Reverse engineering of a REST API

using PowerShell and Chrome

Fabian Bader – Aequitas Integration

about_Author

- @fabian_bader auf Twitter
- Senior Consultant @ Aequitas Integration
- Mitbegründer der @HHPSUG
- fabian [at] cloudbrothers.info

What‘s an RESTful API

- Representational State Transfer (REST) architecture was defined by Roy Fielding in the year 2000
- Core principals are
  - Client-Server Architecture
  - Stateless
  - Uniform Interface
- In most cases a RESTful webservice returns JSON or XML

https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm

Requesting and Changing Data

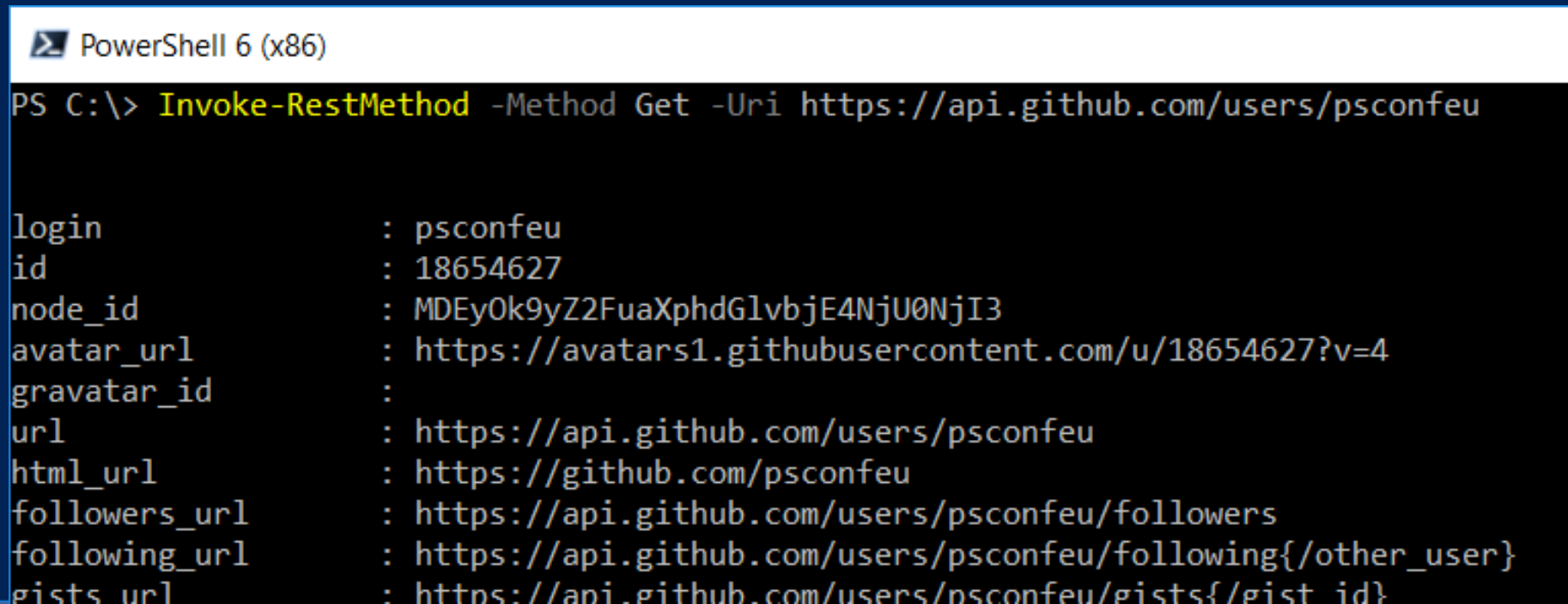
- A request consist, at a minimum, out of two parts of information
  - Request URL
  - Request Method
- The Request URL describes what element or collection of elements you are referring to
- The Request Method describes which action you want to perform

Request Method

GET	Retrieves data. Does not change anything
POST	Transmits data to a resource
PUT	Create a new ressource with the data provided
DELETE	Delete the ressource
OPTIONS	Check which methods are available for this resource

Example of a request

- Request URL
  https://api.github.com/users/psconfeu
- Request Method
  Get

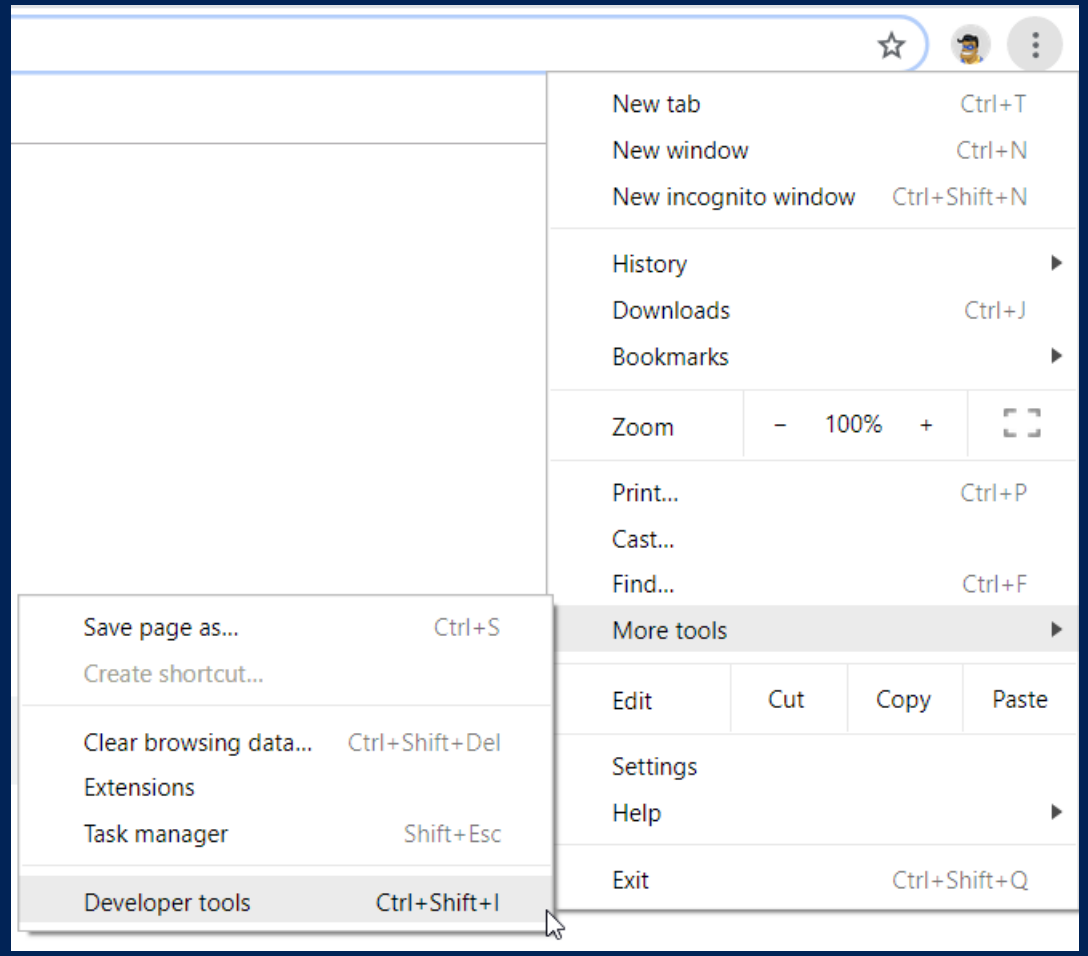


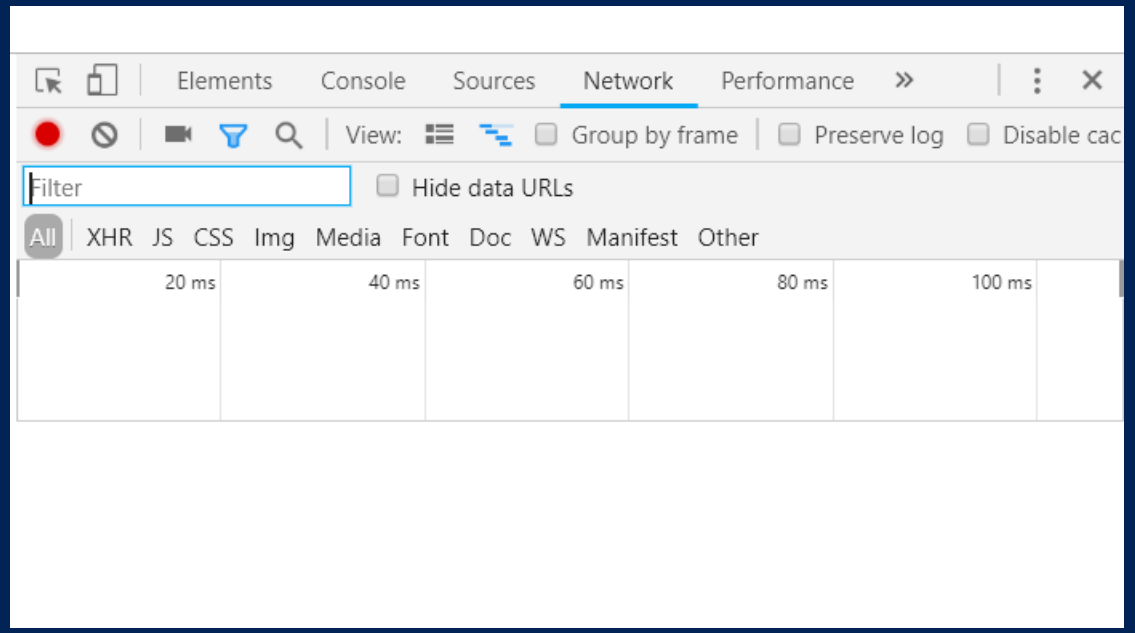
Reverse engineering of a RESTful API

- Tools we need

- Google Chrome Developer Tool
- PowerShell
- Visual Studio Code

Chrome Developer Tools





Demo

Reverse engineering of a Azure REST API call

Azure Authentication – Bearer Token


```
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIng1dCI6Imk2bEdrM0ZaenhSY1ViMkMzbkVRN3N5SEpsWSIsImtpZCI6Imk2bEdrM0ZaenhSY1ViMkMzbkVRN3N5SEpsWSJ9.
eyJhdWQiOiJodHRwczovL21hbmFnZW1lbnQuY29yZS53aW5kb3dzLm5ldC8iLCJpc3MiOiJodHRwczovL3N0cy53aW5kb3dzLm5ldC9iMGFlNTI0NC00Mzc3LTQ1NjktYmM2M
y00ZTM5YWJjYTI0MWIvIiwiaWF0IjoxNTM5MjYyMDA2LCJuYmYiOjE1MzkyNjIwMDYsImV4cCI6MTUzOTI2NTkwNiwiYWNyIjoiMSIsImFpbyI6IkFWUUFxLzhKQUFBQU9rY0
V3YUJvVFFvbmZSSXVUMlIvK2Rm[redacted]kSk5ZL3pYK0lQUzBydDZkMVQrSzQ
vaXNFPSIsImFtciI6WyJwd2QiL[redacted]ZGFjciI6IjAiLCJkZXZpY2VpZCI6
ImMxOTNmN2EwLTllODQtNDg4NC[redacted]XZlbl9uYW1lIjoiRmFiaWFuIiwiZ
3JvdXBzIjpbIjc0ZTY2ZDc1LWU[redacted]UiOiJGYWJpYW4gQmFkZXIiLCJvaW
QiOiIyNWZhZmU1Ny01YWFhLTQ2[redacted]faW1wZXJzb25hdGlvbiIsInN1YiI
6InF6V3VIc011N1RPdVIzQTdrT[redacted]ZTM5YWJjYTI0MWIiLCJ1bmlxdWVf
bmFtZSI6ImZhYmlhbkBiYWRlci[redacted]WRCQUEiLCJ2ZXIiOiIxLjAiLCJ3a
WRzIjpbIjYyZTkwMzk0LTY5ZjU[redacted]1g1Uw86ZBg6-xFgUu9BWf7CgI6KB
Q9OY80W0iKEN4aRsWksIBhV1GR[redacted]32pNPE5Xt9efDT4IfyEYcqFtaA6s
3l5QoVN5cw-9QTbZW0a3ArnV3r01cEc-oiccbg2vp3PnCMaQHmcWIDV8hfap0DfiPgQHN3g1TpeA2O0kV0J8_4if6NTSSHcUC0WRXieTh6Oft2HTQojykzaWcLpSmA5bKbkX6
UUfwg
```

Azure Authentication

- Bearer Token is needed
- Thanks to Stéphane Lapointe we can use Get-AzureRmCachedAccessToken
- Leverages the AzureRmProfileProvider to extract the Bearer Token from the current AzureRM context
- https://gallery.technet.microsoft.com/scriptcenter/Easily-obtain-AccessToken-3ba6e593#content

- Or use an Application with a Service Principal in Azure AD

Demo

Rebuild the Call with PowerShell

Invoke-WebRequest vs. Invoke-RestMethod

PowerShell 6 (x86)

```
PS C:\> Invoke-WebRequest  -Method Get -Uri https://api.github.com/users/psconfeu


StatusCode        : 200
StatusDescription : OK
Content           : {"login":"psconfeu","id":18654627,"node_id"
                    :"MDEyOk9yZ2FuaXphdGlvbjE4NjU0NjI3","avatar
                    _url":"https://avatars1.githubusercontent.c
                    om/u/18654627?v=4","gravatar_id":"","url":"
                    https://api.github.com/users...
RawContent        : HTTP/1.1 200 OK
                    Server: GitHub.com
                    Date: Thu, 11 Oct 2018 12:37:48 GMT
                    Status: 200 OK
                    X-RateLimit-Limit: 60
                    X-RateLimit-Remaining: 50
                    X-RateLimit-Reset: 1539262570
                    Cache-Control: public, max-ag...
Headers           : {[Server, System.String[]], [Date,
                    System.String[]], [Status,
                    System.String[]], [X-RateLimit-Limit,
                    System.String[]]...}
Images            : {}
InputFields       : {}
Links             : {}
RawContentLength  : 1227
RelationLink      : {}
```

PowerShell 6 (x86)

```
PS C:\> Invoke-RestMethod -Method Get -Uri https://api.github.com/users/psconfeu


login             : psconfeu
id                : 18654627
node_id           : MDEyOk9yZ2FuaXphdGlvbjE4NjU0NjI3
avatar_url        : https://avatars1.githubusercontent.com/u/
                    18654627?v=4
gravatar_id       :
url               : https://api.github.com/users/psconfeu
html_url          : https://github.com/psconfeu
followers_url     : https://api.github.com/users/psconfeu/fol
                    lowers
following_url     : https://api.github.com/users/psconfeu/fol
                    lowing{/other_user}
gists_url         : https://api.github.com/users/psconfeu/gis
                    ts{/gist_id}
starred_url       : https://api.github.com/users/psconfeu/sta
                    rred{/owner}{/repo}
subscriptions_url : https://api.github.com/users/psconfeu/sub
                    scriptions
organizations_url : https://api.github.com/users/psconfeu/org
                    s
repos_url         : https://api.github.com/users/psconfeu/rep
                    os
events_url        : https://api.github.com/users/psconfeu/eve
                    nts{/privacy}
received_events_url : https://api.github.com/users/psconfeu/rec
```

TLS v1.2

Windows PowerShell

```
[I ♥ PS [NoAdmin][C:\]] Invoke-RestMethod -Method Get -Uri https://api.github.com/users/psconfeu
Invoke-RestMethod : The request was aborted: Could not create SSL/TLS secure channel.
At line:1 char:1
```

```
# Force TLS 1.2
[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12
```

- Only a problem with Windows PowerShell

Pageination support

- Windows PowerShell only returns the first Page of an REST API response
- PowerShell Core introduces FollowRelLink and MaximumFollowRelLink

Differences between 5.1 and 6.0

- Way too much to cover in this talk
- Amazing Blog series by Mark Kraus (@markekraus)
  „PowerShell Core Web Cmdlets in Depth”
  - Part 1 – Under the hood
  - Part 2 – Missing and Deprecated Features
  - Part 3 – New Features