Technischer Hands-on Workshop

Julian Ladisch, VZG

Ingolf Kuss, hbz

3. FOLIO-Tage Bremen 2019

11. April 2019

API-Zugriffe im Browser beobachten (Strg+Umschalt+I, Network)
API-Dokumentation: https://dev.folio.org/reference/api/
API-Zugriff per curl auf der Kommandozeile

- (für Windows ist eine Möglichkeit https://gitforwindows.org/ , enthält Bash und curl)

API-Zugriff per Postman: https://www.getpostman.com/

- https://gist.github.com/julianladisch/b5239149fc8cbe0f474d2464c7e01108

Login-Token
Zugriffsberechtigungen (User Permissions)
Per API Datensatz lesen, ändern, erstellen, löschen (User)

Die folgenden Shellskripte und Dateien gibt es zum Download auf
https://gist.github.com/julianladisch/9f08c6225ab73c6e6e53fae65037e996

user-aachen.json

```
{
  "username" : "aachen",
  "id" : "7c7a0099-b443-415c-ab5f-34c3bc65f50e",
  "barcode" : "112",
  "active" : true,
  "patronGroup" : "f2bf7b08-6ead-4498-9227-b379b72d4a4d",
  "proxyFor" : [ ],
  "personal" : {
    "lastName" : "Aachen",
    "firstName" : "Annika",
    "email" : "annika.aachen@example.org",
    "addresses" : [ ],
    "preferredContactTypeId" : "002"
  }
}
```

Datensatz anlegen (POST = INSERT)
post-user-aachen

```
OKAPI="${OKAPI:-https://folio-demo.gbv.de/okapi}"

TOKEN=$( curl -s -S -D - -H "X-Okapi-Tenant: diku" -H "Content-type:
application/json" -H "Accept: application/json" \
  -d '{"tenant":"diku","username":"diku_admin","password":"admin"}'
$OKAPI/authn/login \
  | grep -i "^x-okapi-token: " )

OPT=(-s -S -D - -H "$TOKEN" -H "X-Okapi-Tenant: diku" -H "Content-type:
application/json" -H "Accept: application/json")

RESULT=$( curl "${OPT[@]}" -d @user-aachen.json $OKAPI/users)
echo "$RESULT"
ID=$( echo "$RESULT" | grep -i "^Location:" | cut -d" " -f2 )
curl "${OPT[@]}" -d "{ \"id\" : \"$ID\", \"userId\" : \"$ID\",
\"permissions\" : [] }" $OKAPI/perms/users
```

Datensätze abrufen (GET = SELECT)
get-users

```
OKAPI="${OKAPI:-https://folio-demo.gbv.de/okapi}"

TOKEN=$( curl -s -S -D - -H "X-Okapi-Tenant: diku" -H "Content-type:
application/json" -H "Accept: application/json" -d
'{"tenant":"diku","username":"diku_admin","password":"admin"}'
$OKAPI/authn/login | grep -i "^x-okapi-token: " )

curl -s -S -D - -H "$TOKEN" -H "X-Okapi-Tenant: diku" -H "Content-type:
application/json" -H "Accept: text/plain" $OKAPI/users?$@
```

Aufruf mit
`./get-users query=username==aa*`

Datensatz ändern (PUT = UPDATE)
put-user-aachen

```
OKAPI="${OKAPI:-https://folio-demo.gbv.de/okapi}"

TOKEN=$( curl -s -S -D - -H "X-Okapi-Tenant: diku" -H "Content-type:
application/json" -H "Accept: application/json" \
  -d '{"tenant":"diku","username":"diku_admin","password":"admin"}'
$OKAPI/authn/login \
  | grep -i "^x-okapi-token: " )

OPT=(-s -S -D - -H "$TOKEN" -H "X-Okapi-Tenant: diku" -H "Content-type:
application/json" -H "Accept: text/plain")

curl "${OPT[@]}" -X PUT -d @user-aachen.json $OKAPI/users/7c7a0099-b443-
415c-ab5f-34c3bc65f50e
```

Datensatz löschen (DELETE)
delete-user-aachen

```
OKAPI="${OKAPI:-https://folio-demo.gbv.de/okapi}"

TOKEN=$( curl -s -S -D - -H "X-Okapi-Tenant: diku" -H "Content-type:
application/json" -H "Accept: application/json" \
  -d '{"tenant":"diku","username":"diku_admin","password":"admin"}'
$OKAPI/authn/login \
  | grep -i "^x-okapi-token: " )

OPT=(-s -S -D - -H "$TOKEN" -H "X-Okapi-Tenant: diku" -H "Content-type:
application/json" -H "Accept: text/plain")

ID="7c7a0099-b443-415c-ab5f-34c3bc65f50e"
curl "${OPT[@]}" -X DELETE $OKAPI/users/$ID
curl "${OPT[@]}" -X DELETE $OKAPI/perms/users/$ID
```

Load and save users, user permissions, and logins

https://github.com/jemiller0/Folio

Die Architektur von FOLIO:

- https://www.gbv.de/Verbundzentrale/Publikationen/publikationen-der-vzg-2018/pdf/Ladisch_180830_VK_folio_architektur.pdf

Module in unterschiedlichen Programmiersprachen

- Java - https://github.com/folio-org/mod-circulation-storage
- Groovy/Grails - https://github.com/folio-org/mod-agreements
- JavaScript V8/Node.js - https://github.com/folio-org/mod-graphql

Typischer Aufbau einer App:

- Schema in der Datenbank
- Storage-Modul für Datenbankzugriff https://github.com/folio-org/mod-circulation-storage
- Business-Logic-Modul https://github.com/folio-org/mod-circulation
- Browser-Frontend https://github.com/folio-org/ui-checkin

Docker-Container für programmiersprachenunabhängiges Deployment und Kapselung

- Für Java openjdk8 basierend auf Debian Stretch oder Alpine:
  - https://github.com/folio-org/folio-tools/tree/master/folio-java-docker/openjdk8

Okapi hat besondere Unterstützung für Dockercontainer

- Download gewünschter Version vom Repository:
  - https://hub.docker.com/u/folioci/ und https://hub.docker.com/u/folioorg/
- Aber: Module können auch ohne Dockercontainer betrieben werden
  - Okapi reicht es, wenn URLs der API bekannt sind.

Installation eines kompletten FOLIO-Systems erfolgt vorzugsweise automatisiert per Ansible.

- https://github.com/folio-org/folio-ansible
- Zum Nachvollziehen gibt es auch eine Anleitung für eine manuelle Installation:
  - https://github.com/folio-org/folio-install/blob/master/single-server.md

Fertige Systeme, die als Vagrant-Box heruntergeladen werden können:

- https://github.com/folio-org/folio-ansible/blob/master/doc/index.md
- https://app.vagrantup.com/folio