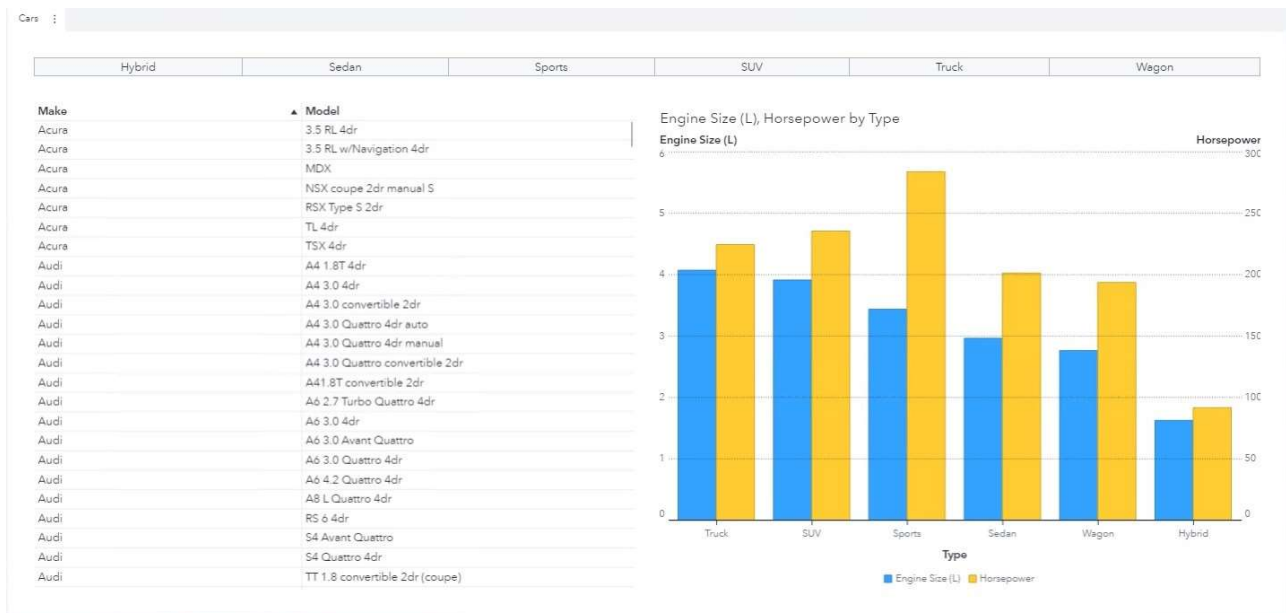


Viya Rest-API: Parametrisieren und Verteilen von VA-Berichten

Mike Heidemann - Technical Architect

Vorbereitung

Initialer Report

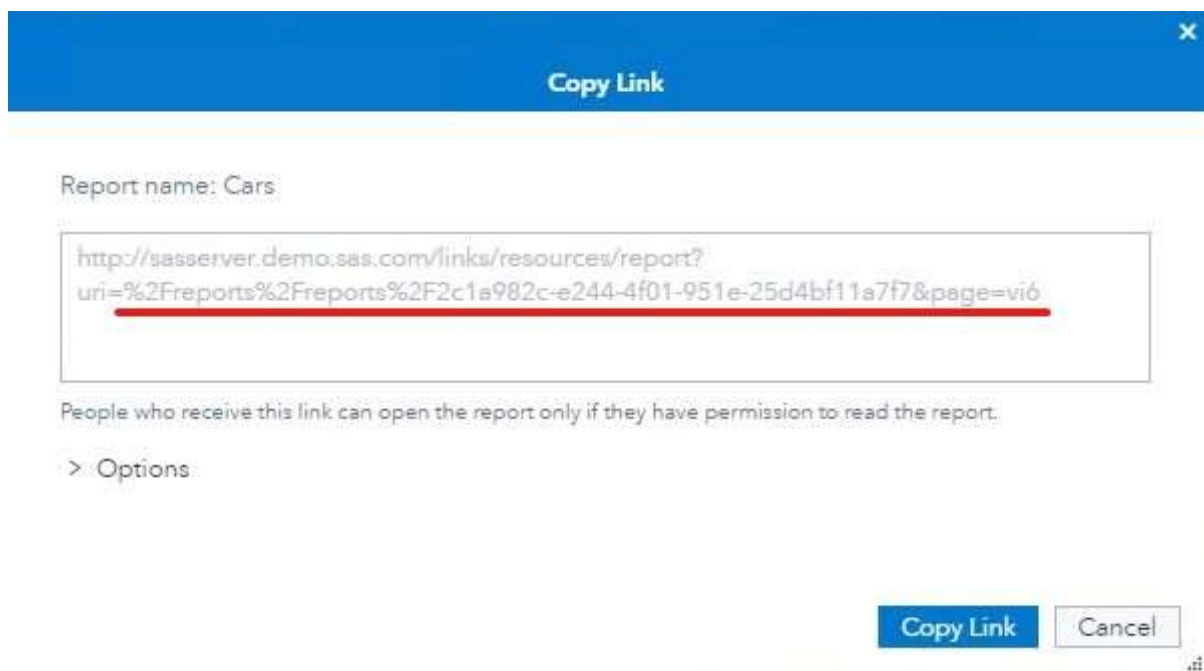


Auslesen der Report-URI

Eine einfache Methode die Report-URI auszulesen wird durch den Report-Viewer bereitgestellt. Einfach im Menü oben rechts den Punkt "Copy Link" auswählen:



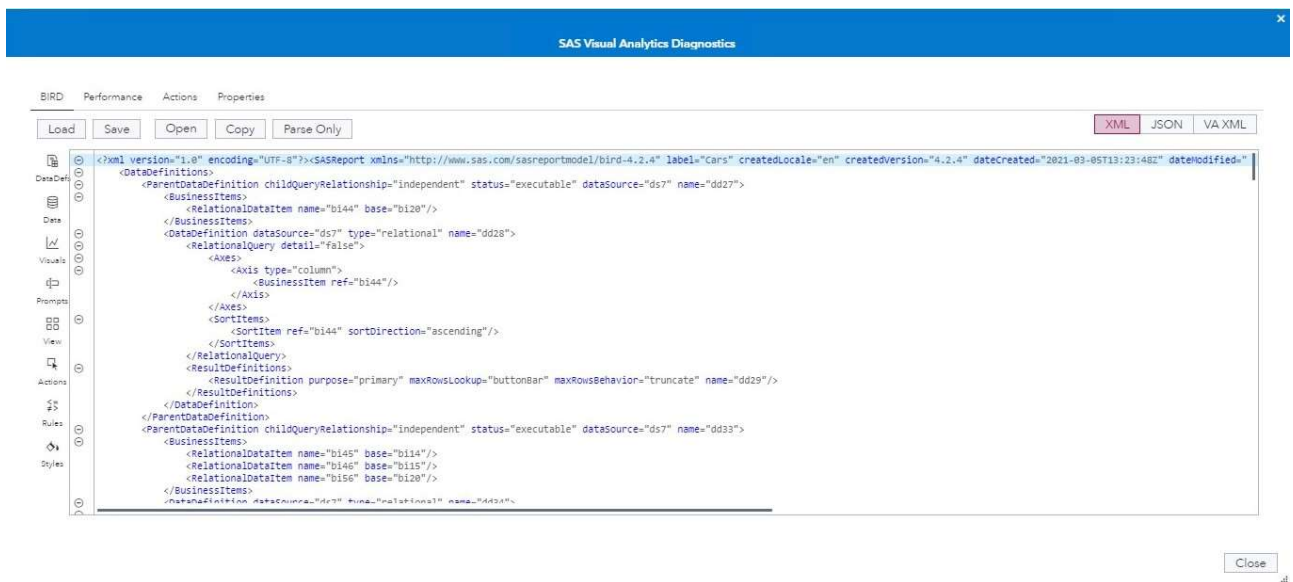
Im darauf erscheinenden Dialog kann die Report-URI leicht ausgelesen werden. Achtung: Die "/" werden URL-Decoded als "%2F" angezeigt



/reports/reports/2c1a982c-e244-4f01-951e-25d4bf11a7f7

Anmerkung: Bei der oben angezeigten Report-URI ist bereits die Seite "page=vi6" mitgegeben. Diese kann auch unter "Views" in Bird-Editor ausgelesen werden

Damit später Parameter von außen angesteuert werden können, benötigen wir dessen interne Namen. Diese können über die Oberfläche z. B. mit dem Bird-Editor ausgelesen werden. Dazu bitte die Tastenkombination "Ctrl+Alt+B" drücken während der Bericht geöffnet ist.



Über die linken Abschnittsflächen kann zu dem "Prompts" Abschnitt gewechselt werden, und dort die am oberen Bildschirmrand befindliche "ButtonBar" mit dessen internen Namen ausgelesen werden.



Der Prompt hat in diesem Beispiel den Namen "pr30"

Hands-On

Welche Schritte werden wir durchführen:

1. Anmelden um einen Authentication Token zu erhalten
2. Eine PDF-Style Definition erstellen
3. Ein "ReportPackage" unseres Reports erstellen
4. Mit Hilfe des "ReportPackage" ein PDF generieren und herunterladen
5. Ein weiteres "ReportsPackage" unseren Reports erstellen, diesmal mit einer Vorfilterung über die "ButtonBar"
6. Ein weiteres PDF auf Basis des zweiten "ReportPackage" erstellen

Erzeugen und abrufen eines Baseline-PDF Berichtes

Für die Authentifizierung verwenden wir das "sas-admin-cli" und kopieren den Authentifizierungstoken aus dem Benutzerprofil:

```
cd /opt/sas/viya/home/bin
./sas-admin auth login -u sasdmo -p Orion123
export token=$(cat ~/.sas/credentials.json | grep access-token |
cut -d\" -f 4)
echo $token

#=>
# eyJhbGciOiJSUzI...k3BCxmCDDA
```

Für die weiteren Abfragen verwenden wir das Linux Standard-Werkzeug cURL. Alternativ kann auch jeder andere Client verwendet werden (SAS, Python, ...).

```
export reportPage=\&page\=vi6
export reportUri=/reports/reports/2c1a982c-e244-4f01-951e-
25d4bf11a7f7
export server=sasserver.demo.sas.com
export reportPrompt=pr30
```

Bevor mit der Erstellung einer PDF-Datei begonnen werden kann, muss eine PDF-Style Definition erzeugt werden. Dieses wird in den nachfolgenden Abfragen verwendet und gibt die Ausgabeformatierung vor.

```
cat << EOF > /tmp/pdf.style
{
  "version": 1,
  "utcOffsetInSeconds": "-14400",
  "orientation": "landscape",
  "paperType": "letter",
  "marginTop": "0.25in",
  "marginBottom": "0.25in",
  "marginLeft": "0.25in",
  "marginRight": "0.25in",
  "fitOption": "none",
  "tableOfContentsDisplay": "thumbnails",
  "showPageNumbers": "false",
  "showEmptyRowsAndColumns": "false",
```

```
"showAppendix": "false",  
"showCoverPage": "false",  
"coverPageDateTime": "Thursday, March 25, 2021 01:00:00 PM",  
"reportElements": "vi6",  
"showPrintedTableOfContents": "false"  
}  
EOF
```

1. Schritt: Erzeugen eines PDF-Packages.

Wichtig bei der nachfolgenden Abfrage ist das Schlüsselwort
„dataLevel=baseline“. Bei Verwendung dieser Richtlinie wird ein
Standardbericht erzeugt.

```

curl -s -k -L -X POST "http://${server}/reportPackages/jobs?
dataLevel=baseline&reportUri=${reportUri}${reportPage}&wait=15" \
-H "Accept: application/vnd.sas.report.package.job+json" -H
"Content-Type: application/vnd.sas.report.rendered+json" \
-H "Authorization: Bearer $token"

#=>
{"version":2,"id":"583f923f-012a-4bfd-8c82-
b550ec2e9e37","creationTimeStamp":"2021-03-
05T13:55:16.835Z","startTimeStamp":"2021-03-
05T13:55:16.838Z","completionTimeStamp":"2021-03-
05T13:55:17.512Z","state":"completed","source":"/reports/reports/2c
1a982c-e244-4f01-951e-25d4bf11a7f7","createdBy":"sasdemo","links":
[{"method":"GET","rel":"self","href":"/reportPackages/jobs/583f923f-
012a-4bfd-8c82-b550ec2e9e37","uri":"/reportPackages/jobs/583f923f-
012a-4bfd-8c82-
b550ec2e9e37","type":"application/vnd.sas.report.package.job"},
{"method":"GET","rel":"reportPackage","href":"/reportPackages/packa
ges/b_ec13c3ec-1467-4f92-a210-
1c051faa5232","uri":"/reportPackages/packages/b_ec13c3ec-1467-4f92-
a210-1c051faa5232","type":"application/vnd.sas.report.package"}]]}

## Alternative in einem Schritt mit jq:
## -----
export repPackage=$(curl -s -k -L -X POST
"http://${server}/reportPackages/jobs?
dataLevel=baseline&reportUri=${reportUri}${reportPage}&wait=15" -H
"Accept: application/vnd.sas.report.package.job+json" -H "Content-
Type: application/vnd.sas.report.rendered+json" -H "Authorization:
Bearer $token" | jq -r '.links[] |
select(.rel=="reportPackage").uri' )
echo $repPackage

```

Aus dem Response kann die neu generierte reportPackage-Uri ausgelesen werden, welche im zweiten Schritt benötigt wird:

```

export repPackage=/reportPackages/packages/b_ec13c3ec-1467-4f92-
a210-1c051faa5232

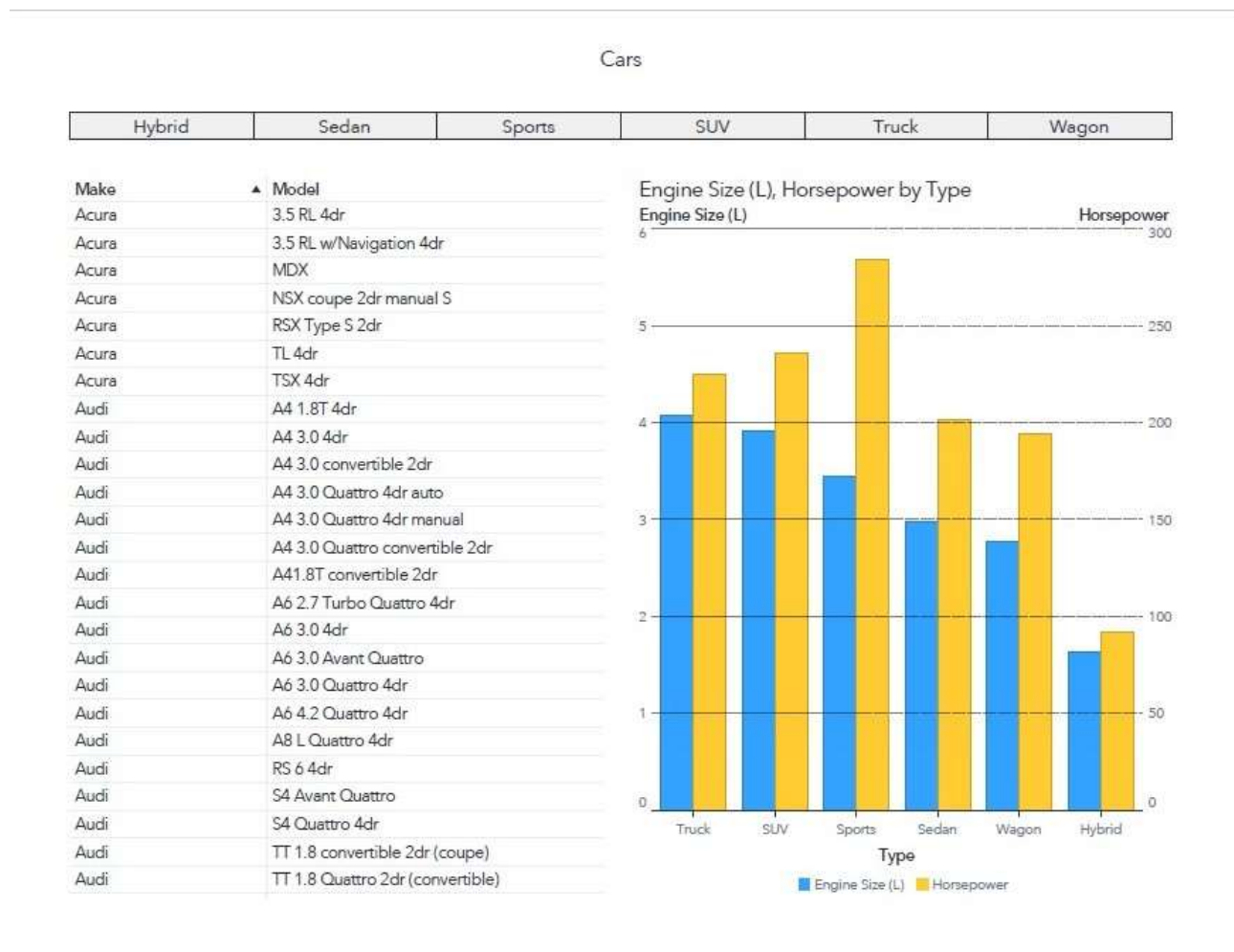
```

2. Schritt: Generieren und Herunterladen der PDF-Datei

Mit der reportPackage-Uri und der zuvor erstellten PDF-Style Definition kann nun das gewünschte PDF generiert und heruntergeladen werden. Auch für diesen Aufruf wird wieder cURL verwendet.

```
curl -X POST "http://${server}/reportRenderer/reports/?  
packageUri=${repPackage}" -H "Accept: application/pdf" -H  
"Authorization: Bearer $token" -H "Content-Type:  
application/vnd.sas.report.rendering.options+json;version=2" --data  
@/tmp/pdf.style -o /tmp/cars_baseline.pdf
```

Ergebnis-PDF



Erzeugen eines parametrisierten PDF-Berichtes

Als Erweiterung zu dem vorherigen Anwendungsfall wird jetzt ein bestimmter Fahrzeugtyp anhand der "ButtonBar" vorgefiltert. Dazu wird eine weitere Konfigurationsdatei benötigt, "custom_conf.xml". Diese Datei enthält die Werte der zu selektierenden Parameter des jeweiligen "Prompts". Es sollen in dem zweiten Bericht nur

Fahrzeuge vom Typ "Sports" angezeigt werden. Über die Konfigurationsdatei können auch mehrere Parameterwerte verarbeitet werden.

```
cat << EOF > /tmp/custom_conf.xml
<SASReportState>
  <Parameters>
    <Parameter prompt="$reportPrompt"
dataType="string">'Sports'</Parameter>
  </Parameters>
</SASReportState>
EOF
```

1. Schritt: Erzeugen eines PDF-Packages.

Wichtig bei der nachfolgenden Abfrage ist das Schlüsselwort

"dataLevel=custom". Bei Verwendung dieser Richtlinie wird ein angepasster Bericht erzeugt. Die Anpassungen werden in der Konfigurationsdatei "custom_conf.xml" beschrieben.


```
curl -s -k -L -X POST "http://${server}/reportPackages/jobs?
dataLevel=custom&reportUri=${reportUri}${reportPage}&wait=15
" -H "Accept: application/vnd.sas.report.package.job+json" -
H "Content-Type: application/vnd.sas.report.state+xml" -H
"Authorization: Bearer $token" --data @/tmp/custom_conf.xml
```

`#=>`

```
{"version":2,"id":"ccf320c0-ebd3-4554-90e2-
d71bfe461a27","creationTimeStamp":"2021-03-
05T14:27:34.250Z","startTimeStamp":"2021-03-
05T14:27:34.252Z","completionTimeStamp":"2021-03-
05T14:27:35.002Z","state":"completed","source":"/reports/rep
orts/2c1a982c-e244-4f01-951e-
25d4bf11a7f7","createdBy":"sasdemo","links":
[{"method":"GET","rel":"self","href":"/reportPackages/jobs/c
cf320c0-ebd3-4554-90e2-
d71bfe461a27","uri":"/reportPackages/jobs/ccf320c0-ebd3-
4554-90e2-
d71bfe461a27","type":"application/vnd.sas.report.package.job
"},
{"method":"GET","rel":"reportPackage","href":"/reportPackage
s/packages/c_203ae454-4a1a-40a3-9ee5-
b7e0de1e8d49","uri":"/reportPackages/packages/c_203ae454-
4a1a-40a3-9ee5-
b7e0de1e8d49","type":"application/vnd.sas.report.package"}]}
```

`## Alternative in einem Schritt mit jq:`

`## -----`

```
export repPackage2=$(curl -s -k -L -X POST
"http://${server}/reportPackages/jobs?
dataLevel=custom&reportUri=${reportUri}${reportPage}&wait=15
" -H "Accept: application/vnd.sas.report.package.job+json" -
H "Content-Type: application/vnd.sas.report.state+xml" -H
"Authorization: Bearer $token" --data @/tmp/custom_conf.xml
| jq -r '.links[] | select(.rel=="reportPackage").uri')
echo $repPackage2
```

Aus dem Response kann die zweite generierte reportPackage-Uri ausgelesen werden, welche im zweiten Schritt benötigt wird:

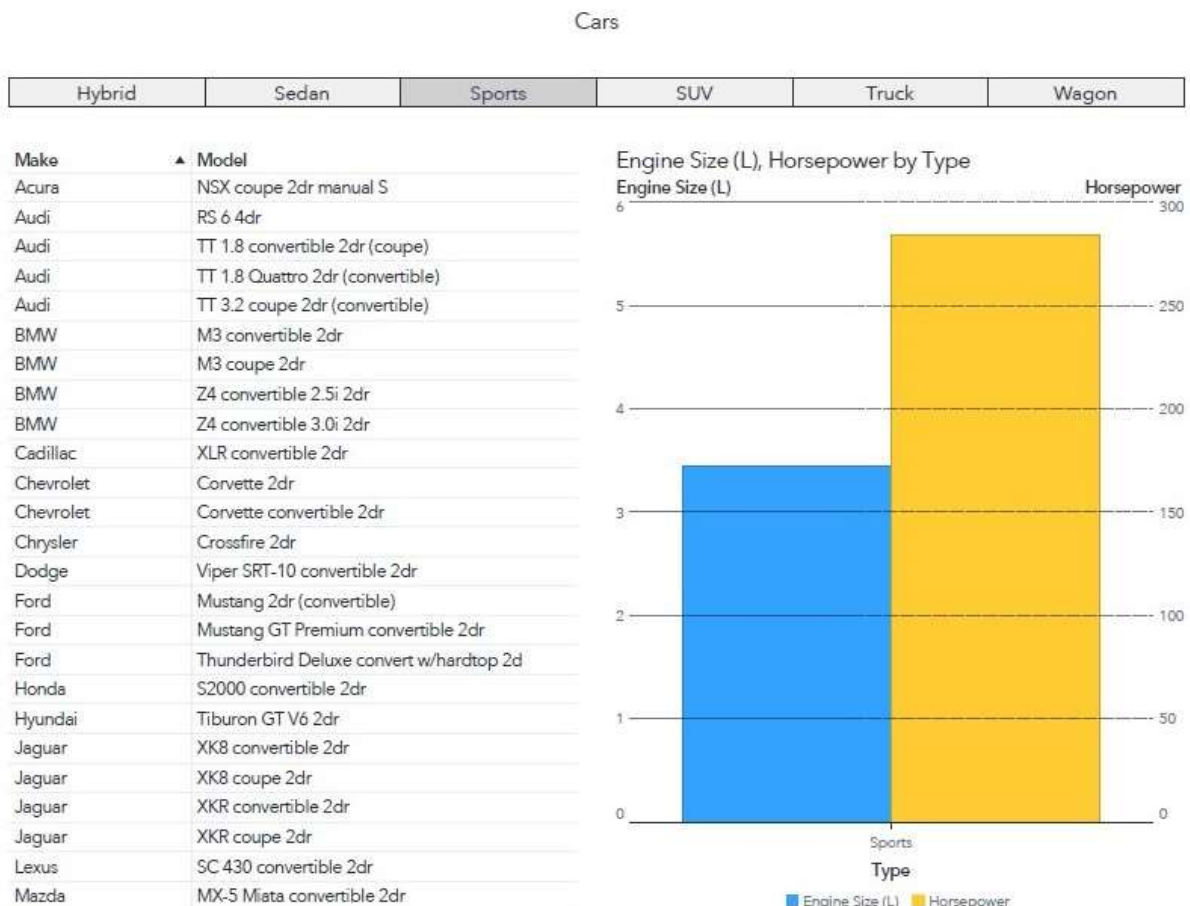
```
export repPackage2=/reportPackages/packages/c_203ae454-4a1a-40a3-9ee5-b7e0de1e8d49
```

2. Schritt: Generieren und Herunterladen der PDF-Datei

Mit der reportPackage-Uri und der zuvor erstellten PDF-Style Definition kann nun das zweite PDF generiert und heruntergeladen werden. Auch für diesen Aufruf wird wieder cURL verwendet.

```
curl -X POST "http://${server}/reportRenderer/reports/?packageUri=${repPackage2}" -H "Accept: application/pdf" -H "Authorization: Bearer $token" -H "Content-Type: application/vnd.sas.report.rendering.options+json;version=2" --data @/tmp/pdf.style -o /tmp/cars_custom.pdf
```

Ergebnis-PDF



Anhang

Drucken von seitenübergreifenden Inhalten

In den Eigenschaften der PDF-Datei ist das Setzen der Option

„*fitOption=expandClippedContentInAppendix*“

möglich. Damit werden abgeschnittene Inhalte nachträglich ans Ende der PDF-Datei angefügt. Falls mit dieser Option nicht das gewünschte Ergebnis erzielt wird, kann auch einfach die Dimension des PDF vergrößert werden.

```
cat << EOF > /tmp/pdf_big.style
{
  "version": 3,
  "utcOffsetInSeconds": "-14400",
  "orientation": "portrait",
  "paperType": "custom",
  "paperHeight": "80cm",
  "paperWidth": "21cm",
  "marginTop": "0.25in",
  "marginBottom": "0.25in",
  "marginLeft": "0.25in",
  "marginRight": "0.25in",
  "fitOption": "none",
  "tableOfContentsDisplay": "thumbnails",
  "showPageNumbers": "false",
  "showEmptyRowsAndColumns": "false",
  "showAppendix": "false",
  "showCoverPage": "false",
  "coverPageDateTime": "Tuesday, April 5, 2016 10:41:25 AM",
  "reportElements": "vi6",
  "showPrintedTableOfContents": "false"
}
EOF
```

Links

Proc http

Wie bereits angesprochen könne alle Beispiele welche mit Linux cmd „cURL“ gezeigt wurden ebenfalls mit den Procedure „Proc HTTP“ umgesetzt werden.

<https://go.documentation.sas.com/?docsetId=proc&docsetTarget=n0bdg5vmrpyi7jn1pbgbje2atoov.htm&docsetVersion=9.4&locale=en>

JQ

Werden häufig Abfragen unter Linux mit JSON Dateien durchgeführt, empfiehlt es sich ein Blick auf JQ zu werfen:

<https://stedolan.github.io/jq/>

SAS -Developers

<https://developer.sas.com/home.html>