



ORACLE APEX BLOGS — By Saravanan Ananth

How to handle API blob responses and download zip files with a loading indicator using APEX API and javascript in oracle APEX

 [saravananananth](#)

 [June 6, 2024](#)

 [Leave a comment](#)

1.Overview

This document explains about how to handle blob responses from application server/API and download in zip format using APEX API and javascript in oracle APEX.

2.Technologies and Tools Used

The following technology has been used to handle blob responses and download files in Zip archive with loading indicator using APEX API in Oracle APEX,

- Oracle Apex
- PL/SQL
- Javascript

3.Use Case

Privacy & Cookies: This site uses cookies. By continuing to use this website, you agree to their use.

To find out more, including how to control cookies, see here: [Cookie Policy](#)

Close and accept

expected functionality.

Using this approach, we can push all the responses receiving from the API/file server into array variable and convert into ZIP file using APEX API and custom javascript to maintain promises with loader widget.

4.Steps with Screenshot

Steps to be followed,

Step 1: Create required API to receive blob response and fetch that response using APEX API packages. Here I have used jasper API which will return the blob response from the application server.

Example URL:

[https://IP.HOST:PORT/JasperReportsIntegration/report?
& repName=bs& repFormat=pdf& dataSource=pd¶meter1=
1](https://IP.HOST:PORT/JasperReportsIntegration/report?& repName=bs& repFormat=pdf& dataSource=pd¶meter1=1)

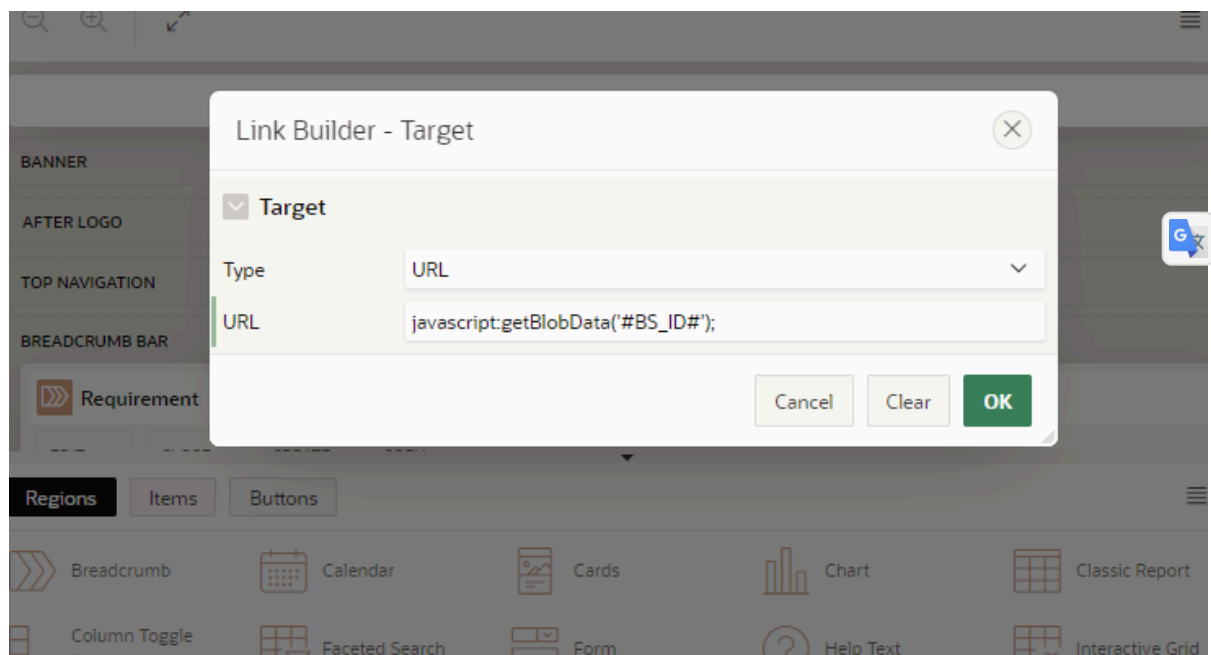
We can use any type of endpoint URL which will return blob response for processing.

Step 2: Create Interactive report with two columns, first column will hold the input parameter to send as an argument for API(end point URL) and second column will hold the link to call the custom function to fetch response and convert it as ZIP file archive.

Privacy & Cookies: This site uses cookies. By continuing to use this website, you agree to their use.

To find out more, including how to control cookies, see here: [Cookie Policy](#)

Id	Zip
1	
2	
3	
3	
4	
24	
44	



Link column target

Step 3: Paste the following code in Function and global variable declaration region(below code will show spinner until Zip generation process has been completed)

Code:

```
async function getBlobData(bsid){
var flag;
var popup = apex.widget.waitPopup();
var ajaxResponse=await apex.server.process('Download',{
```

Privacy & Cookies: This site uses cookies. By continuing to use this website, you agree to their use.

To find out more, including how to control cookies, see here: [Cookie Policy](#).

```
if (data.pie){
```

```

flag='Y';
popup.remove();
}
return flag;
}
});
if (ajaxResponse=='Y'){
popup.remove();
}
}
async function downloadBlob(id) {
var flag;
var popup = apex.widget.waitPopup();
try {
var data = await apex.server.process('Download', {
x01: bsid
});
if (data.file) {
var binaryData = atob(data.file);
var arrayBuffer = new ArrayBuffer(binaryData.length);
var uint8Array = new Uint8Array(arrayBuffer);
for (var i = 0; i < binaryData.length; i++) { uint8Array[i] =
binaryData.charCodeAt(i);

}

var blob=new Blob([uint8Array],

{ type: 'application/zip; charset=utf-8' });

const url = URL.createObjectURL(blob);

var newElement=document.createElement('a');

newElement.setAttribute('href',url);
newElement.setAttribute('download',bsid+'.zip');
document.body.appendChild(newElement);

```

Privacy & Cookies: This site uses cookies. By continuing to use this website, you agree to their use.

To find out more, including how to control cookies, see here: [Cookie Policy](#)

```

}

else { flag = 'N'; } }

catch (error)

{

console.error('Error occurred during server process:', error); flag= 'error';

}

finally {

popup.remove(); apex.message.showPageSuccess("ZIP Downloaded");

}

return flag;

}

async function down(lv_id){ var result=await
downloadBlob(v_id).then(result => {
console.log('Result:', result);
});
}

```

Step 4: Create AJAX callback process and paste the following code

Code:

declare

Privacy & Cookies: This site uses cookies. By continuing to use this website, you agree to their use.

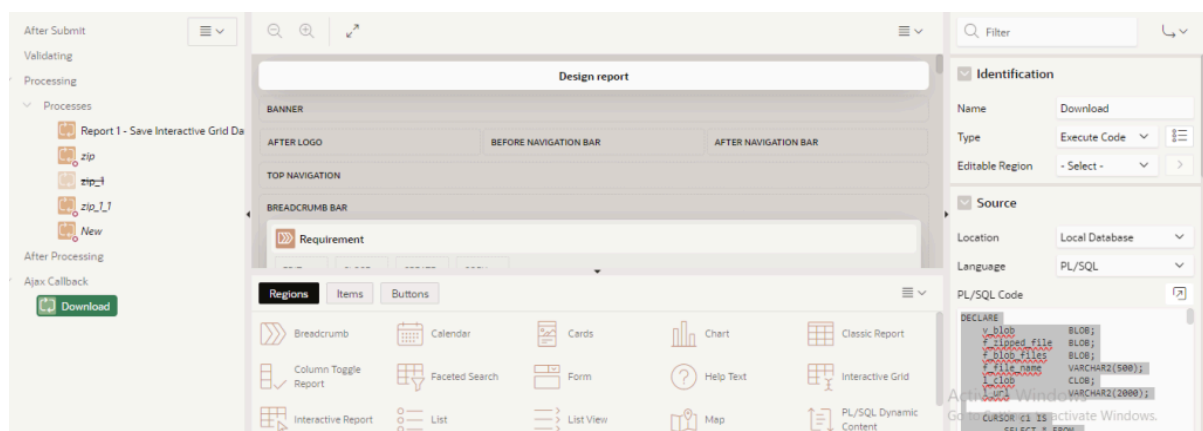
To find out more, including how to control cookies, see here: [Cookie Policy](#)

```
l_file_name varchar2(500),
```

```

l_clob clob;
l_url varchar2(2000);
cursor c1 is
select * from pdf_response where id = apex_application.g_x01;
begin
for i in c1 loop
v_blob := apex_web_service.make_rest_request_b(
p_url=> 'https://host:port/jasperreportsintegration/report?
&_repname=bs&_repformat=pdf&_datasource=pd&parameter1=' || id,
p_http_method => 'get'
);
apex_zip.add_file(
p_zipped_blob => f_zipped_file,
p_file_name => i.file_name,
p_content => v_blob
);
end loop;
apex_zip.finish(
p_zipped_blob => f_zipped_file
);
l_clob := apex_web_service.blob2clobbase64(f_zipped_file);
apex_json.open_object;
apex_json.write(p_name => file, p_value => l_clob);
apex_json.close_object;
exception
when others then
null;
end;

```



Privacy & Cookies: This site uses cookies. By continuing to use this website, you agree to their use.

To find out more, including how to control cookies, see here: [Cookie Policy](#)

1.zip - ZIP archive, unpacked size 8,272,196 bytes

Name	Size	Packed	Type	Modified	CRC32
..			File folder		
_1690374950573...	86,601	78,272	Adobe Acrobat Do...	31-05-2024 15:27	D4F9B7A6
_1690374950573...	86,601	78,274	Adobe Acrobat Do...	31-05-2024 15:27	A3AC7FF6
_1690374950573...	86,601	78,273	Adobe Acrobat Do...	31-05-2024 15:27	3597F26E
images.png-PDI...	4,232	3,241	Adobe Acrobat Do...	31-05-2024 15:27	61FCF1AC
images.png-TS...	4,232	3,242	Adobe Acrobat Do...	31-05-2024 15:27	9A2B9017
images.png-TS...	4,232	3,243	Adobe Acrobat Do...	31-05-2024 15:27	3802C9A3
MicrosoftTeams...	31,748	28,247	Adobe Acrobat Do...	31-05-2024 15:26	EA7F227B
MicrosoftTeams...	31,748	28,246	Adobe Acrobat Do...	31-05-2024 15:27	A7FF5765
MicrosoftTeams...	31,748	28,246	Adobe Acrobat Do...	31-05-2024 15:27	973550AA
OCID2023CP.jpg...	17,982	15,901	Adobe Acrobat Do...	31-05-2024 15:27	F4E2EF6A
OCID2023CP.jpg...	17,982	15,902	Adobe Acrobat Do...	31-05-2024 15:27	B4A35598
OCID2023CP.jpg...	17,982	15,901	Adobe Acrobat Do...	31-05-2024 15:27	D6266D96
pexels-suliman-...	1,949,424	1,883,726	Adobe Acrobat Do...	31-05-2024 15:27	8A223A5D
pexels-suliman-...	1,949,424	1,883,727	Adobe Acrobat Do...	31-05-2024 15:27	E2FF3818
pexels-suliman-...	1,949,424	1,883,727	Adobe Acrobat Do...	31-05-2024 15:27	B9BFB0FB
pexels-suliman-...	1,949,424	1,883,729	Adobe Acrobat Do...	31-05-2024 15:27	BDD2E466

Conclusion:

Above approach will help to download blob responses in Zip file format using combination of PL/SQL and Javascript, same can be achieved only using PL/SQL code by setting header, mimetype and file manipulation can be done by using inbuilt packages like DBMS_lob and wpg_docload to download blob contents. This can be written as centralized code in application process and called wherever required as URL. Since to show the wait notification(spinner) in the front end I have used javascript and APEX widget as part of this functionality.

Happy Coding...!!!

Leave a comment

ORACLE APEX BLOGS, ®

Privacy & Cookies: This site uses cookies. By continuing to use this website, you agree to their use.

To find out more, including how to control cookies, see here: [Cookie Policy](#)