The zGui (r)evolution
–
First hands on experience and best practices

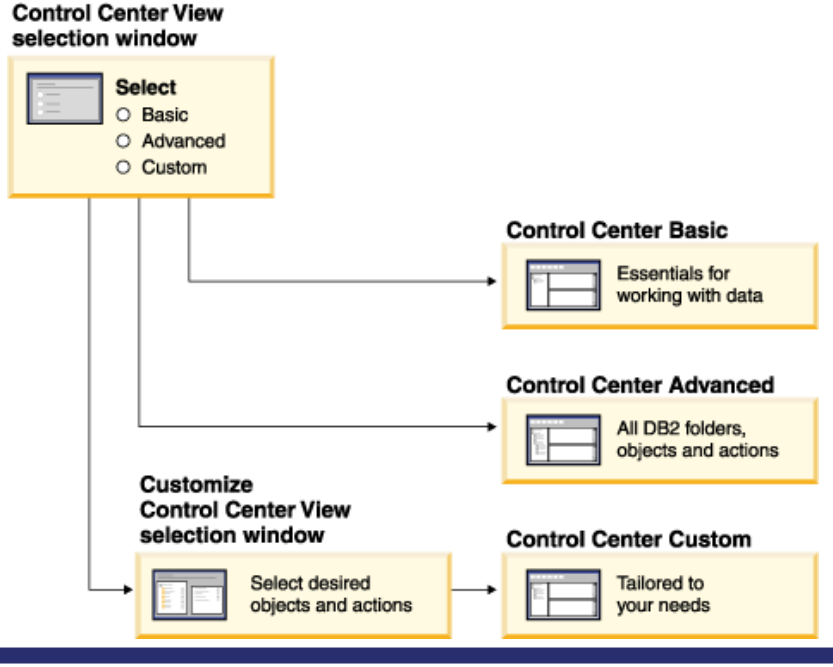
Ulf Heinrich
SEGUS Inc

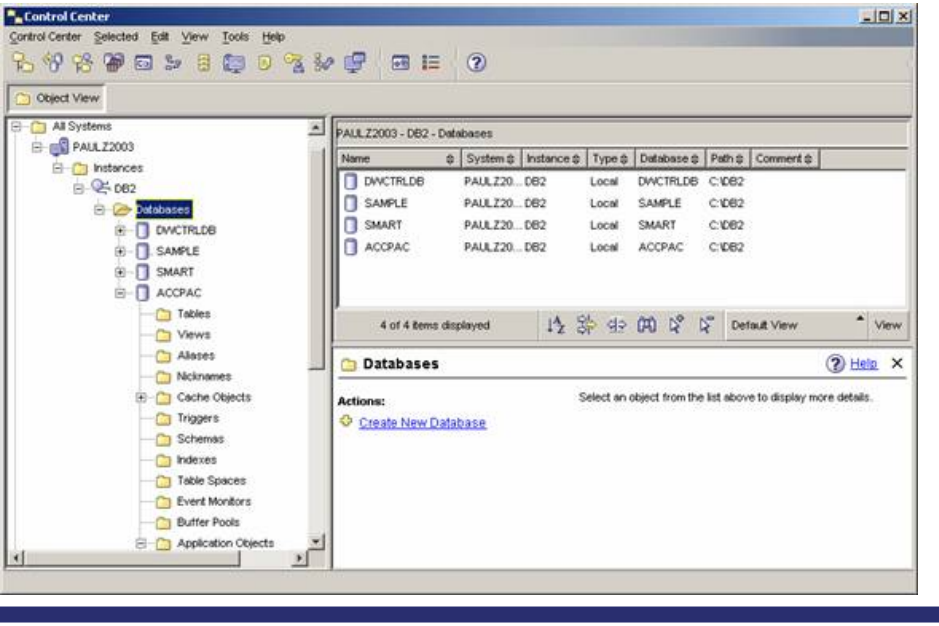
Agenda

- GUIs in the past
- Zowe ecosystem overview
- Zowe differentiation to prior GUIs
- Zowe components
- Zowe examples
- Hands on usage based on a cloning example
- Summary of experience

GUIs in the past

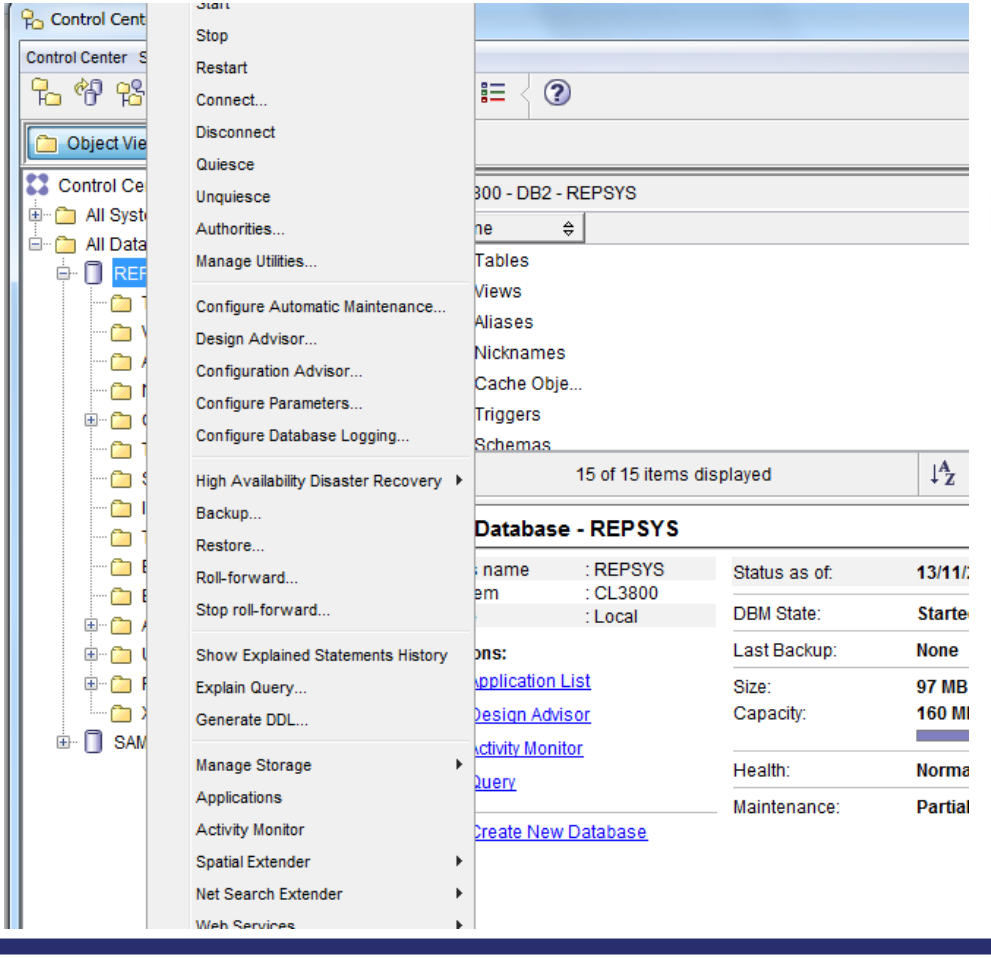
- Db2 Control Center (Db2cc)
  - Introduced with Db2 LUW 5, but also able to connect to Db2 z/OS
  - A Windows/Linux fat client using Db2 connect and stored procedures
  - Manages and administers Db2 systems and objects





GUIs in the past

- Db2 Control Center (Db2cc)
  - Can also open other centers to
    - optimize queries, jobs, and scripts
    - perform data warehousing tasks
    - create stored procedures
    - work with DB2 and IMS commands



GUIs in the past

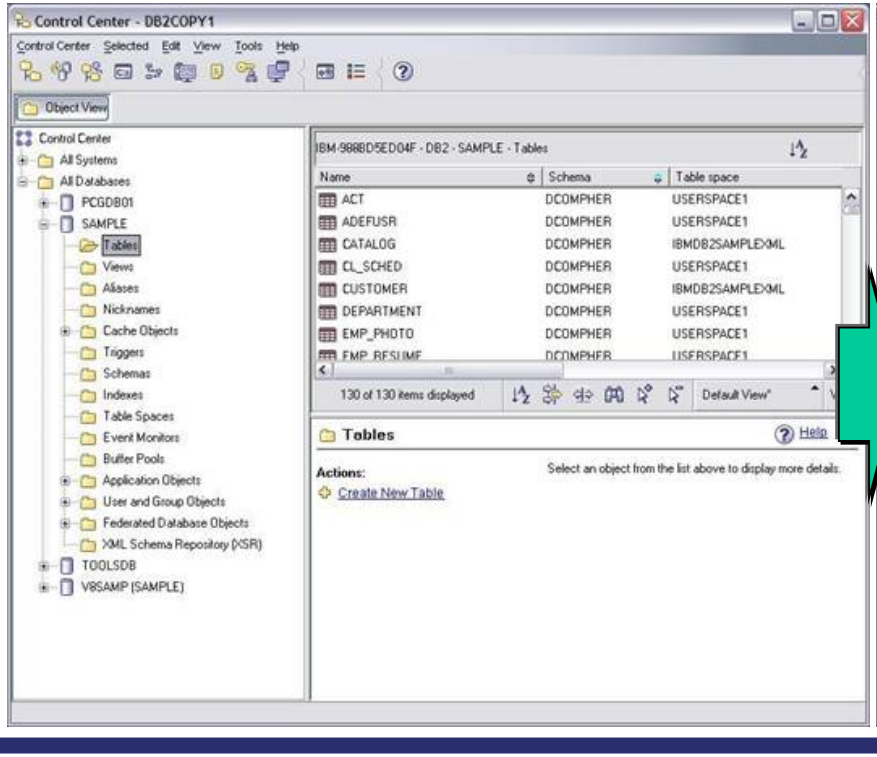
- Db2 Control Center (Db2cc)
  - More and more features and functions added over time:
    - Activity Monitor
    - Command Editor
    - Configuration Assistant
    - Control Center and associated wizards and advisors
    - Control Center plug-in extensions
    - Event Analyzer
    - Health Center
    - Indoubt Transaction Monitor
    - Journal
    - License Center
    - Memory Visualizer
    - Query Patroller Center
    - Satellite Administration Center
    - Task Center
    - User interface to access Spatial Extender functionality
    - User interface to Visual Explain

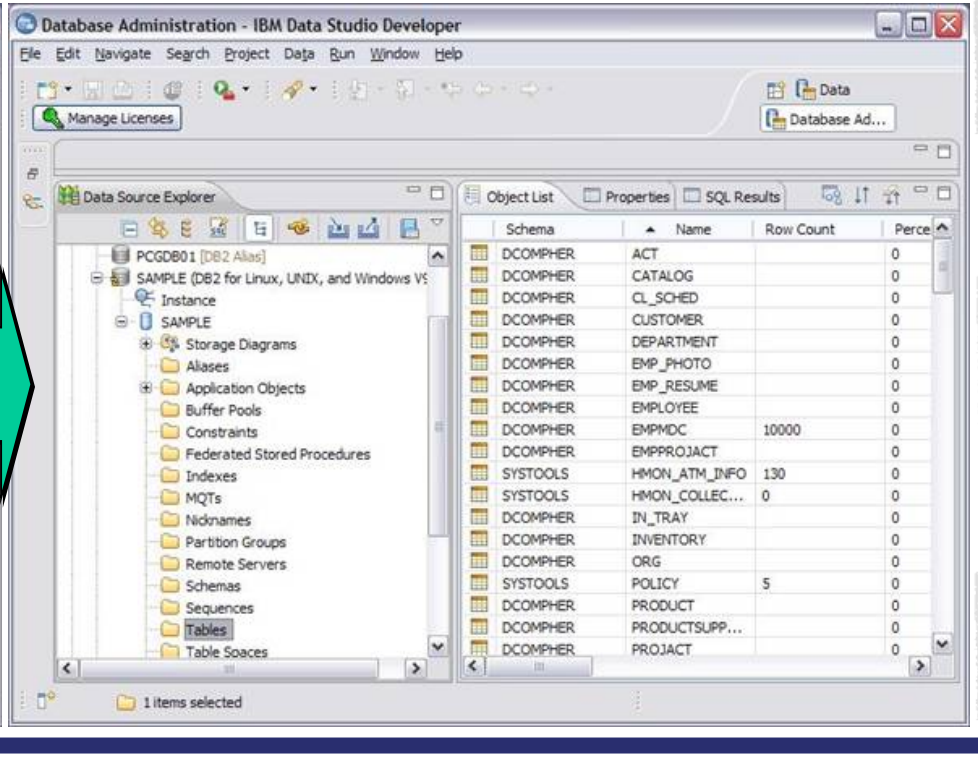
GUIs in the past

- Db2 Control Center (Db2cc)
  - ...along with wizards and advisors:
    - Control Center and associated wizards and advisors
      - Alter Database Partition Group wizard
      - Backup wizard
      - Configuration advisor
      - Configure Database Logging wizard
      - Configure Multisite Update wizard
      - Create Cache Table wizard
      - Create Database wizard
      - Create Federated Objects wizard (Also known as Create Nicknames wizard)
      - Create Table Space wizard
      - Create Table wizard
      - Design advisor
      - Drop Partition launchpad
      - Health Alert Notification
      - Health Indicator Configuration launchpad
      - Load wizard
      - Recommendation advisor
      - Redistribute Data wizard
      - Restore wizard
      - Set Up Activity Monitor wizard
      - Set Up High Availability Disaster Recovery (HADR) Databases wizard
      - Storage Management Setup launchpad
      - Troubleshooting wizard

GUIs in the past

- Db2 Control Center (Db2cc)
  - Deprecated with Db2 LUW 9.7 and Db2 z/OS 10.1
  - Db2cc successor: Data Studio

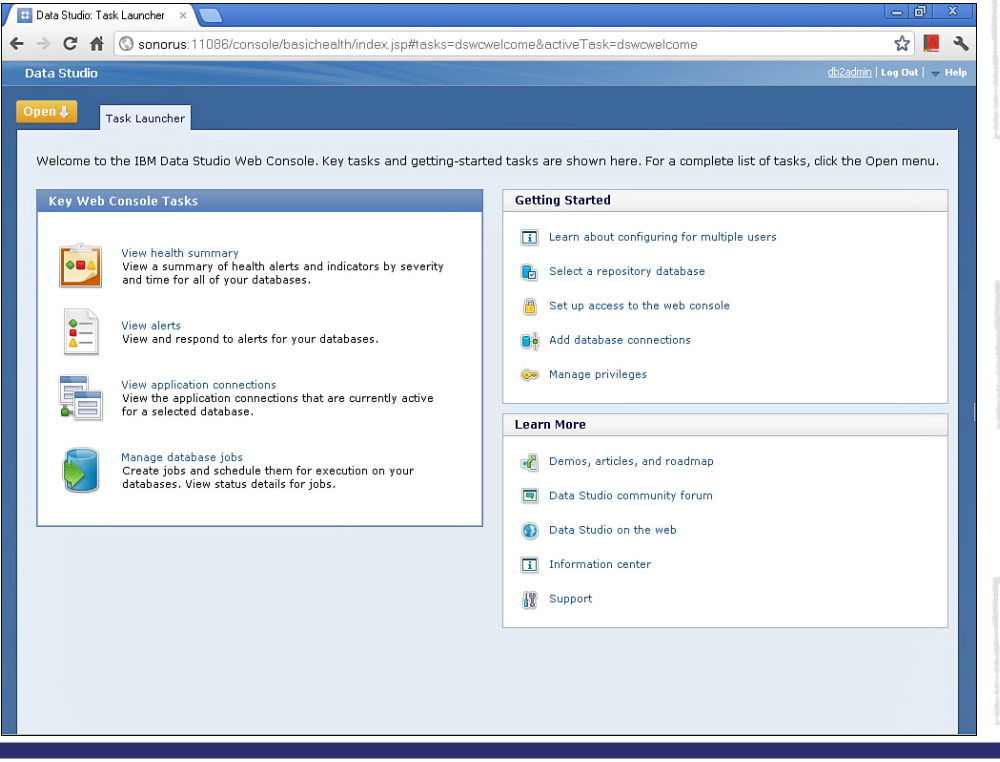




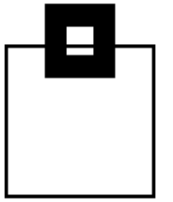
GUIs in the past

- Db2 Data Studio (Db2DS)
  - A Windows/Linux EclipsePlugin using Java Db2 connection

- Db2 Data Studio
  Web Console (Db2DSWC)
  - A Client/Server architecture, that enables web browser access



GUIs in the past



- Db2cc successor: Data Studio
  - True for most of the Db2cc tools, except:
    - Activity Monitor, Event Analyzer, Health Center, Web Console, Memory Visualizer, Query Patroller Center
      → InfoSphere Optim Performance Manager
    - Configuration Assistant
      → InfoSphere Optim Configuration Manager
  - With more complex licensing assoiciated:
    - InfoSphere Optim Performance Manager Extended Insight is a separately priced feature for InfoSphere Optim Performance Manager (part of InfoSphere Optim Performance Manager EE)
    - Data Studio consist of three components
    - The Index Advisor and Query Advisor require an InfoSphere Optim Query Workload Tuner license
    - Db2 Data Studio (Db2DS) renamed and bundled into Optim in 2009

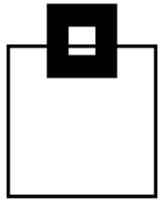
GUIs in the past

- Then Db2 Data Server Manager was introduced* and customers were confused whether this is a DS successor/replacement
  - Some IBMers said yes, some insisted they address different people:
    - DS is intended for developers
    - DSM is intended for DBAs
  - Unfortunately some DS features are not maintained with Db2 12 CD

  - Digging deeper indicates lots of the prior GUI Eclipse stuff and components "borrowed" from Db2DSWC
  - However, the labs are saying it is "very much a rewrite of the front end, but the smarts have been passed onto this next generation"

*in July 2010 also z/OS Management Facility for system programmers

GUIs in the past



- Bottom line/downside for ISVPs and customers:
  - Familiar UIs continue to be changed
  - Used features deprecated, or slightly shifted into other UIs
  - No single/common point of control
    - → ISPF still the one and only true (Db2) z/OS UI that stays reliably solid over the years
    - → ISPF still the one and only true (Db2) z/OS UI that is supported by IBM AND ISVs

Zowe ecosystem overview

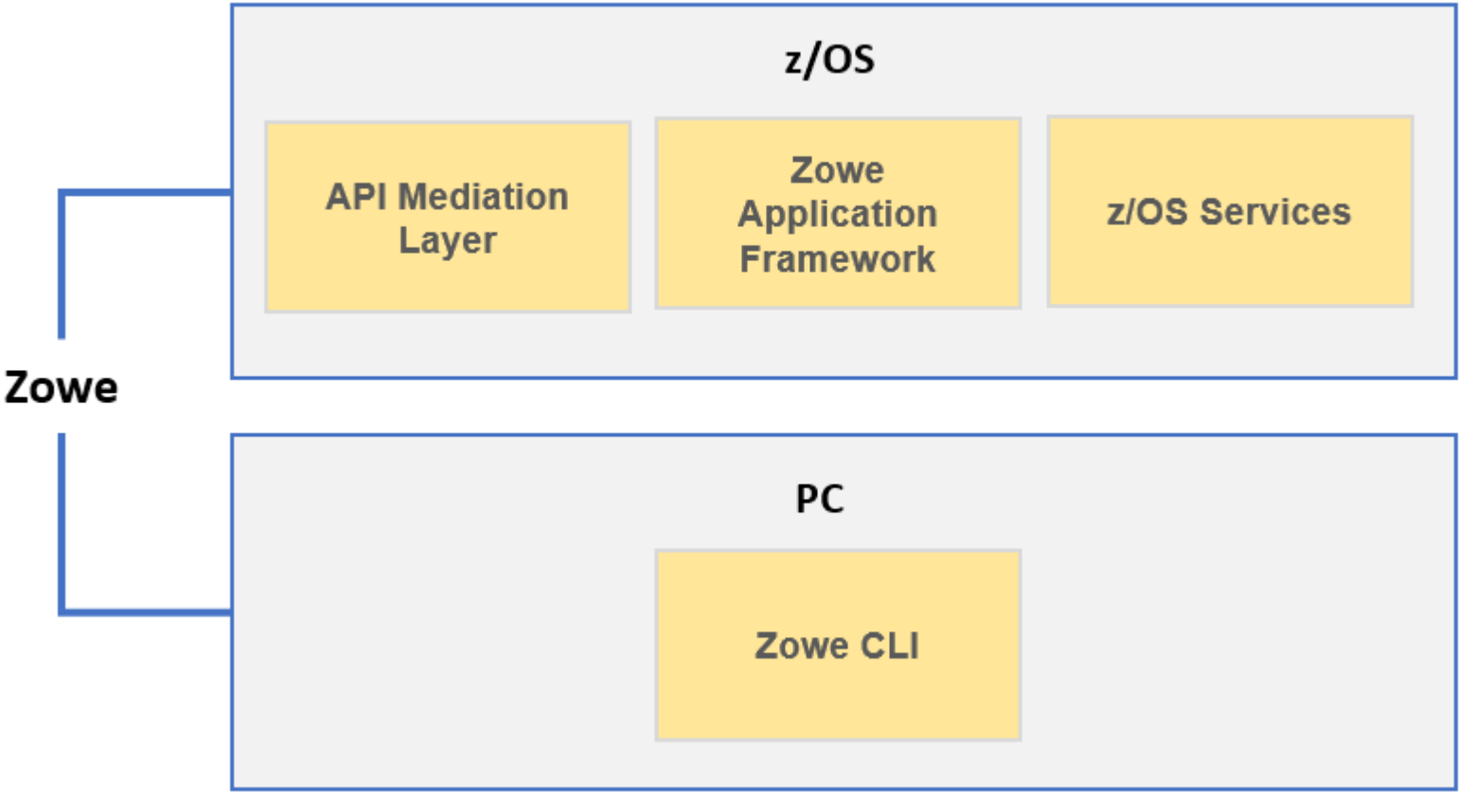
- At the SHARE 2018 conference, IBM, Rocket Software and CA Technolgies (now BROADCOM) announced Zowe – THE z ecosystem
  - Open source project licensed under EPL 2.0
  - Extensible framework
  - Fuses and unites „old“, solid mainframe UI (tn3270, VT) with latest UI (HTML5, JS, TS, CLI)
  - Based on and exploiting proven, rock solid technology (RLF, SAF, USS)
  - Introduces REST APIs, ESM microservices, discovery services, …

  - Addresses
    - Application Developers
    - System Programmers
    - DBAs
    - DevOps Architects

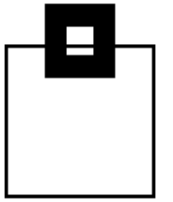
Zowe ecosystem overview

- Zowe is four major components:
  1. Application Framework
     The web UI that works with the underlying REST APIs presenting and bundling information in a modern, powerfull full screen mode
  2. z/OS Services
     Providing z/OS RESTful web service and deployment architecture for z/OS microservices
  3. Zowe CLI
     Allowing to interact with the mainframe to efficiently build z/OS applications
  4. API Mediation Layer
     Central point for all mainframe service REST APIs of the ecosystem

Zowe ecosystem overview



Zowe differentiation to prior GUIs



Zowe is

- the very first open source project on z/OS
- an extensible, common framework for existing and new applications
- designed to make the mainframe an agile, integrated platform
- *a* **THE** common UI for senior mainframe staff and new workforce
- a unified framework that merges proven and latest technology

...to

- demystify the mainframe and attract new people
- reduce the learning curve and improve productivity
- enhance integration and consumability
- simplify the architecture and reduce operational costs
- improve co-existance with a modern, platform-neutral interface

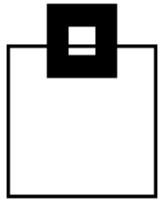
Zowe differentiation to prior GUIs

Zowe is vendor independent:

- Open source project under the Open Mainframe Project
- Free to be used under the Eclipse Public License 2.0
- Open, extensible interfaces of the code
- IBM, Rocket and BROADCOM (fka. CA) are founding members

→ Use, change and contribute

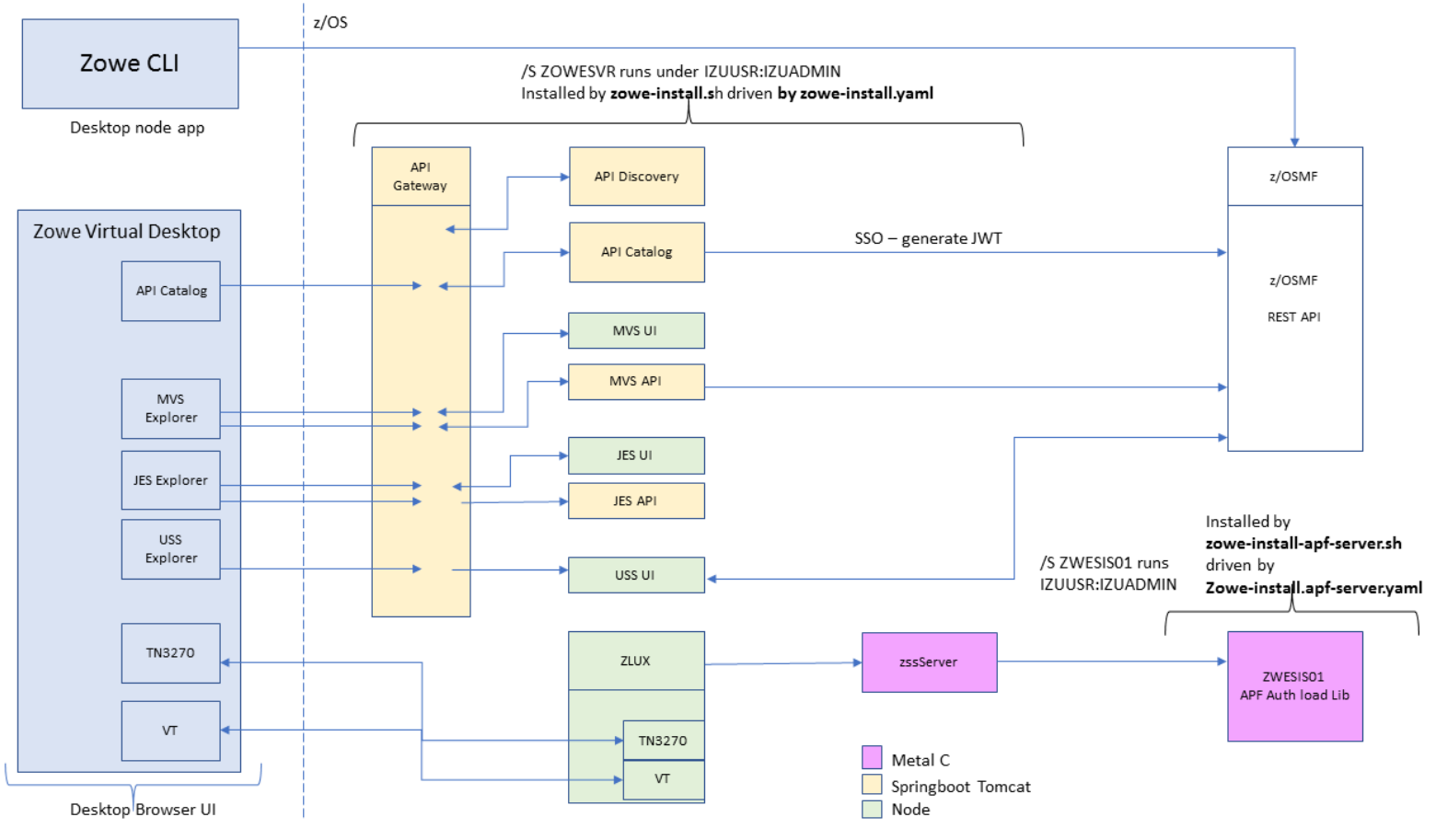
Zowe differentiation to prior GUIs



Zowe integrates nicely into an existing environment:

- Security management: SAF – System Authorization Facility
  - Controlling access by RACF, or other security products, like ACF2
- Resource management: RLF – Resource Limit Facility
  - Control processor usage of Db2 queries
- z/OS and USS support:
  - Explore JES, MVS, USS
  - Access and interact with subsystems like Db2, CICS
  - Browse and edit data sets
  - Execute JCL, Shell and z/OS commands, bash and z/OS scripts
- Platform independent browser technology:
  - HTML5, CSS, JS, TS, …
- Platform independent CLI
  - Node.js, npm, IDEs, Jenkins, TravisCI, …

Zowe components



Zowe components

- Zowe Application framework is four major components:
  1. Desktop
     Browser based web desktop
  2. Application Server
     Web services framework plus proxy applications that communicates with z/OS services and components
  3. ZSS Server
     REST services to support the Application Server
  4. Application plug-ins
     Included and addable applications to access the mainframe and to perform various tasks, e.g.
     - Dataset editor and browser (z/OS and USS)
     - Workflows
     - z/OS subsystem browser (JES, CICS, Db2, IMS, …)
     - …

Zowe components

- Zowe z/OS services contain the following core components:
  1. z/OS dataset services
     list, browse, edit, create, delete, … datasets and members
  2. z/OS job services
     list, browse, submit jobs

- A full list of capabilities of the RESTful API can be listed via the API catalog
  - The Open API Specification describes the APIs and allows to use any standard-based REST API developer tool, or API management process
  - APIs can be used by any application
  - z/OS services are running as microservices with a Spring Boot embedded Tomcat stack

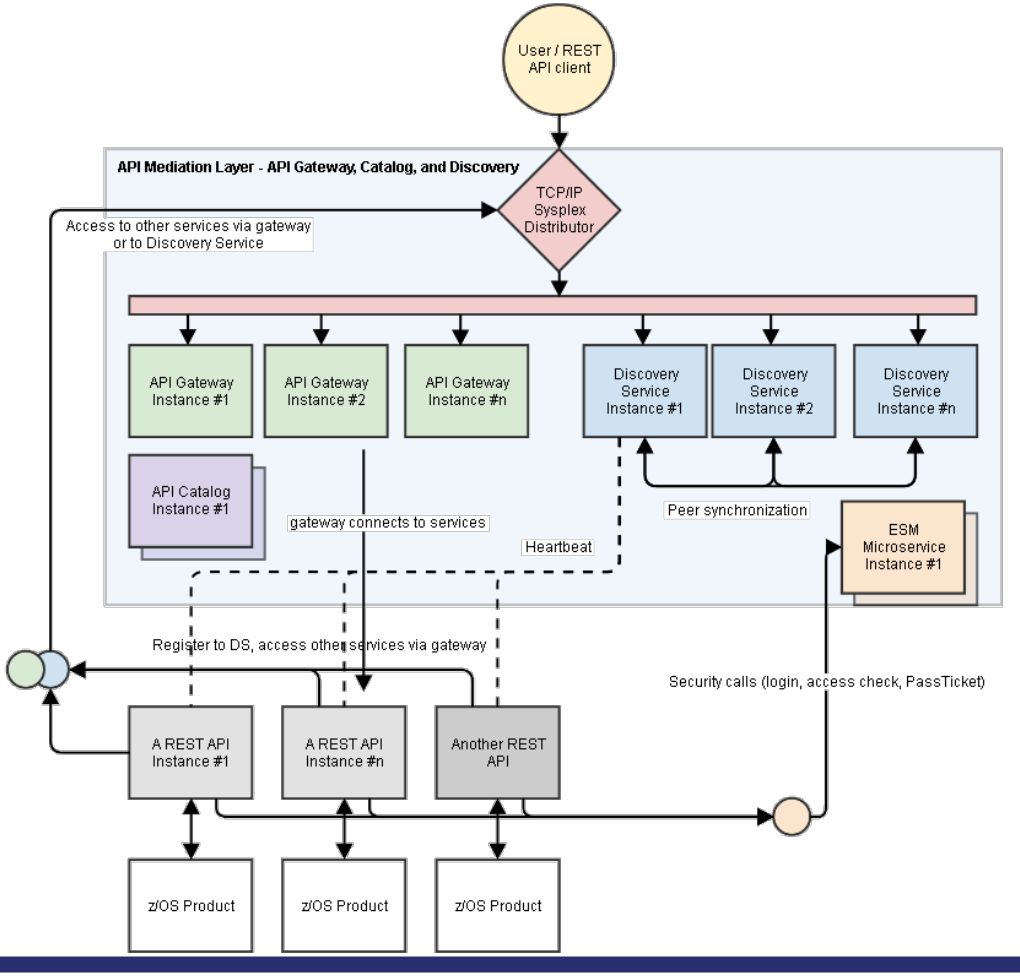
Zowe components

- Zowe CLI comes with the following capabilities:
  - Interact with files:
    - Create, edit, download, and upload data sets
  - Submit jobs:
    - Submit JCL from data sets or local storage, monitor the status, and view/download the output
  - Execute commands:
    - Issue TSO, or z/OS console commands
  - Integrated scripts:
    - Define scripts that do both mainframe and local tasks
  - Return JSON documents:
    - Return the data in JSON format to be used in other programming languages

Zowe components

- Zowe API mediation layer consists of the following components:
  - API gateway:
    - Clients interact with microservices behind a reverse proxy forwarding requests to the appropriate service
    - The gateway is built on Netflix Zuul and Spring Boot technology
  - Discovery services:
    - Accepts the REST service announcements and serves active ones
    - The service is built on Netflix Eureka and Spring Boot technology
  - API catalog:
    - UI catalog of published APIs along with their documentation (Swagger) and status
    - Services can be implemented by multiple instances for high-availability or scalability
  - ESM microservices:
    - Authenticates and authorizes users with mainframe credentials

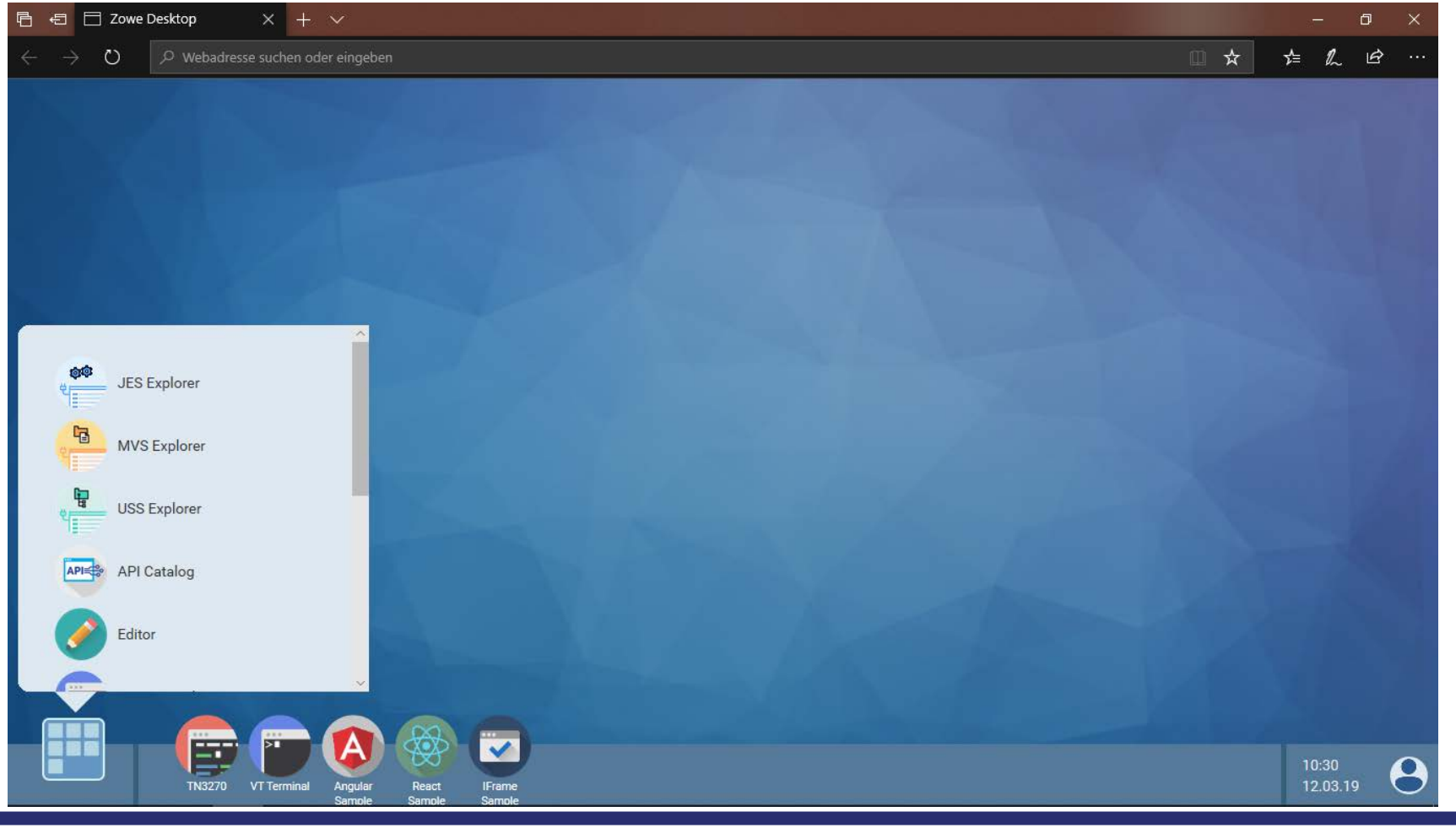
Zowe components



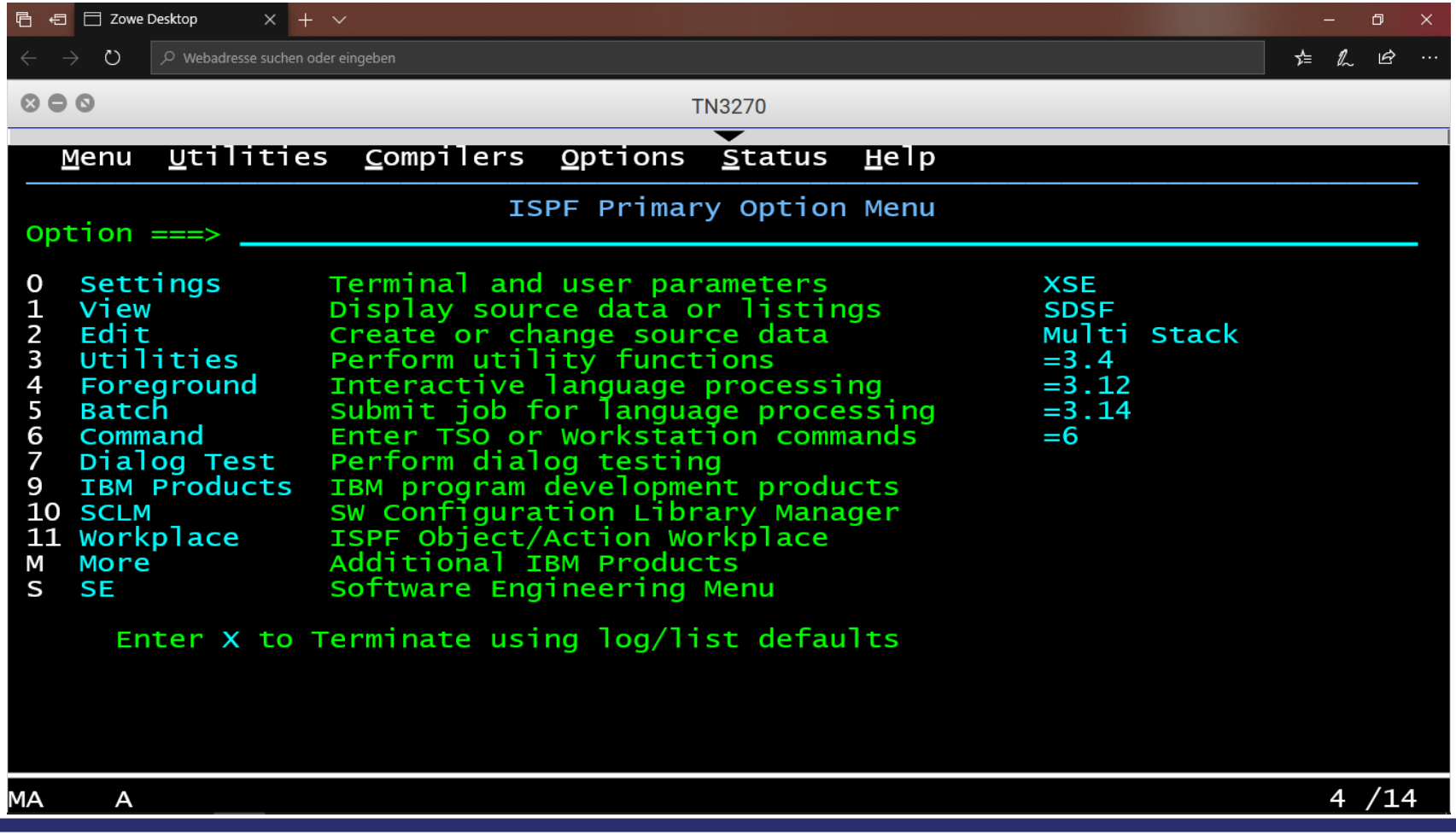
Zowe components @ github.com

- zowe-cli - Zowe CLI
- ztrial-scenarios - This repo tracks the zTrial scenarios for Zowe.
- zowe-common-c - C Libraries for various OS & Networking needs
- zlux-app-server - A collection of build, deploy, and run scripts & configuration files for running a simple zLUX server.
- zlux - The top-level superproject for zLUX. zLUX includes the Zowe Desktop framework in addition to several built-in apps and an example server implementation.
- docs-site - Documentation for the Zowe project
- community - Community Engagement - Contribution Guidelines, Meeting Minutes, and more
- zowe-cli-db2-plugin - DB2 Plugin for the Zowe CLI
- zowe-cli-cics-plugin - CICS Plugin for the Zowe CLI
- zowe-cli-sample-plugin - Plugin Tutorial for Zowe CLI
- perf-timing - Performance tests
- api-layer - Zowe API Mediation Layer
- sample-trial-app
- zowe-install-packaging - Packaging repository for the Zowe Install scripts and files
- imperative - Imperative CLI Framework
- vscode-extension-for-zowe - Visual Studio Code Plug-in for Zowe, which lets users interact with z/OS data sets on a remote mainframe instance. Powered by Zowe CLI.
- cpu_usage_sample - An example of a Spring Boot application
- zowe-install-test - Perform Zowe installation and smoke test
- zlux-server-framework - Contains essential zLUX proxy server components including SSO and service catalogs
- ztrial-sample-cli-plugin
- zlux-build - Repository for common build scripts among various superprojects
- explorer-jes-fvt - Functional tests for jes explorer
- explorer-jes
- explorer-mvs
- explorer-uss
- explorer-ui-server - Simple HTTPS web server, used by explorer UI plugins
- data-sets - Repo for the springboot based data set APIs
- jobs - Repo for the jobs api controller and code
- explorer-api-common - common repo for explorer api projects
- zlux-app-manager - zLUX Framework components for management of zLUX Apps. Used for window managers or web layouts.
- zlc - Zowe Leadership Committee collaboration
- vt-ng2 - A simple USS/Unix/VT terminal emulator written in Angular and Javascript
- tn3270-ng2 - A TN3270 emulator written in Angular and Javascript
- zss - Zowe Secure Services Server for enabling low-level microservices
- zlux-ng2 - Angular Hosting Environment for the zLUX Framework's web components
- zss-auth - Auth handler for App server to connect to ZSS through standard ZSS login
- db-browser - A database viewer and editor for working with a variety of databases within the Zowe Desktop
- db-browser-db2 - db2 module for db-browser App for Zowe
- jupyter-app - A Zowe App for displaying Jupyter
- zos-subsystems - An example app showing z/OS infrastructure
- workshop-starter-app - An App to provide at the start of a workshop session to showcase Zowe App development & App-to-App communication
- file-transfer-app - An App for transfering files to and from a mainframe
- zosmf-auth - Auth handler for App server to connect to z/OSMF through standard z/OSMF login
- zlux-workflow
- zlux-shared - zLUX framework components that are utilized both by the server and in the web browser
- zlux-platform
- zlux-editor - A simple editor in a browser
- sample-react-app - Sample to showcase a react app that natively can be presented into the Zowe desktop
- sample-iframe-app
- sample-angular-app
- spring-boot-jzos-sample - An example of a Spring Boot sample to be statically linked into the API Gateway
- zowe-promote-publish - Zowe Pipeline to Promote and Publish a PAX Candidate
- release-management - Material and activities related to release management
- zowe-cli-standalone-package - Jenkins pipeline which generates a Zowe CLI ZIP containing the base CLI and Zowe plugins.
- sample-node-api - A sample node js api for finding cars and accounts for a dealership
- sample-trial-react-app - Sample React App
- zowe-cli-version-controller - Main controller and maintainer of the versioning scheme
- zlux-grid
- jenkins-slave-images
- zlux-file-explorer
- orion-editor-component
- zlux-widgets
- zlux-file-properties
- explorer-server-tests
- explorer-server - Explorer Server component contribution
- workshop-user-browser-app - Starter files & a tutorial README to get started on building a simple Zowe App
- explorer-server-auth
- taskManager - Shows running services / processes on the z/OS Sysplex Served by Zowe
- zowe.github.io - Testing GitHub Pages for Community WebSite as an Alternative to Wordpress
- zowe-cli-sample-scripts - Demo scripts for the Zowe CLI
- Onboarding-scripts - Template scripts for extenders to onboard their products with
- explorer-utilities- Explorer shared utilities project
- zowe-cli-profile-migration - Zowe CLI Profile Migration Tool
- docs-site-temp
- explorer-injector
- webui-scenarios - Several sample projects that create WebUI's that integrate into Zowe
- explorer-model - The Explorer server model project

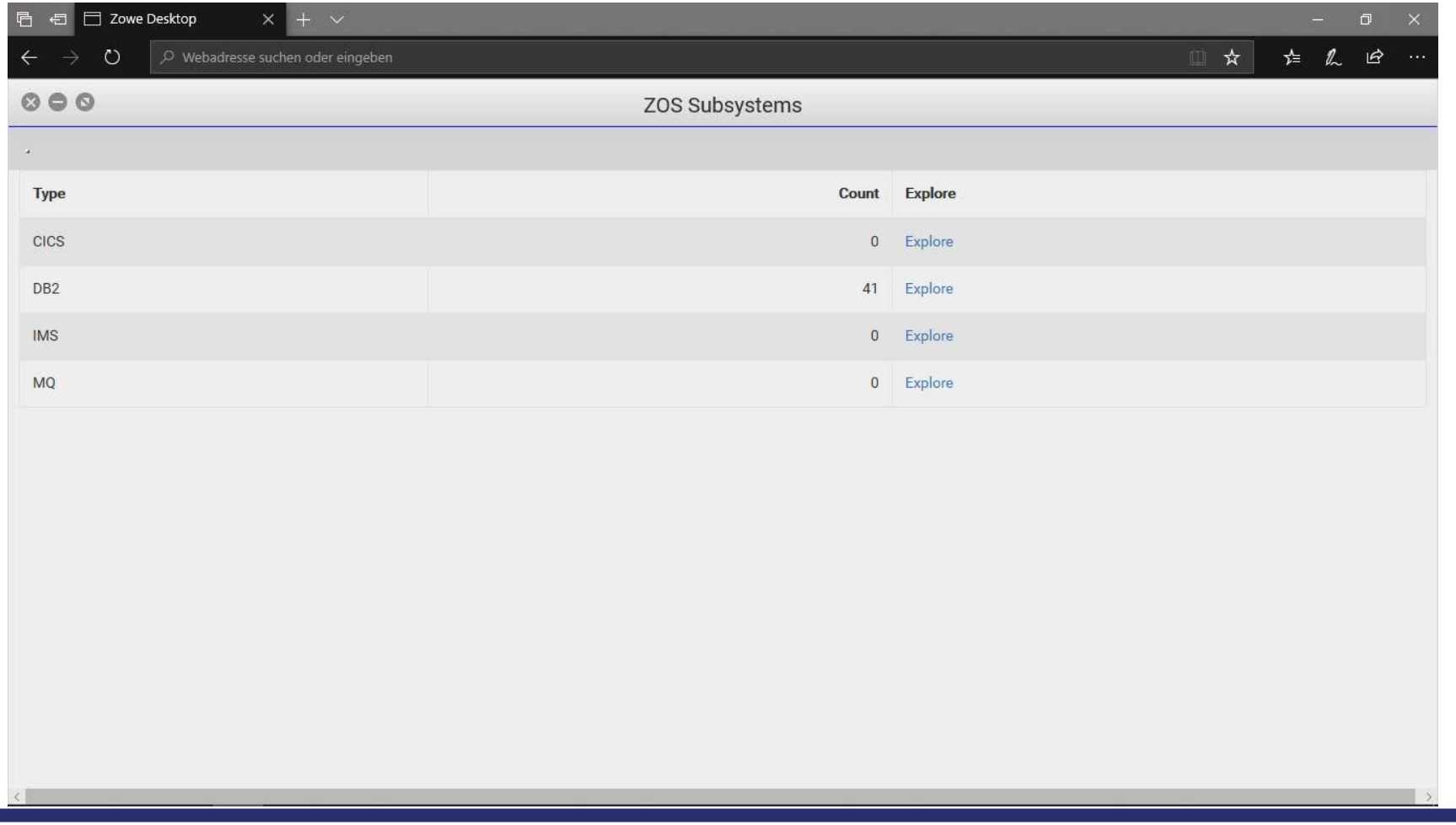
Zowe examples – the Zowe desktop



Zowe examples – the tn3270 app ☺

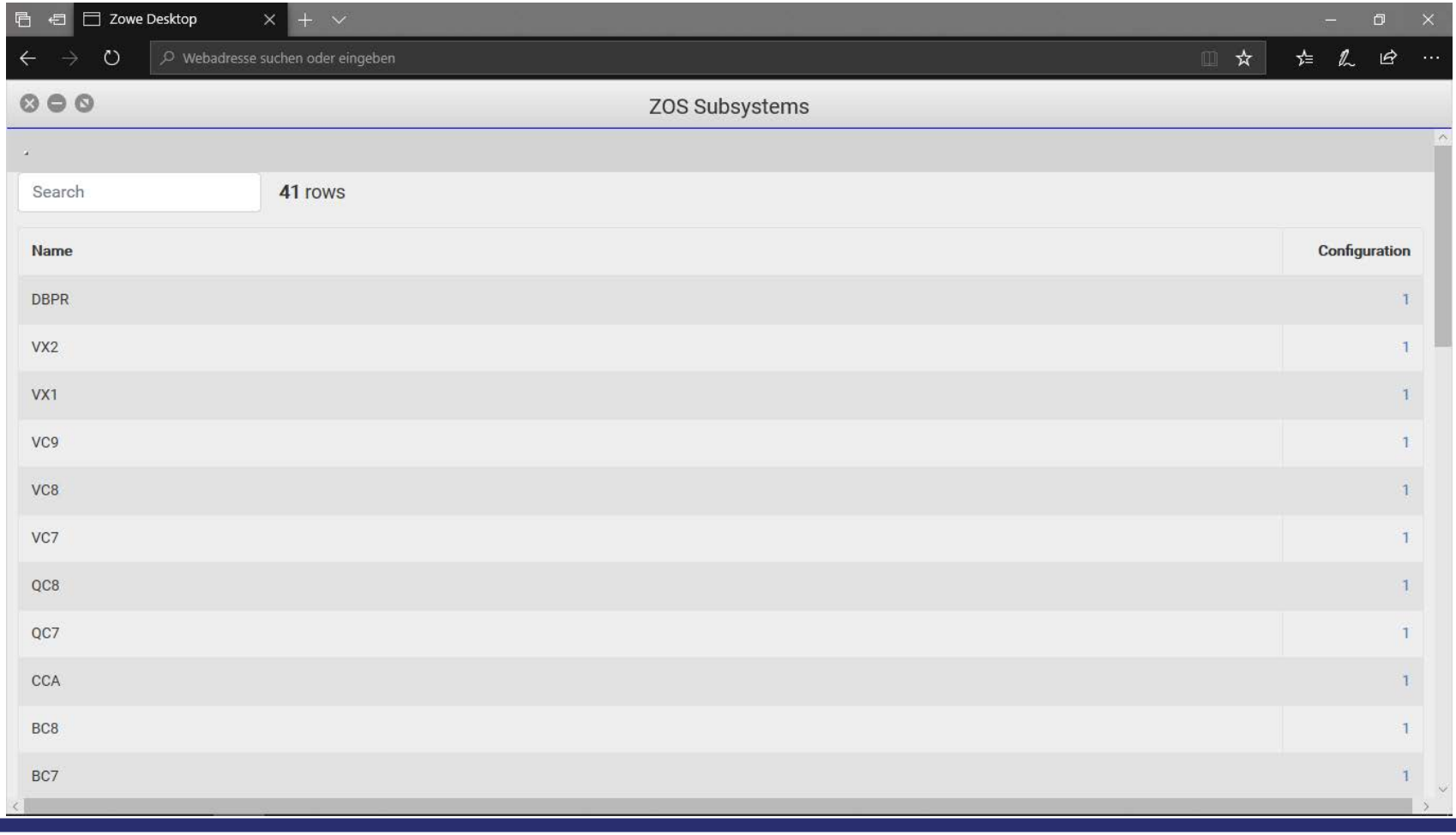


Zowe examples – z/OS Subsystems



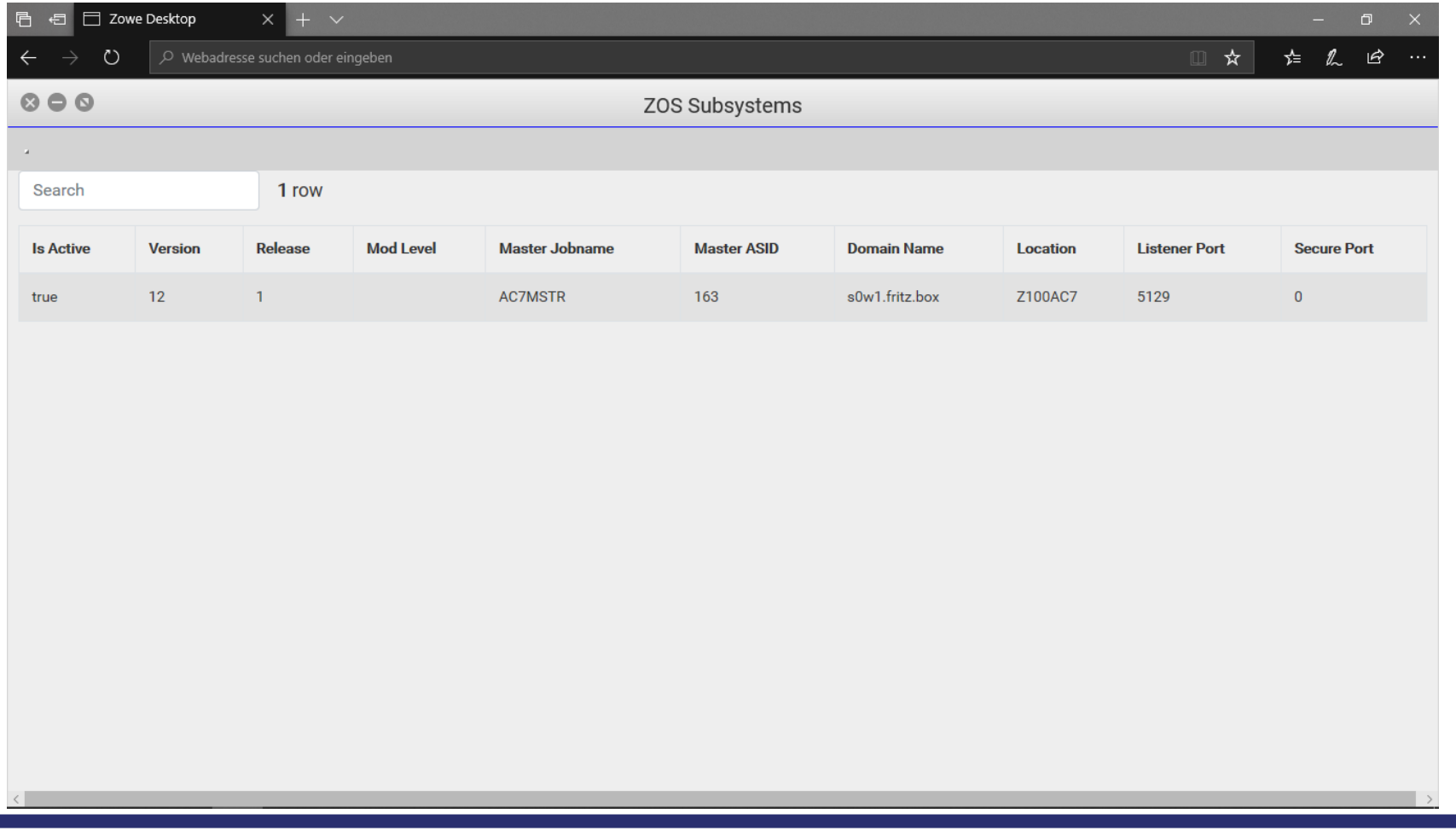
Type	Count	Explore
CICS	0	Explore
DB2	41	Explore
IMS	0	Explore
MQ	0	Explore

Zowe examples – z/OS Subsystems



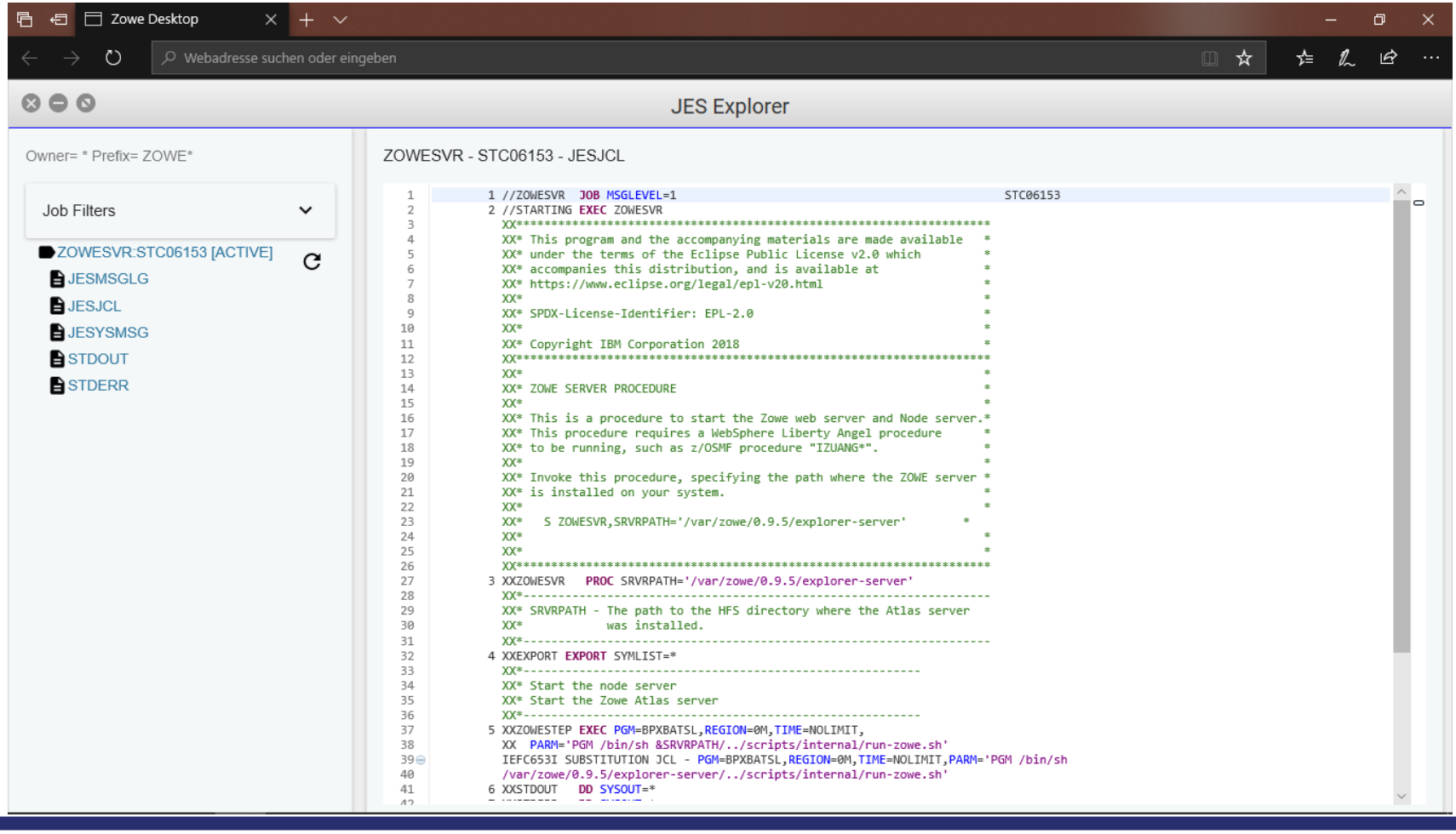
Name	Configuration
DBPR	1
VX2	1
VX1	1
VC9	1
VC8	1
VC7	1
QC8	1
QC7	1
CCA	1
BC8	1
BC7	1

Zowe examples – z/OS Subsystems

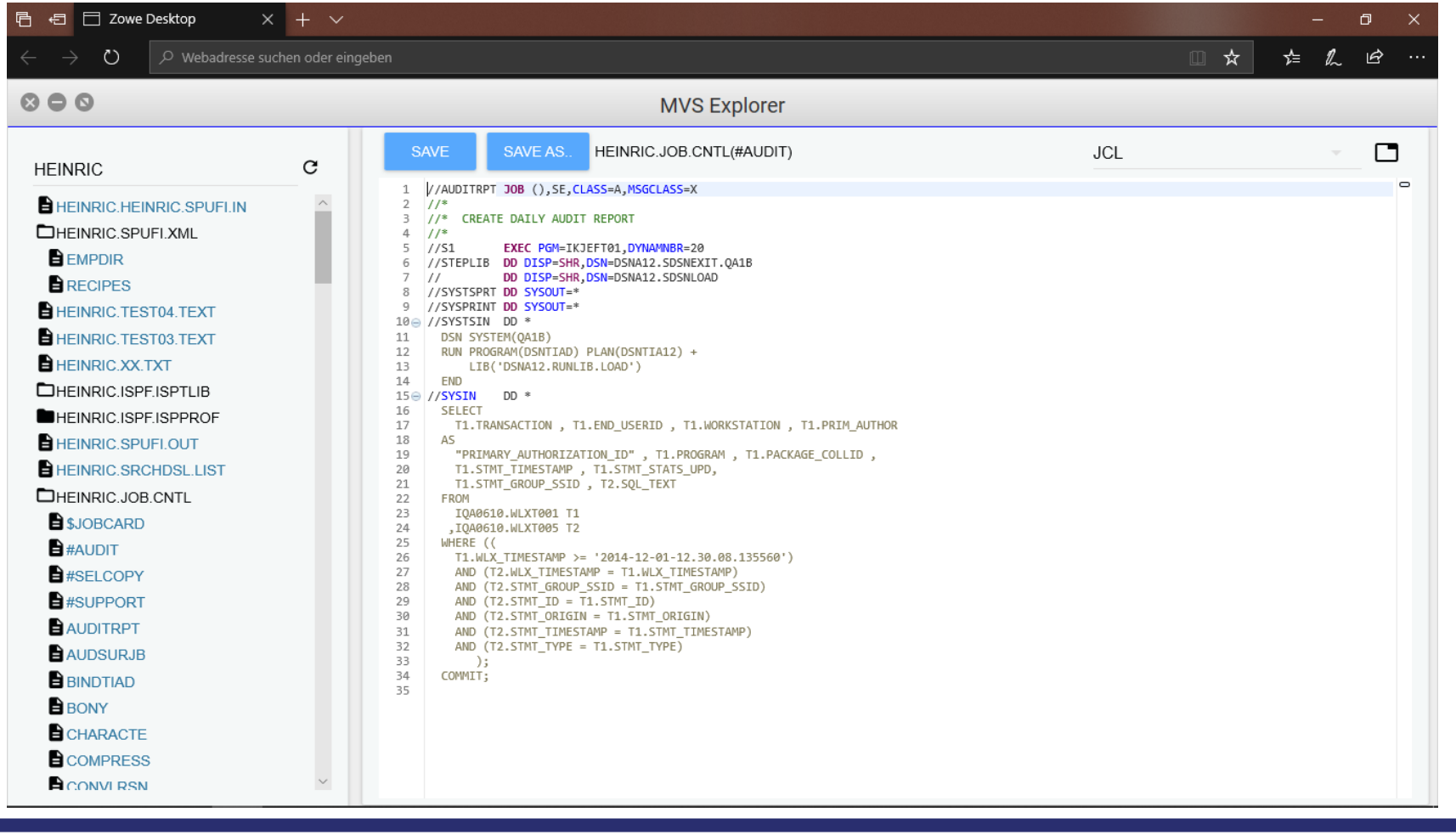


Is Active	Version	Release	Mod Level	Master Jobname	Master ASID	Domain Name	Location	Listener Port	Secure Port
true	12	1		AC7MSTR	163	s0w1.fritz.box	Z100AC7	5129	0

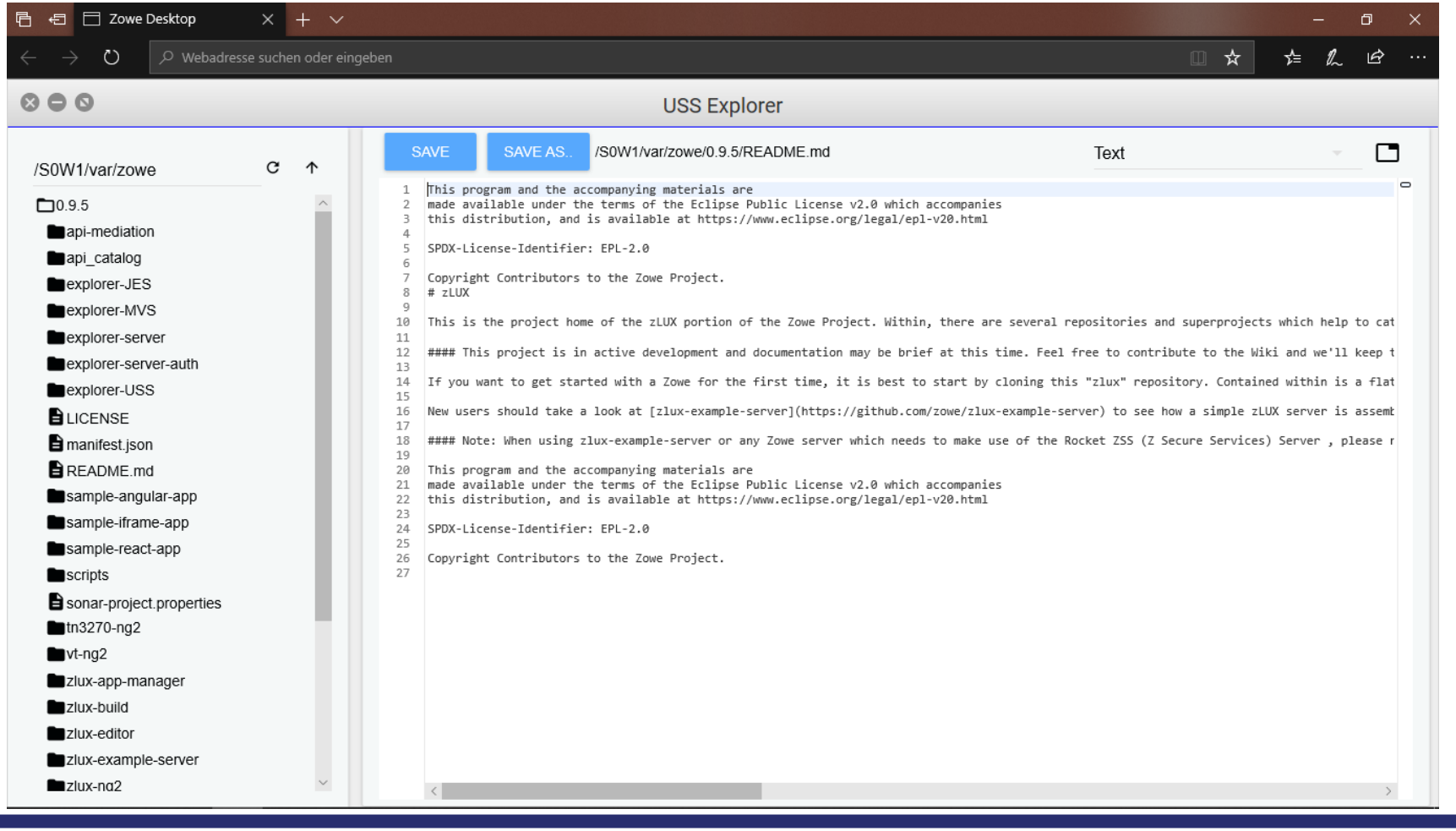
Zowe examples – the JES Explorer



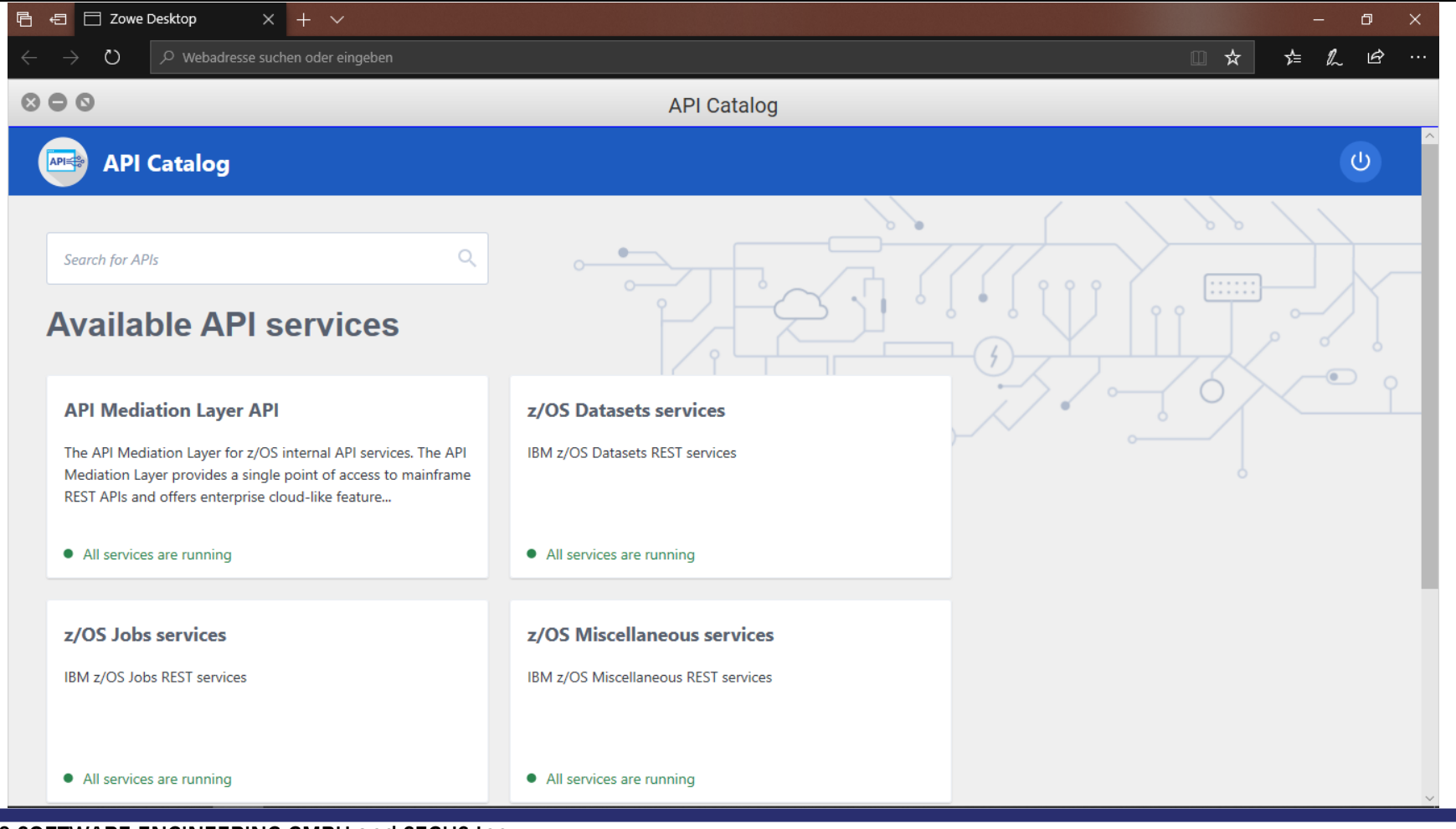
Zowe examples – the MVS Explorer



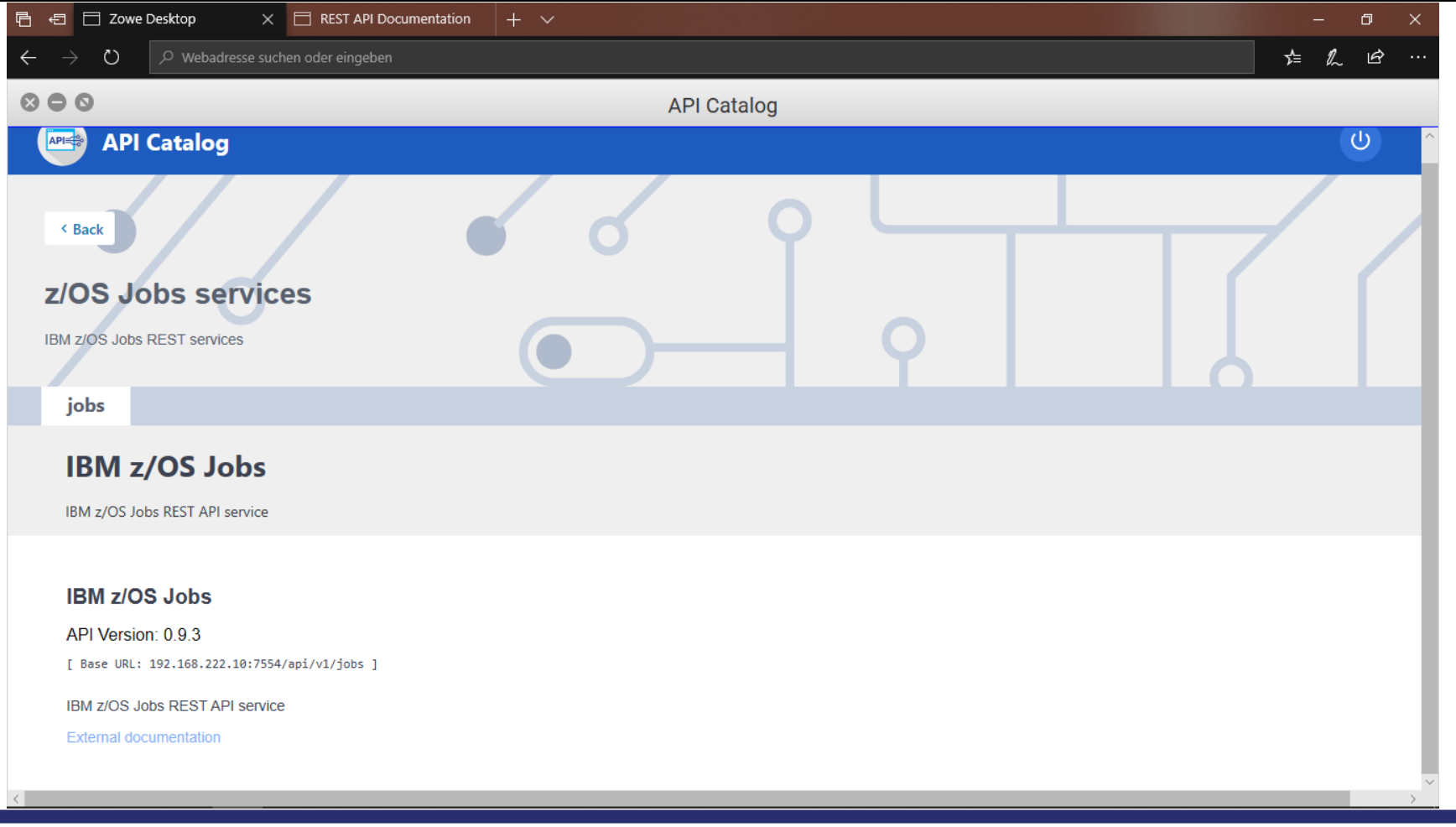
Zowe examples – the USS Explorer



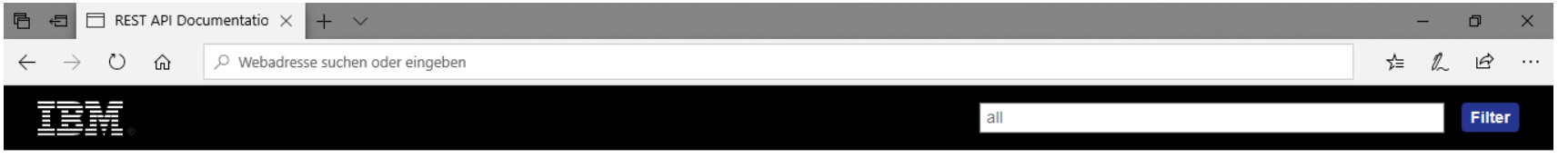
Zowe examples – the API Catalog



Zowe examples – the API Catalog



Zowe examples – the API Catalog

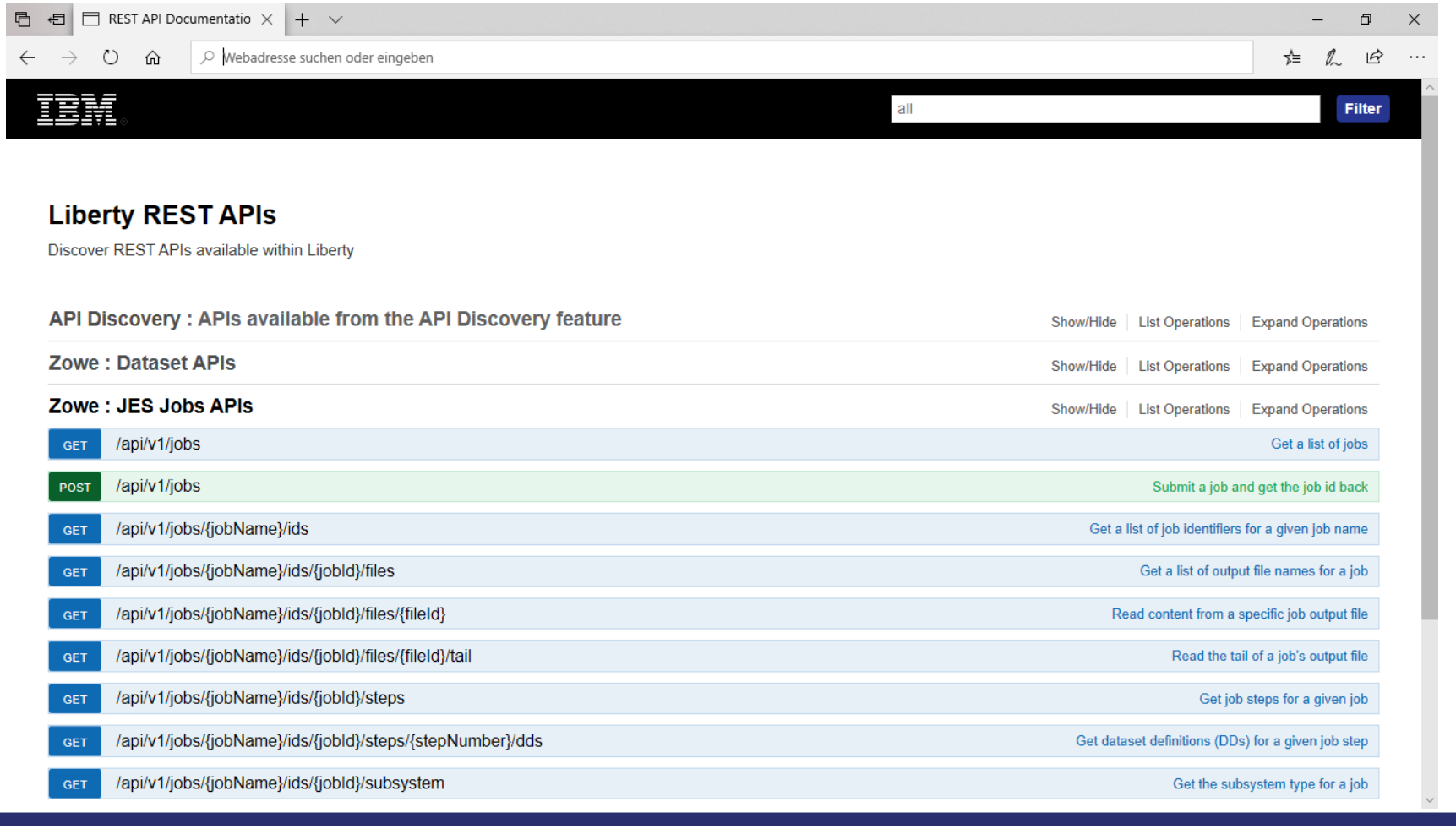


Liberty REST APIs

Discover REST APIs available within Liberty

API Discovery : APIs available from the API Discovery feature	Show/Hide	List Operations	Expand Operations
Zowe : Dataset APIs	Show/Hide	List Operations	Expand Operations
Zowe : JES Jobs APIs	Show/Hide	List Operations	Expand Operations
Zowe : System APIs	Show/Hide	List Operations	Expand Operations
Zowe : USS Files APIs	Show/Hide	List Operations	Expand Operations
Zowe : zOS System APIs	Show/Hide	List Operations	Expand Operations

Zowe examples – the API Catalog



Zowe examples – the API Catalog

REST API Documentatio

Webadresse suchen oder eingeben

Zowe : JES Jobs APIs

Show/Hide | List Operations | Expand Operations

GET /api/v1/jobs — Get a list of jobs

Implementation Notes

This API returns the a list of jobs for a given prefix and owner.

Response Class (Status 200)

Ok

Model | Example Value


```
[
  {
    "name": "string",
    "jobInstances": [
      {
        "jobId": "string",
        "jobName": "string",
        "owner": "string",
        "type": "string",
        "status": "ACTIVE",
```


Response Content Type application/json

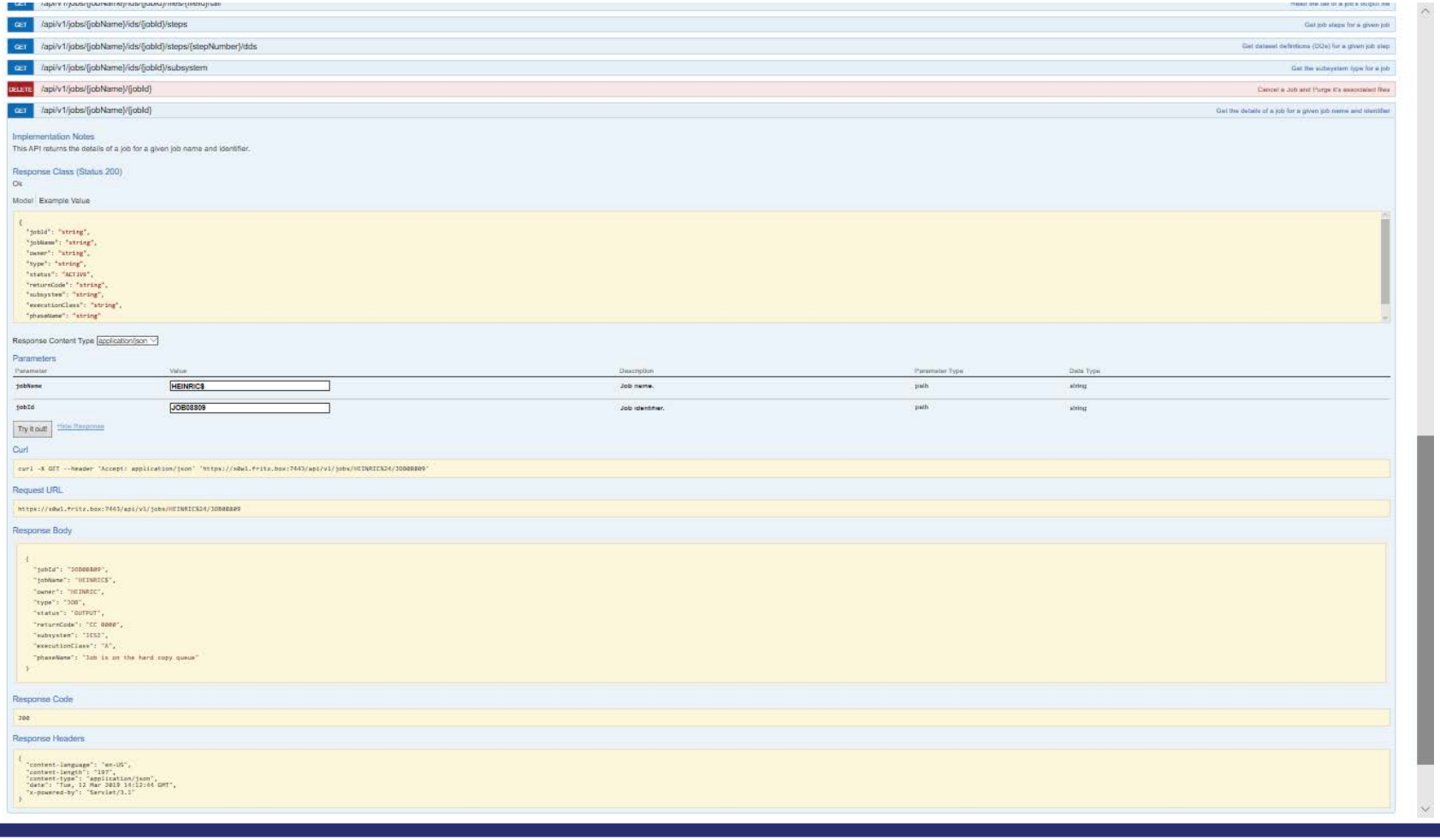
Parameters

Parameter	Value	Description	Parameter Type	Data Type
prefix	*	Job name prefix. If omitted, defaults to '*'.	query	string
owner		Job owner. Defaults to requester's userid.	query	string
status		Job status to filter on, defaults to ALL.	query	string

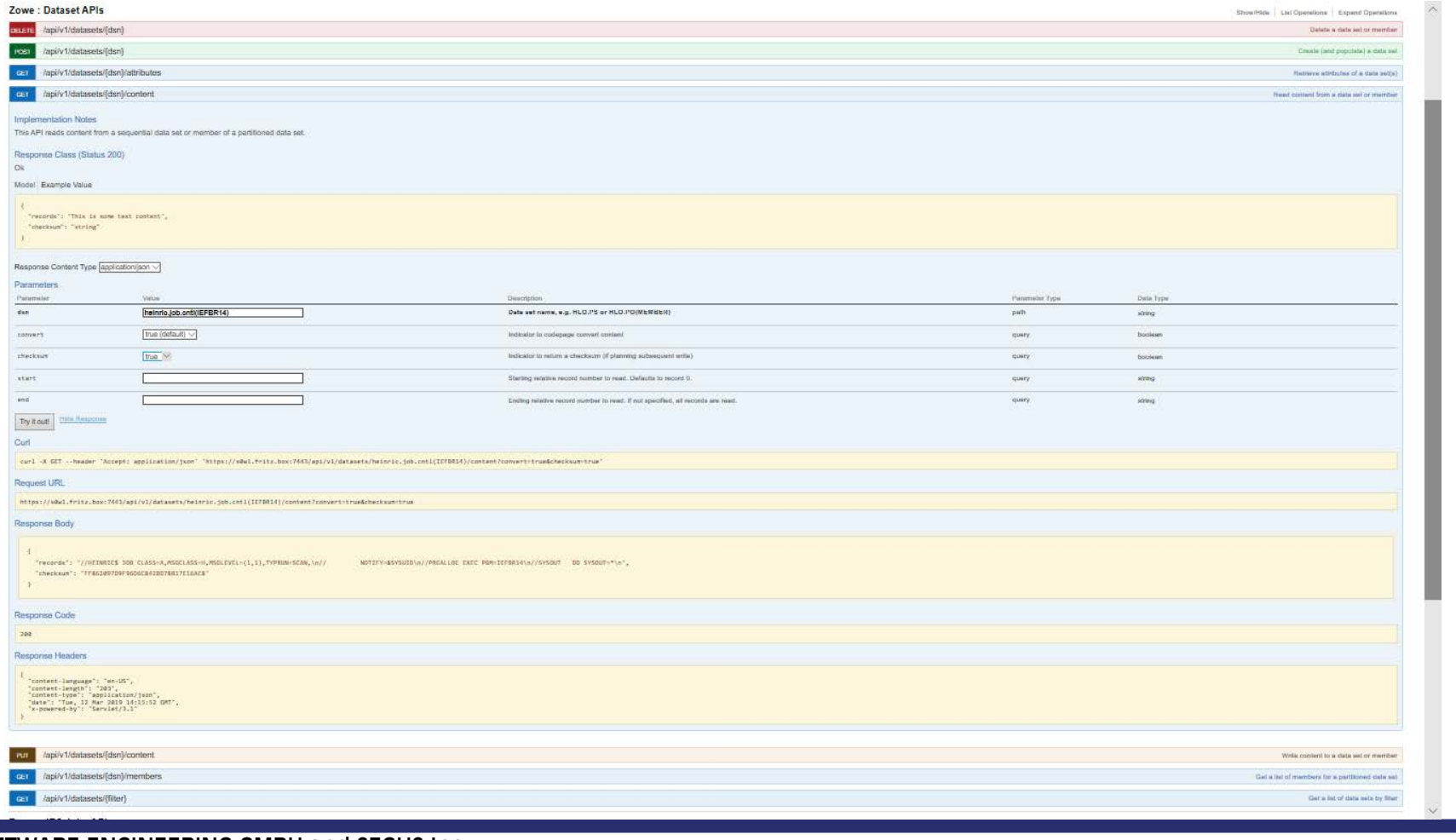
Try it out!

Zowe examples – the API Catalog

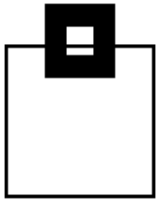
Zowe examples – the API Catalog



Zowe examples – the API Catalog



Zowe examples – the API Catalog



PUT /api/v1/datasets/{dsn}/content — Write content to a data set or member

Implementation Notes

This API writes content to a sequential data set or partitioned data set member.

Parameters

Parameter	Value	Description	Parameter Type	Data Type
dsn	heinric.job.cntl(iefbr14)	Dataset name	path	string
body	{ "records": "//HEINRIC$ JOB CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1),\n// NOTIFY=&SYSUID\n//PREALLOC EXEC PGM=IEFBR14\n//SYSOUT DD SYSOUT=*\n",	Request content (content-type:application/json) in the form: {"records":"data Content","checksum":"checksum_value"} If checksum is passed and it does not match the checksum returned by a previous read, it is deemed a concurrent update has occured, and the write fails.	body	string

Parameter content type: application/json

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Ok		

Try it out! Hide Response

Curl

```
curl -X PUT --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{ \
  "records": "//HEINRIC$ JOB CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1),\n//        NOTIFY=%26SYSUID\n//PREALLOC EXEC PGM=IEFBR14\n//SYSOUT   DD SYSOUT=*\n", \
  "checksum": "FF862097D9F96D6C842BD78817E16AE8" \
 }' 'https://s0w1.fritz.box:7443/api/v1/datasets/heinric.job.cntl(iefbr14)/content'
```

Request URL

```
https://s0w1.fritz.box:7443/api/v1/datasets/heinric.job.cntl(iefbr14)/content
```

Response Body

```
no content
```

Response Code

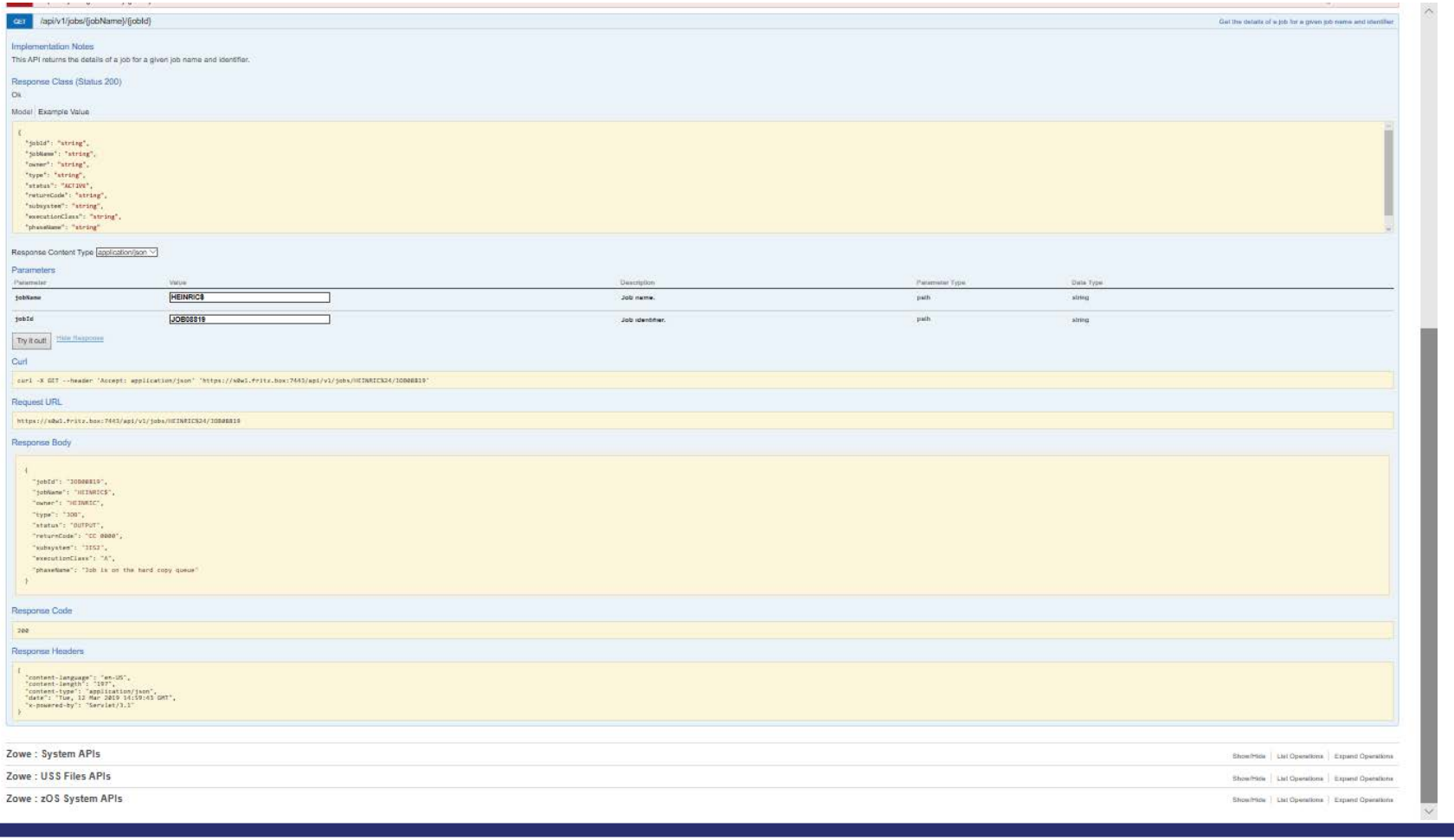
```
200
```

Response Headers

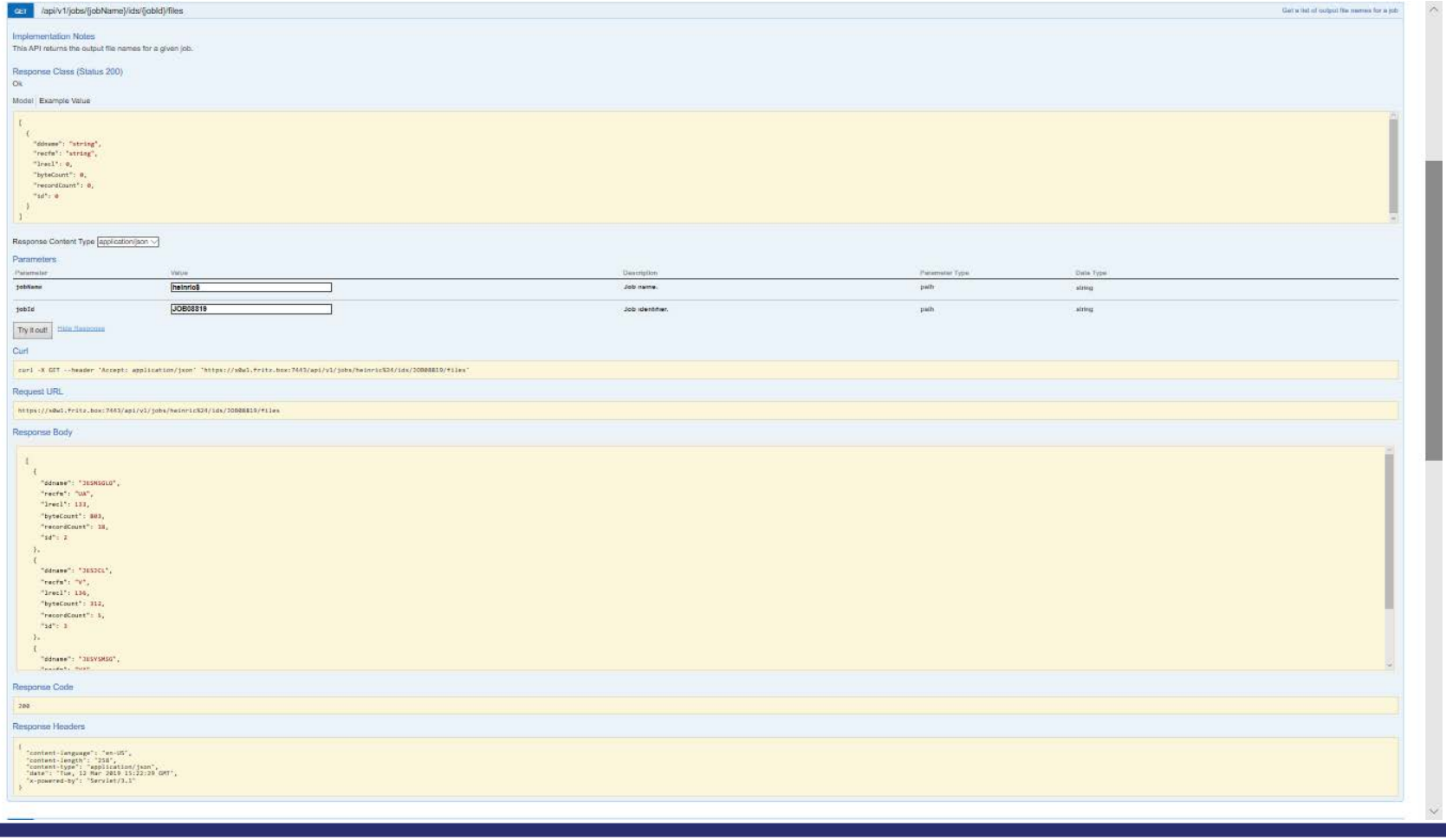
```
{
  "content-language": "en-US",
  "content-length": "0",
  "date": "Tue, 12 Mar 2019 14:19:01 GMT",
  "x-powered-by": "Servlet/3.1"
}
```

Zowe examples – the API Catalog

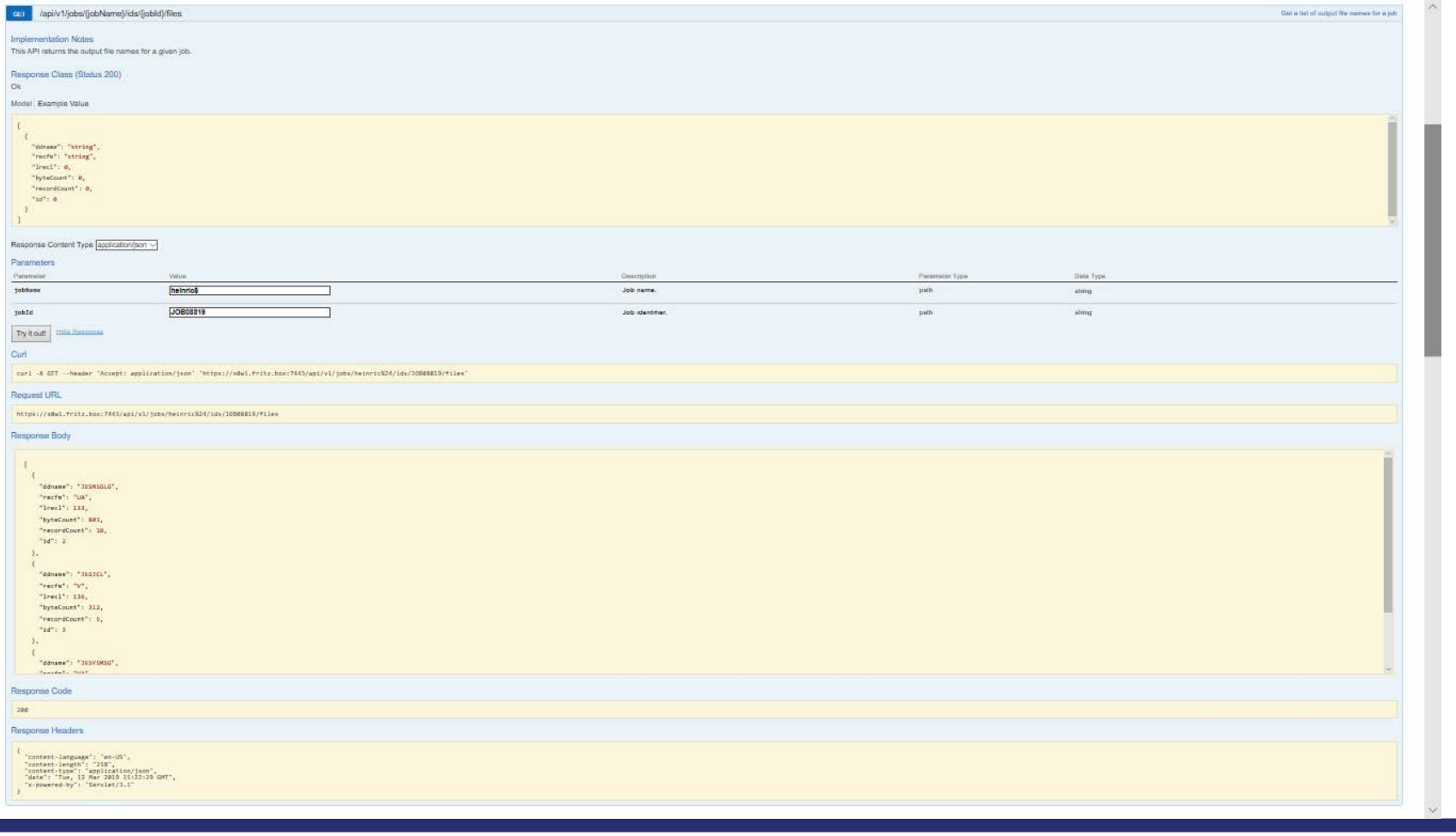
Zowe examples – the API Catalog



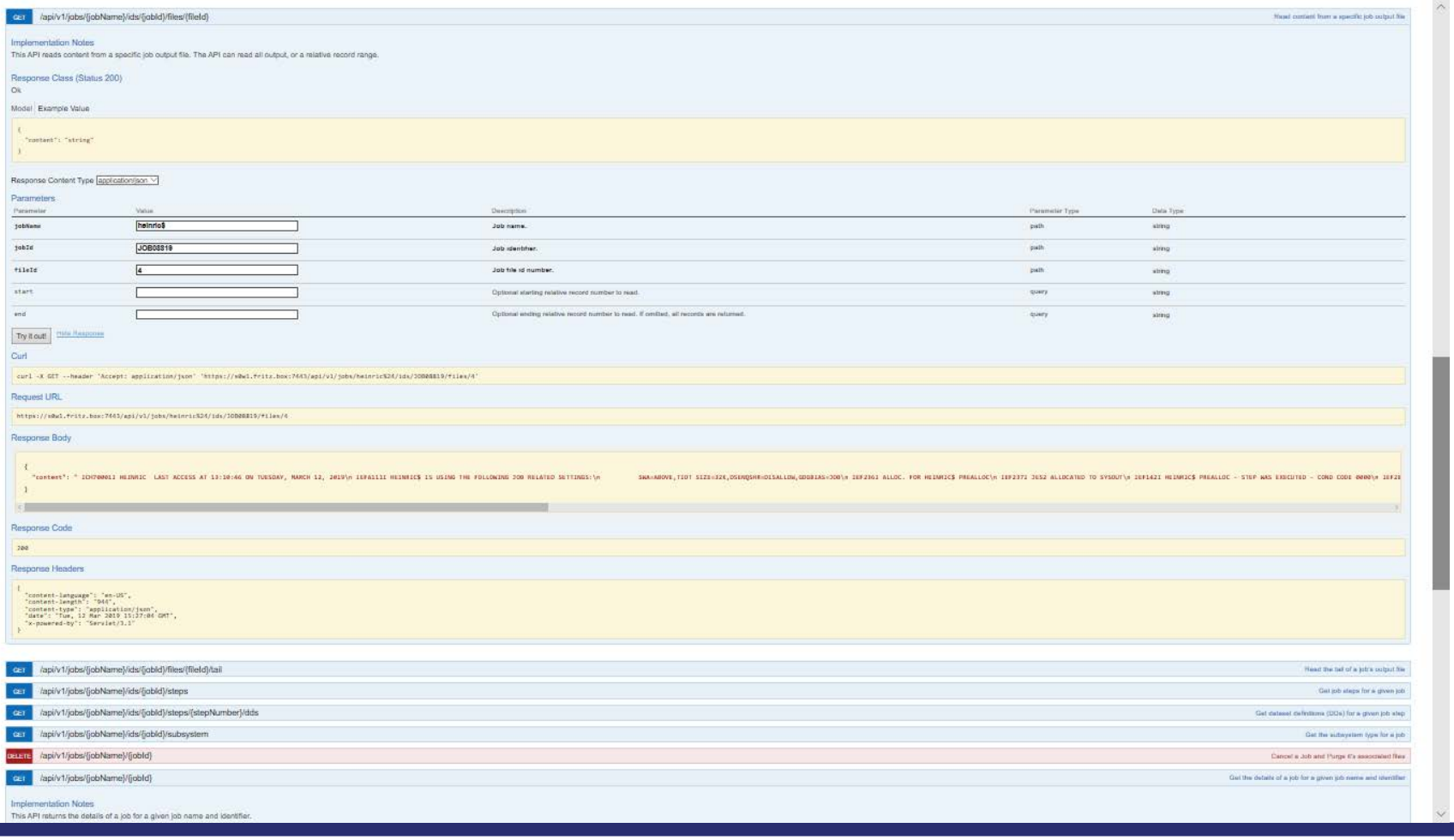
Zowe examples – the API Catalog



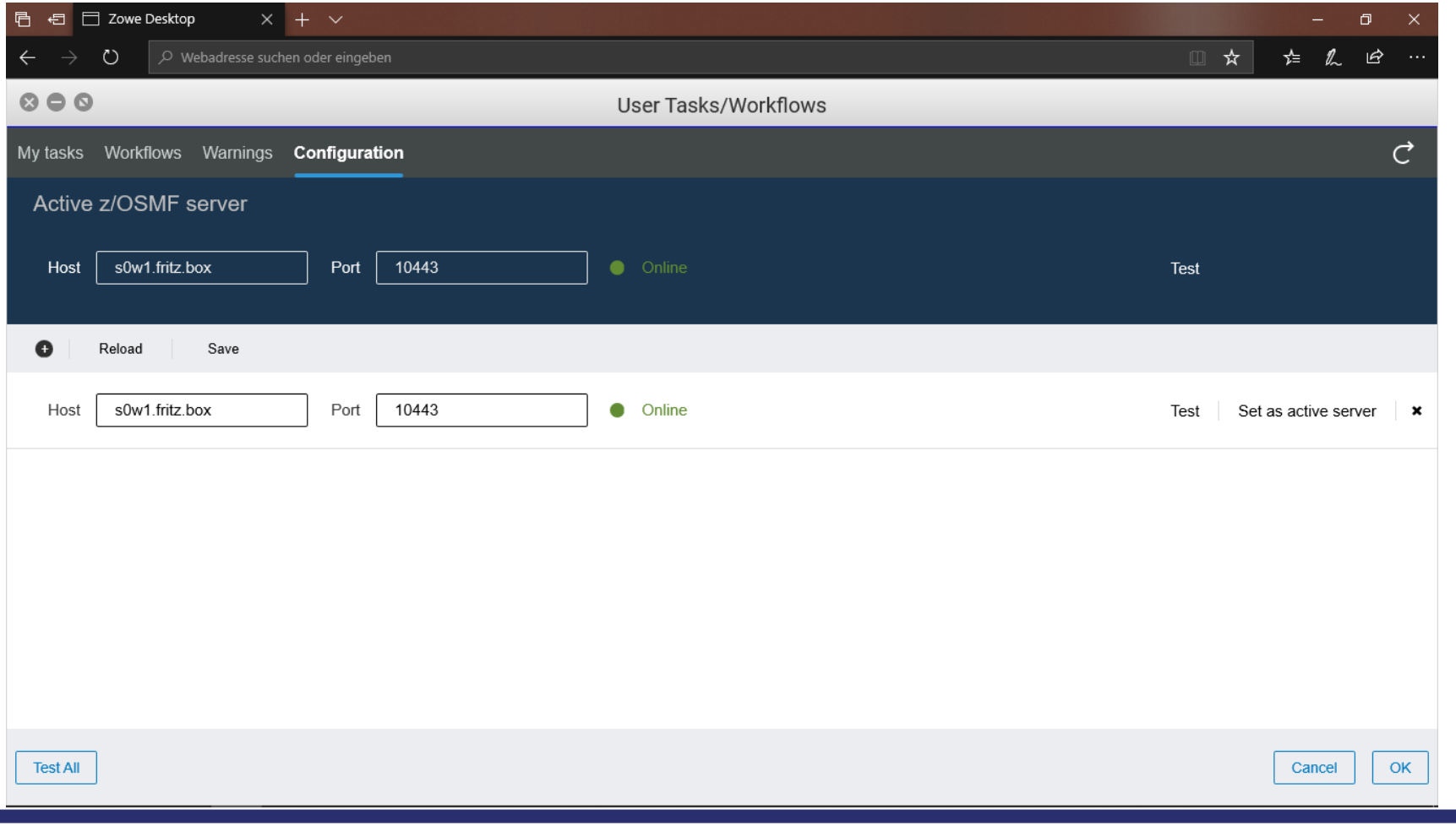
Zowe examples – the API Catalog



Zowe examples – the API Catalog



Zowe examples – User Tasks/Workflows



Zowe examples – User Tasks/Workflows

Zowe Desktop

Webadresse suchen oder eingeben

User Tasks/Workflows

My tasks | Workflows | Warnings | Cor

All Workflows 0

Actions>

Workflow	Status	Progress
No workflows		

0 of 0

0-0 of 0

Create Workflow

Advanced Mode

Name: ContinousDelivery DeploymentCheck for Db2 z/OS

Variables

Name	Value
sourceDb	ACG
targetDb	BCG
testMode	No
localSystemDSN	SE.PRODUCT.EASY0200.CDDC.EXEC
remoteSystemDSN	SE.PRODUCT.EASY0200.CDDC.EXEC
scenario	CDCAFDA3
backupDSN	SE.BACKUP.ACG.DUMP
workloadSet	2019-03-18-11.56.25.524960

Cancel | OK

Zowe examples – the Editor

Zowe Desktop

Webadresse suchen oder eingeben

Editor

File Language Server Language Help

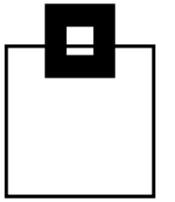
File Explorer

- cfzsrv
- debug
- work
- aopstc
- P390A
- log
- heinric
  - .sh_history
  - .profile
  - .ssh
  - tmp.xml
  - mywf.xml
- dudek
- bin
- usr
- lib
- opt
- samples
- $VERSION
- $SYSNAME
- dev
- tmp

mywf.xml

```
1  <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2  <workflow>
3     <autoTakeOwnership>false</autoTakeOwnership>
4     <workflowInfo>
5        <workflowID scope="none">MyWF</workflowID>
6        <workflowDescription>ContinuousDelivery DeploymentCheck for Db2 z/OS</workflowDescription>
7        <workflowVersion>1.0</workflowVersion>
8        <vendor>SOFTWARE ENGINEERING</vendor>
9        <General/>
10    </workflowInfo>
11    <step name="Prepare" optional="false">
12       <title>Prepare Systems for Cloning</title>
13       <description/>
14       <step name="CDCAFDA3" optional="false">
15          <title>Define datasets</title>
16          <description>PREPARE CLONING (ISC SUPPORT INCLUDED)
17 PROVIDE SCENARIO SPECIFIC DATA SETS
18 PROVIDE CUSTOMIZATION MEMBER HSC#VARS </description>
19          <instructions substitution="false">Generated instruction text for step: PrepareSource
20 Update this field with your own text</instructions>
21          <weight>5</weight>
22          <autoEnable>false</autoEnable>
23          <canMarkAsFailed>false</canMarkAsFailed>
24          <rest>
25             <httpMethod>POST</httpMethod>
26             <uriPath substitution="false">https://s0w1.fritz.box:7443/api/v1/jobs curl -X POST </uriPath>
27             <requestBody substitution="false">{"file":"HEINRIC.JOB.CNTL(IEFBR14)"}</requestBody>
28             <expectedStatusCode>201</expectedStatusCode>
29          </rest>
30             </step>
31    </step>
```

Hands on usage based on a cloning example



- Goal: Run a batch job based Db2 system clonig process out of Zowe

```
Continuous Delivery Deployment Check ----- Scenario Control Menu -------------
Command ===>
MENU=ON SCENARIO=CDCAFDA3 SOURCE=N/A TARGET=N/A OFFLINE COPY - TEST MODE
INTEGRATED FRENAME - NEW TARGET - INTERSYSTEM SCENARIO (SOURCE BASED)

Execute options  1 through 45 in sequence by pressing ENTER.
Enter M to switch between menu display ON or OFF.

Press ENTER to proceed with Select DB2

   ===>   1. Select DB2              - Select source
          2. Select DB2              - Select target
          3. Prepare                 - Define datasets
          4. Build samples           - Generate sample input for new DB2s
          5. Set environment         - General cloning options and sources
          6. Validate variables      - Check customer variables
          7. Validate datasets       - Check installation specific datasets
          8. Gather information      - Get all needed information
          9. WLX Variables           - Source: Get WLX Variables
         10. WLX STC                 - Source: Start WLX STC

MA   A                                                    02/015
```

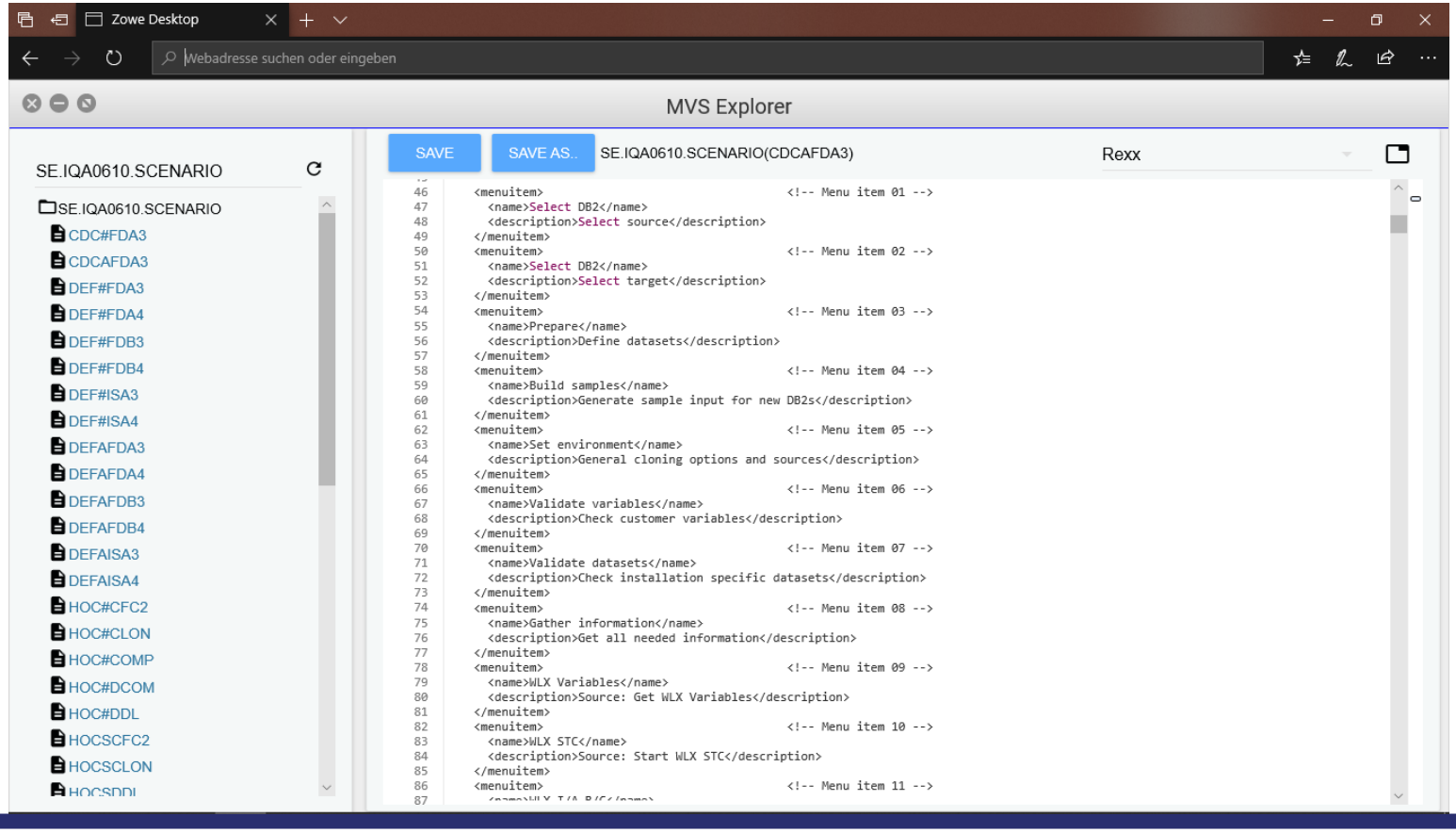
Hands on usage based on a cloning example

- Goal: Run a batch job based Db2 system clonig process out of Zowe

```
Continuous Delivery Deployment Check ----- Scenario Control Menu -------------
Command ===>
MENU=ON SCENARIO=CDCAFDA3 SOURCE=N/A TARGET=N/A OFFLINE COPY - TEST MODE
INTEGRATED FRENAME - NEW TARGET - INTERSYSTEM SCENARIO (SOURCE BASED)

Execute options  1 through 45 in sequence by pressing ENTER.
Enter M to switch between menu display ON or OFF.

Press ENTER to proceed with Select DB2

   ===>   1. Select DB2              - Select source
          2. Select DB2              - Select target
          3. Prepare                 - Define datasets
          4. Build samples           - Generate sample input for new DB2s
          5. Set environment         - General cloning options and sources
          6. Validate variables      - Check customer variables
          7. Validate datasets       - Check installation specific datasets
          8. Gather information      - Get all needed information
          9. WLX Variables           - Source: Get WLX Variables
         10. WLX STC                 - Source: Start WLX STC
MA      A                                                          2 /15
```

Hands on usage based on a cloning example

- The flow of batch jobs is driven by a XML scenario:



Hands on usage based on a cloning example

- The flow of batch jobs is migrated to a Workflow:

Zowe Desktop

Webadresse suchen oder eingeben

Editor

File Language Server Language Help

File Explorer

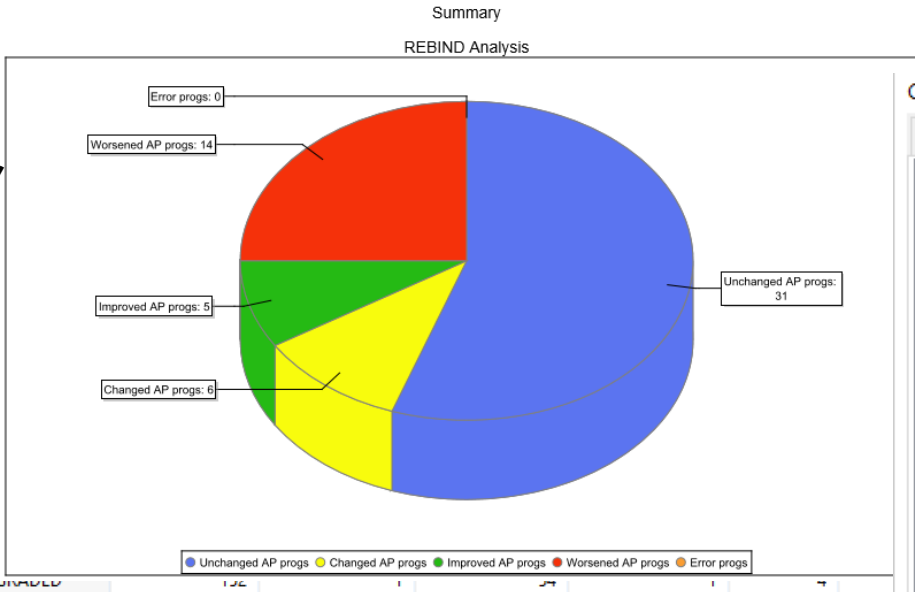
- cfzsrv
- debug
- work
- aopstc
- P390A
- log
- heinric
  - .sh_history
  - .profile
  - .ssh
  - tmp.xml
  - mywf.xml
- dudek
- bin
- usr
- lib
- opt
- samples
- $VERSION
- $SYSNAME
- dev
- tmp

mywf.xml

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<workflow>
  <autoTakeOwnership>false</autoTakeOwnership>
  <workflowInfo>
    <workflowID scope="none">MyWF</workflowID>
    <workflowDescription>ContinuousDelivery DeploymentCheck for Db2 z/OS</workflowDescription>
    <workflowVersion>1.0</workflowVersion>
    <vendor>SOFTWARE ENGINEERING</vendor>
    <General/>
  </workflowInfo>
  <step name="Prepare" optional="false">
    <title>Prepare Systems for Cloning</title>
    <description/>
    <step name="CDCAFDA3" optional="false">
      <title>Define datasets</title>
      <description>PREPARE CLONING (ISC SUPPORT INCLUDED)
PROVIDE SCENARIO SPECIFIC DATA SETS
PROVIDE CUSTOMIZATION MEMBER HSC#VARS </description>
      <instructions substitution="false">Generated instruction text for step: PrepareSource
Update this field with your own text</instructions>
      <weight>5</weight>
      <autoEnable>false</autoEnable>
      <canMarkAsFailed>false</canMarkAsFailed>
      <rest>
        <httpMethod>POST</httpMethod>
        <uriPath substitution="false">https://s0w1.fritz.box:7443/api/v1/jobs curl -X POST </uriPath>
        <requestBody substitution="false">{"file":"HEINRIC.JOB.CNTL(IEFBR14)"}</requestBody>
        <expectedStatusCode>201</expectedStatusCode>
      </rest>
        </step>
  </step>
```

Hands on usage based on a cloning example

Access Path Check – Static & Dynamic SQL Access Path Pre- and/or Post-Check:



STMT No.	Section Number	DSC STMT ID	Im					
2547	13	2547	DE					
796	2	796	DE					
804	3	804	DE					
812	4	812	DE					
820	5	820	DE					
671	2	671	DE					
679	3	679	DE					
687	4	687	DE					
695	5	695	DEGRADED	152	1	54	1	4
3	3	3	DEGRADED	130	1	46	1	6
4	4	4	DEGRADED	95	2	34	1	4
5	5	5	DEGRADED	82	1	29	1	4
6	6	6	DEGRADED	149	1	53	1	4
968	7	968	IMPROVED	171	16	60	6	8
1167	9	1167	IMPROVED	174	10	62	4	10
235	1	235	IMPROVED	178	10	63	4	8
1194	2	1194	IMPROVED	150	191	53	67	12
2409	7	2409	IMPROVED	21	23	8	9	5
1194	2	1194	IMPROVED	175	191	62	67	12

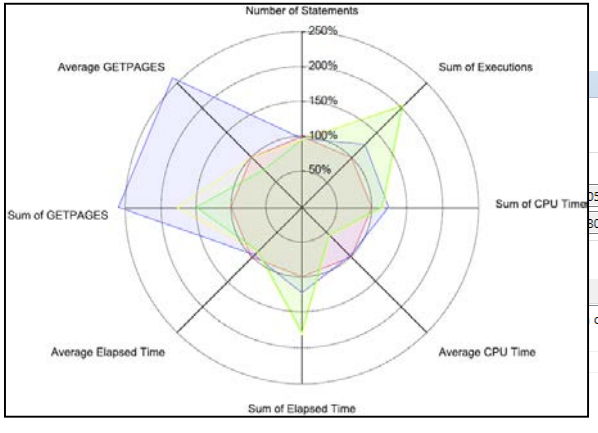
CDDC results summary

WLX | BIX

		All	Invalid	Inoperative
Packages	Analyzed	197	0	0
	Not analyzed	683	22	0
	Improved	16	0	0
	Worsened	19	0	0
	Changed	22	0	0
	Unchanged	140	0	0
Statements static	Analyzed	1730	0	0
	Not analyzed	2810	0	0
	Improved	214	0	0
	Worsened	54	0	0
	Changed	72	0	0
	Unchanged	1390	0	0
Statements dynamic	Analyzed	296	0	0
	Not analyzed	52	0	0
	Improved	3	0	0
	Worsened	34	0	0
	Changed	15	0	0
	Unchanged	244	0	0

Hands on usage based on a cloning example

- Drill down to look into details, when anomalies are detected



	Function level	Catalog level
05	V12R1M500	V12R1M503
30	V12R1M503	V12R1M503

of Number of Statements	Sum of GETPAGES	Sum of Synchronous Buffer Reads	Sum of Rows examined	Sum of Rows processed	Sum of Sorts performed	Sum of Index scans
861	86540	2926	832096	1935	136	165754
874	86713	2909	832408	1951	136	165758

Sum of WF and Tablespace Scans	Sum of Parallel Groups	Sum of Synchronous Buffer Writes	Sum of Elapsed Time	Sum of Wait Latch Request	Sum of Wait Page Latch	Sum of Wait Drain Lock	Sum of Wait Drain Claims	Sum of Wait Log Writer
282	0	79	267413647	221714	150707	9309254	0	1681516
279	0	79	265360495	362544	19190	7138740	0	1636997

Sum of Wait Synchronous IO	Sum of Wait Lock requests	Sum of Wait Synchronous Execution	Sum of Wait Global Locks	Sum of Wait other Thread Read	Sum of Wait other Thread Write	Sum of No RID Limits	Sum of No RID Storage	Sum of no RID WF Storage	Sum of no RID WF Limits
25248113	4249320	0	7867344	6456613	883679	0	0	0	0
21286897	3750693	0	5892333	6395228	586230	0	0	0	0

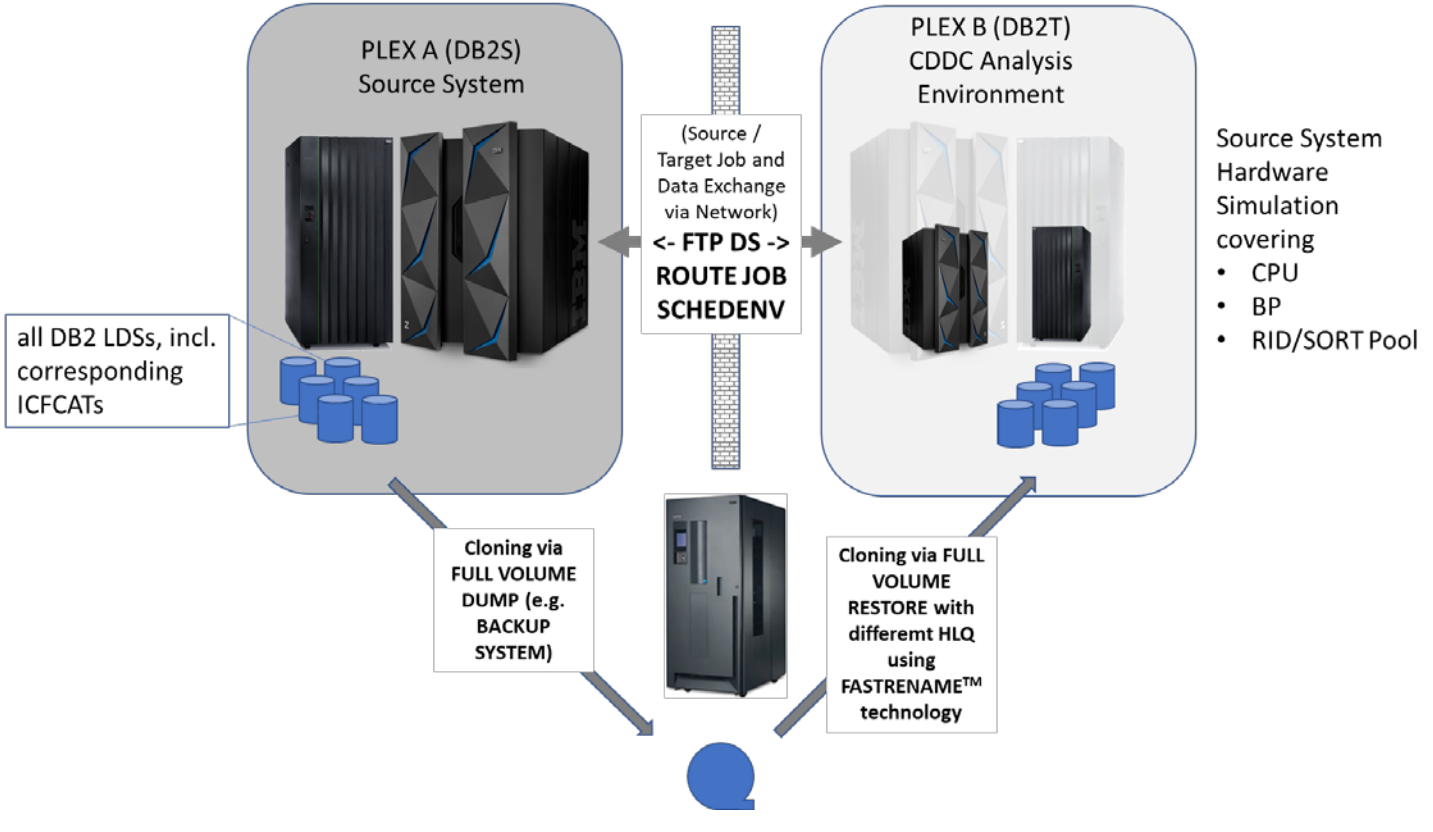
Hands on usage based on a cloning example

- Due to the nature of Zowe anything can be combined with everything, e.g.
  - Console, Shell, Db2 COMMANDs
  - JOBs
  - REXXs
  - Instructions
  - …
- and any information can be accessed:
  - Any type of MVS/USS dat sets
  - Job output
  - …

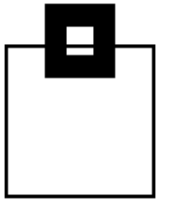
→ This makes the Zowe desktop your single point of control

Hands on usage based on a cloning example

Instant Cloning - Clone based code level checks:



Hands on usage based on a cloning example



- Zowe is perfect for ContinuousDelivery DeploymentCheck for Db2 z/OS:
  - We automatically clone a source Db2 into a target Db2
  - We can apply changes into the target Db2
  - We can replay workload, captured from source
  - We can do before and after comparisons within our clone
  - We can spot differences due to
    - BIF/ICI
    - Application changes
    - Access path changes
  - And we can display the results nicely in a HTML5 GUI

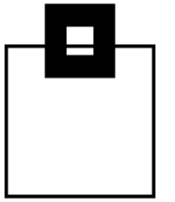
→ The entire process can be fully automated, but customized as needed

Summary of experience

- Starting with Zowe can be challenging, depending on your accessible resources/knowledge:
  - MVS
  - Unix
  - Security
    - Authorization
    - Certificates
  - Tomcat
  - zOSMF
  - …

  , but …

Summary of experience



- It‘s worth it!!!

- We started with quite early (< 1.0) versions, but 1.0.1 was released on the 7th of March
  - → It starts to become solid and certainly ready to look at it

- Use any of your z/OS capabilties as a cloud service
- Make your z/OS system accessible for non ISPFers
- Modernize z/OS applications
- Attract the youngsters to exploit the strength of the z platform

- SEGUS is committed to exploit Zowe with their existing and upcoming tools and to contribute to the new ecosystem.