

Manipulating VA reports using Viya APIs

Torben Skov, SAS Institute



The use case

1 base report

need to copy to 10 reports

and change 8 datasets every time

It will take forever using VA
(and I probably make a few mistakes along the way)

https://developer.sas.com/rest-apis

developer.sas.com/rest-apis

☆

{ sas }
deve/opers

APIs ▾

Open Source ▾

Programming ▾

Solutions ▾

Support ▾

I am looking for...

Sign in

🔍 Filter

Cadence

Any ▾

Categories

☐ Auto Machine Learning

☐ CAS

☐ Compute and Jobs

☐ Data Management

☐ Fraud and Compliance

☐ Health and Life Sciences

☐ Models and Decisions

☐ Platform Administration

☐ Visual Investigator

☐ Visualization and Repo...

REST APIs

SAS Viya REST APIs are designed for enterprise application developers who intend to build on the work of model builders and data scientists. With SAS REST APIs, you can create and access SAS resources using any client technology.

APIs are listed in alphabetical order. Filter the API list using the Categories menu.

Get Started 🔗

📖 Usage Notes

Authentication and Usage Notes

🔗 Examples

REST API GitHub Examples

🌐 SAS Developers Community

A forum for collaboration and knowledge sharin...

Browse REST APIs

👤 Annotations

The Annotations API allows users to associate annotations with resources that other decoupled API manage. An annotation augments a target resource, which is typically a column or table, with information that extends beyond it...

👤 Authorization

Manages authorization rules and makes authorization decisions.

const headers = {
 Accept: 'application/
 Accept-Language: 'st
 Accept-Locale: 'stri
};

fetch('https://www.sas.com/reports
{
 method: 'POST',

100
50
0

OneTwoThree

Copyright © SAS Institute Inc. All rights reserved.

Stuff that has reports

←

→

↺

🏠

🔍 developer.sas.com/rest-apis

☆

🔗

📄

📁

📄

👤

⋮

{ sas }

deve/opers

APIs ▾

Open Source ▾

Programming ▾

Solutions ▾

Support ▾

🗨️

I am looking for... 🔍

Sign in

🔍 Filter

Cadence

Any ▾

Categories

☐ Auto Machine Learning

☐ CAS

☐ Compute and Jobs

☐ Data Management

☐ Fraud and Compliance

☐ Health and Life Sciences

☐ Models and Decisions

☐ Platform Administration

☐ Visual Investigator

☐ Visualization and Repo...

The Reference Data API supports the life cycle of reference data.

📄

Report Images

This API is deprecated as of 2021.2.6 and will be removed in a future release.

📄

Report Transforms

This API creates and manipulates “transforms” of SAS reports that conform to the Business Intelligence Report Definition (BIRD) specification. A transform is a wrapper for a BIRD report. When a transform is part of a request, it ...

👤

SAS Logon

The SAS Logon API provides the standard OAuth protocol endpoints through which clients obtain access tokens to make API calls.

🔗

Score Execution

The Score Execution API is used to produce a score by executing the mapped code generated by score objects using the score definition.

🔗

Subject Contacts

The Subject Contacts API records contacts to subjects and records documents that are produced when executing decisions and business rules.

This API is deprecated as of 2023.10 and will be removed in a future release. The Relationships API manages the storage and discovery of relationships, and the retrieval of networks of relationship data.

📄

Reports

You can use the Reports API to create, read, and update reports. This API enables basic tasks such as moving or renaming reports, as well as editorial tasks such as reading, writing, and updating report content.

📄

Row Sets

The Row Sets API provides a common way to reference rows of rectangular data that are available in the SAS ecosystem. The URLs for this API are the only canonical form of storing data references for SAS Viya.

🔗

Score Definitions

The Score Definitions API is used for creating and maintaining score definitions.

📄

SAS Studio Development

SAS Studio API

🔍

Visual Investigator Alerts

The Visual Investigator Alerts API supports managing alerts in SAS Visual Investigator.




Report specific stuff

Report [🔗](#)

The operations for the report resource.

<div><div>Get</div><div>Get a collection of reports</div></div> <div>Returns a collection of reports with standard paging, filtering, and sorting options.</div> <div>/reports</div>	<div><div>Post</div><div>Create report</div></div> <div>Creates a new report and adds it to a folder as a child.</div> <div>/reports</div>
<div><div>Head</div><div>Check report status</div></div> <div>Returns the headers for a report, including ETag. See Conditional Operations.</div> <div>/reports/{reportId}</div>	<div><div>Get</div><div>Get a report</div></div> <div>Returns the specified report, including ETag header. See Conditional Operations.</div> <div>/reports/{reportId}</div>
<div><div>Head</div><div>Check report resource summary status</div></div> <div>Returns the headers for a report resource summary, including modifiedTimeStamp.</div> <div>/reports/{reportId}</div>	<div><div>Get</div><div>Get resource summary report</div></div> <div>Returns the specified report in resource summary format.</div> <div>/reports/{reportId}</div>
<div><div>Delete</div><div>Delete a report</div></div> <div>Deletes the specified report and its content. Remove the report from the parent folder.</div> <div>/reports/{reportId}</div>	<div><div>Put</div><div>Update a report</div></div> <div>Updates the specified report. Requires an If-Match or If-Unmodified-Since request header. See Conditional Operations.</div> <div>/reports/{reportId}</div>
<div><div>Post</div><div>Validate report name</div></div> <div>Ensures that the name does not exceed the maximum length and, if parentFolderUri is provided, that a report with the name can be added as a member without conflict. The ...</div>	

The report/report endpoint

 APIs ▾ Open Source ▾ Programming ▾ Solutions ▾

REST APIs > Visualization and Reports > Reports > **getReport**

Get a report

Get

/reports/{reportId}

Returns the specified report, including **ETag** header. See [Conditional Operations](#).

Request Samples [↗](#)

Python - Python 3 ▾

```
1 import http.client
2
3 conn = http.client.HTTPConnection("example.com")
4
5 headers = {
6     'If-None-Match': "",
7     'If-Modified-Since': "",
8     'Authorization': "Bearer <access-token-goes-here>",
9     'Accept': "application/json, application/vnd.sas.report+json, application/vnd.sas.error+json"
10 }
11
12 conn.request("GET", "/reports/reports/STRING_VALUE", headers=headers)
13
14 res = conn.getresponse()
15 data = res.read()
16
17 print(data.decode("utf-8"))
```

Response Samples [↗](#)

200 ▾

application/json ▾

```
1 {
2     "id": "f8b66d4b-d67a-4e4e-a66d-b6f06b1820bf",
3     "name": "Sample Report",
4     "description": "Description of a sample report.",
```

https://go.documentation.sas.com/doc/en/pgmsascd/c/9.4_3.5/proc/p1dxmryomzzddkn1t8ljx3pcaxk2.htm

SAS® Help Center

SAS® 9.4 and SAS® Viya® 3.5 Programming DocumentationSAS 9.4 / Viya 3.5

▸ HADOOP Procedure

▸ HDMD Procedure

▾ HTTP Procedure

PROC HTTP Statement

DEBUG Statement

HEADERS Statement

SSLPARMS Statement

Overview: HTTP Procedure

▸ Usage: HTTP Procedure

▾ Examples: HTTP Procedure

Example 1: A Simple GET Request

Example 2: A Simple POST Request

Example 3: Specify a Proxy In the HTTP Request

Example 4: Specifying Input Data as a String

Example 5: Specify A Proxy In a Macro Variable

Example 6: A POST That Contains the

Base SAS Procedures Guide

SyntaxOverviewUsageExamples

Example 1: A Simple GET Request

Features:

METHOD= argument

URL= Argument

OUT= Argument

Details

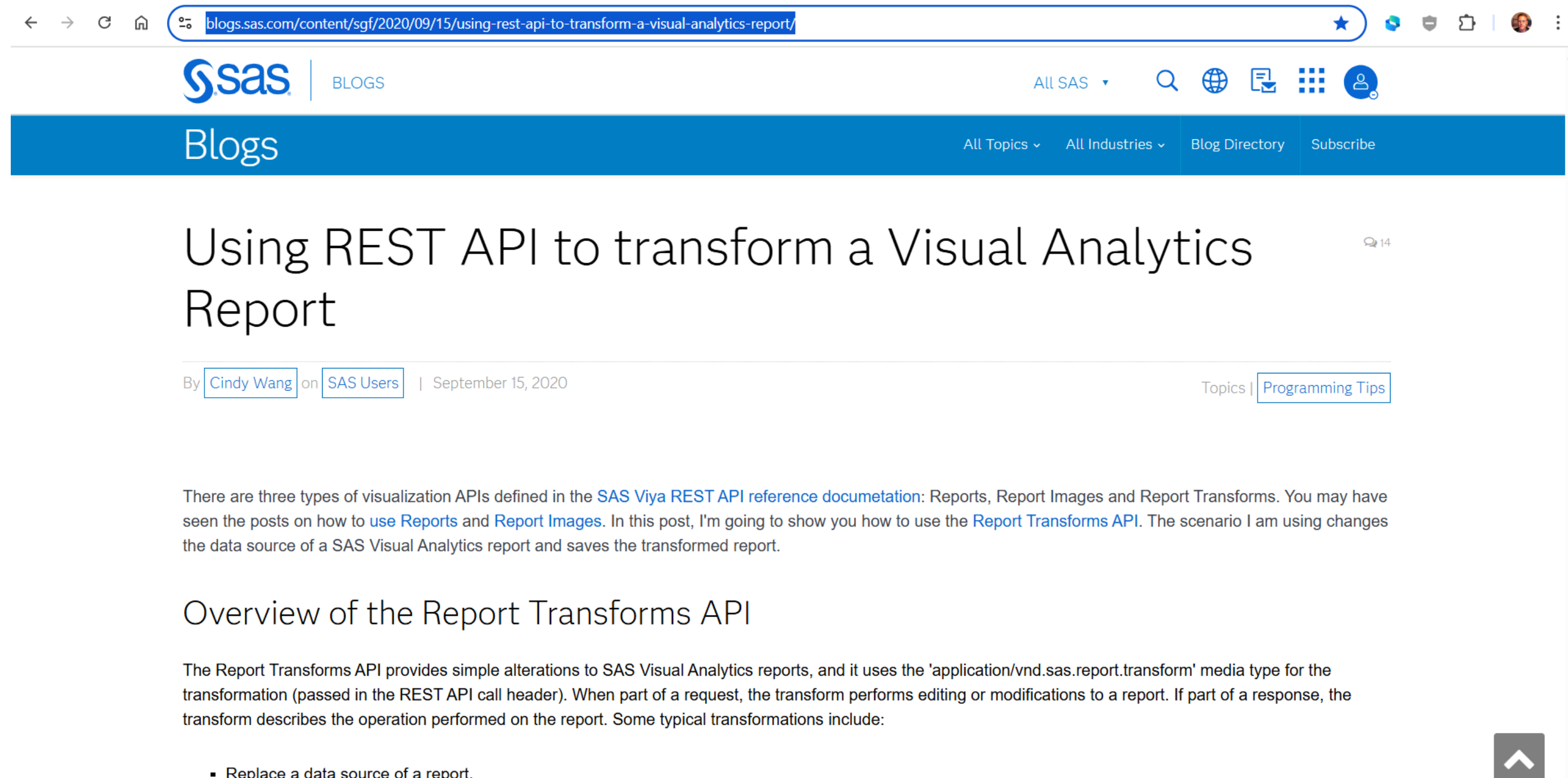
This example makes a GET request to a server on the local network. GET is the simplest and most common request that you can make with PROC HTTP. The METHOD= argument is required, and you must specify a METHOD value when the IN argument is omitted for a PROC HTTP request, making the argument optional. A GET request must specify METHOD=GET.

Program

```
filename resp TEMP;

proc http
  url="http://httpbin.org/get"
  out=resp;
run;
```

https://blogs.sas.com/content/sgf/2020/09/15/using-rest-api-to-transform-a-visual-analytics-report/



The screenshot shows a web browser displaying the SAS Blogs page. The address bar shows the URL: blogs.sas.com/content/sgf/2020/09/15/using-rest-api-to-transform-a-visual-analytics-report/. The page header includes the SAS logo, the word "BLOGS", and navigation links: "All SAS", "All Topics", "All Industries", "Blog Directory", and "Subscribe". The main title of the article is "Using REST API to transform a Visual Analytics Report", with a comment count of 14. The author is Cindy Wang, writing on behalf of SAS Users, dated September 15, 2020. The topic is categorized as "Programming Tips". The article text begins with: "There are three types of visualization APIs defined in the [SAS Viya REST API reference documentation](#): Reports, Report Images and Report Transforms. You may have seen the posts on how to [use Reports](#) and [Report Images](#). In this post, I'm going to show you how to use the [Report Transforms API](#). The scenario I am using changes the data source of a SAS Visual Analytics report and saves the transformed report." The section "Overview of the Report Transforms API" follows, stating: "The Report Transforms API provides simple alterations to SAS Visual Analytics reports, and it uses the 'application/vnd.sas.report.transform' media type for the transformation (passed in the REST API call header). When part of a request, the transform performs editing or modifications to a report. If part of a response, the transform describes the operation performed on the report. Some typical transformations include:" A list item is visible: "▪ Replace a data source of a report."

By [Cindy Wang](#) on [SAS Users](#) | September 15, 2020

Topics | [Programming Tips](#)

There are three types of visualization APIs defined in the [SAS Viya REST API reference documentation](#): Reports, Report Images and Report Transforms. You may have seen the posts on how to [use Reports](#) and [Report Images](#). In this post, I'm going to show you how to use the [Report Transforms API](#). The scenario I am using changes the data source of a SAS Visual Analytics report and saves the transformed report.

Overview of the Report Transforms API

The Report Transforms API provides simple alterations to SAS Visual Analytics reports, and it uses the 'application/vnd.sas.report.transform' media type for the transformation (passed in the REST API call header). When part of a request, the transform performs editing or modifications to a report. If part of a response, the transform describes the operation performed on the report. Some typical transformations include:

- Replace a data source of a report.

https://github.com/sascommunities/sas-users-blog/blob/master/va-reports-transform/report_transform_datasource.sas

The screenshot displays a GitHub web interface for the repository `sascommunities/sas-users-blog`. The browser's address bar shows the URL `github.com/sascommunities/sas-users-blog/blob/master/va-reports-transform/report_transform_datasource.sas`. The repository page includes a navigation bar with links to Platform, Solutions, Resources, Open Source, Enterprise, and Pricing. Below this, the repository name and a 'Public' badge are shown, along with buttons for Notifications, Fork (13), and Star (6). A secondary navigation bar contains links for Code, Issues, Pull requests, Actions, Projects, Security, and Insights. The left sidebar shows a file explorer with a list of folders, including 'HLA', 'calculator-regression-model-deci...', 'create-excel-report-using-sas-pyt...', 'developer.sas.com_2.0', 'empirical-mode-decomposition', 'excel-python-sas-viya', 'hotfix-management', 'improving-detection-level-shifts-...', 'r-studio-with-sas-viya', 'sas-viya-job-execution-service-au...', and 'va-reports-transform'. The main content area shows the file `report_transform_datasource.sas` with a commit message by `jefurbee`: 'Adding report transform API code for Cindy Wang's blog post'. The code is displayed in a light blue theme, showing SAS macro and PROC HTTP code. The code defines a macro `%let BASE_URI=%sysfunc(getoption(SERVICESBASEURL));` and a PROC HTTP block that sends a POST request to a specific URL with a JSON body containing report and data source information.

github.com/sascommunities/sas-users-blog/blob/master/va-reports-transform/report_transform_datasource.sas

Platform Solutions Resources Open Source Enterprise Pricing

Search or jump to... Sign in Sign up

sascommunities / sas-users-blog Public

Notifications Fork 13 Star 6

Code Issues Pull requests Actions Projects Security Insights

Files

master

Go to file

HLA

calculator-regression-model-deci...

create-excel-report-using-sas-pyt...

developer.sas.com_2.0

empirical-mode-decomposition

excel-python-sas-viya

hotfix-management

improving-detection-level-shifts-...

r-studio-with-sas-viya

sas-viya-job-execution-service-au...

va-reports-transform

sas-users-blog / va-reports-transform / report_transform_datasource.sas

jefurbee Adding report transform API code for Cindy Wang's blog post 6f18c95 · 5 years ago History

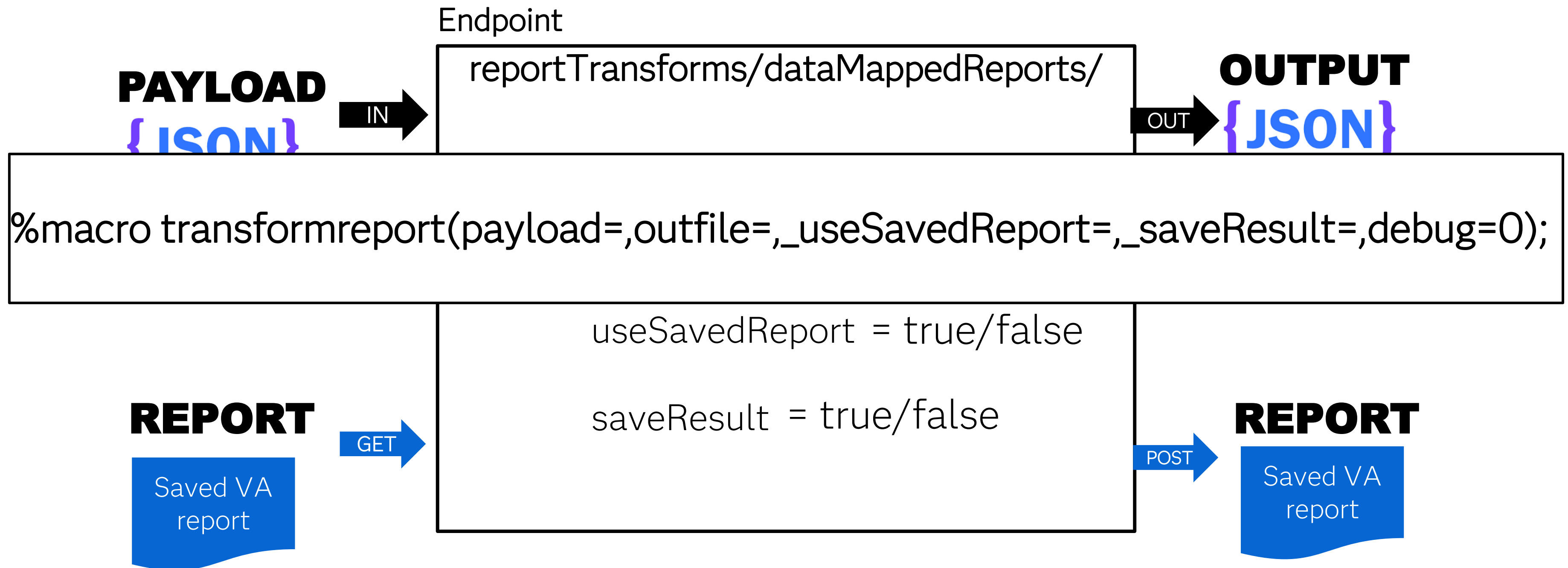
Code Blame 82 lines (77 loc) · 2.15 KB

Raw Copy Download Edit

```
1 %let BASE_URI=%sysfunc(getoption(SERVICESBASEURL));
2 FILENAME tranFile TEMP ENCODING='UTF-8';
3 FILENAME hdrout TEMP ENCODING='UTF-8';
4
5 /* transform the datasource of a report */
6 /* use the tranFile to hold the response body */
7 PROC HTTP METHOD="POST" oauth_bearer=sas_services OUT=tranFile headerout=hdrout
8     URL = "&BASE_URI/reportTransforms/dataMappedReports/?useSavedReport=true&saveResult=true"
9     IN = '
10         {
11             "inputReportUri": "/reports/reports/4e72e34a-f691-46d2-9c5d-859bf6d41d7d",
12             "dataSources": [
13                 {
14                     "namePattern": "serverLibraryTable",
15                     "purpose": "original",
```

Automating it all

Using the endpoint



Constructing the payload

Building blocks

```
"resultReportName": "Skov01",
"resultParentFolderUri": "/folders/folders/71108b56-74a2-4512-bd26-0b7da396b9d6",
"resultReport": {
  "name": "Skov01",
  "description": "TEST report transform"
```

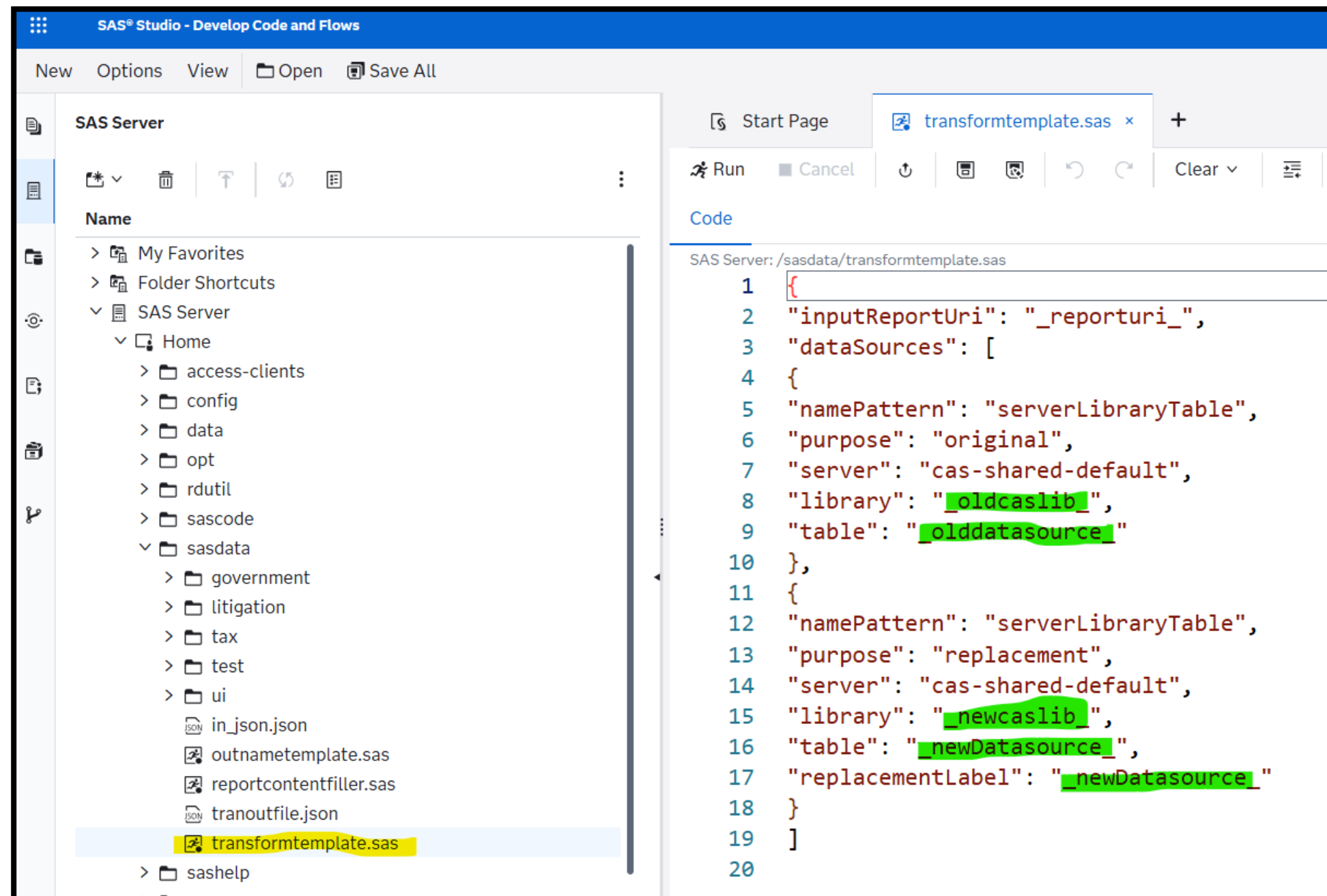
```
"inputReportUri": "/reports/reports/8c821231-9b31-4fe4-94be-e2f680e9dca6"
```

```
"dataSources": [
{
  "namePattern": "serverLibraryTable",
  "purpose": "original",
  "server": "cas-shared-default",
  "library": "public",
  "table": "air"
},
{
  "namePattern": "serverLibraryTable",
  "purpose": "replacement",
  "server": "cas-shared-default",
  "library": "public",
  "table": "air2",
  "replacementLabel": "air2"
}
]
```

```
"reportContent": {
  "@element": "SASReport",
  "xmlns": "http://www.sas.com/sasreportmodel/bird-4.57.0",
  "label": "Basereport",
  "dateCreated": "2025-11-07T07:51:11Z",
  "createdApplicationName": "SAS Visual Analytics",
  "dateModified": "2025-11-17T15:38:39Z",
  "lastModifiedApplicationName": "SAS Visual Analytics",
  "createdVersion": "4.57.0",
  "createdLocale": "en",
  "features": [
    "promptModelV2"
  ],
  "implicitInteractions": [
```

File templates for the different building blocks

Saved template



Read template and substitute

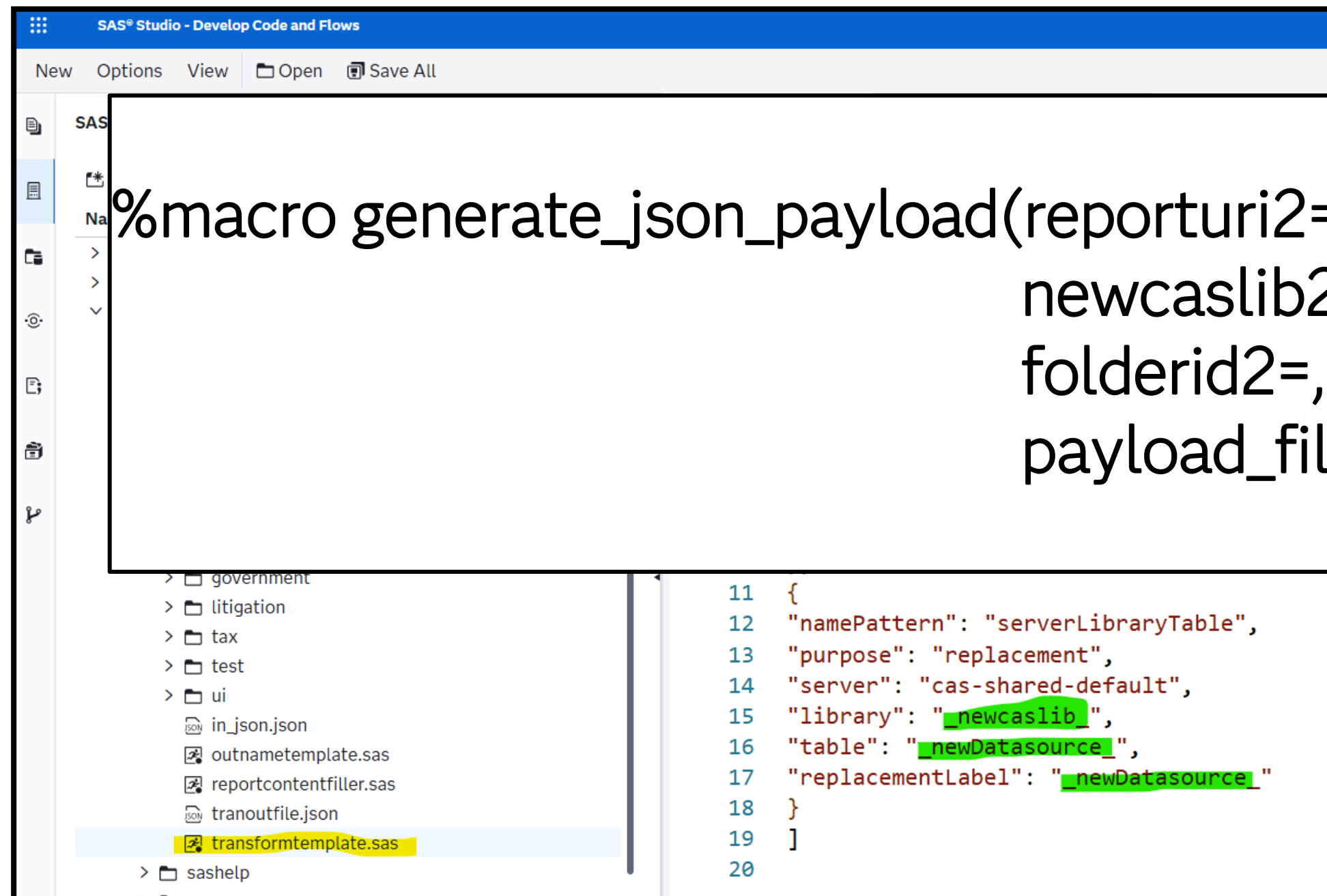
```
filename i_tmplte "&filepath./transformtemplate.sas" ENCODING='UTF-8';  
  
data inputreport;  
    length line $32000;  
    infile i_tmplte truncover;  
    INPUT line $32000. ;  
    line=tranwrd(line, "_reporturi_", "&reporturi2");  
    line=tranwrd(line, "_oldcaslib_", "&oldcaslib2");  
    line=tranwrd(line, "_olddatasource_", "&olddatasource2");  
    line=tranwrd(line, "_newcaslib_", "&newcaslib2");  
    line=tranwrd(line, "_newdatasource_", "&newdatasource2");  
    run;  
  
filename i_tmplte clear;
```

Collect all parts and write payload

```
data _null_;  
    set inputreport namereport reportcontent close;  
    file &payload_fileref.;  
    put line;  
    run;
```

File templates for the different building blocks

Saved template



Read template and substitute

```
filename i_tmplte "&filepath./transformtemplate.sas" ENCODING='UTF-8';  
data inputreport;
```

```
%macro generate_json_payload(reporturi2=,oldcaslib2=,olddatasource2=,  
                             newcaslib2=,newDatasource2=,  
                             folderid2=,newreportname2=,  
                             payload_fileref=,_reportcontentfid=);
```

payload

```
11 {  
12 "namePattern": "serverLibraryTable",  
13 "purpose": "replacement",  
14 "server": "cas-shared-default",  
15 "library": "_newcaslib_",  
16 "table": "newDatasource_",  
17 "replacementLabel": "_newDatasource_"  
18 }  
19 ]  
20
```

```
data _null_;  
    set inputreport namereport reportcontent close;  
    file &payload_fileref.;  
    put line;  
run;
```


Get report content of the json output

Extreemly
Long JSON

```
* read the file byte by byte and insert line breaks for every , (not in quotes;)
```

```
data _null_;
```

```
infile &infile ref recfm=n;
```

```
file outtmp recfm=n;
```

```
retain quotes 0;
```

```
input char $char1;
```

```
if char = '"' then
```

```
else if char =
```

```
put char $char1;
```

```
*only if we are
```

```
if char=',' and
```

```
run;
```

```
* read the new file line by line - output only after we encounter reportContent
```

```
* (Hope the user
```

```
data _null_;
```

```
infile outtmp
```

```
file outtmp2
```

```
retain output
```

```
INPUT line $3
```

```
if outputflag
```

```
if idx > 0 or
```

```
outputfla
```

```
put line;
```

```
end;
```

```
run;
```

```
* read the next file until with have everything between balanced {};
```

```
data _null_;
```

```
infile outtmp2 recfm=n;
```

```
file &outfile ref recfm=n;
```

```
input char $char1. ;
```

```
retain outputflag 0 curlybrackets -1;
```

```
if curlybrackets < 0 and char = '{' then do;
```

```
outputflag=2;
```

```
curlybrackets=0;
```

```
end;
```

```
if outputflag > 0 then do;
```

```
if char = '{' then curlybrackets=curlybrackets+1;
```

```
if char = '}' then curlybrackets=curlybrackets-1;
```

```
end;
```

```
if outputflag > 0 then put char $char1. ;
```

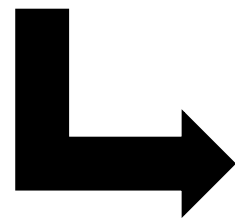
```
if curlybrackets = 0 then outputflag=0;
```

```
run;
```

reportContent JSON (only)

Get report content of the json output

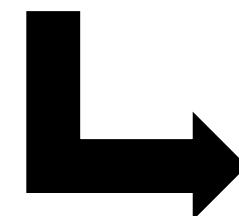
Extreemly
Long JSON



```
* read the file byte by byte and insert line breaks for every , (not in quotes;)
```

```
%macro get_report_content_from_output(infileref=, outfileref=);
```

```
run;
```



reportContent JSON (only)

Macros high-level

```
%transformreport(payload=json_in,  
                 outfile=json_out,  
                 _useSavedReport=true/false,_saveResult=true/false)
```

```
%get_report_content_from_output(infileref=json_out, outfileref=rcontent);
```

```
%generate_json_payload(_reportcontentfid=rcontent,  
                      payload_fileref=json_in,  
                      reporturi2=,  
                      oldcaslib2=,olddatasource2=,  
                      newcaslib2=newDatasource2=,  
                      folderid2=newreportname2= );
```

Automating

Transformation scenarios

Only one dataset to change

Read from saved report, transform, Save new report

Two or more datasets to change

First: Read from saved report, transform, return json

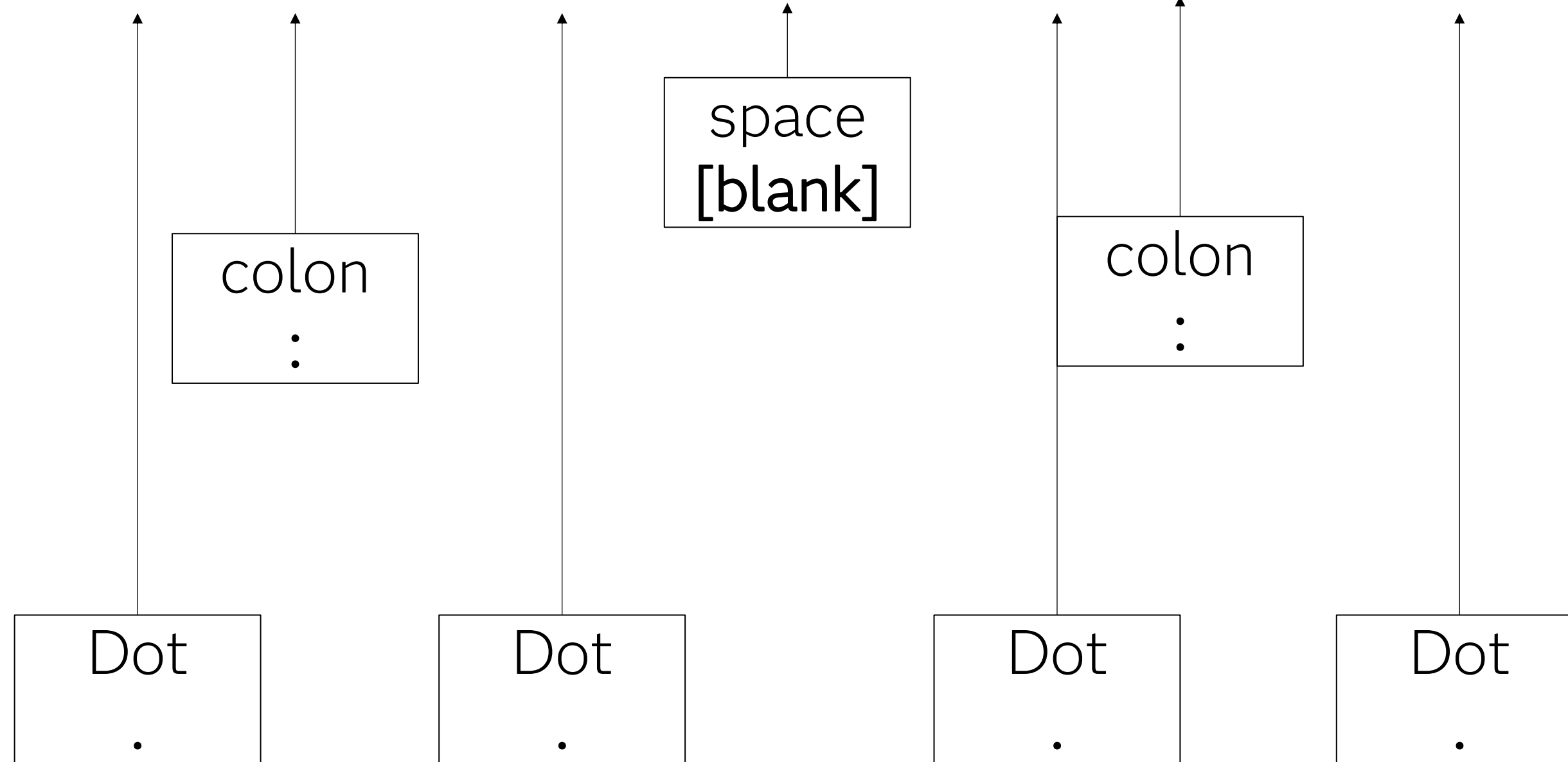
 get transformed content from json, change content, return json

Last: get transformed content from json, transform, save to new report

Data driven

Input:

public.car:public.cars1 public.air:public.air1



Split macro

Loop logic

```
%macro split_macro(substitutions=);
```

```
%let My_count= %eval(%sysfunc(count(&substitutions,%str(:))));
```

 ← Count colons → #substitutions to make

```
%do i=1 %to &My_count; ← Loop
```

```
%if &i = 1 and &i = &My_count %then %do;
```

```
%put read saved => transform => save report
```

```
%end;
```

```
%else %if &i = 1 %then %do;
```

```
%put read saved => transform => json
```

```
%end;
```

```
%else %if &i < &My_count %then %do;
```

```
%put Json => transform => json
```

```
%end;
```

```
%else %if &i = &My_count %then %do;
```

```
%put Json => transform => save report
```

```
%end;
```

```
%end;
```

```
%mend;
```

Loop logic

Actual code

```
%macro split_macro(substitutions=);
```

```
%let My_count= %eval(%sysfunc(count(&substitutions,%str(:))));
```

```
%do i=1 %to &My_count;
```

```
  %if &i = 1 and &i = &My_count %then %do;
```

```
    %put read saved => transform => save report
```

```
  %end;
```

```
%if &i = 1 and &i = &My_count %then %do;
```

```
  %put Only one dataset to transform;
```

```
  %if &debug > 0 %then filename json_in "&filepath./in_json.json" ENCODING='UTF-8'%STR(;;);
```

```
  %else filename json_in TEMP ENCODING='UTF-8'%STR(;;);
```

```
  filename json_out TEMP ENCODING='UTF-8';
```

```
  %generate_json_payload(reporturi2=&reporturi,
```

```
                        oldcaslib2=&oldcaslib,olddatasource2=&olddatasource,newcaslib2=&newcaslib,newDatasource2=&newDatasource,
```

```
                        folderid2=&newfolderuri,
```

```
                        newreportname2=&newreportname,
```

```
                        payload_fileref=json_in);
```

```
  %transformreport(payload=json_in,outfile=json_out,_useSavedReport=true,_saveResult=true);
```

```
  filename json_in clear;
```

```
  filename json_out clear;
```

```
%end;
```

Loop logic

Actual code

```
%macro split_macro(substitutions=);
```

```
%let My_count= %eval(%s
```

```
%do i=1 %to &My_count.
```

```
%if &i = 1 and &i = &M
```

```
%put read saved =>
```

```
%end;
```

```
%else %if &i = 1 %then
```

```
%put read saved =>
```

```
%end;
```

```
%else %if &i < &My_co
```

```
%put Jason => trans
```

```
%end;
```

```
%else %if &i = &My_co
```

```
%put Json => transf
```

```
%end;
```

```
%end;
```

```
%mend;
```

```
%else %if &i = 1 %then %do;  
  %put More than one and this is the first;  
  
  %if &debug > 0 %then filename json_in "&filepath./in_json&i..json" ENCODING='UTF-8'%STR(;;)  
  %else filename json_in TEMP ENCODING='UTF-8'%STR(;;)  
  filename json_out TEMP ENCODING='UTF-8';  
  
  %generate_json_payload(reporturi2=&_reporturi_,oldcaslib2=&oldcaslib,olddatasource2=&olddatasource,  
    newcaslib2=&newcaslib,newDatasource2=&newDatasource,payload_fileref=json_in);  
  
  %transformreport(payload=json_in,outfile=json_out,_useSavedReport=true,_saveResult=false);  
  
  %if &debug > 0 %then filename rcontent "&filepath./reportcontent_json&i..json" ENCODING='UTF-8'%STR(;;)  
  %else filename rcontent TEMP ENCODING='UTF-8'%STR(;;)  
  
  %get_report_content_from_output(infileref=json_out, outfileref=rcontent);  
  
  filename json_in clear;  
  filename json_out clear;  
  
%end;
```


Loop logic

Actual code

```
%macro split_macro(substitutions=);
```

```
%let My_count= %eval(%sysfunc(count(&substitutions,%str(:))));
```

```
%do i=1 %to &My_count;
```

```
%if &i = 1 and &i = &My_count
```

```
%put read saved => tran
```

```
%end;
```

```
%else %if &i = 1 %then %do
```

```
%put read saved => tran
```

```
%end;
```

```
%else %if &i < &My_count
```

```
%put Jason => transform
```

```
%end;
```

```
%else %if &i = &My_count
```

```
%put Json => transform
```

```
%end;
```

```
%end;
```

```
%mend;
```

```
%end;  
%else %if &i < &My_count %then %do;  
  %put More than one and this is NOT the last;  
  
  %if &debug > 0 %then filename json_in "&filepath./in_json&i..json" ENCODING='UTF-8'%STR;;  
  %else filename json_in TEMP ENCODING='UTF-8'%STR;;  
  filename json_out TEMP ENCODING='UTF-8';  
  
  %generate_json_payload(reporturi2=&_reporturi_,oldcaslib2=&oldcaslib,olddatasource2=&olddatasource,  
    newcaslib2=&newcaslib,newDatasource2=&newDatasource,  
    payload_fileref=json_in,  
    _reportcontentfid=rcontent);  
  
  %transformreport(payload=json_in,outfile=json_out,_useSavedReport=false,_saveResult=false);  
  filename rcontent clear;  
  %if &debug > 0 %then filename rcontent "&filepath./reportcontent_json&i..json" ENCODING='UTF-8'%STR;;  
  %else filename rcontent TEMP ENCODING='UTF-8'%STR;;  
  
  %get_report_content_from_output(infileref=json_out, outfileref=rcontent);  
  
  filename json_in clear;  
  filename json_out clear;  
%end;
```

Loop logic

Actual code

```
%macro split_macro(substitutions=);
```

```
%let My_count= %eval(%sysfunc(count(0,substitutions,0,sub(,))));
```

```
%do i=1 %to &My_count.;
```

```
  %if &i = 1 and &i = &My_count %then
```

```
    %put read saved => transform
```

```
%end;
```

```
%else %if &i = 1 %then %do;
```

```
  %put read saved => transform
```

```
%end;
```

```
%else %if &i < &My_count %then
```

```
  %put Jason => transform => js
```

```
%end;
```

```
%else %if &i = &My_count %then
```

```
  %put Json => transform => sav
```

```
%end;
```

```
%end;
```

```
%mend;
```

```
%else %if &i = &My_count %then %do;
  %put More than one and this is the last;

  %if &debug > 0 %then filename json_in "&filepath./in_json&i..json" ENCODING='UTF-8'%STR(;);
  %else filename json_in TEMP ENCODING='UTF-8'%STR(;);

  filename json_out TEMP ENCODING='UTF-8';

  %generate_json_payload(reporturi2=&_reporturi_,oldcaslib2=&oldcaslib,olddatasource2=&olddatasource,
    newcaslib2=&newcaslib,newDatasource2=&newDatasource,
    folderid2=&newfolderuri,
    newreportname2=&newreportname,payload_fileref=json_in,
    reportcontentfid=rcontent);

  %transformreport(payload=json_in,outfile=json_out,_useSavedReport=false,_saveResult=true);

  filename rcontent clear;
  filename json_in clear;
  filename json_out clear;

%end;
```

Trying it out

```
%changeDataSource(reportname=/Public/Report API/Basereport  
  ,newlocation=/Public/Report API  
  ,newreportname=Skov01  
  ,substitutions=public.cars:public.cars2)
```

```
%changeDataSource(reportname=/Public/Report API/Basereport  
  ,newlocation=/Public/Report API  
  ,newreportname=Skov02  
  , substitutions=public.cars:public.cars2 public.class:public.class_bmi);
```

```
%changeDataSource(reportname=/Public/Report API/Basereport  
  ,newlocation=/Public/Report API  
  ,newreportname=Skov03  
  ,substitutions=public.cars:public.cars2 public.class:public.class_bmi public.air:public.air2);
```

