

One API to Rule Them All: The New REST-API

CHECKMK CONFERENCE #6 – MUNICH, APRIL 29, 2020



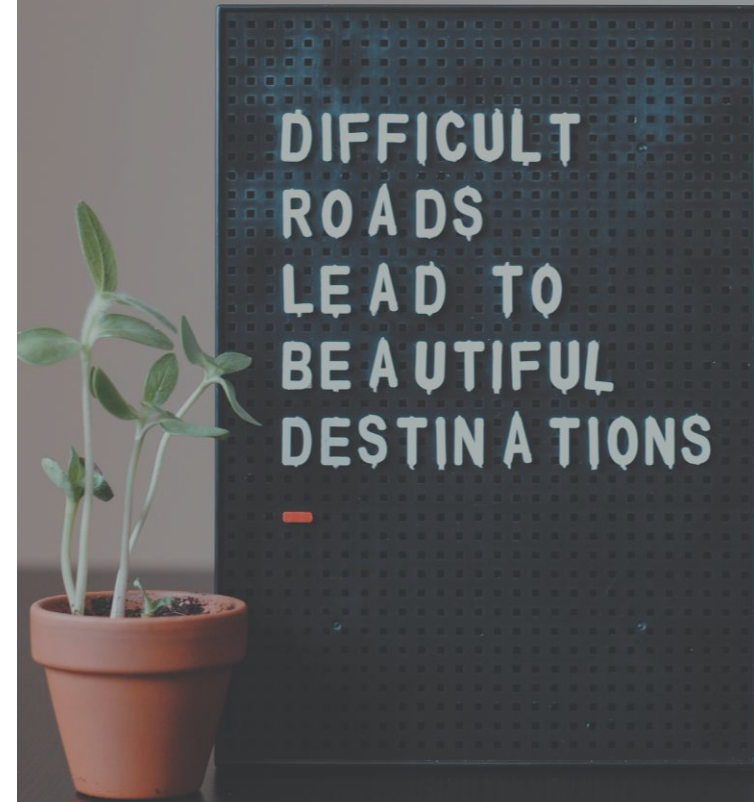
Christoph Rauch

Developer

tribe29

Our vision for the REST-API (1/4)

- An API which covers the entire functionality of the GUI and the command line
 - This is an ambitious goal which we have set ourselves
 - We're currently making progress on a steady pace



Our vision for the REST-API (2/4)

- ◆ The API shall be easy to use by humans
- ◆ This means:
 - ◆ helpful error messages
 - ◆ no undocumented areas
 - ◆ every behavior of the API is completely explained
 - ◆ no guesswork is necessary, ever



Our vision for the REST-API (3/4)

- Automation with the API shall be easy
 - All API URLs (endpoints) are completely documented
 - The documentation is always up to date
 - Example code is provided for each URL
 - It will always be backwards compatible
- Because of this, we automatically generate the documentation and the specification from the source code



Our vision for the REST-API (4/4)

- The API conforms to industry standards
- Because it is based on WWW technology, this means:
 - It's based on HTTP/1.1
 - We use JSON as a transport format
 - We formalize it with an OpenAPI 3.0.x specification file
 - We target Level 3 of the Richardson Maturity Model



Our vision for the REST-API

- ✓ The API covers the entire functionality of the GUI and the command line
- ✓ The API shall be easy to use by humans
- ✓ Automation with the API shall be easy
- ✓ The API conforms to industry standards



Generated documentation

Search...

Introduction

Authentication >

How does it work? >

Client compatibility issues >

ENDPOINTS

Contact-groups >

Folders >

Hosts >

POST Create a host

DELETE Delete a host

GET Show a host

PUT Update a host

Host-groups >

Service-groups >

Miscellaneous >

Agent related endpoints >

RESPONSE SCHEMAS

REQUEST SCHEMAS

Creating a contact group

Creating a folder

Updating a folder

Creating a new host

attribute

folder required

hostname required

nodes

Responses

> 200

> default Any unsuccessful or unexpected result.

POST Create a host

DELETE Delete a host

GET Show a host

PUT Update a host

Host-groups >

Service-groups >

Miscellaneous >

Agent related endpoints >

RESPONSE SCHEMAS

REQUEST SCHEMAS

Creating a contact group

Creating a folder

Updating a folder

Creating a new host

POST

Request sample

Payload

import requests

session = requests.session()

session.headers['Authorization'] = 'Bearer username password'

session.headers['Accept'] = 'application/json'

Replace HOST_NAME and SITE_NAME with the names from your host and site.

API_URL = 'http://HOST_NAME/SITE_NAME/check_mk/api/v0'

resp = session.get(API_URL + '/objects/host/example.com',)

resp.raise_for_status()

Response samples

GET /objects/host/{hostname}

Request samples

requests curl httpie

Copy

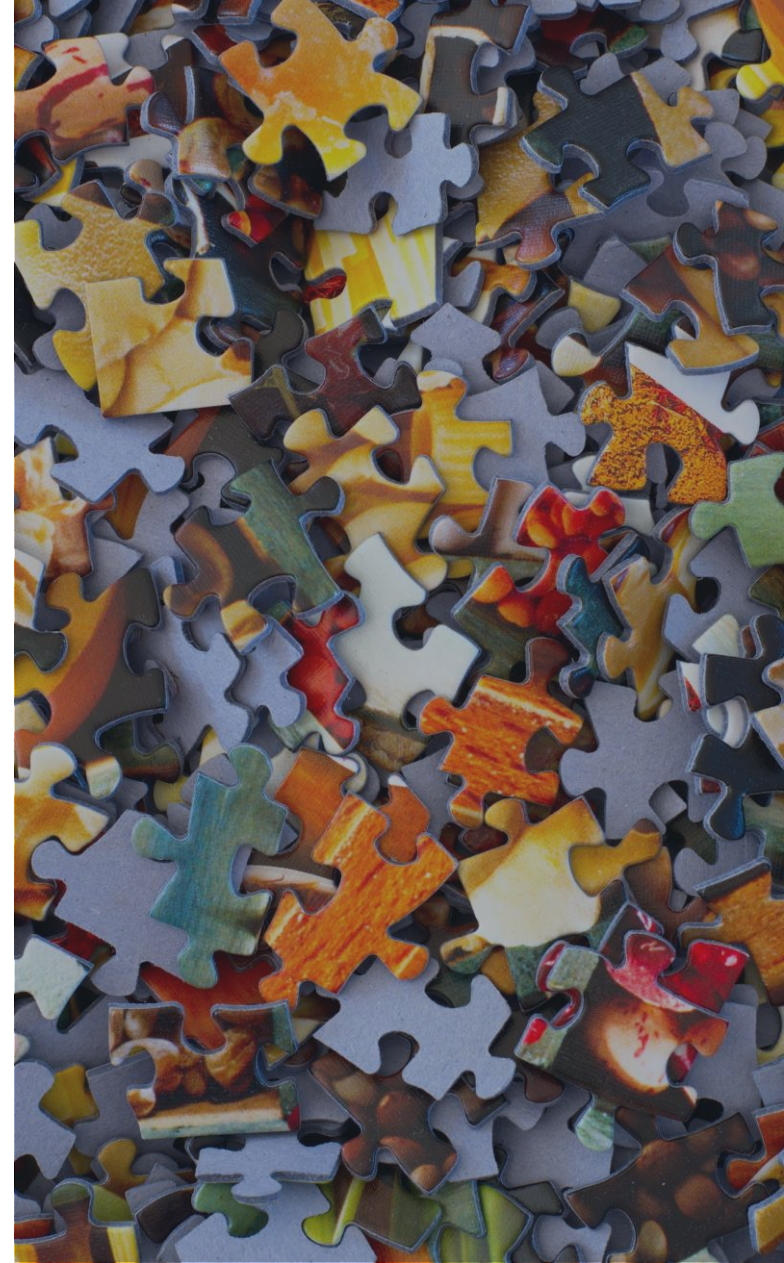
REST-API - current status

- ◆ Major progress:
 - ◆ Hosts, folders, host-groups, service-groups, contact-groups (configuration)
 - ◆ Baking, signing, downloading of agents
- ◆ Development focus for 1.7:
 - ◆ Bakery, BI, Folders, Hosts, Services, Groups, Configuration, Rulesets
 - ◆ Get Livestatus data via the API for Services and Hosts



Demo: A few important terms

- **API endpoint:** The URL with which you can make the Checkmk API do something
- **HTTP method:** Something like a category of the endpoint's behavior, e.g. if they may change data, or are read-only
- **HTTP status code:** Numbers like 2XX (OK), 3XX (see other resource), 4XX (client error), 5XX (server error)
- **API payload:** JSON structure to be sent to the API endpoint

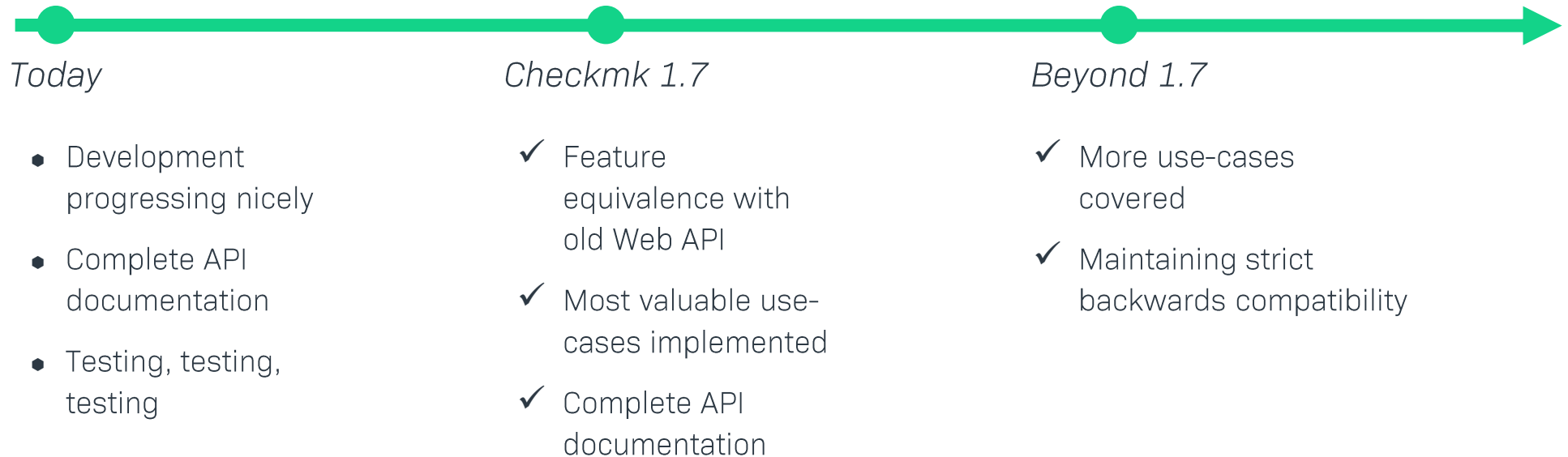


Demo time!

- ◆ Our example:
 - ◆ Synchronize your IT inventory DB (or Excel file) with Checkmk to immediately start monitoring once a new piece of infrastructure is deployed



Timeline for the REST API



Thank you



tribe29 GmbH
Kellerstraße 29
81667 München
Deutschland

Web — tribe29.com
E-Mail — mail@tribe29.com