OPEN
MAINFRAME
PROJECT

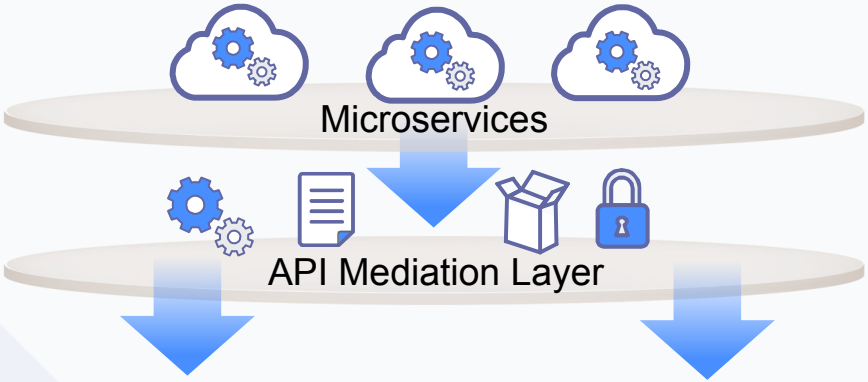
Zowe Overview

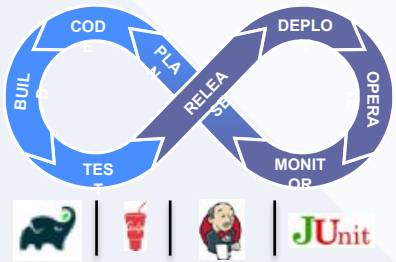
Introducing Zowe

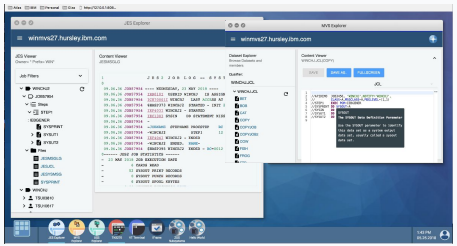
- An extensible framework for connecting applications and tools to mainframe data and applications.
- Aims to make the mainframe an integrated and agile platform within the changing IT architectural landscape.
- First open source project on z/OS. All code is licensed under the Eclipse Public License version 2.0

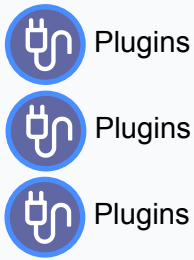
Zowe

Zowe framework and ecosystem at a glance









Ecosystem enablement thru Zowe Conformance

- Vendors are able to build applications that leverage or build on top of the Zowe Framework
- Building on this framework speeds up vendors time to market and enables easier integration with other Zowe Conformant applications
- Complimentary participation for any Open Mainframe Project member organization
- More details at https://www.openmainframeproject.org/projects/zowe/conformance

Getting started with Zowe

Sign up to the Zowe email lists

User List:
https://lists.openmainframeproject.org/g/zowe-user/
Developer List:
https://lists.openmainframeproject.org/g/zowe-dev/

Find other users and developers on Slack

https://slack.openmainframeproject.org
Channels: #zowe-user and #zowe-dev

View the code and contribute

https://github.com/zowe
Contribution Guidelines at
https://github.com/zowe/community#contribute

Zowe is part of the Open Mainframe Project hosted projects ecosystem

Learn about all Open Mainframe projects at https://www.openmainframeproject.org/projects

Sustainability in open source on mainframe happens thanks to Open Mainframe Project member support


Zoss Team, LLC

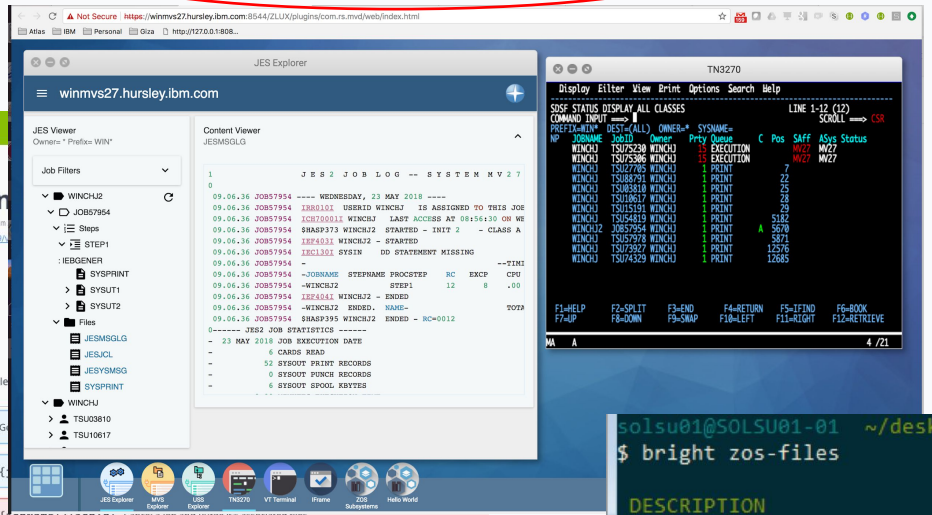
Learn more about the Open Mainframe Project

- Find out more and subscribe to our newsletter at www.openmainframeproject.org
- Check out our projects at https://www.openmainframeproject.org/projects
- See the full landscape of open source on the mainframe at https://l.openmainframeproject.org
- Organizational membership opportunities at https://www.openmainframeproject.org/about/join or email at membership@openmainframeproject.com

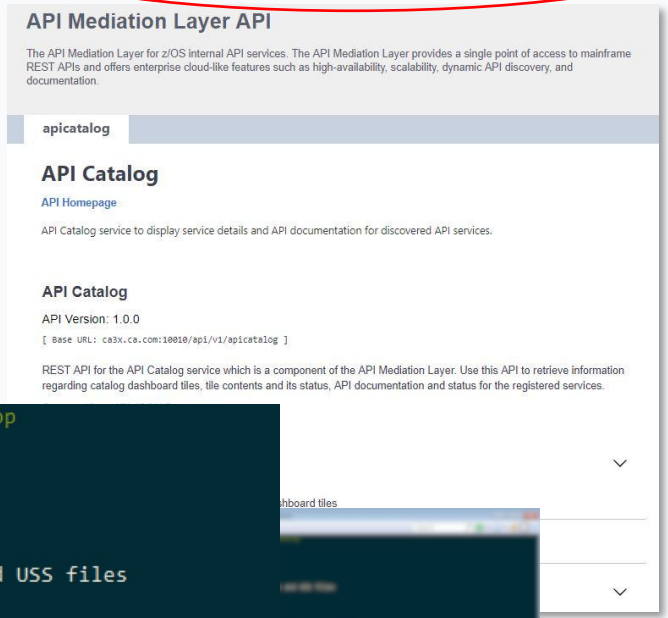
Extra slides

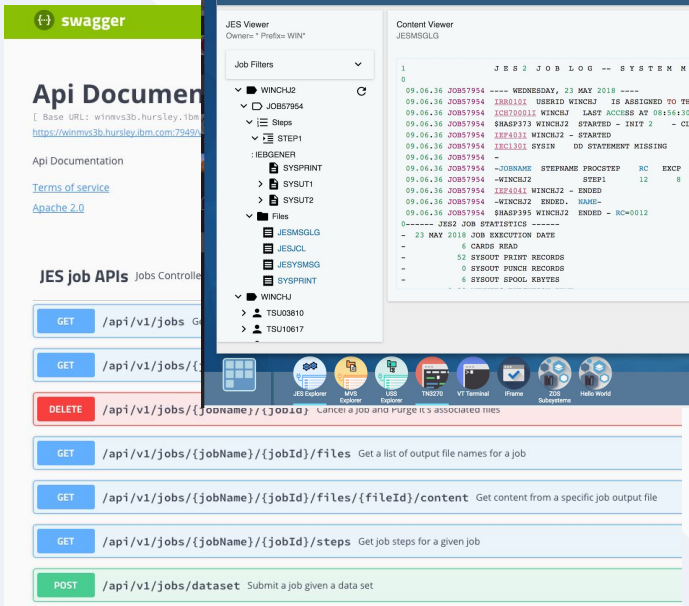
What's in Zowe?

Browser-based Web Desktop

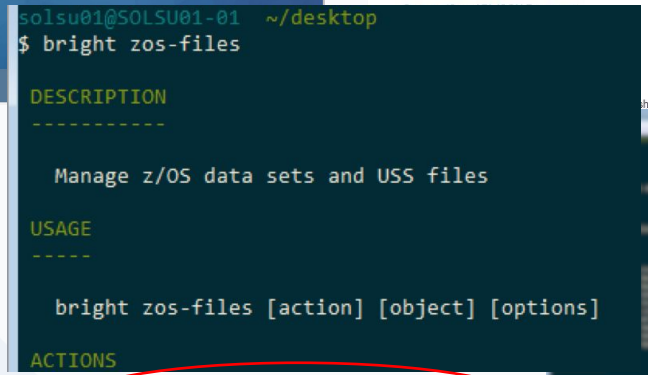


API Mediation Layer
(Gateway, Discovery Service, Catalog)





Swagger-defined z/OS REST APIs

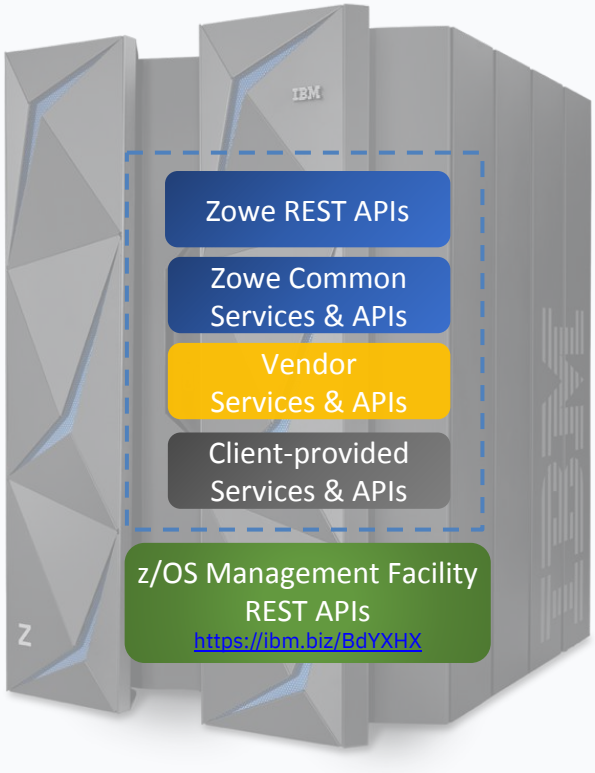


Node.js- based CLI

Zowe REST Services – API economy for deep integration

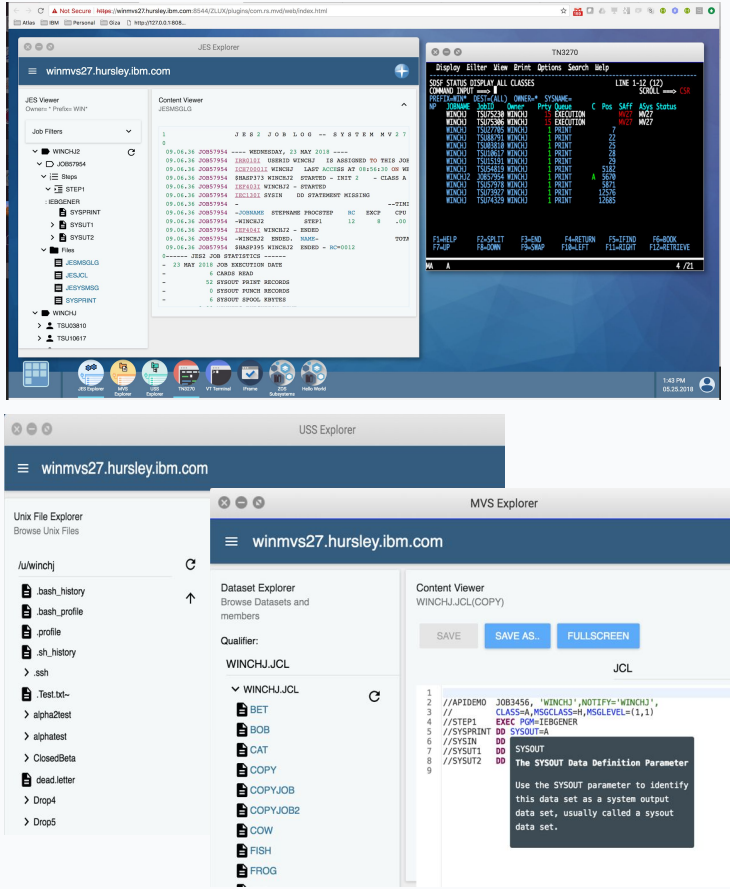
- Industry standard REST interfaces to z/OS resources that are language and platform neutral, stateless and scalable
- Foundational building blocks for system services

  - ***Dataset APIs***
    - Create, read, update, delete, and list data sets
  - ***JES APIs***
    - View the information and files of jobs, and submit and cancel job
  - ***USS APIs***
    - Create, read, update, and delete USS files
  - ***System APIs***
    - View information about PARMLIB, SYSPLEX, and USER



Zowe Web Desktop – An app container in a browser

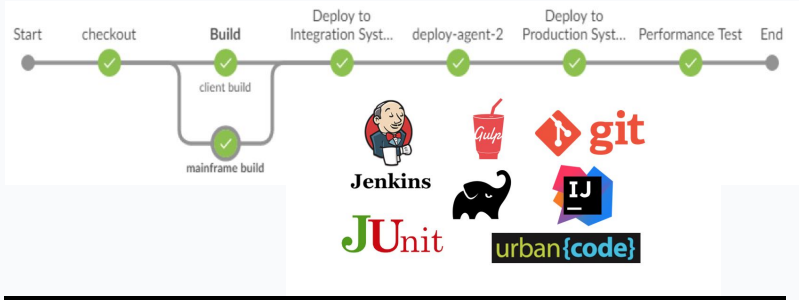
- Known as **zLUX**, the Zowe Web UI is a virtual desktop system that offers a rich and open platform for a web-based mainframe user experience.
  - ***Mainframe Virtual Desktop***
    - A web-based window manager that provides full screen interactive experience
  - ***Zowe Node Server***
    - Runs zLUX; uses Express.js as web service framework for communication between applications and z/OS services and components, pre-reqs Node.js for z/OS
  - ***ZSS Server***
    - Provides secured REST API services
  - ***Application plug-in***
    - Dataservices, Configuration dataservice, URI broker, app-to-app communication, Error reporting UI, Logging utility
  - ***Explorers***
    - JES, MVS, USS explorers
    - Basic editing support for REXX and JCL



Zowe CLI – Enables cloud-like access to mainframe

- Enables app developer and DevOps engineers to interact with the mainframe easily through a CLI from any terminal on Windows, MacOS, Linux
- Easily integrates with IDEs, shell commands, bash scripts, and build tools; installs using NPM

Build | Test | Deploy



- ***Interact with mainframe files***
  Create, edit, download, and upload mainframe files (data sets) directly
- ***Submit jobs***
  Submit JCL from data sets or local storage, monitor status, view and download output automatically
- ***Issue TSO and z/OS console commands***
  Issue TSO and console commands to the mainframe directly
- ***Integrate z/OS actions into scripts***
  Build local scripts that accomplish both mainframe and local tasks
- ***Produce responses as JSON documents***
  Return data in JSON format on request for consumption in other programming languages
- ***CLI Plug-Ins***
  Access to CICS and DB2

```
GROUPS
------

  plugins                Install and manage plug-ins
  profiles               Create and manage configuration profiles
  provisioning | pv      Perform z/OSMF provisioning tasks on Published Templates
                         in the Service Catalog and Provisioned Instances in the
                         Service Registry.
  zos-console | console  Issue z/OS console commands and collect responses
  zos-files | files      Manage z/OS data sets
  zos-jobs | jobs        Manage z/OS jobs
  zos-tso | tso          Issue TSO commands and interact with TSO address spaces
  zosmf                  Interact with z/OSMF

OPTIONS
-------

  --version  | -V (boolean)

     Display the current version of CA Brightside

GLOBAL OPTIONS
--------------

  --response-format-json  | --rfj (boolean)

     Produce the command response as a JSON document

  --help  | -h (boolean)
```

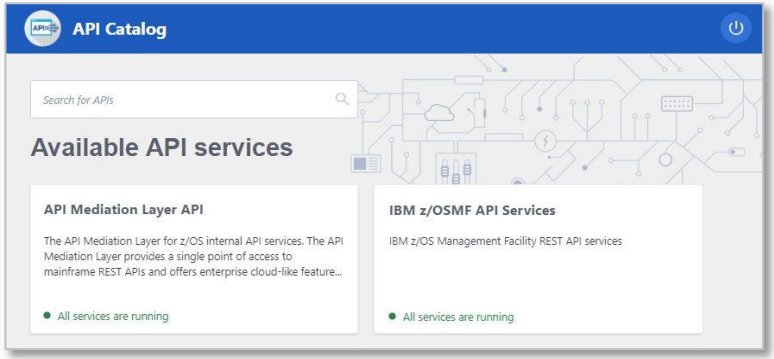
Zowe API Mediation Layer – Gateway to mainframe APIs

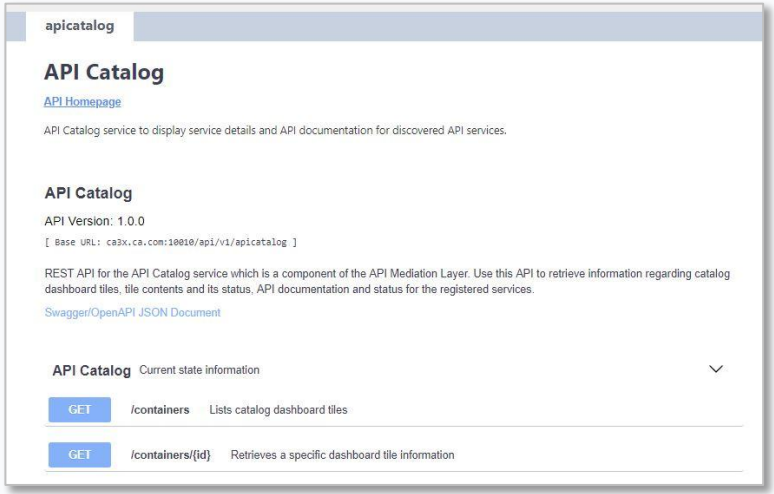
- Enables a single point of access to mainframe APIs with high-availability, scalability, dynamic API discovery, consistent security, “one-time” sign-on experience and unified standard API documentation (OpenAPI / Swagger)

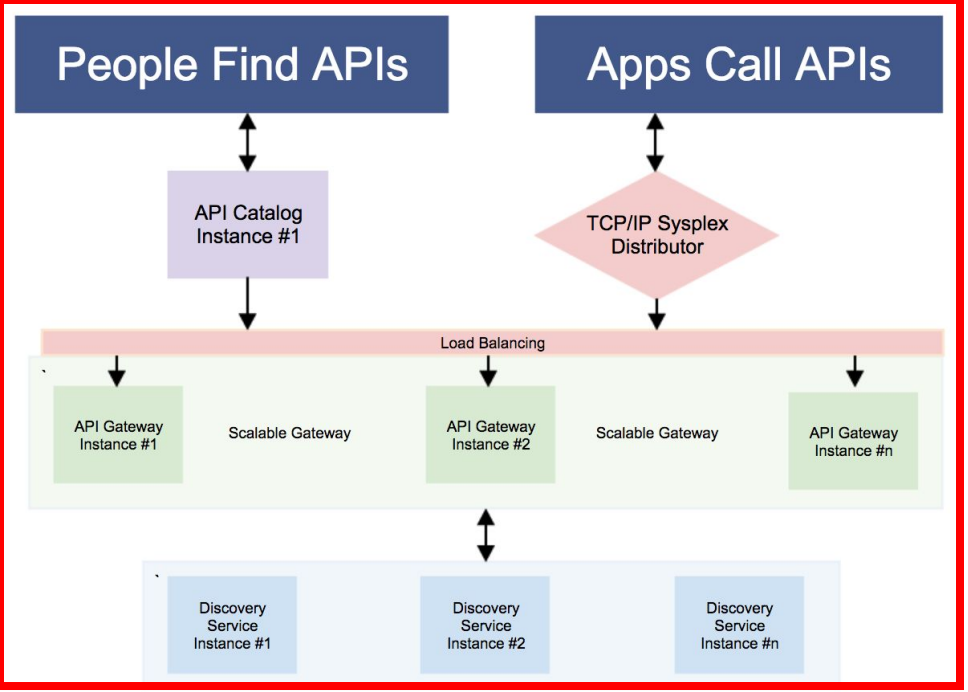
  - ***API Catalog***
    UI Catalog of available APIs with their Swagger doc and service status

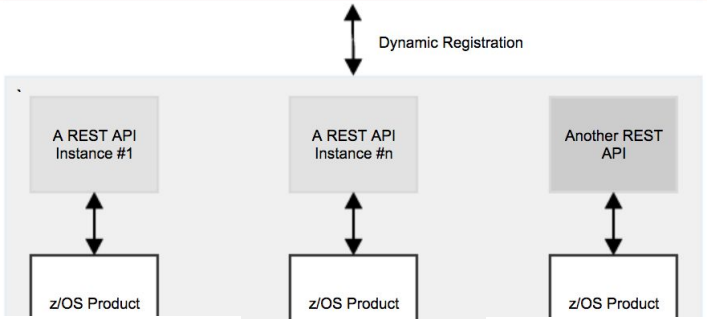
  - ***Gateway***
    Single secure point of entry to an ecosystem of API services. Hides complexity. Highly available. Based on Netflix Zuul.

  - ***Discovery Service***
    Discover APIs across many applications. Repository of active API services. Based on Netflix Eureka.









z/OS Connect z/OSMF Jax-RS App

API Layer Components*

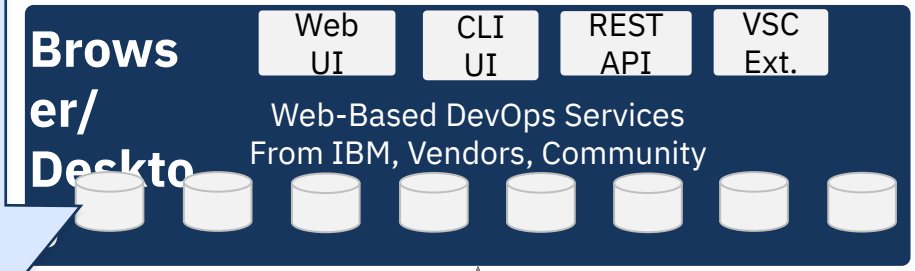
- **API Catalog**
  UI Catalog of available APIs with their Swagger doc and service status
- **API Gateway**
  Single point of entry to an ecosystem of microservices. Hides complexity. Highly available. Based on Netflix Zuul.
- **Discovery Service**
  Discover APIs across many applications. Repository of active services. Based on Netflix Eureka.
- **z/OSMF API**
  Authenticate Zowe users with mainframe credentials

** Separate microservices, might be running as separate address spaces*

Zowe High Level Architecture

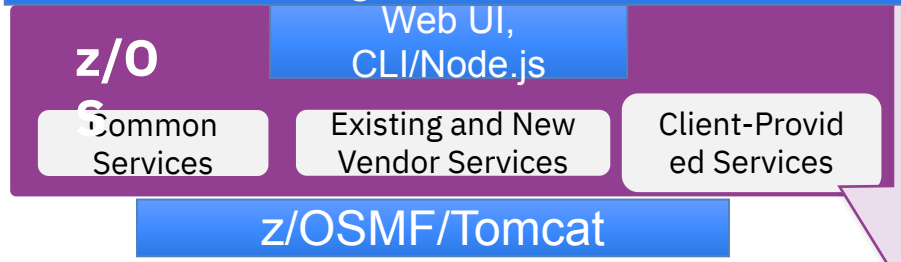
Base Components

- Editor support (REXX/JCL to start)
- CLI
- APIs
- Virtual Desktop – App Container
- VS Code Extension



RESTful

Catalog of RESTful API

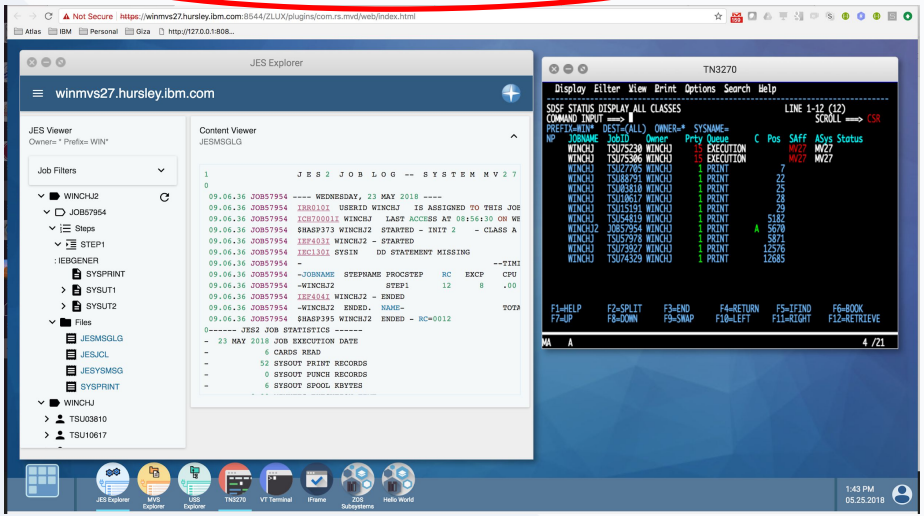


Sample Vendor / Open Source Integration

- ServiceNow
- JIRA
- Jenkins
- Git
- SonarLint

Where is Zowe Extensible?

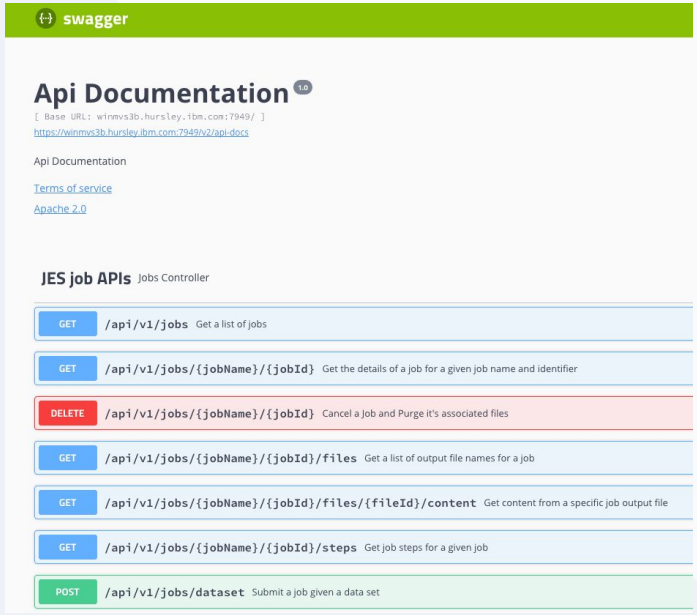
Browser-based Web Desktop



- z/OS Native Web UI for applications
- Launch in context (i.e., right mouse click 3270 to web app)
- App to app communication
- Exploit graphic widgets planned for inclusion

Where is Zowe Extensible?

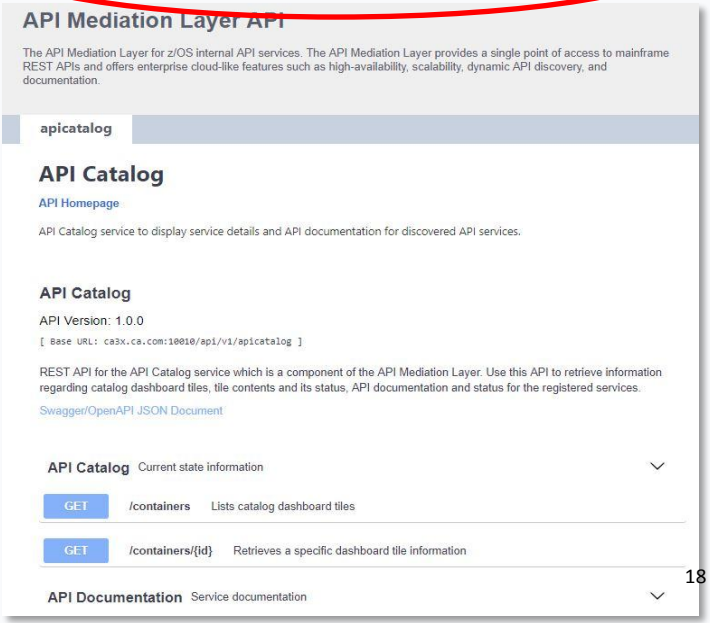
- REST API enable your products
  - REST API for product controls/admin
  - Sharing of information



Swagger-defined z/OS REST APIs

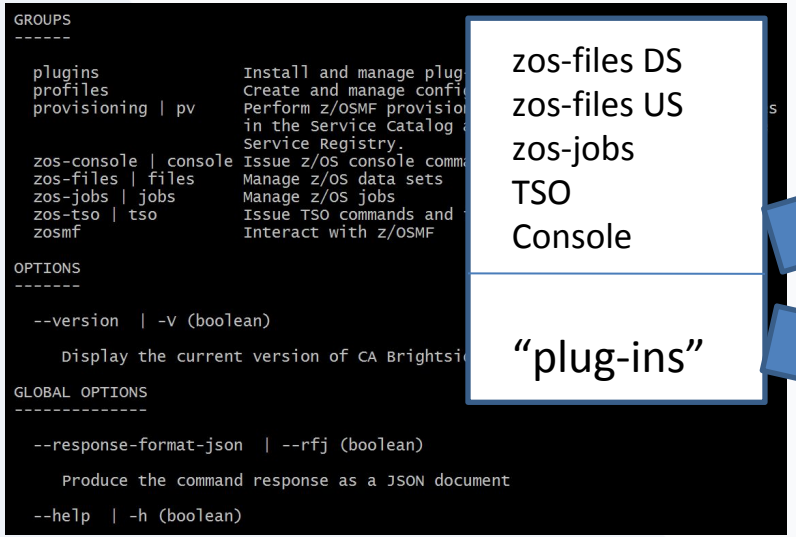
- Opt in to API Mediation
- Participate in Single Sign On, High Availability and Status tracking capabilities

API Mediation Layer
(API Catalog, Discovery Service, Gateway)



Where is Zowe Extensible?

Node.js- based CLI



zos-files DS
zos-files US
zos-jobs
TSO
Console

“plug-ins”

Out of box commands

Custom Extensions

z/OSMF

REST APIs
TSO, Console
JES, MVS. USS

Your application, product, tool, …

Zowe “Demo”

Intro Demo:
https://www.youtube.com/watch?v=NX20ZMRoTtk&feature=share

Visual Studio Code (using command line):
https://www.youtube.com/watch?v=la1_Ss27fn8