



solucionare

Logística Jurídica

Documentação de integração com a WebAPI de Discovery Full

Versão 1.0

Sumário

1. Informações Gerais	4
1.1- Dados de acesso a WebAPI	4
1.2 – Tecnologia da WebAPI.....	4
1.3 – Endereço da documentação técnica online da WebAPI	4
1.4 – Possíveis status de uma pesquisa.....	4
1.5 – Tipos de parâmetros de consulta	4
1.6 – Fluxo	5
2 - Métodos Disponíveis na WebAPI	6
2.1 - Método de cadastro de pesquisa de pessoa física / Jurídica (cadastraPesquisa_PFPJ). ..	6
2.1.1 - Parâmetros de requisição:	6
2.1.2 – Respostas do método:.....	7
2.2 – Método de cadastro de pesquisa de Advogados (cadastraPesquisa_Adv).....	7
2.2.1 - Parâmetros de requisição:	7
2.2.2 – Respostas do método:.....	8
2.3 – Método de cadastro de pesquisa de Assuntos / Tópicos (cadastraPesquisa_Assuntos). ..	8
2.3.1 - Parâmetros de requisição:	8
2.3.2 – Respostas do método:.....	9
2.4 - Método de cadastro de pesquisa por números de processo (cadastraPesquisa_NumProcessos).	9
2.4.1 - Parâmetros de requisição:	9
2.4.2 – Respostas do método:.....	10
2.5 - Método de cadastro de pesquisa preditiva (cadastraPesquisa_Preditiva).....	10
2.5.1 - Parâmetros de requisição:	10
2.5.2 – Respostas do método:.....	10
2.6 – Método para buscar o status das pesquisas cadastradas (buscaStatusPesquisa)	10
2.6.1 - Parâmetros de requisição:	11
2.6.2 – Respostas do método:.....	11
2.7 – Método para buscar o resultado da pesquisa (buscaDadosResultadoPesquisa).....	11
2.7.1 - Parâmetros de requisição:	11
2.7.2 – Respostas do método:.....	12
2.8 – Método para buscar o resultado da pesquisa count (buscaDadosResultadoPesquisa_Count)	13
2.8.1 - Parâmetros de requisição:	13
2.8.2 – Respostas do método:.....	13

2.9 – Método para buscar os movimentos da pesquisa (buscaDadosMovimentoPesquisa) .	14
2.9.1 - Parâmetros de requisição:	14
2.9.2 – Respostas do método:.....	14
2.10 – Método para buscar as publicações da pesquisa (buscaDadosPublicacaoPesquisa) ..	15
2.10.1 - Parâmetros de requisição:	15
2.10.2 – Respostas do método:.....	15
2.11 - Método para confirmar o recebimento das publicações dos processos encontrados na pesquisa (confirmaRecebimentoPublicacaoPesquisa).....	16
2.11.1 - Parâmetros de requisição:	16
2.11.2 – Respostas do método:.....	16
2.12 – Método para buscar o status dos documentos do processo (buscaStatusDocumentoProcesso)	16
2.12.1 - Parâmetros de requisição:	16
2.12.2 – Respostas do método:.....	17
2.13 – Método para buscar os documentos iniciais da pesquisa (buscaDadosDocIniciaisPesquisa)	17
2.13.1 - Parâmetros de requisição:	17
2.13.2 – Respostas do método:.....	17
2.14 - Método para confirmar o recebimento dos documentos dos processos encontrados na pesquisa (confirmaRecebimentoDocIniciaisPesquisa).....	18
2.14.1 - Parâmetros de requisição:	18
2.14.2 – Respostas do método:.....	18
2.15 – Método para buscar o status das pesquisas preditivas cadastradas (buscaStatusPesquisa_Preditiva)	19
2.15.1 - Parâmetros de requisição:	19
2.15.2 – Respostas do método:.....	19
2.16 – Método para buscar o resultado da pesquisa preditiva (buscaDadosResultadoPesquisa_Preditiva)	19
2.16.1 - Parâmetros de requisição:	19
2.16.2 – Respostas do método:.....	20
2.17 – Método para gerar o token e consumir demais métodos da api (autenticaAPI).....	20
2.17.1 - Parâmetros de requisição:	20
2.17.2 – Respostas do método:.....	21

1. Informações Gerais

1.1- Dados de acesso a WebAPI

- Link de acesso:
<http://online.solucionarelj.com.br:9090/WebApiDiscoveryFull>

1.2 – Tecnologia da WebAPI

A API é restfull, ou seja, permite ser consumida tanto em json quanto em xml.

1.3 – Endereço da documentação técnica online da WebAPI

Existe uma documentação com mais exemplos voltada para desenvolvedores que irão consumir a WebAPI.

- Link de acesso:
<http://online.solucionarelj.com.br:9090/WebApiDiscoveryFull/Help>

1.4 – Possíveis status de uma pesquisa

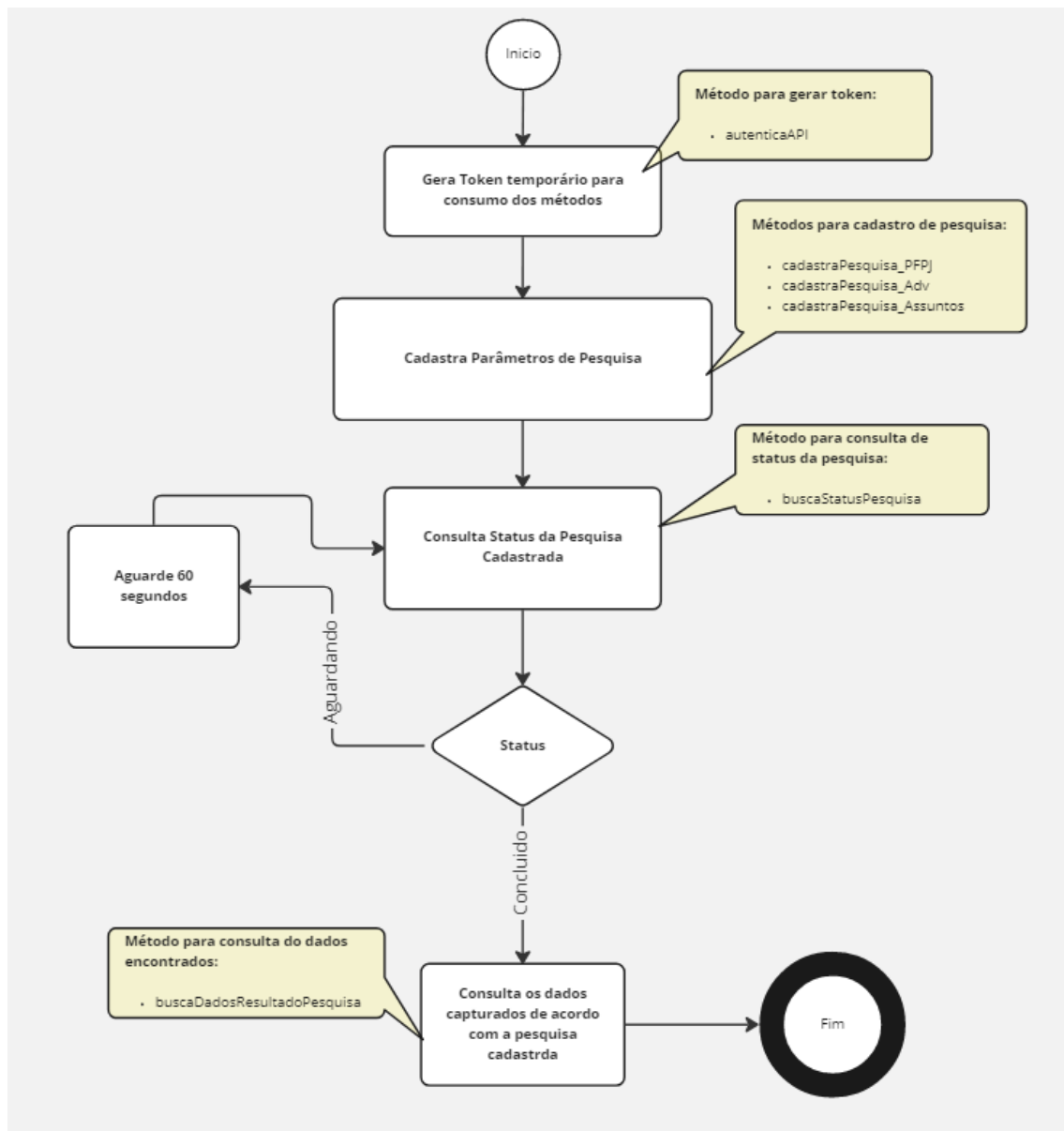
- **1 - Aguardando** – A pesquisa solicitada está em execução e ainda não temos um resultado para ser entregue;
- **2 - Concluído** – A pesquisa solicitada está concluída e os dados já estão prontos para serem entregues;

1.5 – Tipos de parâmetros de consulta.

- **URI** – Parâmetros a ser enviado na URL
- **BODY** – Parâmetro a ser enviado via JSON ou XML

1.6 – Fluxo

- Solicitar uma nova consulta



2 - Métodos Disponíveis na WebAPI

Método	Descrição
cadastraPesquisa_PFPJ	Cadastra uma pesquisa de pessoa física / jurídica
cadastraPesquisa_Adv	Cadastra uma pesquisa de advogados
cadastraPesquisa_Assuntos	Cadastra uma pesquisa de assuntos
cadastraPesquisa_NumProcessos	Cadastra uma pesquisa por números de processo
cadastraPesquisa_Preditiva	Cadastra uma pesquisa preditiva
buscaStatusPesquisa	Busca status da pesquisa cadastrada
buscaDadosResultadoPesquisa	Busca resultado da pesquisa cadastrada
buscaDadosResultadoPesquisa_Count	Busca resultado da pesquisa count cadastrada
buscaDadosMovimentoPesquisa	Busca os andamentos dos processos encontrados na pesquisa
buscaDadosPublicacaoPesquisa	Busca as publicações dos processos encontrados na pesquisa
confirmaRecebimentoPublicacaoPesquisa	Confirma o recebimento das publicações dos processos encontrados na pesquisa
buscaStatusDocumentoProcesso	Busca status dos documentos do processo
buscaDadosDocIniciaisPesquisa	Busca os documentos iniciais dos processos encontrados na pesquisa
confirmaRecebimentoDocIniciaisPesquisa	Confirma o recebimento dos documentos dos processos encontrados na pesquisa
buscaStatusPesquisa_Preditiva	Busca status da pesquisa preditiva cadastrada
buscaDadosResultadoPesquisa_Preditiva	Busca resultado da pesquisa preditiva cadastrada
autenticaAPI	Gera token para consumir demais métodos

2.1 - Método de cadastro de pesquisa de pessoa física / Jurídica (cadastraPesquisa_PFPJ).

Este método permite o cadastro de uma pesquisa por pessoa física / Jurídica.

2.1.1 - Parâmetros de requisição:

Para utilizar o método deverão ser passados os seguintes parâmetros:

HEADER

- **token JWT – Obrigatório – Exemplo:** Authorization: tokenJWT

BODY

- **nome(varchar(500)) – Obrigatório** – Nome que será utilizado durante a pesquisa.
- **buscaParcial(boolean) – Obrigatório** – Para consultas dos termos de forma parcial enviar valor verdadeiro e para consultas dos termos de forma exata enviar valor falso.
- **documento(varchar(50)) - Opcional** – Documento CPF/CNPJ do nome cadastrado.
- **apenasProcessosAtivos(boolean) – Obrigatório** – Para consultas filtrando apenas os processos ativos enviar valor verdadeiro e para consultas normal enviar valor falso.
- **buscarApenasCount(boolean) – Obrigatório** – Para consultas dos termos do tipo contador enviar valor verdadeiro e para consultas dos termos do tipo normal enviar valor falso.

- **assuntos(Array(varchar(1000))) - Opcional** – Assuntos que serão utilizados como filtro na pesquisa dos processos.
Obs.: Pode ser enviado um ou vários.
- **instancia(int) - Obrigatório** – Instância em que deseja realizar a consulta conforme descrito abaixo:
1 – Primeira Instância
2 – Segunda Instância
3 – Instâncias Superiores
4 – Todas as Instâncias acima.
- **abrangencias(Array(varchar(10))) - Obrigatório** – Abrangências da pesquisa.
Obs.: Pode ser enviado um ou vários.
- **dataDistribuicaoProcessoApartirDe (DateTime) - Opcional** – Preenchimento com a data inicial da pesquisa, ou seja, irá trazer processos daquela data em diante.
- **entregarPublicacoes (boolean) - Opcional** – Para consultas de publicações enviar valor verdadeiro, caso não enviado esse parâmetro, não serão retornadas publicações.
- **limiteRetorno(int) - Opcional** – Quantidade máxima de processos que serão retornados na pesquisa, caso não enviado esse parâmetro, serão retornados todos os dados.
Obs.: Só serão aceitos números múltiplos de 1000.
Exemplo: 1000, 2000, 10000 e etc.

2.1.2 – Respostas do método:

Ao cadastrar uma pesquisa será retornado os seguintes dados:

- **codPesquisa(int)** – Código para acompanhamento do status da pesquisa cadastrada.

Obs.: Qualquer retorno que não seja de sucesso, tratado ou não, será retornado uma exceção. (Recomendamos usar o Try Catch capturar a mensagem de erro).

2.2 – Método de cadastro de pesquisa de Advogados (cadastraPesquisa_Adv)

Este método permite o cadastro de uma pesquisa por advogados.

2.2.1 - Parâmetros de requisição:

Para utilizar o método deverão ser passados os seguintes parâmetros:

HEADER

- **token JWT – Obrigatório – Exemplo:** Authorization: tokenJWT

BODY

- **nome(varchar(500)) – Obrigatório** – Nome que será utilizado durante a pesquisa.
- **numOab(int) - Opcional** – Número da OAB do advogado.
- **ufOab(varchar(2)) - Opcional** – UF da OAB do Advogado.
- **apenasProcessosAtivos(boolean) – Obrigatório** – Para consultas filtrando apenas os processos ativos enviar valor verdadeiro e para consultas normal enviar valor falso.
- **buscarApenasCount(boolean) – Obrigatório** – Para consultas dos termos do tipo contador enviar valor verdadeiro e para consultas dos termos do tipo normal enviar valor falso.

- **assuntos(Array(varchar(1000))) - Opcional** – Assuntos que serão utilizados como filtro na pesquisa dos processos.
Obs.: Pode ser enviado um ou vários
- **instancia(int) - Obrigatório** – Instância em que deseja realizar a consulta conforme descrito abaixo:
 - 1 – Primeira Instância
 - 2 – Segunda Instância
 - 3 – Instâncias Superiores
 - 4 – Todas as Instâncias acima.
- **abrangencias(Array(varchar(10))) - Obrigatório** – Abrangências da pesquisa.
Obs.: Pode ser enviado um ou vários.
- **dataDistribuicaoProcessoApartirDe (DateTime) - Opcional** – Preenchimento com a data inicial da pesquisa, ou seja, irá trazer processos daquela data em diante.
- **entregarPublicacoes (boolean) - Opcional** – Para consultas de publicações enviar valor verdadeiro, caso não enviado esse parâmetro, não serão retornadas publicações.
- **limiteRetorno(int) - Opcional** – Quantidade máxima de processos que serão retornados na pesquisa, caso não enviado esse parâmetro, serão retornados todos os dados.
Obs.: Só serão aceitos números múltiplos de 1000.
Exemplo: 1000, 2000, 10000 e etc.

2.2.2 – Respostas do método:

Ao cadastrar uma pesquisa será retornado os seguintes dados:

- **codPesquisa(int)** – Código para acompanhamento do status da pesquisa cadastrada.

Obs.: Qualquer retorno que não seja de sucesso, tratado ou não, será retornado uma exceção. (Recomendamos usar o Try Catch capturar a mensagem de erro).

2.3 – Método de cadastro de pesquisa de Assuntos / Tópicos (cadastraPesquisa_Assuntos)

Este método permite o cadastro de uma pesquisa por assuntos / Tópicos.

2.3.1 - Parâmetros de requisição:

Para utilizar o método deverão ser passados os seguintes parâmetros:

HEADER

- **token JWT – Obrigatório – Exemplo:** Authorization: tokenJWT

BODY

- **assuntos(Array(varchar(1000))) - Obrigatório** – Assuntos que serão utilizados como filtro na pesquisa dos processos.
Obs.: Pode ser enviado um ou vários
- **assuntoTramiteProcesso(boolean) – Obrigatório** – Será pesquisado esses assuntos como parte do conteúdo do processo caso venha verdadeiro esse parâmetro.
- **apenasProcessosAtivos(boolean) – Obrigatório** – Para consultas filtrando apenas os processos ativos enviar valor verdadeiro e para consultas normal enviar valor falso.
- **buscarApenasCount(boolean) – Obrigatório** – Para consultas dos termos do tipo contador enviar valor verdadeiro e para consultas dos termos do tipo normal enviar valor falso.

- **instancia(int) - Obrigatório** – Instância em que deseja realizar a consulta conforme descrito abaixo:
1 – Primeira Instância.
2 – Segunda Instância.
3 – Instâncias Superiores.
4 – Todas as Instâncias acima.
- **abrangencias(Array(varchar(10))) - Obrigatório** – Abrangências da pesquisa.
Obs.: Pode ser enviado um ou vários.
- **dataDistribuicaoProcessoApartirDe (DateTime) - Opcional** – Preenchimento com a data inicial da pesquisa, ou seja, irá trazer processos daquela data em diante.
- **entregarPublicacoes (boolean) - Opcional** – Para consultas de publicações enviar valor verdadeiro, caso não enviado esse parâmetro, não serão retornadas publicações.
- **limiteRetorno(int) - Opcional** – Quantidade máxima de processos que serão retornados na pesquisa, caso não enviado esse parâmetro, serão retornados todos os dados.
Obs.: Só serão aceitos números múltiplos de 1000.
Exemplo: 1000, 2000, 10000 e etc.

2.3.2 – Respostas do método:

Ao cadastrar uma pesquisa será retornado os seguintes dados:

- **codPesquisa(int)** – Código para acompanhamento do status da pesquisa cadastrada.

Obs.: Qualquer retorno que não seja de sucesso, tratado ou não, será retornado uma exceção. (Recomendamos usar o Try Catch capturar a mensagem de erro).

2.4 - Método de cadastro de pesquisa por números de processo (cadastraPesquisa_NumProcessos).

Este método permite o cadastro de uma pesquisa por números de processo.

2.4.1 - Parâmetros de requisição:

Para utilizar o método deverão ser passados os seguintes parâmetros:

HEADER

- **token JWT – Obrigatório – Exemplo:** Authorization: tokenJWT

BODY

- **instancia(int) - Obrigatório** – Instância em que deseja realizar a consulta conforme descrito abaixo:
1 – Primeira Instância
2 – Segunda Instância
3 – Instâncias Superiores
4 – Todas as Instâncias acima.
- **listaNumProcessos(Array(varchar(1000))) - Obrigatório** – Números de processo que serão utilizados como filtro na pesquisa dos processos.
Obs.: Pode ser enviado um ou vários.
- **entregarPublicacoes (boolean) - Opcional** – Para consultas de publicações enviar valor verdadeiro, caso não enviado esse parâmetro, não serão retornadas publicações.
- **entregarDocIniciais (boolean) - Opcional** – Para busca de documentos iniciais enviar valor verdadeiro, caso não enviado esse parâmetro, não serão retornados documentos.

- **abrangencias(Array(varchar(10)))** - **Obrigatório** – Números de da pesquisa.
Obs.: Pode ser enviado um ou até 5.000 números de processos.

2.4.2 – Respostas do método:

Ao cadastrar uma pesquisa será retornado os seguintes dados:

- **codPesquisa(int)** – Código para acompanhamento do status da pesquisa cadastrada.

Obs.: Qualquer retorno que não seja de sucesso, tratado ou não, será retornado uma exceção. (Recomendamos usar o Try Catch capturar a mensagem de erro).

2.5 - Método de cadastro de pesquisa preditiva (cadastraPesquisa_Preditiva).

Este método permite o cadastro de uma pesquisa preditiva.

2.5.1 - Parâmetros de requisição:

Para utilizar o método deverão ser passados os seguintes parâmetros:

HEADER

- **token JWT – Obrigatório – Exemplo:** Authorization: tokenJWT

BODY

- **listProcessos(Array(ListaNumProcesso))** - **Obrigatório** – Objeto contendo número de processo e instância que serão utilizados como filtro na pesquisa preditiva.

ListaNumProcesso			
Campo	Descrição	Tipo	Tamanho
instancia	1 – Primeira Instância	Int	None
	2 – Segunda Instância		
	3 – Instâncias Superiores		
	4 – Todas as Instâncias acima		
numeroProcesso	Número do processo	Varchar	(50)

Obs.: Pode ser enviado um ou vários.

2.5.2 – Respostas do método:

Ao cadastrar uma pesquisa preditiva será retornado os seguintes dados:

- **codPesquisa(int)** – Código para acompanhamento do status da pesquisa cadastrada.

Obs.: Qualquer retorno que não seja de sucesso, tratado ou não, será retornado uma exceção. (Recomendamos usar o Try Catch capturar a mensagem de erro).

2.6 – Método para buscar o status das pesquisas cadastradas (buscaStatusPesquisa)

Retorna o status das pesquisas cadastradas anteriormente.

2.6.1 - Parâmetros de requisição:

Para utilizar o método deverão ser passados os seguintes parâmetros:

HEADER

- **token JWT – Obrigatório – Exemplo:** Authorization: tokenJWT

BODY

- **codPesquisa(Array (int) – Obrigatório** – Código da pesquisa retornado nos métodos de cadastro.
Obs.: Pode ser enviado um ou vários.

2.6.2 – Respostas do método:

Ao cadastrar uma pesquisa será retornado os seguintes dados:

- **codPesquisa(int)** – Código para acompanhamento do status da pesquisa cadastrada.
- **codStatus(int)** – Retorna o status atual da pesquisa cadastrada.
- **descricao varchar(200))** - Retorna à descrição do status da pesquisa cadastrada.
- **codStatusMovimento(int)** – Retorna o status atual dos movimentos da pesquisa cadastrada.
- **descricaoMovimento varchar(200))** - Retorna à descrição do status dos movimentos da pesquisa cadastrada.
- **codStatusPublicacao(int)** – Retorna o status atual das publicações da pesquisa cadastrada.
- **descricaoPublicacao varchar(200))** - Retorna à descrição do status das publicações da pesquisa cadastrada.
- **codStatusDocIniciais(int)** – Retorna o status atual dos documentos da pesquisa cadastrada.
- **descricaoDocIniciais varchar(200))** - Retorna à descrição do status dos documentos da pesquisa cadastrada.

Possíveis valores para todos os status:

- 1 – Aguardando;
- 2 – Concluído;
- 9 – Não requisitado;

Obs.: Qualquer retorno que não seja de sucesso, tratado ou não, será retornado uma exceção. (Recomendamos usar o Try Catch capturar a mensagem de erro).

2.7 – Método para buscar o resultado da pesquisa (buscaDadosResultadoPesquisa)

Este método retorna todos os dados e processos encontrados na pesquisa solicitada.

2.7.1 - Parâmetros de requisição:

Para utilizar o método deverão ser passados os seguintes parâmetros:

HEADER

- **token JWT – Obrigatório – Exemplo:** Authorization: tokenJWT

BODY

- **codPesquisa(int) – Obrigatório** – Código da pesquisa retornado nos métodos de cadastro.

2.7.2 – Respostas do método:

Ao executar o método será retornado uma lista com os dados encontrados na pesquisa solicitada:

Campo	Descrição	Tipo	Tamanho	Pode Nulo
codProcesso	Identificador do processo	Int	None.	Não
numeroProcessoFormatado	Número CNJ com formatação	Varchar	50	Nao
instancia	Instancia	Int	None.	Não
valorCausa	Valor da Causa	Double	None.	Sim
assuntos	Assuntos do processo	Lista do tipo varchar(250)	None.	Sim
capaProcesso	Capa do processo	Objeto tipo capaProcesso	None.	Não
partes	Partes do processo	Lista do tipo partes	None.	Sim
advogados	Advogados do processo	Lista do tipo advogados	None.	Sim

capaProcesso

Campo	Descrição	Tipo	Tamanho
siglaTribunal	Sigla do Tribunal	Varchar	50
relator	Relator	Varchar	250
dataDistribuicao	Data de Distribuição	dateTime	None.
dataAutuacao	Data de Autuação	dateTime	None.
orgaoJulgador	Órgão Julgador	Varchar	250
classeCnj	Classe CNJ do processo	Varchar	200
segmento	Segmento	Varchar	Max
uf	Estado	Varchar	2
unidadeOrigem	Unidade de Origem	Varchar	Max
statusProcesso	Status do Processo	Varchar	Max
dataArquivamento	Data do Arquivamento	datetime	None.
ramoDireito	Ramo do Direito	Varchar	Max

partes

Campo	Descrição	Tipo	Tamanho
codParte	Identificador da parte	Int	None.
tipo	Tipo de parte	Varchar	50

polo	Polo	Varchar	50
nome	Nome	Varchar	500
cnpj	CNPJ da parte	Varchar	20
cpf	CPF da parte	Varchar	20

advogados

Campo	Descrição	Tipo	Tamanho
codParte	Identificador da parte	Int	None.
nome	Nome do Advogado	Varchar	50
ufOab	UF da OAB	Varchar	2
numeroOAB	Número da OAB	Int	None.
polo	Polo do Advogado	varchar	

2.8 – Método para buscar o resultado da pesquisa count (buscaDadosResultadoPesquisa_Count)

Este método retorna todos os dados e processos encontrados na pesquisa do tipo count solicitada.

2.8.1 - Parâmetros de requisição:

Para utilizar o método deverão ser passados os seguintes parâmetros:

HEADER

- **token JWT – Obrigatório – Exemplo:** Authorization: tokenJWT

BODY

- **codPesquisa(int) – Obrigatório** – Código da pesquisa count retornado nos métodos de cadastro.

2.8.2 – Respostas do método:

Ao executar o método será retornado uma lista com os dados encontrados na pesquisa count solicitada:

Campo	Descrição	Tipo	Tamanho	Pode Nulo
codProcesso	Identificador do processo	Int	None.	Não
quantidade	Quantidade de processos	Int	None	Nao
totalPorTribunal	Total por tribunal	Objeto tipo totalPorTribunal	None.	Não
totalPorJustica	Total por justiça	Lista do tipo totalPorJustica	None.	Sim
totalPorano	Total por ano	Lista do tipo totalPorAno	None.	Sim

totalPorTribunal

Campo	Descrição	Tipo	Tamanho
-------	-----------	------	---------

siglaTribunal	Sigla do Tribunal	Varchar	Max
quantidade	Quantidade	Int	None

totalPorJustica

Campo	Descrição	Tipo	Tamanho
justica	Justiça	Varchar	Max
quantidade	Quantidade	Int	None

totalPorAno

Campo	Descrição	Tipo	Tamanho
ano	Ano	Int	None
quantidade	Quantidade	Int	None

2.9 – Método para buscar os movimentos da pesquisa (buscaDadosMovimentoPesquisa)

Este método retorna todos os andamentos dos processos encontrados na pesquisa solicitada.

2.9.1 - Parâmetros de requisição:

Para utilizar o método deverão ser passados os seguintes parâmetros:

HEADER

- **token JWT – Obrigatório – Exemplo:** Authorization: tokenJWT

BODY

- **codPesquisa(int) – Obrigatório** – Código da pesquisa retornado nos métodos de cadastro.

2.9.2 – Respostas do método:

Ao executar o método será retornado uma lista com todos os andamentos dos processos encontrados na pesquisa solicitada:

Campo	Descrição	Tipo	Tamanho	Pode Nulo
codProcesso	Identificador do processo	Int	None.	Não
DataAndamento	Data do andamento	dateTime	none	Sim
ConteudoAndamento	Conteúdo do andamento	Varchar	Max.	Não

Obs.: Qualquer retorno que não seja de sucesso, tratado ou não, será retornado uma exceção. (Recomendamos usar o Try Catch capturar a mensagem de erro).

2.10 – Método para buscar as publicações da pesquisa (buscaDadosPublicacaoPesquisa)

Este método retorna as publicações dos processos encontrados na pesquisa solicitada.

2.10.1 - Parâmetros de requisição:

Para utilizar o método deverão ser passados os seguintes parâmetros:

HEADER

- **token JWT – Obrigatório – Exemplo:** Authorization: tokenJWT

BODY

- **codPesquisa(int) – Obrigatório** – Código da pesquisa retornado nos métodos de cadastro.

2.10.2 – Respostas do método:

Ao executar o método será retornado uma lista com as publicações dos processos encontrados na pesquisa solicitada:

Obs.: Esse método retornar no máximo 1000 publicações, ou seja, caso possua mais que 1000 publicações para receber, o método deverá ser consumido mais de uma vez, após, confirmar o recebimento das publicações usando o método (confirmaRecebimentoPublicacaoPesquisa)

O valor máximo das publicações pode ser alterado, indicamos que siga o fluxograma para que não haja erros na integração, caso o valor seja alterado.

Campo	Descrição	Tipo	Tamanho	Pode Nulo
codPublicacoesProcessos	Identificador da publicação do processo	Int	None.	Não
codPesquisa	Código da pesquisa	Int	None	Nao
codProcesso	Identificador do processo	Int	None.	Não
numProcesso	Número CNJ com formatação	Varchar	50.	Não
dataDisponibilizacao	Data da disponibilização	dateTime	None.	Não
dataPublicacao	Data da publicação	dateTime	None.	Não
nomeDiario	Nome do Diário	Varchar	300.	Sim
orgaoCompleto	Nome completo do órgão	Varchar	400.	Sim
conteudoComplementar	Conteúdo complementar	Varchar	Max	Sim

conteudo	Conteúdo	Varchar	Max	Sim
----------	----------	---------	-----	-----

Obs.: Qualquer retorno que não seja de sucesso, tratado ou não, será retornado uma exceção. (Recomendamos usar o Try Catch capturar a mensagem de erro).

2.11 - Método para confirmar o recebimento das publicações dos processos encontrados na pesquisa (confirmaRecebimentoPublicacaoPesquisa).

Este método confirma o recebimento das publicações dos processos encontrados na pesquisa. Com isso, as publicações confirmadas não serão mais enviadas.

2.11.1 - Parâmetros de requisição:

Para utilizar o método deverão ser passados os seguintes parâmetros:

HEADER

- **token JWT – Obrigatório – Exemplo:** Authorization: tokenJWT

BODY

- **codPesquisa(int) – Obrigatório** – Código da pesquisa retornado nos métodos de cadastro.
- **codPublicaçõesProcessos(Array (int)) – Obrigatório** – Código da publicação recebida e armazenada.
Obs.: Pode ser enviado um ou vários.

2.11.2 – Respostas do método:

Ao confirmar o recebimento de uma publicação será:

- **true**

Obs.: Qualquer retorno que não seja de sucesso, tratado ou não, será retornado uma exceção. (Recomendamos usar o Try Catch capturar a mensagem de erro).

2.12 – Método para buscar o status dos documentos do processo (buscaStatusDocumentoProcesso)

Retorna o status dos documentos dos processos.

2.12.1 - Parâmetros de requisição:

Para utilizar o método deverão ser passados os seguintes parâmetros:

HEADER

- **token JWT – Obrigatório – Exemplo:** Authorization: tokenJWT

BODY

- **codPesquisa(Array (int) – Obrigatório** – Código da pesquisa retornado nos métodos de cadastro.
Obs.: Pode ser enviado um ou vários.

2.12.2 – Respostas do método:

Ao cadastrar uma pesquisa será retornado os seguintes dados:

- **codPesquisa(int)** – Código para acompanhamento do status da pesquisa cadastrada.
- **codProcesso(int)** – Código identificador do processo.
- **codStatus(int)** – Retorna o status atual do documento do processo.
- **descricaoStatus(varchar(200))** - Retorna à descrição do status do documento do processo.
- **detalhamentoStatus(varchar(2000))** – Retorna o detalhamento do status do documento do processo.

Possíveis valores para todos os status:

- 1 – Aguardando;
- 2 – Concluído;
- 9 – Processo não encontrado;

Obs.: Qualquer retorno que não seja de sucesso, tratado ou não, será retornado uma exceção. (Recomendamos usar o Try Catch capturar a mensagem de erro).

2.13 – Método para buscar os documentos iniciais da pesquisa (buscaDadosDocIniciaisPesquisa)

Este método retorna os documentos iniciais dos processos encontrados na pesquisa solicitada.

2.13.1 - Parâmetros de requisição:

Para utilizar o método deverão ser passados os seguintes parâmetros:

HEADER

- **token JWT – Obrigatório – Exemplo:** Authorization: tokenJWT

BODY

- **codPesquisa(int) – Obrigatório** – Código da pesquisa retornado nos métodos de cadastro.

2.13.2 – Respostas do método:

Ao executar o método será retornada uma lista com os documentos iniciais dos processos encontrados na pesquisa solicitada:

Obs.: Esse método retornar no máximo 1000 documentos, ou seja, caso possua mais que 1000 documentos para receber, o método deverá ser consumido mais de uma vez, após, confirmar o recebimento dos documentos usando o método (confirmaRecebimentoDocIniciaisPesquisa).

O valor máximo dos documentos pode ser alterado, indicamos que siga o fluxograma para que não haja erros na integração, caso o valor seja alterado.

Campo	Descrição	Tipo	Tamanho	Pode Nulo
codDocIniciais	Identificador do documento do processo	Int	None.	Não
codPesquisa	Código da pesquisa	Int	None	Nao
codProcesso	Identificador do processo	Int	None.	Não
linkDocumentosIniciais	Link de download do documento	Varchar	Max.	Não

Obs.: Qualquer retorno que não seja de sucesso, tratado ou não, será retornado uma exceção. (Recomendamos usar o Try Catch capturar a mensagem de erro).

2.14 - Método para confirmar o recebimento dos documentos dos processos encontrados na pesquisa (**confirmaRecebimentoDocIniciaisPesquisa**).

Este método confirma o recebimento dos documentos iniciais dos processos encontrados na pesquisa. Com isso, os documentos confirmadas não serão mais enviados.

2.14.1 - Parâmetros de requisição:

Para utilizar o método deverão ser passados os seguintes parâmetros:

HEADER

- **token JWT – Obrigatório – Exemplo:** Authorization: tokenJWT

BODY

- **codPesquisa(int) – Obrigatório** – Código da pesquisa retornado nos métodos de cadastro.
- **codDocIniciais (Array (int)) – Obrigatório** – Código do documento recebido e armazenado.
Obs.: Pode ser enviado um ou vários.

2.14.2 – Respostas do método:

Ao confirmar o recebimento de um documento será:

- **true**

Obs.: Qualquer retorno que não seja de sucesso, tratado ou não, será retornado uma exceção. (Recomendamos usar o Try Catch capturar a mensagem de erro).

2.15 – Método para buscar o status das pesquisas preditivas cadastradas (buscaStatusPesquisa_Preditiva)

Retorna o status das pesquisas preditivas cadastradas anteriormente.

2.15.1 - Parâmetros de requisição:

Para utilizar o método deverão ser passados os seguintes parâmetros:

HEADER

- **token JWT – Obrigatório – Exemplo:** Authorization: tokenJWT

BODY

- **codPesquisa(Array (int) – Obrigatório** – Código da pesquisa preditiva retornado nos métodos de cadastro.
Obs.: Pode ser enviado um ou vários.

2.15.2 – Respostas do método:

Ao buscar o status de uma pesquisa será retornado os seguintes dados:

- **codPesquisa(int)** – Código para acompanhamento do status da pesquisa cadastrada.
- **codStatus(int)** – Retorna o status atual da pesquisa cadastrada.
- **descricao(varchar(200))** - Retorna à descrição do status da pesquisa cadastrada.

Possíveis valores para todos os status:

- 1 – Aguardando;
- 2 – Concluído;
- 9 – Não requisitado;

Obs.: Qualquer retorno que não seja de sucesso, tratado ou não, será retornado uma exceção. (Recomendamos usar o Try Catch capturar a mensagem de erro).

2.16 – Método para buscar o resultado da pesquisa preditiva (buscaDadosResultadoPesquisa_Preditiva)

Este método retorna todos os dados e processos encontrados na pesquisa preditiva solicitada.

2.16.1 - Parâmetros de requisição:

Para utilizar o método deverão ser passados os seguintes parâmetros:

HEADER

- **token JWT – Obrigatório – Exemplo:** Authorization: tokenJWT

BODY

- **codPesquisa(int) – Obrigatório** – Código da pesquisa preditiva retornado nos métodos de cadastro.

2.16.2 – Respostas do método:

Ao executar o método será retornada uma lista com os dados encontrados na pesquisa preditiva solicitada:

Campo	Descrição	Tipo	Tamanho	Pode Nulo
codProcesso	Identificador do processo	Int	None.	Não
numeroProcesso	Número CNJ	Varchar	50	Nao
instancia	Instancia	Int	None.	Não
dadosPreditivo	Dados preditivo	Objeto tipo dadosPreditivo	None.	Não
dadosProcessoEncontrado	Informa se o processo foi encontrado	Boolean	None.	Não

dadosPreditivo

Campo	Descrição	Tipo	Tamanho
tribunal	Tribunal	Varchar	Max
orgaoJulgador	Órgão Julgador	Varchar	Max
classeCnj	Classe CNJ do processo	Varchar	Max
assuntos	Assuntos do processo	Lista do tipo varchar	250
listJulgamentos	Dados do julgamento	Lista do tipo listJulgamentos	None
mediaValorCausa	Média do valor da causa	Double	Max
mediaValorExecucao	Média do valor execução	Double	None
mediaDiasTransitoJulgado	Média de dias trânsito julgado	Double	None
qtdProcessosAnalisados	Quantidade de processos analisado	Int	None

listJulgamento

Campo	Descrição	Tipo	Tamanho
codJulgamento	Identificador do julgamento	Int	None
tipoDecisao	Tipo da decisão	Varchar	500
mediaDiasAteDecisao	Média dias até decisão	Double	None
Probabilidade	Probabilidade	Double	None

2.17 – Método para gerar o token e consumir demais métodos da api (autenticaAPI)

Este método retorna token JWT.

2.17.1 - Parâmetros de requisição:

Para utilizar o método deverão ser passados os seguintes parâmetros:

BODY

- **nomeRelacional(varchar(100)) – Obrigatório** – Nome do perfil do cliente junto a Solucionare
- **token(varchar(50)) – Obrigatório** – Senha padrão para autenticação no WebService(deve ser solicitado caso ainda não tenha).

2.17.2 – Respostas do método:

Ao consumir o método será retornado os seguintes dados:

- **(string)** – token JWT

Obs.: Qualquer retorno que não seja de sucesso, tratado ou não, será retornado uma exceção. (Recomendamos usar o Try Catch capturar a mensagem de erro).