetro	Descrição	Obrigatoriedade	Formato	Tamanho	Exemplo
versao	versão da consulta a ser realizada	O	N	1	Esse valor deverá ser '1' pois é a versão correspondente a essa consulta
token	string com o token de acesso da empresa cliente fornecida pela API	O	AN	ND	\$token = "20384sT%3065";



8. Parâmetro para consultar coletas

1. Esse Webservice consulta as coletas criadas no portal de assinatura, retornando seus status, os parâmetros deve seguir a tabela e enviado via JSON. Exemplo solicitar Collection do Postman.

2. Retorno WebService API:

Primeiro: parâmetro atributo "sucesso" que retorna TRUE caso tudo estiver correto, ou FALSE (ANEXO 1 - MENSAGENS DE ERRO) caso algo dê errado:

```
$sucesso = $objeto->sucesso;
```

```
$mensagem = $objeto->mensagem;
```

Segundo: parâmetro de atributo "mensagem" que retorna o que foi solicitado

```
$mensagem = $objeto->mensagem;
```

\$mensagem = será dado retorno em JSON conforme exemplo.

Caso \$sucesso seja FALSE, retornará uma mensagem de acordo com o erro

PARÂMETROS PARA CONSULTAR PROPÓSITO DE ASSINATURA					
Parâmetro	Descrição	Obrigatoriedade	Formato	Tamanho	Exemplo
versao	versão da consulta a ser realizada	O	N	1	Esse valor deverá ser '1' pois é a versão correspondente a essa consulta
token	string com o token de acesso da empresa cliente fornecida pela API	O	AN	ND	\$token = "20384sT%3065";
coletaId	Array com as coletas que quer saber o status	OP	NA	ND	\$ColetasId['1','200','350','5']

9. Processo de homologação

Para iniciar as operações em ambiente de produção, o desenvolvedor deverá contatar o suporte técnico da API e realizar essa solicitação, o mesmo receberá um documento para informar alguns dados da contratante. Será realizada uma bateria de testes antes da migração dos ambientes. Após aprovação em ambiente de homologação a API notificará a liberação do ambiente de produção e, a partir da liberação da mesma o ambiente de homologação ficará indisponível automaticamente.

Caso seja necessária alguma alteração no cadastro da contratante o desenvolvedor deverá notificar a API através do suporte técnico para que as modificações sejam realizadas.

SUPORTE TÉCNICO

Em caso de dúvidas a respeito do desenvolvimento a API disponibiliza os seguintes canais para suporte:

E-mail: suporte@pelainternet.com.br
Fone: (48) 3045-6900

Anexo 1 – Mensagem de erro

Listagem de erros previsíveis no WS de criação de coleta.

WSColeta2

1. Erro ao comunicar com o WS para criar uma coleta
2. O nome da coleção deve ter pelo menos 4 caracteres
3. Data limite inválida!
4. Data Vigência inválida!
5. Nenhum documento incluído!
6. Nenhum signatário incluído!
7. Nenhum CPF de criador incluído!
8. CPF do criador da coleta não encontrado!
9. Nome do documento inválido!
10. Hash base64 do documento inválido!
11. Conteúdo original do documento inválido!
12. Tipo inválido de arquivo!
13. Tipo de documento inexistente!
14. Nenhum tipo de documento ativo encontrado!
15. Ocorreu um erro ao carimbar o documento!
16. CPF do signatário inválido.
17. CPF do signatário contém um tamanho inválido!
18. CPF do signatário não encontrado!
19. Propósito inexistente.
20. Nenhum propósito ativo encontrado.
21. Tipo de documento inexistente.
22. É necessário incluir um documento com o CPF do signatário sendo o nome do arquivo!
23. Ocorreu um problema ao notificar os signatários: " MSG do erro de envio"

Lista de erros previsíveis no WS de Proposito de Assinatura

WSProposito2

- Ocorreu um erro ao obter a requisição!
- Token necessário!
- Este Token expirou. Entre em contato com o suporte!
- Token invalido!
- Cliente não encontrado!

Lista de erro previsíveis no WS de Tipo de Documento

WSTipoDocumento2

- Ocorreu um erro ao obter a requisição!
- Token necessário!
- Este Token expirou. Entre em contato com o suporte!
- Cliente não encontrado!

WSDocumentoAssinado2

- Ocorreu um erro ao obter a requisição
- Cliente não encontrado
- Token necessário!