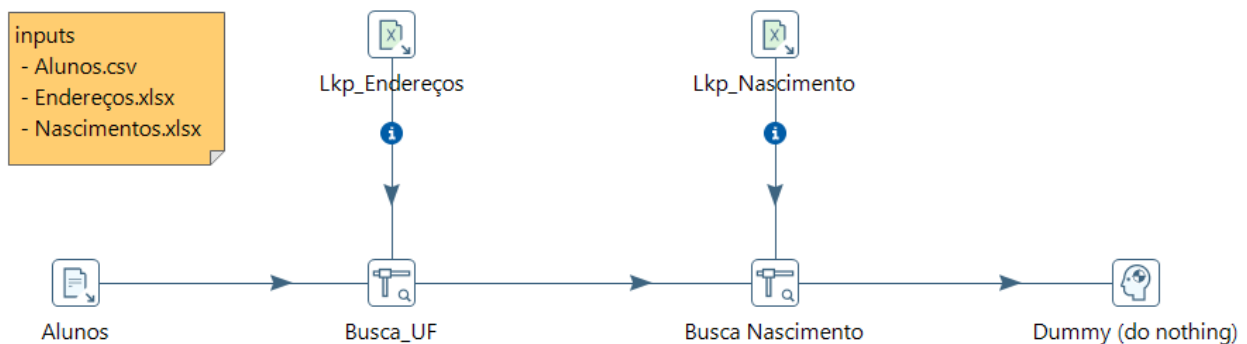


## Exercício 07 – Stream Lookup

**7.1** Crie uma nova transformação, adicionando os inputs “alunos.csv”, “endereços.xlsx” e “nascimentos.xlsx”. Em seguida, adicione o step “Stream lookup” para busca de endereços e outro step “Stream lookup” para busca de data de nascimento, direcionando o resultado para um step “Dummy”. Salve e execute.



## 7.2 Configuração da Lookup “Busca\_UF”:

Stream lookup

Step name:

Lookup step:

The key(s) to look up the value(s):

#	Field	LookupField
1	id	id

Specify the fields to retrieve:

#	Field	New name	Default	Type
1	Estado			None

Preserve memory (costs CPU) ☒

Key and value are exactly one integer field ☐

Use sorted list (i.s.o. hashtable) ☐

Buttons: Help, OK, Cancel, Get Fields, Get lookup fields

### 7.3 Configuração da Lookup “Busca\_Nascimento”:

Stream lookup

Step name:

Lookup step:

The key(s) to look up the value(s):

#	Field	LookupField
1	id	id

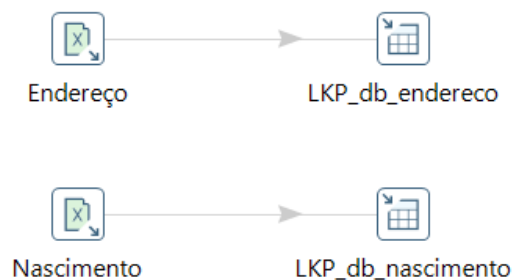
Specify the fields to retrieve :

#	Field	New name	Default	Type
1	data_nascimento			Date

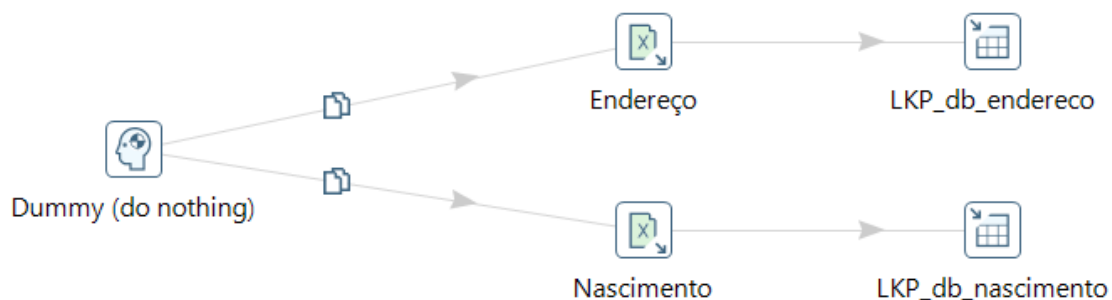
☒ Preserve memory (costs CPU)  
☐ Key and value are exactly one integer field  
☐ Use sorted list (i.s.o. hashtable)

## Exercício 08 – Database lookup

**8.1** Crie uma nova transformação, adicionando os inputs “endereços.xlsx” e “nascimentos.xlsx” e os grave em duas tabelas no Postgres (labs.lkp\_db\_endereco e labs.lkp\_db\_nascimento).



**8.2** Após execução com sucesso, adicione um step “Dummy” ao início, desabilitando todos os HOPS:



**8.3** CNa mesma transformação, adicione um novo fluxo tendo como entrada o arquivo “alunos.csv”. Em série, adicione dois steps “Database lookup”, um para buscar endereços e outro para buscar idade. Configure as lookups e grave a saída em um step “Dummy”:



**8.4** Configuração lookupu “Busca Endereco”:

Database lookup

Step name:

Connection:

Lookup schema:

Lookup table:

Enable cache? ☐

Cache size in rows (0=cache):

Load all data from table ☐

The key(s) to look up the value(s):

#	Table field	Comparator	Field1	Field2
1	id	=	id	

Values to return from the lookup table:

#	Field	New name	Default	Type
1	estado			None

Do not pass the row if the lookup fails ☐

Fail on multiple results? ☐

Order by:

## 8.5 Configuração lookup “Busca Idade”:

Database lookup

Step name:

Connection:

Lookup schema:

Lookup table:

Enable cache? ☐

Cache size in rows (0=cache):

Load all data from table ☐

The key(s) to look up the value(s):

#	Table field	Comparator	Field1	Field2
1	id	=	id	

Values to return from the lookup table:

#	Field	New name	Default	Type
1	idade			None

Do not pass the row if the lookup fails ☐

Fail on multiple results? ☐

Order by:

## 8.6 Salve, execute e observe evolução e performance da transformação na aba “Execution Results”:

Execution Results													
<input type="button" value="Logging"/> <input type="button" value="Execution History"/> <input type="button" value="Step Metrics"/> <input type="button" value="Performance Graph"/> <input type="button" value="Metrics"/> <input type="button" value="Preview data"/>													
#	Stepname	Copynr	Read	Written	Input	Output	Updated	Rejected	Errors	Active	Time	Speed (r/s)	input/output
1	Alunos	0	0	16306	16308	0	0	0	0	Running	5.4s	3,035	0/9900
2	Busca_Endereco	0	6406	6405	6405	0	0	0	0	Running	5.4s	1,192	9900/101
3	Busca_Idade	0	6304	6303	6303	0	0	0	0	Running	5.4s	1,173	101/100
4	Fim	0	6203	6203	0	0	0	0	0	Running	5.4s	1,155	100/0

## Exercício 09 – Rest client com API JSON

**9.1** Crie uma nova transformação, adicionando o step “Generate rows”, e o preencha conforme descrito abaixo:

Opção Limit : 1

Campo Name : url

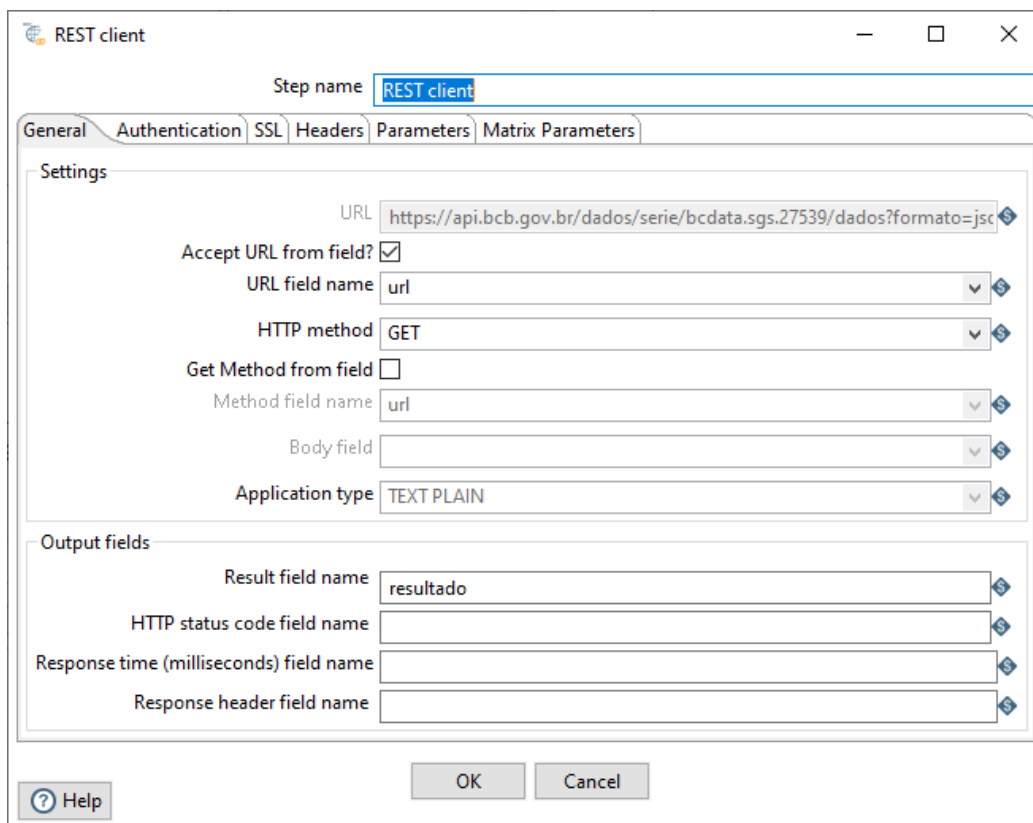
Campo Type : String

Campo Value : <https://api.bcb.gov.br/dados/serie/bcdata.sgs.27539/dados?formato=json>

Campo Set Empty String : N

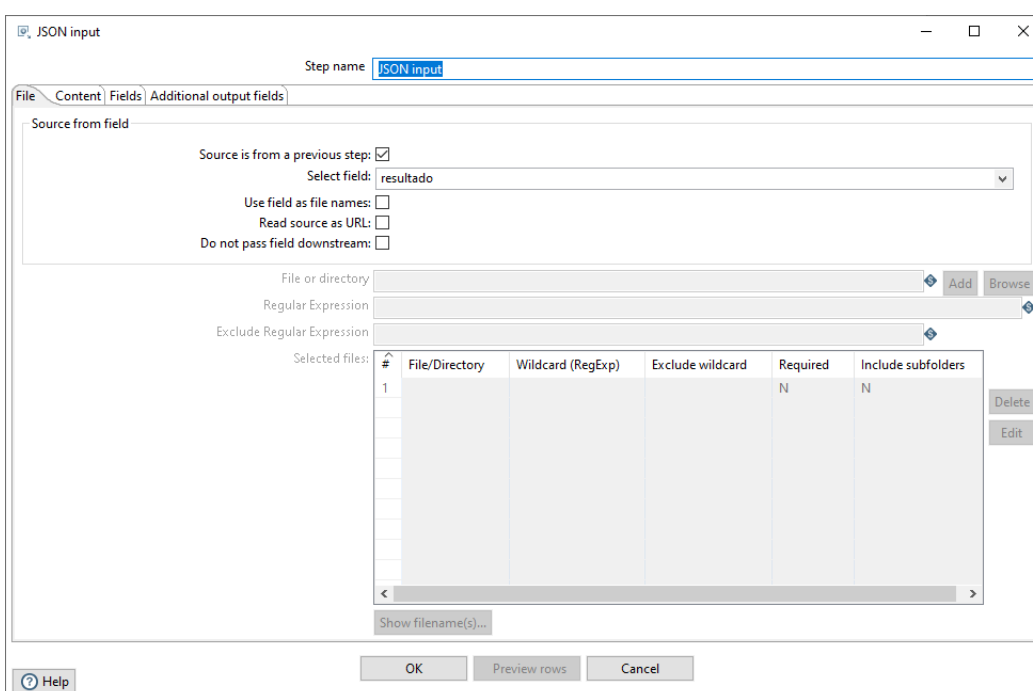
[illegible]

**9.2** Na saída do step “Generate rows”, adicione um step “Rest client” e o configure conforme a seguir:



The screenshot shows the 'REST client' configuration window. The 'Step name' is 'REST client'. The 'General' tab is selected. Under 'Settings', the 'URL' is 'https://api.bcb.gov.br/dados/serie/bcdata.sgs.27539/dados?formato=json', 'Accept URL from field?' is checked, 'URL field name' is 'url', 'HTTP method' is 'GET', 'Get Method from field' is unchecked, 'Method field name' is 'url', 'Body field' is empty, and 'Application type' is 'TEXT PLAIN'. Under 'Output fields', 'Result field name' is 'resultado', and the other three fields are empty. At the bottom are 'Help', 'OK', and 'Cancel' buttons.

**9.3** Na saída do step “Rest client”, adicione o step “JSON input” e o configure conforme a seguir na aba “File”:

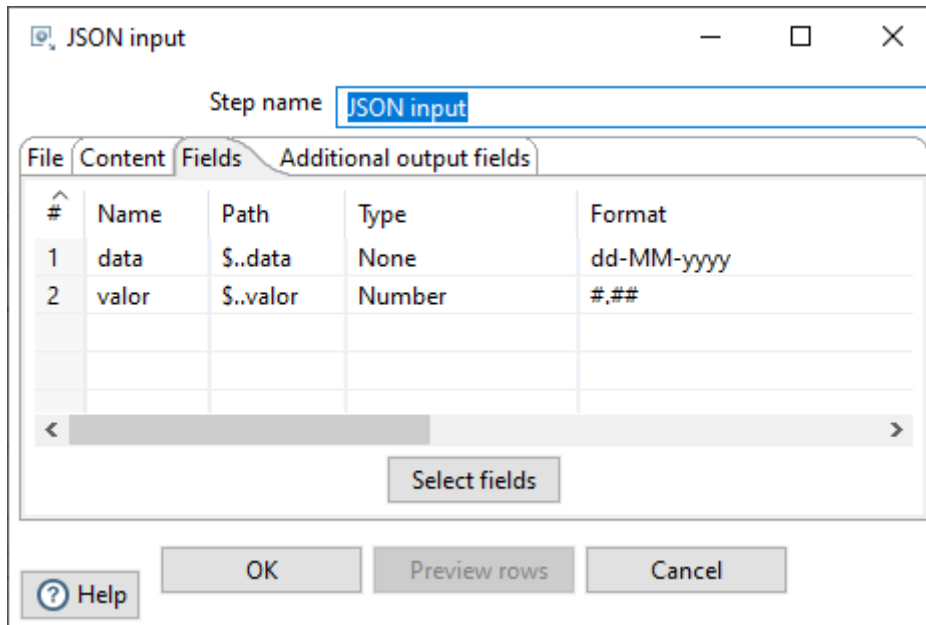


The screenshot shows the 'JSON input' configuration window. The 'Step name' is 'JSON input'. The 'File' tab is selected. Under 'Source from field', 'Source is from a previous step:' is checked, 'Select field:' is 'resultado', 'Use field as file names:' is unchecked, 'Read source as URL:' is unchecked, and 'Do not pass field downstream:' is unchecked. Below, there are fields for 'File or directory', 'Regular Expression', and 'Exclude Regular Expression'. A table titled 'Selected files' is shown with one row:

#	File/Directory	Wildcard (RegExp)	Exclude wildcard	Required	Include subfolders
1				N	N

At the bottom are 'Help', 'OK', 'Preview rows', and 'Cancel' buttons.

**9.4** Mude para a aba “Fields” e proceda com a configuração a seguir:



JSON input

Step name:

File Content **Fields** Additional output fields

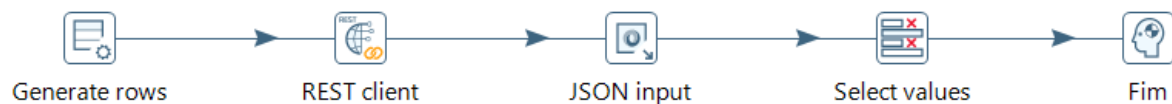
#	Name	Path	Type	Format
1	data	\$..data	None	dd-MM-yyyy
2	valor	\$..valor	Number	#,##

<  >

Select fields

Help OK Preview rows Cancel

**9.5** Remova os campos desnecessários (step “Select values”) e grave a saída em um step do tipo “Dummy”.



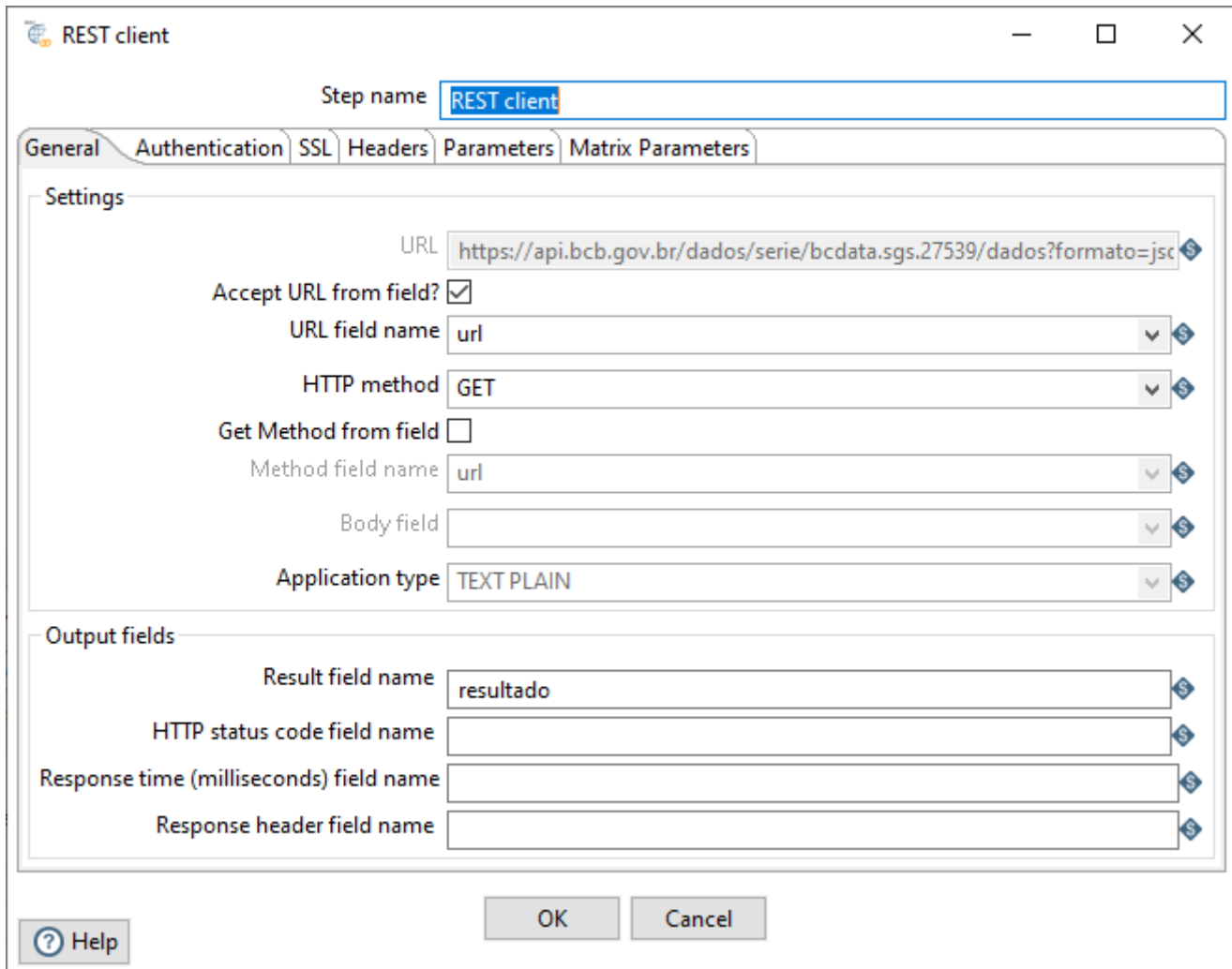


## Exercício 10 – Rest client com API XML

**10.1** Crie uma nova transformação, chamada “10 – Busca API Rest com XML”. Adicione o step “Generate Rows”, gere apenas 1 linha com o valor indicado abaixo:

Value: <https://www.camara.leg.br/sitcamaraws/deputados.asmx/ObterDeputados>

**10.2** Adicione o step “REST Client” e faça um HOP entre a saída do “Generate Rows” e a entrada deste novo step. Configure a aba “General” tal qual indicado:



REST client

Step name: REST client

General | Authentication | SSL | Headers | Parameters | Matrix Parameters

Settings

URL: <https://api.bcb.gov.br/dados/serie/bcdata.sgs.27539/dados?formato=json>

Accept URL from field? ☒

URL field name: url

HTTP method: GET

Get Method from field? ☐

Method field name: url

Body field:

Application type: TEXT PLAIN

Output fields

Result field name: resultado

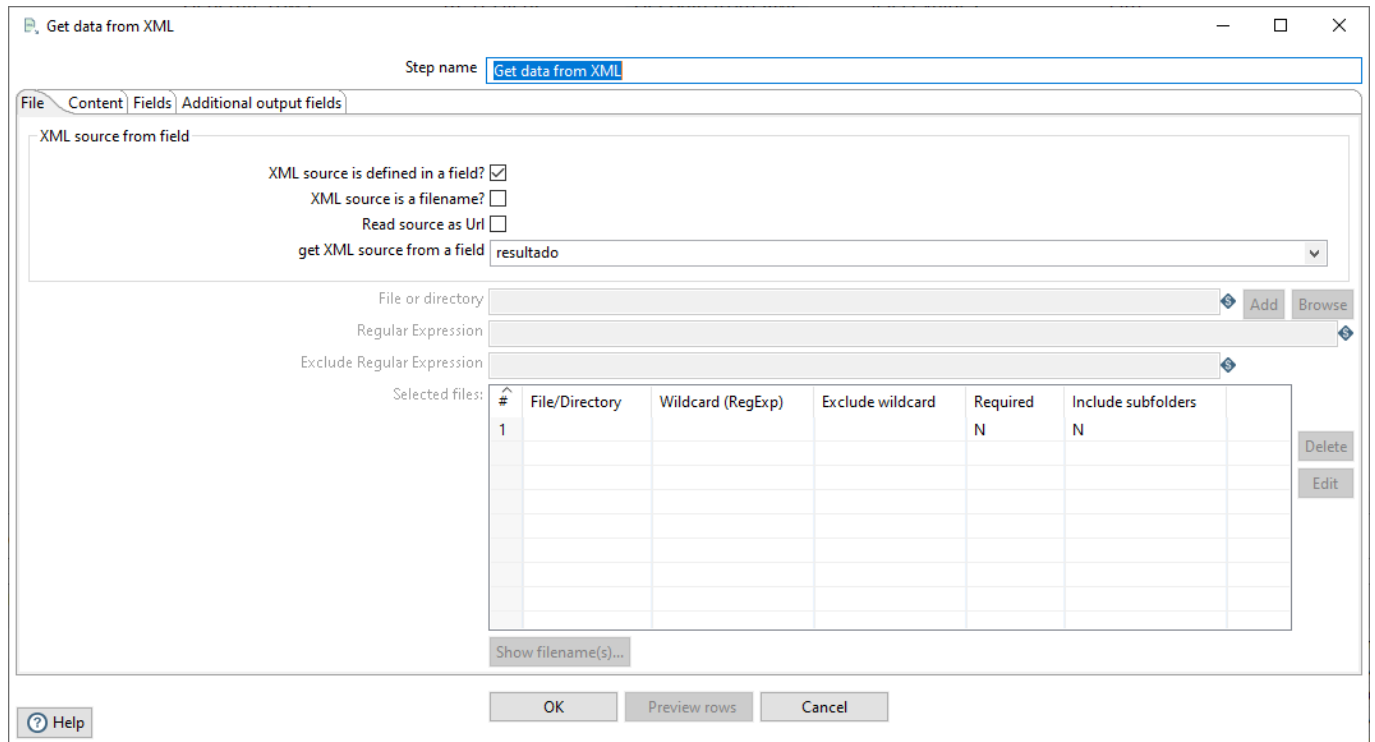
HTTP status code field name:

Response time (milliseconds) field name:

Response header field name:

Help OK Cancel

**10.3** Adicione o step “Get data from XML”, e conecte à saída do step “Rest client”. Na aba “File”, indique que a origem dos dados vem de um campo do step anterior e selecione este campo:



Step name: Get data from XML

File | Content | Fields | Additional output fields

XML source from field

XML source is defined in a field? ☒

XML source is a filename? ☐

Read source as Uri ☐

get XML source from a field: resultado

File or directory: [text field] Add Browse

Regular Expression: [text field]

Exclude Regular Expression: [text field]

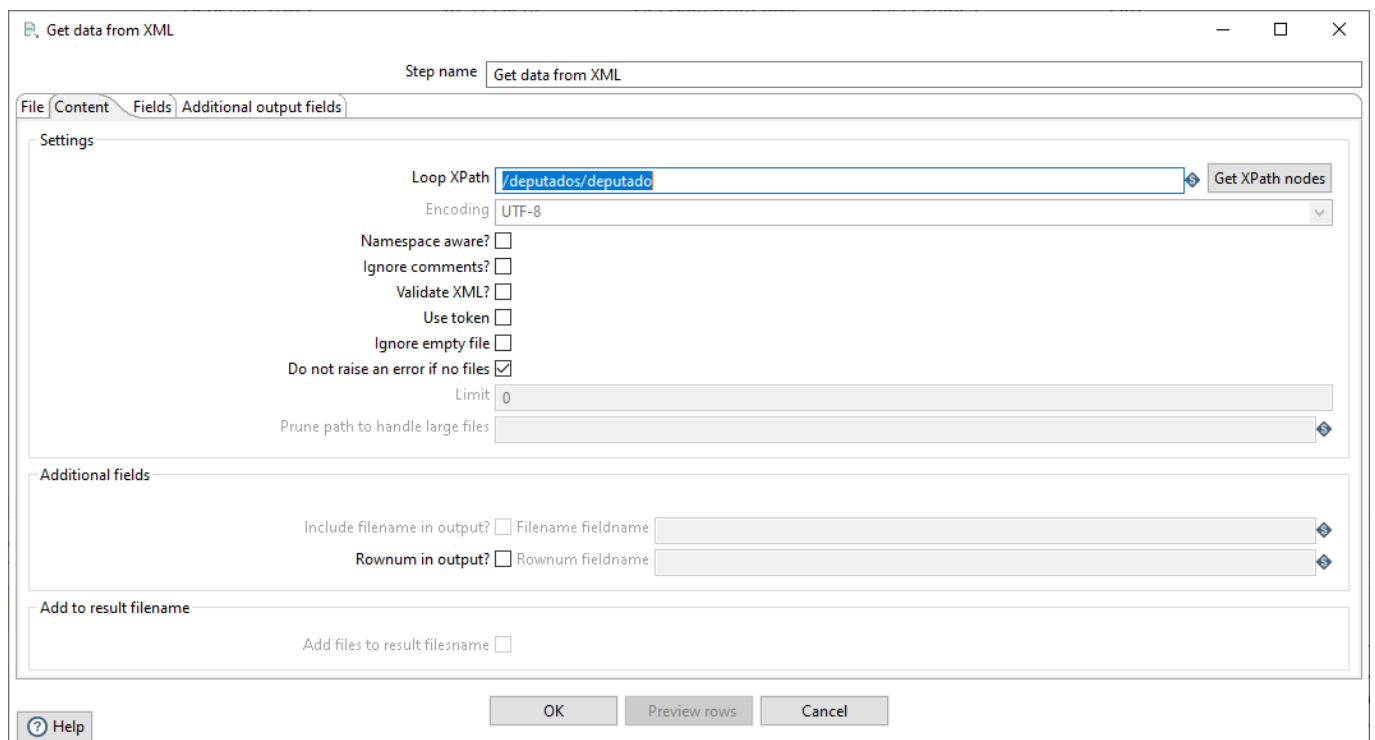
Selected files:

#	File/Directory	Wildcard (RegExp)	Exclude wildcard	Required	Include subfolders
1				N	N

Show filename(s)...

Help OK Preview rows Cancel

**10.4** Na aba “Content”, insira manualmente o XPath “/deputados/deputado”:



Step name: Get data from XML

File | Content | Fields | Additional output fields

Settings

Loop XPath: /deputados/deputado Get XPath nodes

Encoding: UTF-8

Namespace aware? ☐

Ignore comments? ☐

Validate XML? ☐

Use token ☐

Ignore empty file ☐

Do not raise an error if no files ☒

Limit: 0

Prune path to handle large files: [text field]

Additional fields

Include filename in output? ☐ Filename fieldname: [text field]

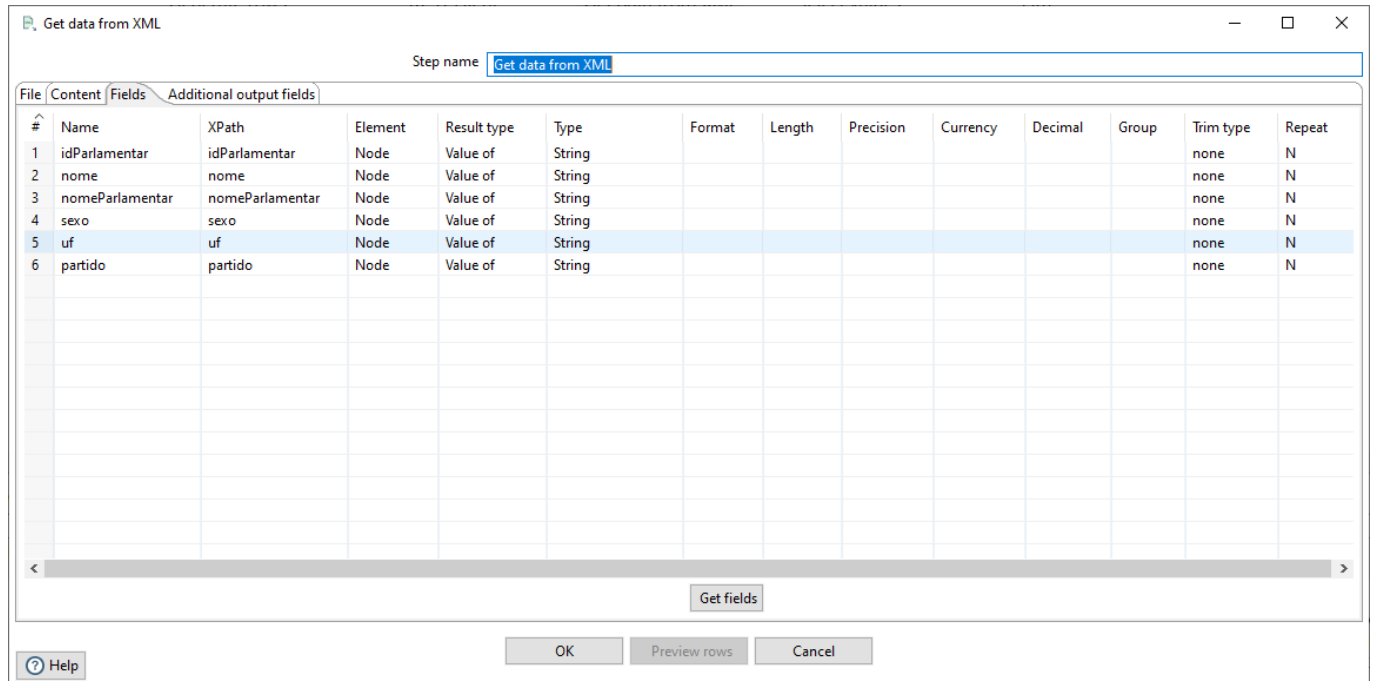
Rownum in output? ☐ Rownum fieldname: [text field]

Add to result filename

Add files to result filename ☐

Help OK Preview rows Cancel

**10.5** Na aba “Fields”, insira manualmente os campos identificados na imagem. Marque todos como “Element” do tipo “Node”, “Type” como “String”:

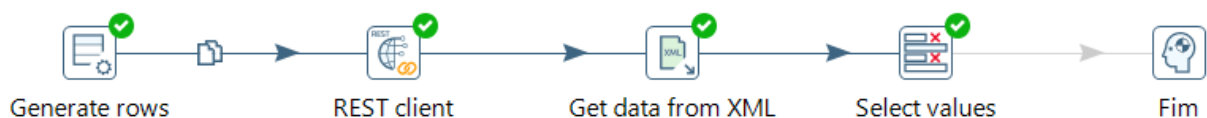


#	Name	XPath	Element	Result type	Type	Format	Length	Precision	Currency	Decimal	Group	Trim type	Repeat
1	idParlamentar	idParlamentar	Node	Value of	String							none	N
2	nome	nome	Node	Value of	String							none	N
3	nomeParlamentar	nomeParlamentar	Node	Value of	String							none	N
4	sexo	sexo	Node	Value of	String							none	N
5	uf	uf	Node	Value of	String							none	N
6	partido	partido	Node	Value of	String							none	N

Obs.: Para identificar os campos e os tipos de dados que o método devolve, basta acessar o link a seguir:

<https://www2.camara.leg.br/transparencia/dados-abertos/dados-abertos-legislativo/webservices/deputados/obterdeputados>

**10.6** Remova os campos “url” e “resultado”, adicionando um “Dummy” para concluir a transformação:



### BÔNUS:

Como uma extensão a este exercício, substitua o step dummy por uma gravação em arquivo texto e tabela, gerando as mesmas informações em ambas as saídas.

## Exercício 11 – API Rest com TOKEN

**11.1** Crie uma nova transformação, chamada “11 – Busca API Rest com TOKEN”. Para esta etapa, será necessário criar uma conta em um portal que oferece uma API pública que servirá ao propósito deste exercício. Portanto, acesse o endereço a seguir para criar seu perfil e obter suas credenciais, que serão posteriormente inseridas no Pentaho Data Integrator:

Acessar o endereço: <https://www.atlas.jor.br/plataforma/api/utilizarAPI/>



**11.2** Clique em “Faça o cadastro” e siga os passos orientados pelo portal:

### CADASTRO

O primeiro passo para acessar a API do **Atlas da Notícia** é fazendo um cadastro, criando seu próprio login e senha, que serão utilizados como credenciais de uso do nosso banco de dados.

[Faça o cadastro](#)

**11.3** Após concluir o cadastro, uma tela informando acesso negado será apresentada, mas isto é esperado; prossiga para o passo seguinte:

Atlas da Notícia

SOBRE O ATLAS

NEWSLETTER

FORMULÁRIO

PESQUISAS

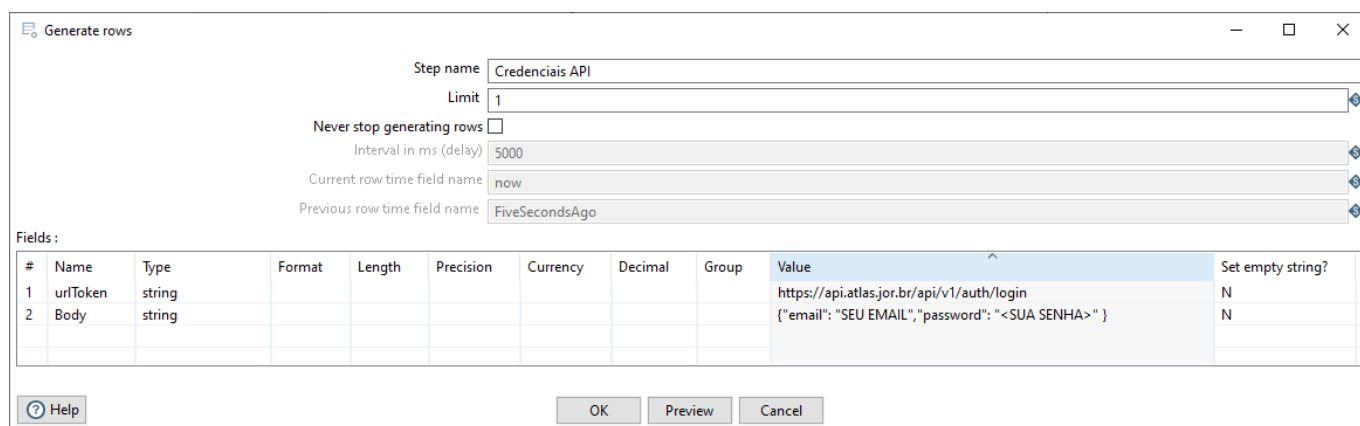
DOCUMENTAÇÃO DA API

ENGLISH

**Você não está autorizado a acessar esta rota**

Voltar

**11.4** Vá para o Pentaho, e acesse a transformação criada para este exercício; insira um step “Generate rows” e o configure conforme imagem a seguir:



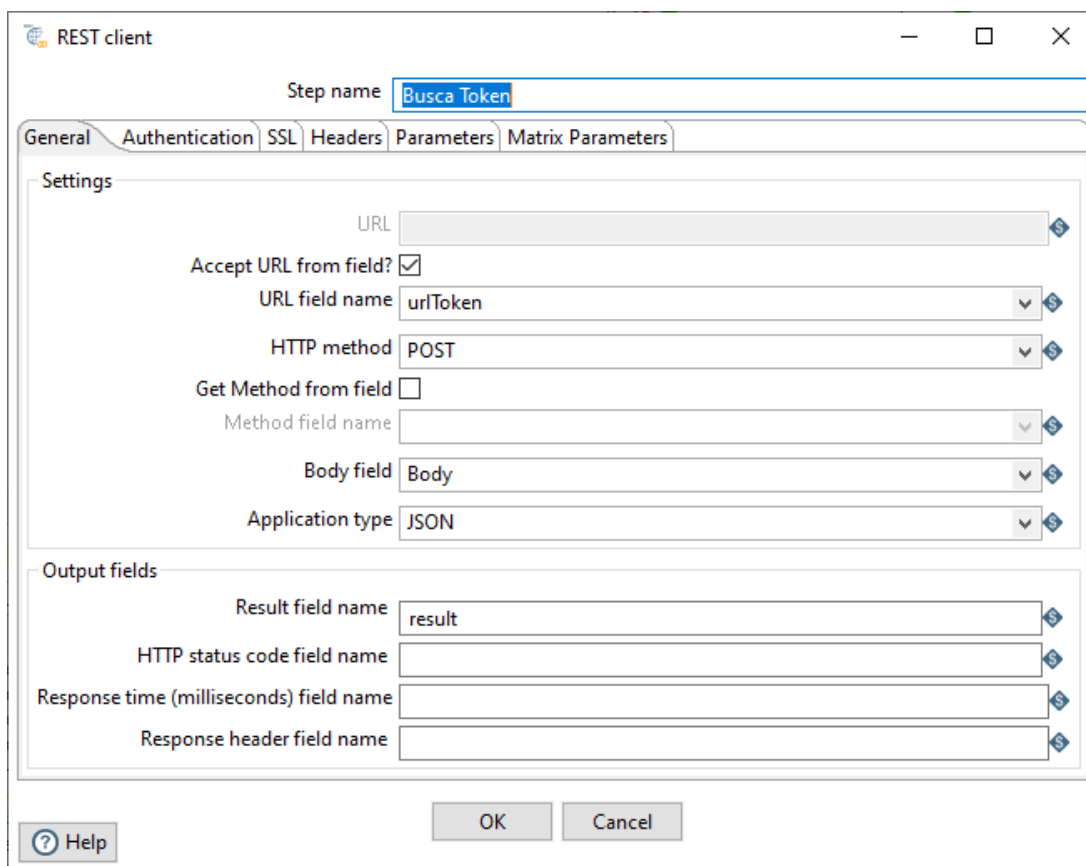
#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Value	Set empty string?
1	urlToken	string							https://api.atlas.jor.br/api/v1/auth/login	N
2	Body	string							{\"email\": \"<seu email>\", \"password\": \"<sua senha>\" }	N

Onde:

Valor para urlToken: <https://api.atlas.jor.br/api/v1/auth/login>

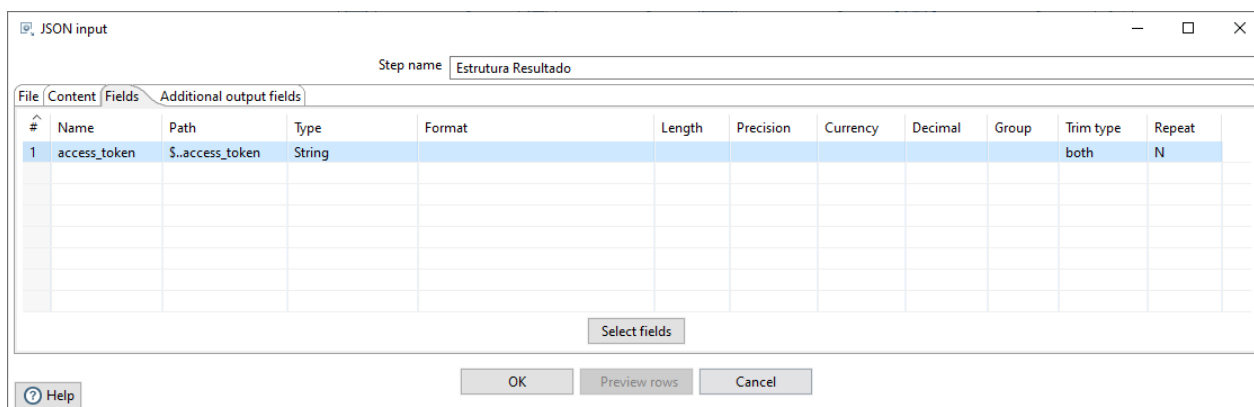
Valor para Body: {\"email\": \"<seu email>\", \"password\": \"<sua senha>\" }

**11.5** Adicione o step “REST client”, o configure para receber os dados gerados pelo step anterior, tanto o “urlToken” quanto o “Body”; observe o método “POST” sendo utilizado; na saída deste step, mantenha apenas o campo “result”:



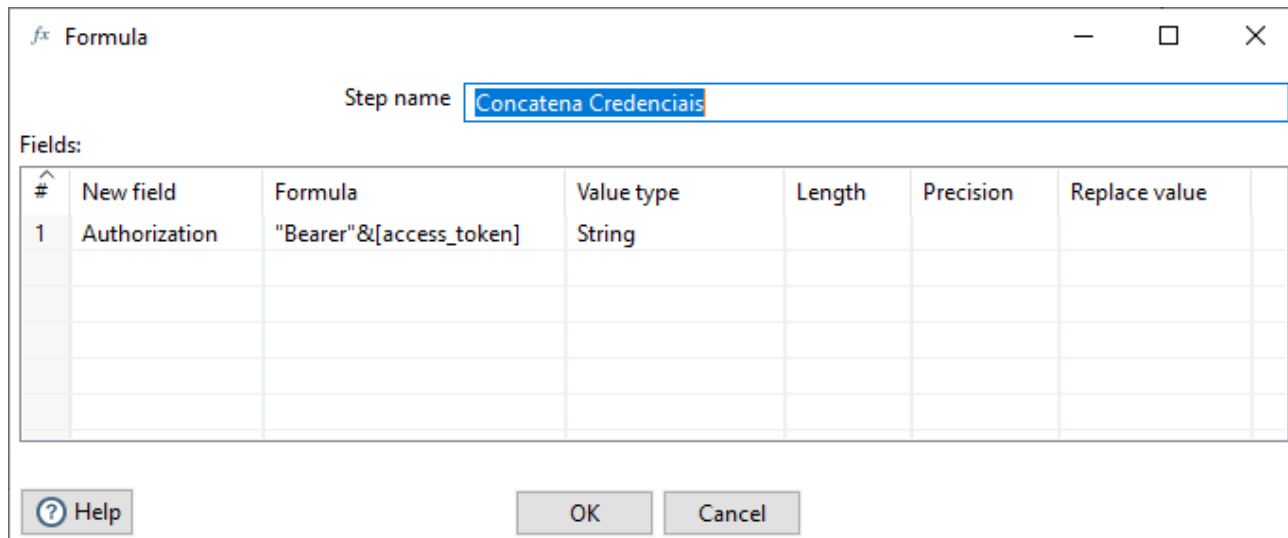
**11.6** Adicione novo step “JSON Input”; configure a entrada de dado proveniente do step anterior; adicione manualmente os campos necessários à leitura do resultado, conforme instruções a seguir:

Name: access\_token  
Path: \$.access\_token  
Type: String  
Trim Type: Both



#	Name	Path	Type	Format	Length	Precision	Currency	Decimal	Group	Trim type	Repeat
1	access_token	\$.access_token	String							both	N

**11.7** Insira o step “Formula”, para realizar a concatenação dos dados de credenciais; criar o campo “Authorization”, com o conteúdo de fórmula ‘ “Bearer”&” “[access\_token] ‘:



fx Formula

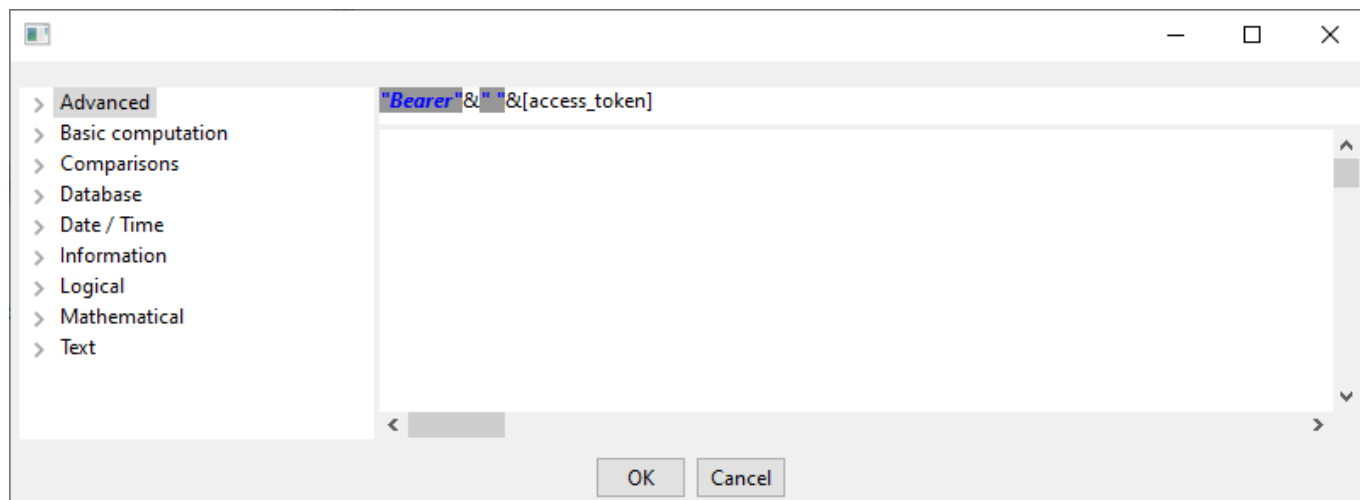
Step name: Concatena Credenciais

Fields:

#	New field	Formula	Value type	Length	Precision	Replace value
1	Authorization	"Bearer"&[access_token]	String			

Help OK Cancel

Criação do campo “Authorization”



> Advanced "Bearer"&[access\_token]

> Basic computation

> Comparisons

> Database

> Date / Time

> Information

> Logical

> Mathematical

> Text

OK Cancel

Configuração do step “Formula”

### 11.8 Adicione um segundo step “REST Client”; adicione a URL de busca dos dados:

URL: <https://api.atlas.ior.br/api/v1/data/analytic?estado=DF>

HTTP Method: GET

The screenshot shows the 'Settings' dialog box for a REST client. The 'Step name' is 'Retorna Dados'. The 'General' tab is selected. The 'URL' is 'https://api.atlas.jor.br/api/v1/data/analytic?estado=DF'. The 'HTTP method' is 'GET'. The 'Application type' is 'TEXT PLAIN'. The 'Output fields' section is empty.

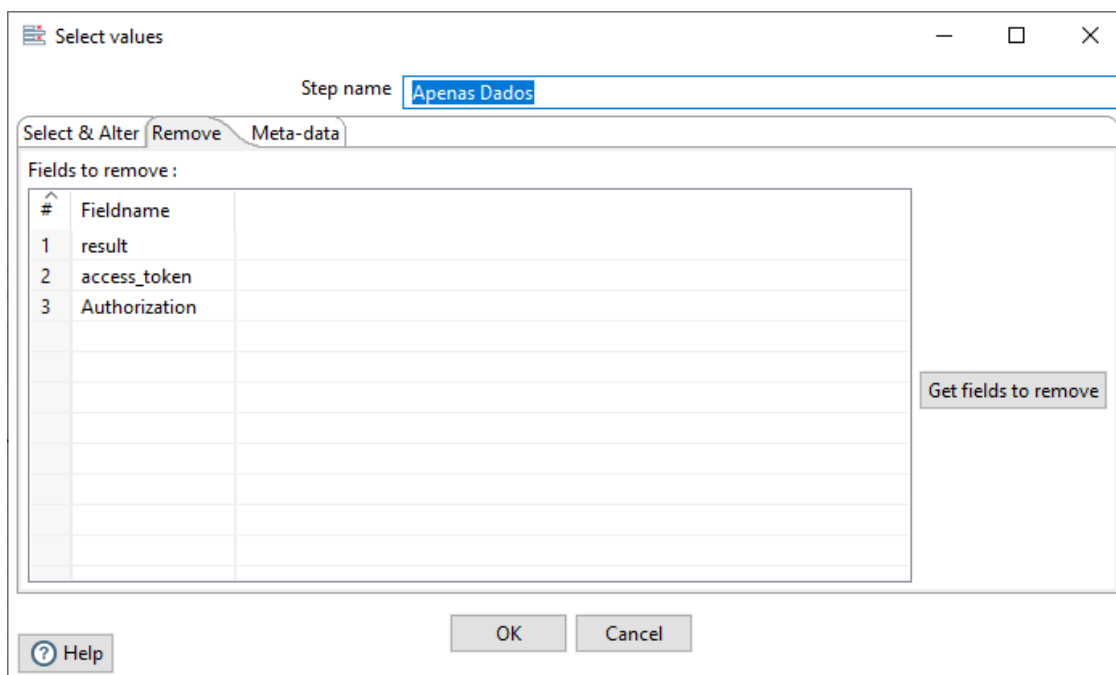
### Configuração geral do step

[illegible]

## Configuração do header



## 11.9 Remova do fluxo todos os campos à exceção do campo “Dados”:



Select values

Step name:

Select & Alter Remove Meta-data

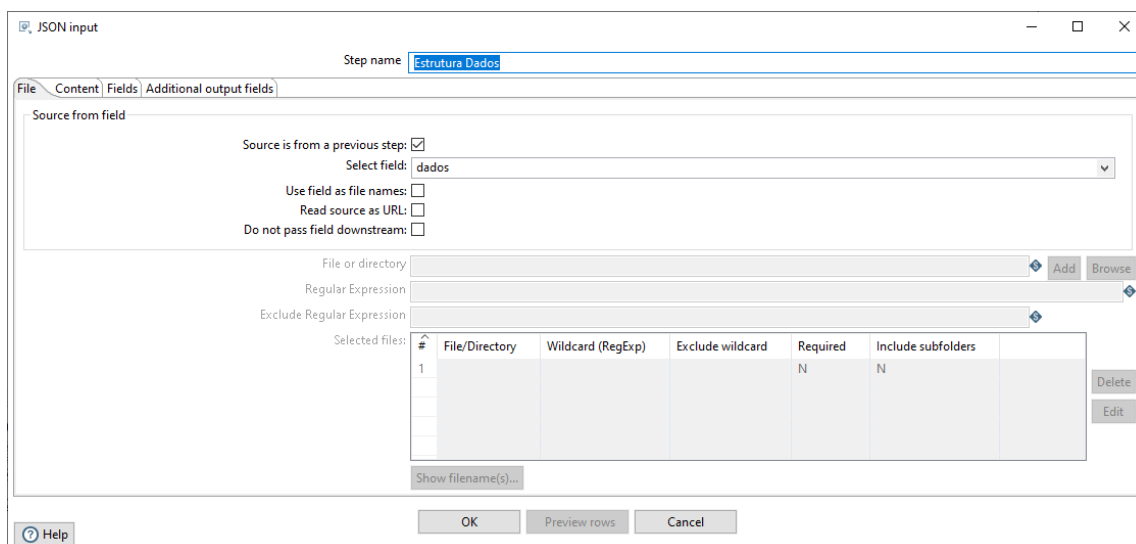
Fields to remove :

#	Fieldname
1	result
2	access_token
3	Authorization

Get fields to remove

Help OK Cancel

## 11.10 Em um novo step “JSON input”, configure os campos provenientes da saída da consulta REST:



JSON input

Step name:

File Content Fields Additional output fields

Source from field

Source is from a previous step: ☒

Select field:

Use field as file names: ☐

Read source as URL: ☐

Do not pass field downstream: ☐

File or directory  Add Browse

Regular Expression

Exclude Regular Expression

Selected files:

#	File/Directory	Wildcard (RegExp)	Exclude wildcard	Required	Include subfolders
1				N	N

Show filename(s)...

Help OK Preview rows Cancel

### 11.11 Configure manualmente os campos que serão retornados pela consulta, a partir da lista de campos disponíveis na documentação da API:

Documentação API: <https://www.atlas.jor.br/plataforma/api/utilizarAPI/>

```
1  [
2    {
3      "id": 2215,
4      "nome_veiculo": "DIARIO DE RONDONIA - VOZ REGIONAL",
5      "media_source_id": 8,
6      "fonte": "Secom/PR",
7      "segment_id": 1,
8      "segmento": "Impresso",
9      "city_id": 3854,
10     "cidade": "Ariquemes",
11     "cod_mun_ibge": "1100023",
12     "state_id": 19,
13     "estado": "RO".
```

JSON input

Step name:

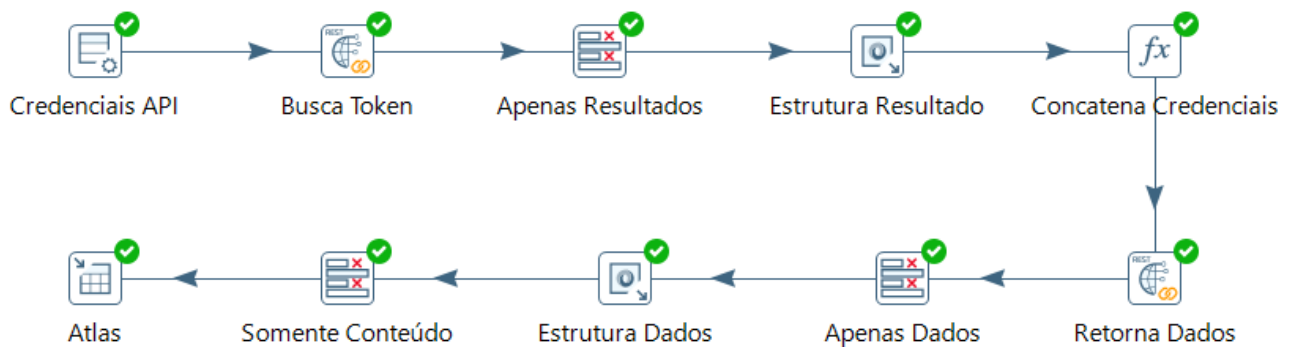
#	Name	Path	Type	Format	Length	Precision	Currency	Decimal	Group	Trim type	Repeat
1	id	\$.id	Integer							both	N
2	nome_veiculo	\$.nome_veiculo	String							both	N
3	fonte	\$.fonte	String							both	N
4	segmento	\$.segmento	String							both	N
5	cidade	\$.cidade	String							both	N
6	estado	\$.estado	String							both	N

Select fields

OK Preview rows Cancel

Help

**11.12** Após a construção de sua transformação, salve, execute e analise os dados gerados a cada step, para acompanhar as transformações etapa a etapa:

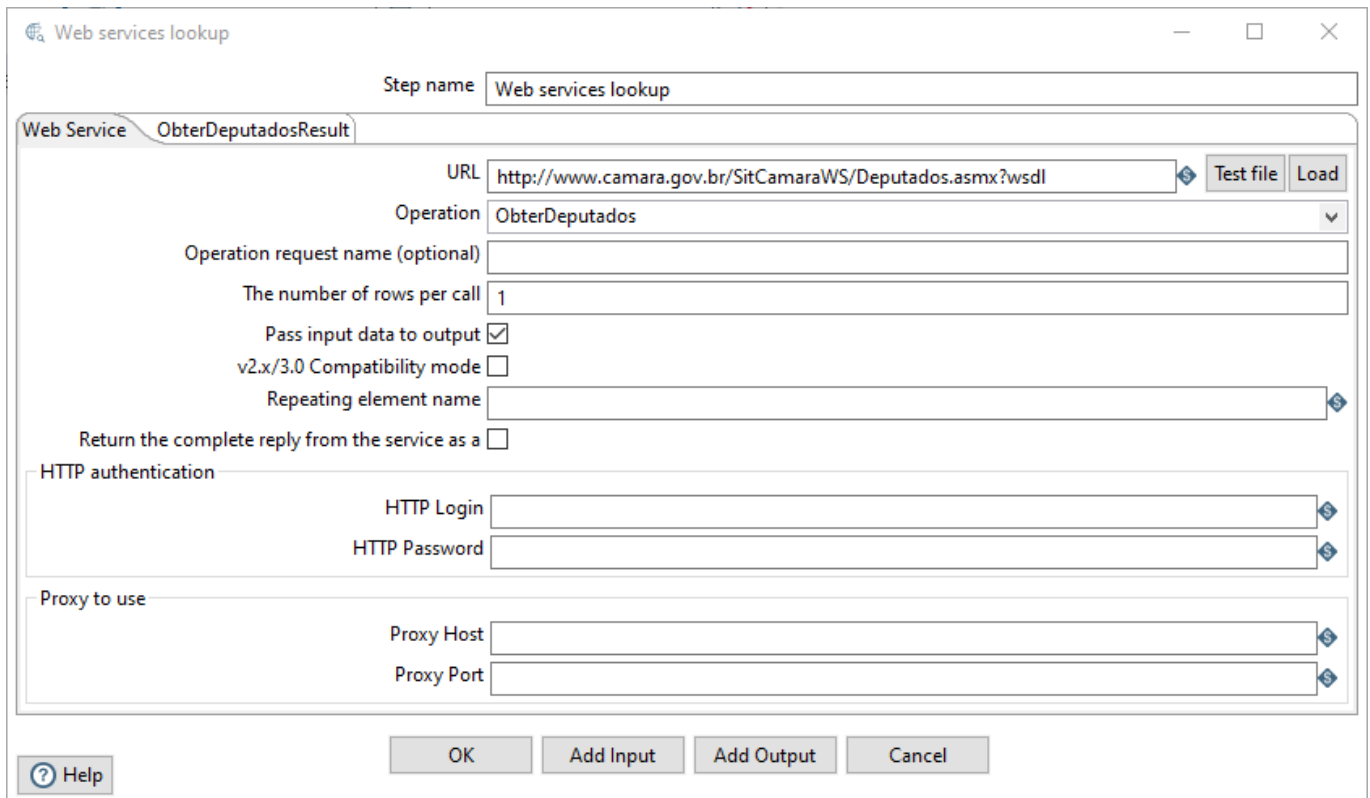


### BÔNUS:

Como uma extensão a este exercício, crie uma tabela chamada “ATLAS” e grave todos registros cujos nomes dos veículos estejam preenchidos, e grave o restante em um arquivo csv.

## Exercício 12 – Webservices Lookup

**12.1** Crie uma nova transformação, chamada “12 – Busca Webservices Lookup”, adicionando o step “Webservices lookup”. Na aba “Web Service”, preencha o campo “URL” com o endereço <http://www.camara.gov.br/SitCamaraWS/Deputados.asmx?wsdl>; Clique em “Load” para trazer os métodos disponíveis no Webservice; em “Operation”, selecione o método “ObterDeputados”; reduza o valor de “The number of rows per call” para 1, e marque “Pass input data to output”:



Web services lookup

Step name: Web services lookup

Web Service: ObterDeputadosResult

URL: <http://www.camara.gov.br/SitCamaraWS/Deputados.asmx?wsdl> [Test file] [Load]

Operation: ObterDeputados

Operation request name (optional):

The number of rows per call: 1

Pass input data to output: ☒

v2.x/3.0 Compatibility mode: ☐

Repeating element name:

Return the complete reply from the service as a: ☐

HTTP authentication

HTTP Login:

HTTP Password:

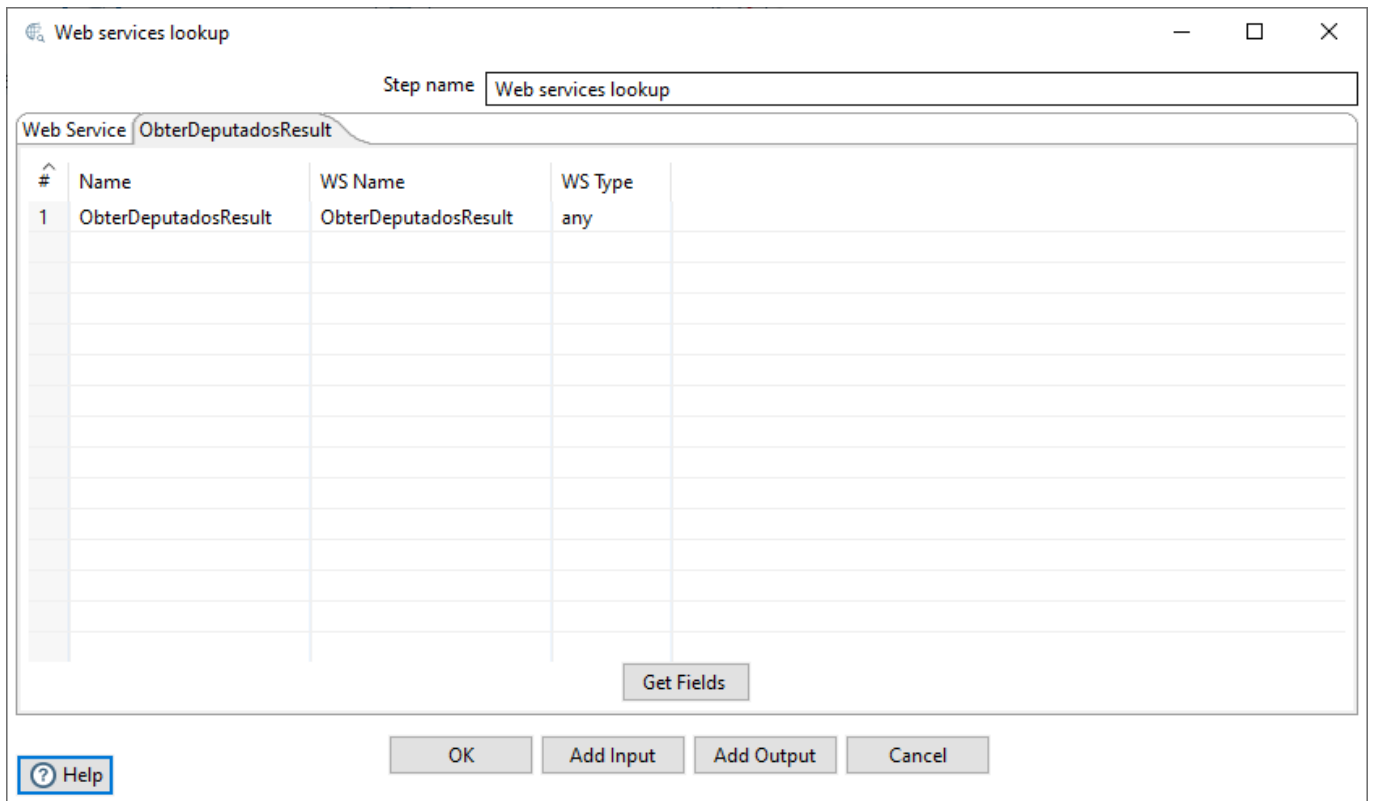
Proxy to use

Proxy Host:

Proxy Port:

[?] Help [OK] [Add Input] [Add Output] [Cancel]

## 12.2 Vá para a aba “ObterDeputadosResult” e traga os campos disponíveis:



Web services lookup

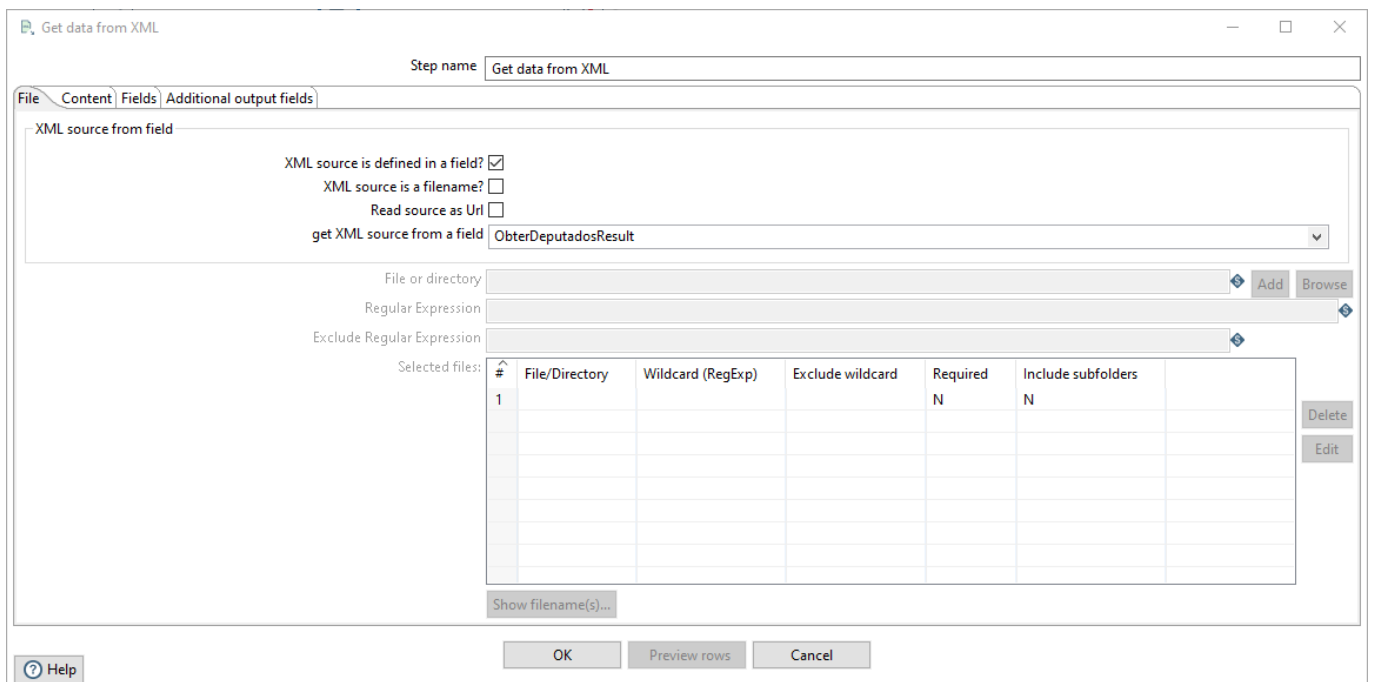
Step name: Web services lookup

#	Name	WS Name	WS Type
1	ObterDeputadosResult	ObterDeputadosResult	any

Get Fields

Help OK Add Input Add Output Cancel

## 12.3 Adicione o step de leitura XML, e na aba “File”, configure conforme imagem a seguir:



Get data from XML

Step name: Get data from XML

File Content Fields Additional output fields

XML source from field

XML source is defined in a field? ☒

XML source is a filename? ☐

Read source as Uri ☐

get XML source from a field: ObterDeputadosResult

File or directory: Add Browse

Regular Expression: Add

Exclude Regular Expression: Add

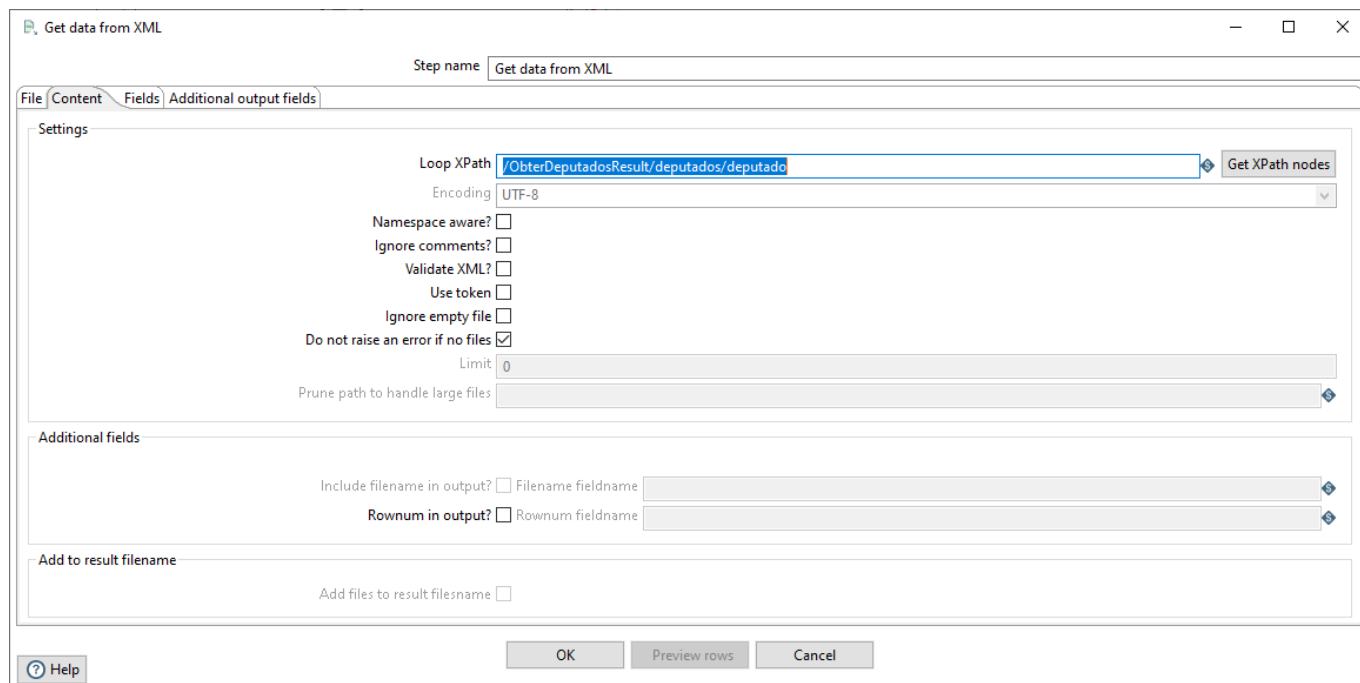
#	File/Directory	Wildcard (RegExp)	Exclude wildcard	Required	Include subfolders
1				N	N

Show filename(s)...

Help OK Preview rows Cancel

**12.4** Na aba “Content”, adicione manualmente o “Loop XPath”, conforme texto a seguir:

**Loop XPath: /ObterDeputadosResult/deputados/deputado**



Step name: Get data from XML

File Content Fields Additional output fields

Settings

Loop XPath: /ObterDeputadosResult/deputados/deputado Get XPath nodes

Encoding: UTF-8

Namespace aware? ☐

Ignore comments? ☐

Validate XML? ☐

Use token? ☐

Ignore empty file? ☐

Do not raise an error if no files? ☒

Limit: 0

Prune path to handle large files

Additional fields

Include filename in output? ☐ Filename fieldname

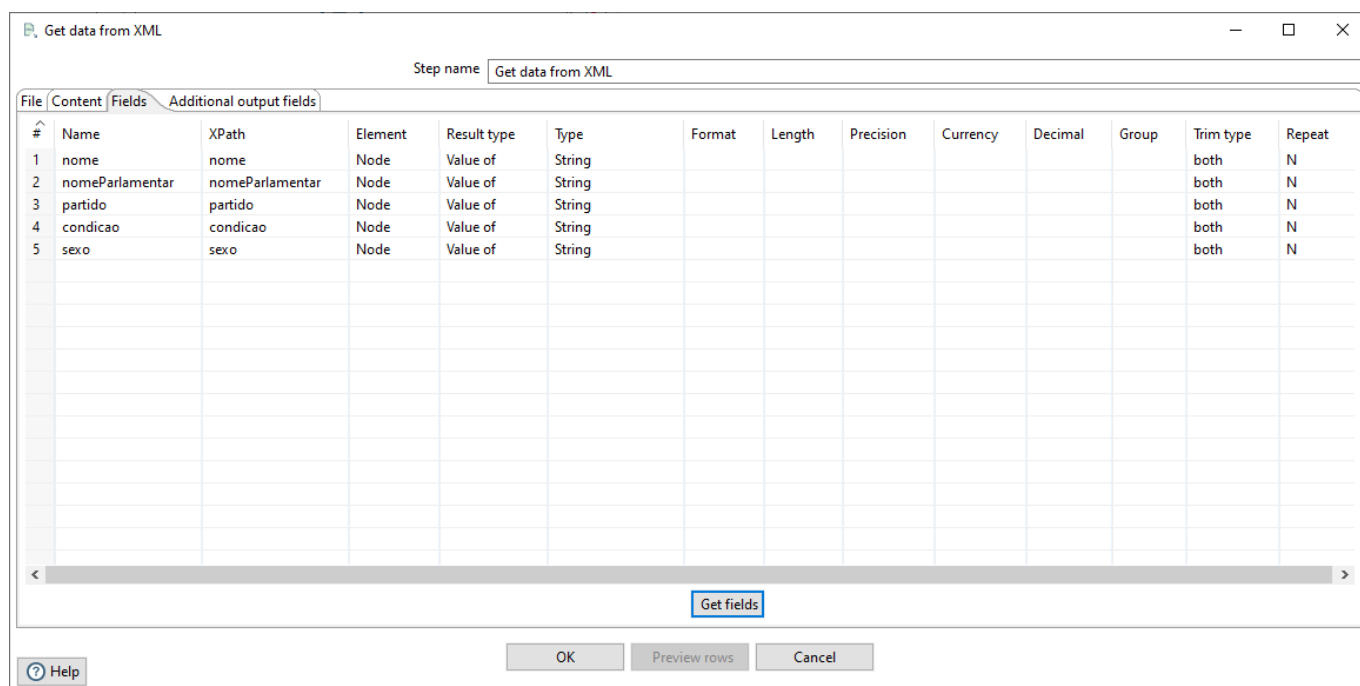
Rownum in output? ☐ Rownum fieldname

Add to result filename

Add files to result filename ☐

OK Preview rows Cancel

**12.5** Na aba “Fields”, adicione manualmente os campos, conforme imagem a seguir:



Step name: Get data from XML

File Content Fields Additional output fields

#	Name	XPath	Element	Result type	Type	Format	Length	Precision	Currency	Decimal	Group	Trim type	Repeat
1	nome	nome	Node	Value of	String							both	N
2	nomeParlamentar	nomeParlamentar	Node	Value of	String							both	N
3	partido	partido	Node	Value of	String							both	N
4	condicao	condicao	Node	Value of	String							both	N
5	sexo	sexo	Node	Value of	String							both	N

Get fields

OK Preview rows Cancel

**12.6** Remova o campo “ObterDeputadosResult”, crie uma tabela para armazenar os resultados e a atualize quando houver alteração nos dados:

