

REST APIs mit SAS nutzen

Andreas Menrath
Software Architekt

Agenda

- Was ist REST?
- Wie geht das mit SAS?
 - PROC HTTP
 - Libname Json
 - PROC JSON
- Live-Demo
- Fazit

TURNING **DATA** INTO **VALUE**

HMS Analytical Software GmbH is providing highly specialized services to regulated industries in the area of data science and software engineering.

FACTS & FIGURES



1989 established

150+ employees

~17 Mio. € revenue in 2022

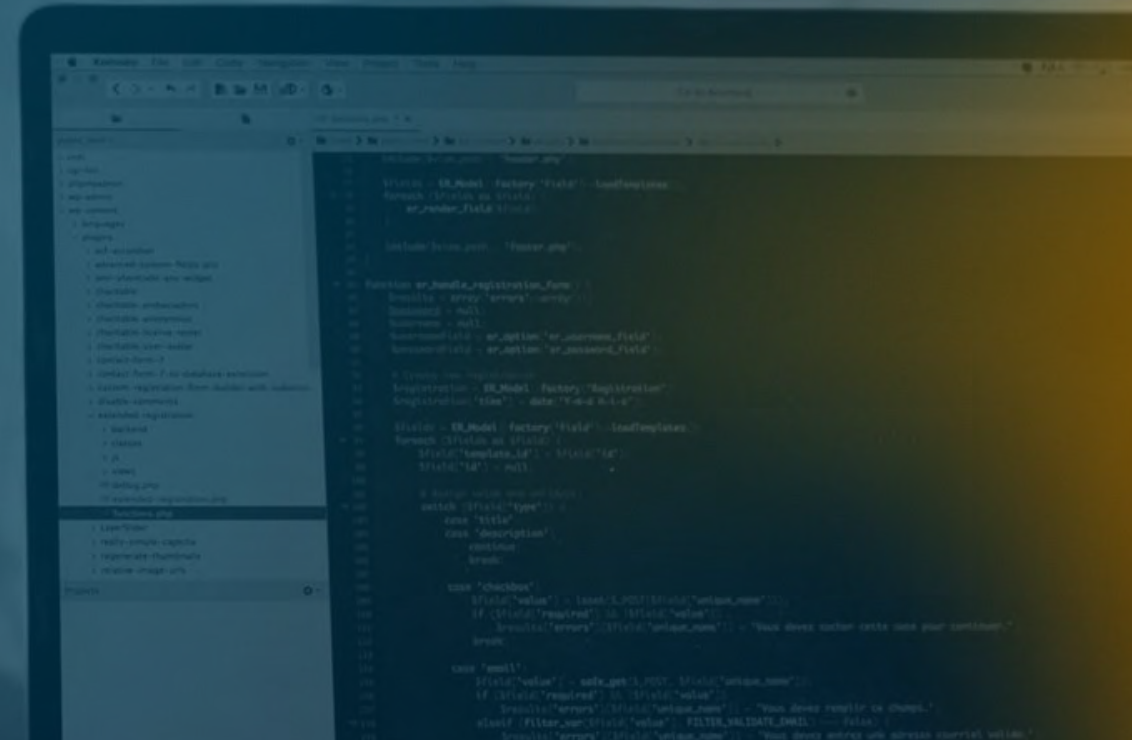
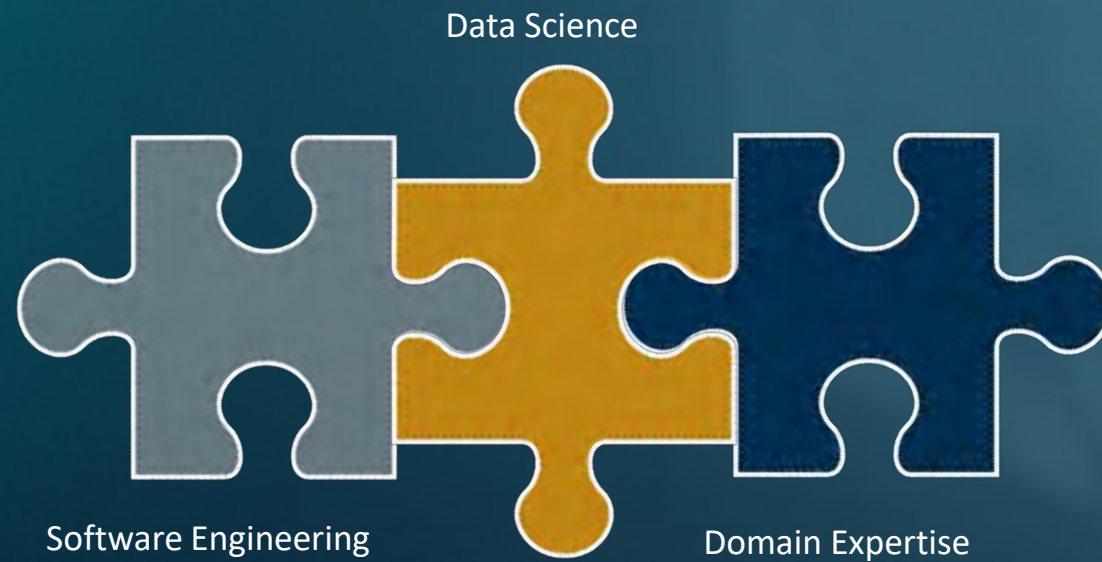
HQ in Heidelberg

Offices in Germany

Focus on regulated industries

Worldwide Network and Clients

THE HMS DIFFERENTIATOR



Full-Service Boutique Vendor in the analytical space combining crucial competencies in the area of **Data Science** with **Software Engineering** and **Domain Expertise**.

EXPERT NETWORK and TECH NETWORK



Microsoft
Partner

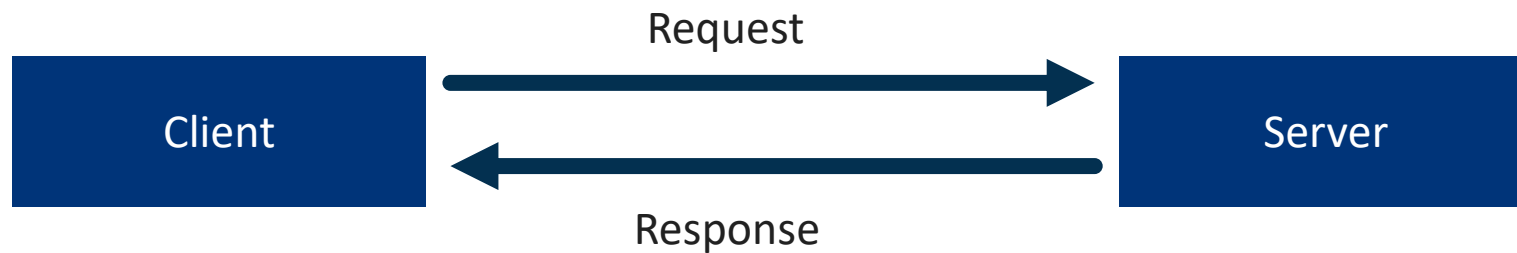


- Gold DevOps
- Gold Data Platform
- Gold Data Analytics
- Gold Cloud Platform
- Gold Application Development

Was ist REST?

Was ist REST?

- REST: „Representational State Transfer“
- basiert auf HTTP
- Client-Server Kommunikation
- zustandslos



Aufbau einer Anfrage (1)

- URL

[https:// analytical-software.de/api/my-endpoint?parameter1=x](https://analytical-software.de/api/my-endpoint?parameter1=x)

Schema Server URL

Endpoint

Abfrage

- HTTP Verb

| Verb | SQL | Beschreibung |
|--------|--------|----------------|
| GET | SELECT | Abfrage |
| POST | INSERT | Erzeugung |
| PUT | UPDATE | Aktualisierung |
| DELETE | DELETE | Löschen |

Aufbau einer Anfrage (2)

- HTTP Header (= Metadaten)

| Beispiel | Bedeutung |
|--------------------------------|-------------------------------------|
| Content-Type: application/json | Content vom Typ Json wird geschickt |
| Accept: application/json | Rückgabe vom Typ Json wird erwartet |
| Content-Length: 32 | Größe des Payloads |
| api-key: abcdef123 | Authentifizierung über API Key |
| Authorization: Bearer <token> | Authentifizierung über JWT |

- Body (=Payload)
 - Json
 - XML
 - Webformular
 - Binärdaten

Aufbau einer Antwort

- HTTP Status Code

| Statuscode | Bedeutung |
|------------|--------------------------------|
| 200 | OK |
| 201 | Ressource erfolgreich erstellt |
| 2xx | Erfolgreich |
| 3xx | Umleitung |
| 4xx | Client-Fehler |
| 5xx | Server-Fehler |

- HTTP Header

- HTTP Body

- Json
- XML
- Binärdatei

Authentifizierung

- Per Header
 - API-Key
 - Bearer Token (OAuth 2)
- Basic: Username + Passwort
- NTLM (Kerberos)

Aufbau von Json

- Elemente
 - Zahl
 - Zeichenkette / String
 - Bool (true/false)
 - Nullwert
 - Array []
 - Objekt {}
- Elemente lassen sich in beliebiger Tiefe schachteln

```
{
  "firstName": "John",
  "lastName": "Smith",
  "isAlive": true,
  "age": 27,
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": "10021-3100"
  },
  "phoneNumbers": [
    {
      "type": "home",
      "number": "212 555-1234"
    },
    {
      "type": "office",
      "number": "646 555-4567"
    }
  ],
  "children": [
    "Catherine",
    "Thomas",
    "Trevor"
  ],
  "spouse": null
}
```

Bitte mit



PROC HTTP

```
proc http
```

```
url="http://httpbin.org/get"
```

```
method="GET"
```

```
in=reqbody
```

```
out=respbody
```

```
headerin=reqhead
```

```
headerout=resphead
```

```
query= (
```

```
    "param1"="value1"
```

```
    "param2"="value2"
```

```
)
```

```
ct="application/json"
```

```
;
```

```
run;
```

PROC HTTP

- Nützliche Statements:
 - debug
 - headers
- Nützliche Makrovariablen:
 - SYS_PROCHTTP_STATUS_CODE
 - SYS_PROCHTTP_STATUS_PHRASE

```
proc http
    url="http://httpbin.org/get"
    method="GET"
    out=respbody
;

debug level=3;

headers "param1"="value1"
        "param2"="value2";

run;
```


Libname JSON

```
/* einfachstes Beispiel */
```

```
libname myref json "C:\...\my.json";
```

```
/* mit fileref + Erstellung Mapping Datei */
```

```
libname myref json  
    fileref=mytemp  
    automap=create  
    map=tempmap;
```

```
/* spätere Nutzung der Mapping Datei */
```

```
libname myref json  
    fileref=mytemp  
    map=tempmap;
```

PROC JSON

Syntax

```
PROC JSON OUT=fileref | "external-file" <options>;  
    EXPORT <libref>SAS-data-set <(SAS-data-set-options)> </options>;  
    WRITE VALUES value(s) </options>;  
    WRITE OPEN type;  
    WRITE CLOSE;
```

Summary of Optional Arguments

FMTCHARACTER | NOFMTCHARACTER

determines whether to apply a character SAS format to the resulting output if a character SAS format is associated with a SAS data set variable.

FMTDATETIME | NOFMTDATETIME

determines whether to apply a date, time, or datetime SAS format to the resulting output if a date, time, or datetime SAS format is associated with a SAS data set variable.

FMTNUMERIC | NOFMTNUMERIC

determines whether to apply a numeric SAS format to the resulting output if a numeric SAS format is associated with a SAS data set variable.

KEYS | NOKEYS

determines whether exported observations are written as JSON objects or as JSON arrays.

PRETTY | NOPRETTY

determines how to format the JSON output.

SASTAGS | NOSASTAGS

determines whether to include or suppress SAS metadata when using the EXPORT statement.

SCAN | NOSCAN

determines whether PROC JSON scans and encodes input strings to ensure that only characters that are acceptable are exported to the JSON output file.

TRIMBLANKS | NOTRIMBLANKS

determines whether to remove or retain trailing blanks from the end of character data in the JSON output.

Demo



"Dieses Foto" von Unbekannter Autor ist lizenziert gemäß [CC BY-SA-NC](#)

Fazit



"Dieses Foto" von Unbekannter Autor ist lizenziert gemäß [CC BY-SA](#)

Fazit

PROC HTTP:

- ausgereift
- für den alltäglichen Gebrauch geeignet

Libname JSON:

- fehleranfällig
- SAS session crashes während Entwicklung sind möglich

PROC JSON:

- kompliziert
- man kann nicht immer das gewünschte Zielformat erstellen
- Als Alternative Data Step in Betracht ziehen

Fragen?

Andreas Menrath
Software Architekt



Vielen Dank für die Aufmerksamkeit!



Andreas Menrath

Software Architect

Connect with me on

Linked in



HMS 
analytical software