



Microsoft

Integración de Sharepoint con Pentaho Data Integration

EJEMPLO PRÁCTICO

1. INTRODUCCIÓN

En este documento se recogen algunos ejemplos de integración entre Sharepoint y Pentaho Data Integration (PDI). Sharepoint es ampliamente utilizado por las organizaciones para diferentes fines, en particular, almacenar datos a través de listas o ficheros de todo tipo.

A partir de lo indicado en el punto anterior, Sharepoint se convierte en una fuente de datos más dentro de los sistemas analíticos.

En el presente documento, se recogen tres ejemplos:

- Obtención de datos de listas
- Obtención de datos contenidos en ficheros CSV
- Descarga de ficheros

2. PASOS PREVIOS

Antes de hacer peticiones a la API, hay que realizar una serie de pasos previos, como los que se describen a continuación:

1. Ir a la página

https://tenant.sharepoint.com/sites/test_integracion_pdi/_layouts/15/appregnew.aspx para registrar una aplicación nueva de Sharepoint.

(La URL genérica es

https://<YourSharePointCollectionURL>/_layouts/15/appregnew.aspx.

The screenshot shows the 'App Information' section of the SharePoint application registration page. It includes fields for Client Id, Client Secret, Title, App Domain, and Redirect URI, each with a corresponding 'Generate' button and a numbered instruction. The 'Create' and 'Cancel' buttons are at the top right.

App Information
The app's information, including app id, secret, title, hosting url and redirect url.

Client Id: 1. Click here to generate Client Id

Client Secret: 2. Click here to generate Client Secret

Title: 3. Enter Title for your APP

App Domain: 4. Enter the domain of your SharePoint Online Site Collection
Example: "www.contoso.com"

Redirect URI: 5. Enter your Site Collection URL
Example: "https://tenant.sharepoint.com/sites/test_integracion_pdi/_layouts/15/appregnew.aspx"

Rellenar los campos que aparecen en la imagen y hacer clic en “Crear”.

2. Una vez creada la aplicación, aparecerán en pantalla los siguientes datos:

The app identifier has been successfully created.

Client Id: 7a7a12d2-330f-4171-8100-0077100
Client Secret: 9VNnkLgC
Title: Postman_APITest
App Domain:
Redirect URI:

Copiar estos datos en un fichero de texto, ya que los vamos a necesitar para pasos posteriores.

3. Para conceder permisos a la aplicación, debemos ir a la página https://tenant.sharepoint.com/sites/test_integracion_pdi/layouts/15/appinv.aspx
(La URL genérica es: *https://<YourSharePointCollectionURL>/_layouts/15/appinv.aspx*).

App Id and Title

The app's identity and its title.

App Id:

Lookup

1. Enter the Client ID that was generated in the previous step and click on Lookup button. This will fetch all your app details.

Title:

App Domain:

Example: "www.contoso.com"

Redirect URL:

Example: "https://www.contoso.com/default.aspx"

App's Permission Request XML

The permission required by the app.

Permission Request XML:

2. Add below XML to grant "Full Control" access to your App to communicate with SharePoint.

```
<AppPermissionRequests AllowAppOnlyPolicy="true">
  <AppPermissionRequest Scope="http://sharepoint/content/sitecollection/web" Right="FullControl" />
</AppPermissionRequests>
```

3. Click on Create button

Create

Cancel

En el campo de solicitud de permisos XML, copiar lo siguiente:

```
<AppPermissionRequests AllowAppOnlyPolicy="true">
  <AppPermissionRequest
    Scope="http://sharepoint/content/sitecollection/web"
    Right="FullControl" />
</AppPermissionRequests>
```

4. Por último, para obtener el id del tenant, tenemos que acceder a la siguiente web:
https://tenant.sharepoint.com/sites/test_integracion_pdi/layouts/15/appprincipals.aspx. (La URL genérica es:
https://<YourSharePointCollectionURL>/_layouts/15/appprincipals.aspx).

PT Prasad Team Site EDIT LINKS

Site Settings · Site Collection App Permissions ⓘ

Home	App Display Name ⓘ
Conversations	Postman_APTTest
Documents	
Notebook	
Pages	
Recent	
Inquiry	

App Identifier

The value before @ is Client Id

The value after @ is Tenant Id

i:01.t/ms.sp.ext/7a7a12c...@335344d...

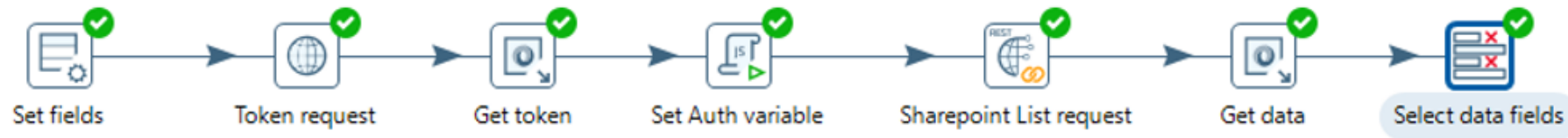
Client Id

Tenant Id

Copiamos el TenantId (la parte que va a continuación del @) en el fichero de texto donde teníamos los datos copiados en los pasos anteriores.

3. OBTENER DATOS DE UNA LISTA

La estructura de la transformación de PDI es la siguiente:



El proceso consiste en obtener el token en primer lugar y luego, realizar la petición a la API de Sharepoint y seleccionar los campos deseados para poder utilizarlos posteriormente.

A continuación, se describe detalladamente cada uno de los pasos.

1. Generar los campos necesarios asignándoles su valor correspondiente para poder utilizarlos en los pasos posteriores:

Step name									
Set fields									
Limit									
1									
Never stop generating rows ☐									
Interval in ms (delay)									
5000									
Current row time field name									
now									
Previous row time field name									
FiveSecondsAgo									
Fields:									
#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Value
1	url_token	String							https://accounts.accesscontrol.windows.net/aa82e677-ed4-4f52-8609-25b997f58368/tokens/OAuth/2/
2	Content-Type	String							application/x-www-form-urlencoded
3	grant_type	String							client_credentials
4	client_id	String							586031a3-63b6-4ae5-83ce-c3212e63ba3d@aa82e677-ed4-4f52-8609-25b997f58368
5	client_secret	String							TZ4kOxUD92fuvzqAVMwYEAdQPX7T/n4k+eZ9aeZII0E=
6	resource	String							00000003-0000-0ff1-ce00-000000000000/stratebi.sharepoint.com@aa82e677-ed4-4f52-8609-25b997f58368
7	url_sharepoint	String							https://stratebi.sharepoint.com/sites/test_integracion_pdi/_api/web/lists/GetByTitle('ListaProductos')/items
8	Content-Type sharepoint	String							application/json;odata=verbose
9	Accept	String							application/json;odata=verbose

- **url_token:** URL de la petición para la obtención del token.
Valor: <https://accounts.accesscontrol.windows.net/aa82e677-ed4-4f52-8609-25b997f58368/tokens/OAuth/2/>
(La URL genérica es:
https://accounts.accesscontrol.windows.net/tenant_id/tokens/OAuth/2/).
- **Content-Type:** header de la petición para obtener el token.
Valor: *application/x-www-form-urlencoded*.
- **grant_type:** parámetro del body de la petición para obtener el token.
Valor: *client_credentials*.

- **client_id:** parámetro del body de la petición para obtener el token.
Valor: *586031a3-63b6-4ae5-83ce-c3212e63ba3d@aa82e677-ede4-4f52-8609-25b997f58368*.
(El valor genérico es *client_id@tenant_id*).
- **client_secret:** parámetro del body de la petición para obtener el token.
Valor: *TZHkOxUD92fuvzqAYMwYEAdQPX7T/t4k+eZ9aeZlI0E=*
(El valor es el del campo *client_secret* obtenido en el paso 2 de la sección de pasos previos).
- **resource:** parámetro del body de la petición para obtener el token.
Valor: *00000003-0000-0ff1-ce00-000000000000/tenant.sharepoint.com@aa82e677-ede4-4f52-8609-25b997f58368*.
(El valor genérico es:
00000003-0000-0ff1-ce00-000000000000/YourSharepointOnlineDomainName@tenant_id).

- **url_sharepoint:** URL de la petición para la obtención de los datos, en este caso, de la **lista**.

Valor:

[https://tenant.sharepoint.com/sites/test_integracion_pdi/_api/web/lists/GetByTitle\('ListaProductos'\)/items](https://tenant.sharepoint.com/sites/test_integracion_pdi/_api/web/lists/GetByTitle('ListaProductos')/items).

(El valor genérico es:

[https://tenant_id/sites/site_id/_api/web/lists/GetByTitle\(name_list\)/items](https://tenant_id/sites/site_id/_api/web/lists/GetByTitle(name_list)/items)).

- **Content-Type sharepoint:** header para la petición de obtención de datos

Valor: *application/[json;odata=verbose](#)*.

- **Accept:** header de la petición de obtención de datos

Valor: *application/[json;odata=verbose](#)*.

HTTP post

Step name Token request

General Fields

Settings

URL

Accept URL from field? ☒

URL field name url_token

Encoding UTF-8

Request entity field

Post a file ☐

Connection timeout 10000

Socket timeout 10000

Connection close wait time -1

Output fields

Result field name result_token

HTTP status code field name

Response time (milliseconds) field name

Response header field name

HTTP authentication

HTTP Login

HTTP Password

HTTP post

Step name

General Fields

Body (Header) Parameters :

#	Name	Parameter	Put in Header?
1	Content-Type	Content-Type	Y
2	grant_type	grant_type	N
3	client_id	client_id	N
4	client_secret	client_secret	N
5	resource	resource	N

Query parameters

#	Name	Value
1		

Marcamos el campo Content-Type como **Header**.

3. Copiar y guardar en el campo **result_token** el valor del token.

JSON input

Step name **Get token**

File Content Fields Additional output fields												
#	Name	Path	Type	Format	Length	Precision	Currency	Decimal	Group	Trim type	Repeat	
1	token	\$.access_token	String							none	N	

File Content Fields Additional output fields					
Source from field					
Source is from a previous step: ☒					
Select field: result_token					
Use field as file names: ☐					
Read source as URL: ☐					
Do not pass field downstream: ☐					
File or directory					
Regular Expression					
Exclude Regular Expression					
Selected files:					
#	File/Directory	Wildcard (RegExp)	Exclude wildcard	Required	Include subfolders
1				N	N

4. Creación de la variable “Auth” para formar el campo de Authorization con la estructura correcta.

Modified JavaScript value

Step name

Java script functions :

- > Transform Scripts
- > Transform Constants
- > Transform Functions
- ▼ Input fields
 - ▶ url_token
 - ▶ Content-Type
 - ▶ grant_type
 - ▶ client_id
 - ▶ client_secret
 - ▶ resource
 - ▶ url_sharepoint
 - ▶ Content-Type sharepoint
 - ▶ Accept
 - ▶ result_token
 - ▶ token
- ▼ Output fields
 - Please use the 'Replace value 'Fieldname'

Java script :

Script 1 ✕

```
var Auth = "Bearer"+" "+token;
```

Linerr: 0

Compatibility mode? ☐ Optimization level

Fields

#	Fieldname	Rename to	Type	Length	Precision	Replace value 'Fieldname' or 'Rename to'
1	Auth		String			N

5. Realizar la petición para obtener los valores de la lista.

 REST client

Step name

General Authentication SSL Headers Parameters Matrix Parameters

Settings

URL

Accept URL from field? ☒

URL field name

HTTP method

Get Method from field ☐

Method field name

Body field

Application type

Output fields

Result field name

HTTP status code field name

Response time (milliseconds) field name

Response header field name

Step name

General Authentication SSL Headers Parameters Matrix Parameters			
Header			
#	Field	Name	
1	Auth	Authorization	
2	Accept	Accept	
3	Content-Type sharepoint	Content-Type	

6. Guardar los datos de los campos deseados.

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

Regular Expression

Exclude Regular Expression

Selected files:

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

JSON input

Step name

File Content Fields Additional output fields

#	Name	Path	Type	Format	Length	Precision	Currency	Decimal	Group	Trim type	Repeat
1	Title	\$.Title	String							none	N
2	Familia	\$.field_2	String							none	N
3	Subfamilia	\$.field_1	String							none	N

7. Seleccionar solo los campos necesarios.

 Select values

Step name

Select & Alter Remove Meta-data

Fields :

#	Fieldname	Rename to	Length	Precision
1	Title			
2	Familia			
3	Subfamilia			

En la siguiente imagen podemos observar los datos obtenidos de la lista de Sharepoint:

 Examine preview data

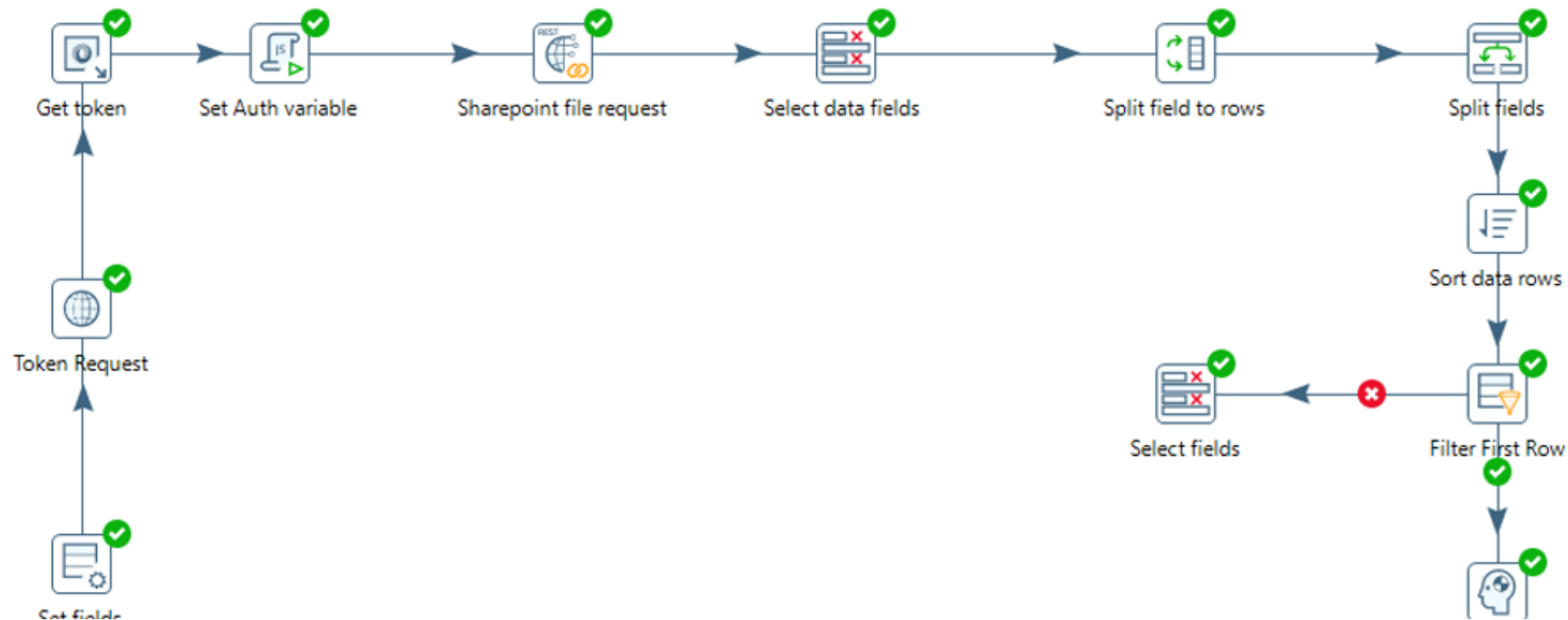
Rows of step: Select data fields (8 rows)

#	Title	Familia	Subfamilia
1	bici	Deportes	Vehiculos deportivos
2	skate	Deportes	Vehiculos deportivos
3	balon futbol	Deportes	Balones
4	balon voley	Deportes	Balones
5	red voley	Deportes	Utensilios de entrenamiento
6	conos	Deportes	Utensilios de entrenamiento
7	balon de baloncesto	Deportes	Pelotas
8	camiseta manga corta sport	Deportes	Ropa

4. OBTENER DATOS DE UN FICHERO

4.1. Obtener y tratar datos de un fichero csv en una transformación

La estructura de la transformación de PDI es la siguiente:



A continuación, se describe detalladamente cada uno de los pasos.

- Generate rows

Step name

Set fields

Limit

1

Never stop generating rows

☐

Interval in ms (delay)

5000

Current row time field name

now

Previous row time field name

FiveSecondsAgo

Fields:

#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Value
1	url_token	String							https://accounts.accesscontrol.windows.net/aa82e677-ed4-4f52-8609-25b997f58368/tokens/OAuth/2/
2	Content-Type	String							application/x-www-form-urlencoded
3	grant_type	String							client_credentials
4	client_id	String							586031a3-63b6-4ae5-83ce-c3212e63ba3d@aa82e677-ed4-4f52-8609-25b997f58368
5	client_secret	String							TZHkOxUD92fuvzqAYMwYEAQXPX7T/t4k+eZ9aeZII0E=
6	resource	String							00000003-0000-0ff1-ce00-000000000000/stratebi.sharepoint.com@aa82e677-ed4-4f52-8609-25b997f58368
7	url_sharepoint_file	String							https://[redacted].sharepoint.com/sites/test_integracion_pdi/_api/web/GetFolderByServerRelativeUrl('/sites/test_inte
8	Content-Type sharepoint	String							application/json;odata=verbose
9	Accept	String							application/json;odata=verbose

- **url_token:** URL de la petición para la obtención del token.
Valor: <https://accounts.accesscontrol.windows.net/aa82e677-ed4-4f52-8609-25b997f58368/tokens/OAuth/2/>.
(La URL genérica es:
https://accounts.accesscontrol.windows.net/tenant_id/tokens/OAuth/2/).
- **Content-Type:** header de la petición para obtener el token.
Valor: *application/x-www-form-urlencoded*.
- **grant_type:** parámetro del body de la petición para obtener el token.
Valor: *client_credentials*.
- **client_id:** parámetro del body de la petición para obtener el token.
Valor: *586031a3-63b6-4ae5-83ce-c3212e63ba3d@aa82e677-ed4-4f52-8609-25b997f58368*.
(El valor genérico es *client_id@tenant_id*).
- **client_secret:** parámetro del body de la petición para obtener el token.
Valor: *TZHkOxUD92fuvzqAYMwYEAQdQPX7T/t4k+eZ9aeZlI0E=*.
(El valor es el del campo *client_secret* obtenido en el paso 2 de la sección de pasos previos).

- **resource:** parámetro del body de la petición para obtener el token.
Valor: 00000003-0000-0ff1-ce00-000000000000/tenant.sharepoint.com@aa82e677-ed4-4f52-8609-25b997f58368.
(El valor genérico es:
`00000003-0000-0ff1-ce00-000000000000/YourSharepointOnlineDomainName@tenant_id`).
- **url_sharepoint_file:** URL de la petición para la obtención de los datos, en este caso, del fichero "Customers.csv".
Valor:
[https://tenant.sharepoint.com/sites/test_integracion_pdi/_api/web/FolderByServerRelativeUrl\('/sites/test_integracion_pdi/Documentos%20compartidos'\)/Files\('Customers.csv'\)/\\$value](https://tenant.sharepoint.com/sites/test_integracion_pdi/_api/web/FolderByServerRelativeUrl('/sites/test_integracion_pdi/Documentos%20compartidos')/Files('Customers.csv')/$value) .
(El valor genérico es:
`https://tenant_id/sites/site_id/_api/web/FolderByServerRelativeUrl('folder_path')/Files('file_name')/$value`).
- **Content-Type sharepoint:** header para la petición de obtención de datos
Valor: `application/json;odata=verbose`.

Step name		Token request
GeneralFields		
Settings		
URL		
Accept URL from field?		☒
URL field name		url_token
Encoding		UTF-8
Request entity field		
Post a file		☐
Connection timeout		10000
Socket timeout		10000
Connection close wait time		-1
Output fields		
Result field name		result_token
HTTP status code field name		
Response time (milliseconds) field name		
Response header field name		
HTTP authentication		
HTTP Login		
HTTP Password		
Proxy to use		
Proxy Host		
Proxy Port		

HTTP post

Step name

General Fields

Body (Header) Parameters :

#	Name	Parameter	Put in Header?
1	Content-Type	Content-Type	Y
2	grant_type	grant_type	N
3	client_id	client_id	N
4	client_secret	client_secret	N
5	resource	resource	N

Query parameters

#	Name	Value
1		

Marcamos el campo Content-Type como **Header**.

3. Copiar y guardar en el campo **result_token** el valor del token.

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

Regular Expression

Exclude Regular Expression

Selected files:

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

4. Creación de la variable “Auth” para formar el campo de Authorization con la estructura correcta.

Modified JavaScript value

Step name

Java script functions :

- > Transform Scripts
- > Transform Constants
- > Transform Functions
- ✓ Input fields
 - url_token
 - Content-Type
 - grant_type
 - client_id
 - client_secret
 - resource
 - url_sharepoint
 - Content-Type sharepoint
 - Accept
 - result_token
 - token
- ✓ Output fields
 - Please use the 'Replace value 'Fieldname'

Java script :

Script 1

```
var Auth = "Bearer"+" "+token;
```

Linerr: 0

Compatibility mode? ☐ Optimization level

Fields

#	Fieldname	Rename to	Type	Length	Precision	Replace value 'Fieldname' or 'Rename to'
1	Auth		String			N

5. Realizar la petición a la API para obtener los datos del fichero.

REST client

Step name

General Authentication SSL Headers Parameters Matrix Parameters

Settings

URL

Accept URL from field? ☒

URL field name

HTTP method

Get Method from field ☐

Method field name

Body field

Application type

Output fields

Result field name

HTTP status code field name

Response time (milliseconds) field name

Response header field name

Step name

General Authentication SSL Headers Parameters Matrix Parameters

Header

#	Field	Name	
1	Auth	Authorization	
2	Accept	Accept	
3	Content-Type sharepoint	Content-Type	

6. Renombrar el campo “result_sharepoint” a “result”.

 Select values

Step name

Select & Alter

Remove

Meta-data

Fields :

#	Fieldname	Rename to	Length	Precision	
1	result_sharepoint	result			

7. Separar cada una de las filas de datos por el delimitador de salto de línea y añadir el campo "rownum".

 Split field to rows — □ ×

Step name

Field to split  

Delimiter 

Delimiter is a Regular Expression ☒

New field name 

Additional fields

Include rownum in output? ☒ Rownum fieldname 

Reset Rownum at each input row? ☐

stratebi
Analytics and Big Data Consulting

Downloaded from <http://ajph.org/> on November 10, 2015

10. Filtrar la primera fila para que no aparezca el nombre de los campos dos veces y la última porque se reciben valores nulos.

 Filter rows — □ ×

Step name:

Send 'true' data to step:

Send 'false' data to step:

The condition:

☐

rownum = [1]

OR

(
AND Markets_CustomerNumber IS NULL
Markets_Territory IS NULL
)

? Help OK Cancel

Si se lee la primera o la última fila, se desecha. En caso contrario, se recogen los datos.

 Dummy (do nothing) — □ ×

Step name:

? Help OK Cancel

 Select values

Step name

Select & Alter Remove Meta-data

Fields :

#	Fieldname	Rename to	Length	Precision	
1	Markets_CustomerNumber				
2	Markets_Territory				
3	Markets_Country				
4	Markets_State				
5	Markets_City				
6	Customers_CustomerName				
7	Customers_ContactFirstName				
8	Customers_ContactLastName				
9	Customers_Phone				
10	Customers_AddressLine1				
11	Customers_CreditLimit				

El resultado es el siguiente:

Rows of step: Select fields (98 rows)

#	Markets_CustomerNumber	Markets_Territory	Markets_Country	Markets_State	Markets_City	Customers_CustomerName	Customers_ContactFirstName	Customers_ContactLastName	Customers_ContactPhone
1	103	EMEA	France	<null>	Nantes	Atelier graphique	Carine	Schmitt	40.32.255
2	112	NA	USA	NV	Las Vegas	Signal Gift Stores	Sue	King	70255518
3	114	APAC	Australia	Victoria	Melbourne	Australian Collectors, Co.	Peter	Ferguson	03 9520 4
4	119	EMEA	France	<null>	Nantes	La Rochelle Gifts	Janine	Labrun	40.67.855
5	121	EMEA	Norway	<null>	Stavern	Baane Mini Imports	Jonas	Bergulfsen	07-98 951
6	124	NA	USA	CA	San Rafael	Mini Gifts Distributors Ltd.	Valarie	Nelson	41555514
7	128	EMEA	Germany	<null>	Frankfurt	Blauer See Auto, Co.	Roland	Keitel	+49 69 61
8	129	NA	USA	CA	San Francisco	Mini Wheels Co.	Julie	Murphy	65055557
9	131	NA	USA	NY	NYC	Land of Toys Inc.	Kwai	Yu	21255578
10	141	EMEA	Spain	<null>	Madrid	Euro+ Shopping Channel	Diego	Freyre	(91) 555 1
11	144	EMEA	Sweden	<null>	Lulea00e5	Volvo Model Replicas, Co	Christina	Berghund	0921-12 1
12	145	EMEA	Denmark	<null>	Kobenhavn	Danish Wholesale Imports	Jytte	Petersen	31 12 355
13	146	EMEA	France	<null>	Lyon	Saveley & Henriot, Co.	Mary	Saveley	78.32.555
14	148	Japan	Singapore	<null>	Singapore	Dragon Souvenirs, Ltd.	Eric	Natividad	+65 221 1
15	151	NA	USA	NY	NYC	Muscle Machine Inc	Jeff	Young	21255574
16	157	NA	USA	PA	Allentown	Diecast Classics Inc.	Kyung	Yu	21555515
17	161	NA	USA	CA	Burlingame	Technics Stores Inc.	Juni	Hirano	65055568
18	166	APAC	Singapore	<null>	Singapore	Handji Gifts& Co	Wendy	Victorino	+65 224 1
19	167	EMEA	Norway	<null>	Bergen	Herkku Gifts	Veysel	Oeztan	+47 2267
20	171	EMEA	France	<null>	Lille	Daedalus Designs Imports	Martine	Rancu00e9	20.16.155
21	172	EMEA	France	<null>	Paris	La Corne D'abondance, Co.	Marie	Bertrand	(1) 42.34.
22	173	NA	USA	MA	Cambridge	Cambridge Collectables Co.	Kyung	Tseng	61755555
23	175	NA	USA	CT	Bridgewater	Gift Depot Inc.	Julie	King	20355525
24	177	Japan	Japan	Osaka	Osaka	Osaka Souvenirs Co.	Mory	Kentary	+81 06 61
25	181	NA	USA	NY	NYC	Vitachrome Inc.	Michael	Frick	21255515
26	186	EMEA	Finland	<null>	Helsinki	Toys of Finland, Co.	Matti	Karttunen	90-224 81
27	187	EMEA	UK	<null>	Manchester	AV Stores, Co.	Victoria	Ashworth	(171) 555
28	189	EMEA	Ireland	<null>	Dublin	Clover Collections, Co.	Dean	Cassidy	+353 186
29	198	NA	USA	MA	Brickhaven	Auto-Moto Classics Inc.	Leslie	Taylor	61755584
30	201	EMEA	UK	<null>	Liverpool	UK Collectables, Ltd.	Elizabeth	Devon	(171) 555

4.2. Descargar ficheros con Python

Para la realización de este ejemplo, primero hay que crear una serie de ficheros ([config.json](#), [sharepoint.py](#), [project.py](#)).

Config.json: fichero donde se establecen las propiedades de user, [password](#), url, site y doc_library.

```
{ } config.json > ...
1  {
2      "sharepoint":
3      {
4          "user": '          @stratebi.com',
5          "password": '          ',
6          "url": "https://stratebi.sharepoint.com",
7          "site": "https://stratebi.sharepoint.com/sites/test_integracion_pdi/",
8          "doc_library": "Documentos%20compartidos/Test"
9      }
10 }
```

sharepoint.py: fichero donde se definen las funciones necesarias.

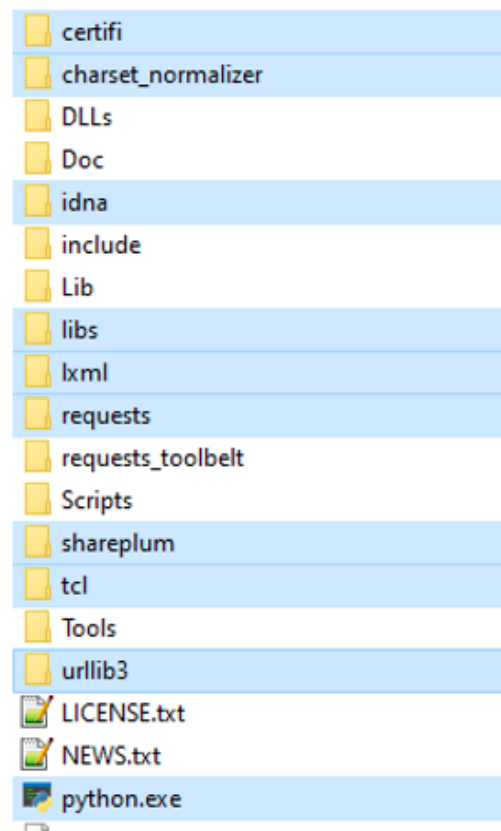
```
sharepoint.py > ...
1  from shareplum import Site, Office365
2  from shareplum.site import Version
3  import json, os
4
5  ROOT_DIR = os.path.dirname(os.path.abspath(__file__))
6  config_path = '\\'.join([ROOT_DIR, 'config.json'])
7
8  with open(config_path) as config_file:
9      config = json.load(config_file)
10     config = config['sharepoint']
11
12  USERNAME = config['user']
13  PASSWORD = config['password']
14  SHAREPOINT_URL = config['url']
15  SHAREPOINT_SITE = config['site']
16  SHAREPOINT_DOC = config['doc_library'] #not necessary if you don't work with folders
17
18  class SharePoint:
19      def auth(self):
20          self.authcookie = Office365(SHAREPOINT_URL, username= USERNAME, password= PASSWORD).GetCookies()
21          self.site = Site(SHAREPOINT_SITE, version=Version.v365, authcookie=self.authcookie)
22          return self.site
23
24      def connect_folder(self, folder_name):
25          self.auth_site = self.auth()
26          self.sharepoint_dir = '\\'.join([SHAREPOINT_DOC, folder_name])
27          self.folder = self.auth_site.Folder(self.sharepoint_dir)
28          return self.folder
29
30      def download_file(self, file_name, folder_name):
31          self._folder = self.connect_folder(folder_name)
32          return self._folder.get_file(file_name)
```

project.py: fichero principal del proyecto al que se llamará desde PDI para descargar el fichero necesario.

```
project.py > ...
1  from sharepoint import SharePoint
2
3  #set file name
4  file_name = 'Customers.csv'
5
6  #set folder name
7  folder_name = 'Python'
8
9  #get file
10 file = SharePoint().download_file(file_name, folder_name)
11
12 #save file
13 with open(file_name, 'wb') as f:
14     f.write(file)
15     f.close()
```


Para poder ejecutar correctamente el fichero **project.py**, necesitamos instalar “shareplum”. Este paso se puede realizar ejecutando *pip install shareplum*.

IMPORTANTE: comprobar que los módulos señalados en la siguiente imagen se encuentran en el mismo directorio que el ejecutable *Python.exe*. En caso de faltar alguno de estos módulos, copiarlos a dicho directorio para poder ejecutar el proceso ETL correctamente en PDI.



Una vez que tenemos todos los módulos correctamente ubicados, podemos pasar al proceso de PDI.

La estructura del **Job** de PDI es la siguiente:



En el paso de Download file, lo único que tenemos que especificar es la ruta donde se encuentra el fichero Python.exe y la ruta donde se encuentra el fichero project.py:

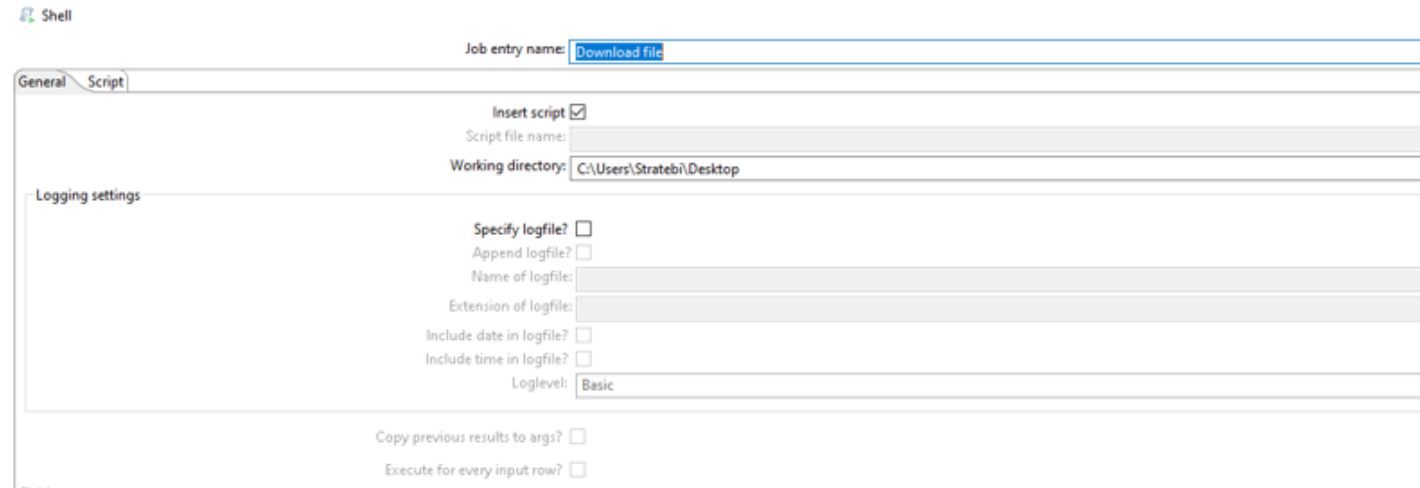
Shell

Job entry name: Download file

General Script

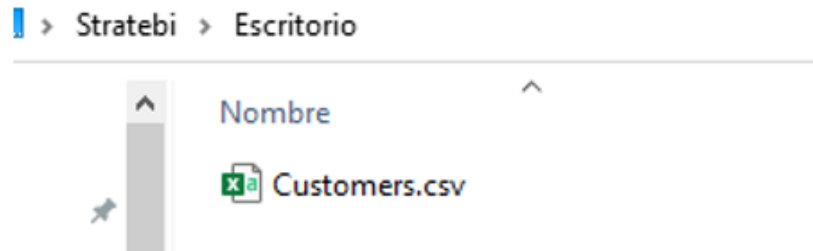
```
"C:\Users\Stratebi\AppData\Local\Programs\Python\Python310\python.exe" "C:\Users\Stratebi\OneDrive - stratebi.com\Escritorio\Python\project.py"
```

En la pestaña de General, podemos especificar la ruta de descarga del fichero en el campo Working Directory.



The screenshot shows the 'General' tab of a job configuration window. At the top, 'Job entry name' is set to 'Download file'. Below this, the 'Working directory' is specified as 'C:\Users\Stratebi\Desktop'. The 'Logging settings' section includes checkboxes for 'Specify logfile?', 'Append logfile?', 'Include date in logfile?', and 'Include time in logfile?', all of which are currently unchecked. The 'Loglevel' is set to 'Basic'. At the bottom, there are checkboxes for 'Copy previous results to args?' and 'Execute for every input row?', both also unchecked.

Tras la ejecución del Job, podemos observar que se encuentra el fichero de Sharepoint descargado en la ruta que hemos especificado en el campo correspondiente:



5. CONCLUSIONES

Como conclusiones, y tal y como se ha podido ver, es fácilmente viable la integración entre dos potentes herramientas, como son Sharepoint y Pentaho Data Integration.

Además, a partir de los ejemplos mostrados, se puede ver la flexibilidad para las diferentes posibilidades de uso en Sharepoint.

Espero que os haya sido de utilidad!!