IBM z/OS Connect

API Requester Introduction and Overview

Mitch Johnson
mitchj@us.ibm.com
Washington System Center

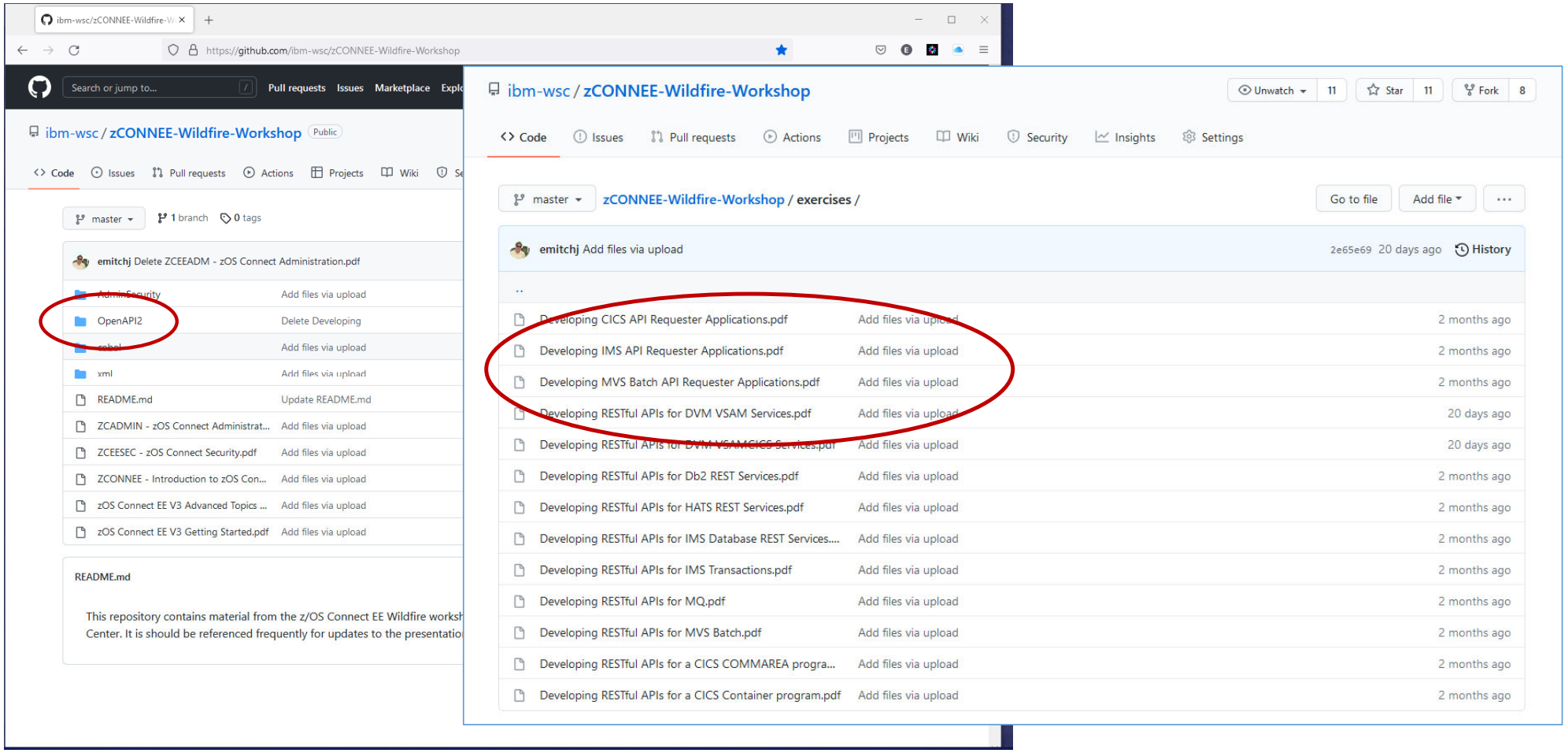
IBM Z
Wildfire Team –
Washington System Center

Notes and Disclaimers

- The information in this presentation was derived from various product documentation web sites.
- Additional information included in this presentation was distilled from years of experience implementing security using RACF with z/OS products like CICS, IMS, Db2, MQ, etc. as well as Java runtimes environments like WebSphere Application Server and WebSphere Application Server Liberty which is commonly called Liberty.
- There will be additional information on slides that will designated as Tech/Tips. These contain information that at perhaps at least interesting and hopefully, useful to the reader.
- **IBM z/OS Connect (OpenAPI 2)** refers the z/OS Connect EE product prior to service level V3.0.55. **IBM z/OS Connect (OpenAPI 3)** refers to the additional functions and features added with service level V3.0.55. Important - servers configured for OpenAPI 2 can will continue to operate as is with service level V3.0.55 and later.
- A z/OS Connect OpenAPI 2, or a z/OS Connect OpenAPI 3 icon will appear on slides where the information is specific to these products. Don't hesitate to ask questions as to why the icon does or does not appear on certain slides.
- The examples, tips, etc. present in this material are based on firsthand experiences and are not necessarily sanctioned by Liberty or z/OS Connect development.

mitchj@us.ibm.com

z/OS Connect Wildfire Github Site https://ibm.biz/BdPRGD



mitchj@us.ibm.com

- **Contact your IBM representative to schedule access to these exercises**

z/OS Connect EE exposes z/OS resources to the “cloud” via RESTful APIs

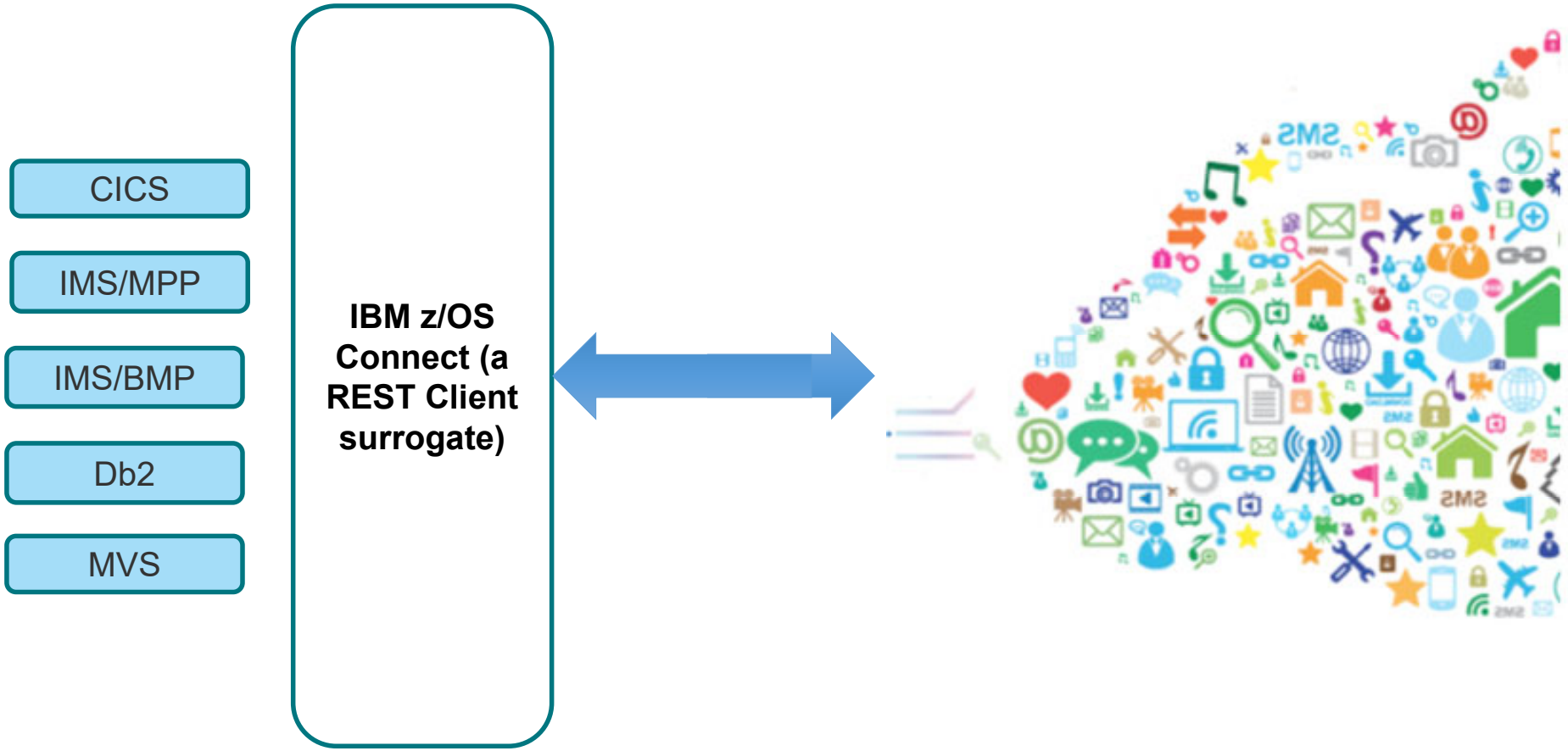
IBM z/OS Connect (an API Gateway)

- CICS
- IMS/TM
- IMS/DB
- Db2
- MQ
- IBM File Manager+
- HATS(3270)
- IBM DVM+
- MVS
- WAS
- Custom*

+ HCL and Rocket Software
*Other Vendors or your own implementation

mitchj@us.ibm.com

z/OS Connect EE exposes external REST APIs in the “cloud” to z/OS applications



mitchj@us.ibm.com

/but_first, what_is_REST?

What makes an API “RESTful”?

mitchj@us.ibm.com

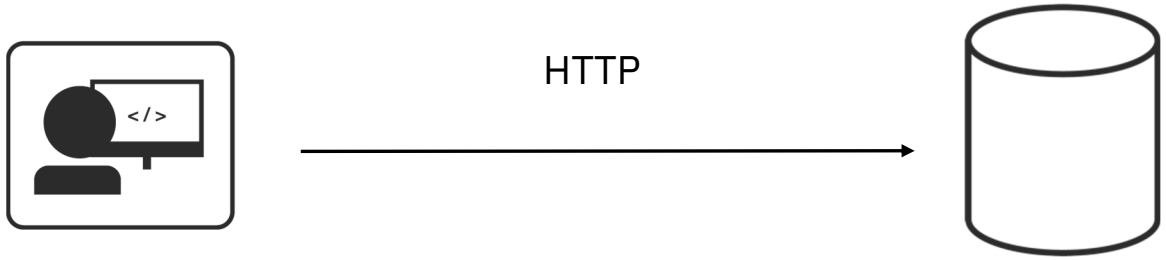
REST is architectural programming style

REST stands for **Re**presentational **S**tate **T**ransfer.

An architectural programming style for **accessing** and **updating** data over the internet.

Typically using HTTP… but not all HTTP interfaces are “RESTful”.

Simple and intuitive for the end consumer (**the developer**).



Roy Fielding defined REST in his 2000 PhD dissertation "Architectural Styles and the Design of Network-based Software Architectures" at UC Irvine. He developed the REST architectural style in parallel with HTTP 1.1 of 1996-1999, based on the existing design of HTTP 1.0 of 1996.

mitchj@us.ibm.com

Key Principles of REST

Use Path and Query parameters to refine the request

Use HTTP verbs for Create, Read, Update, Delete (CRUD) operations

POST
GET
PUT
DELETE

```
http://<host>:<port>/path/parameter?name=value&name=value
```

URI path identifies a resource (or lists of resources)

URL identifies the protocol, host and port and includes the URI Path

Request/Response Body is used to represent the data object

```
GET http://www.acme.com/customers/12345?personalDetails=true
RESPONSE: HTTP 200 OK
BODY { “id” : 12345
       “name” : “Joe Bloggs”,
       “address” : “10 Old Street”,
       “tel” : “01234 123456”,
       “dateOfBirth” : “01/01/1980”,
       “maritalStatus” : “married”,
       “partner” : “http://www.acme.com/customers/12346” }
```

mitchj@us.ibm.com

REST vs RESTful

REST is an architectural style of development having these principles plus..

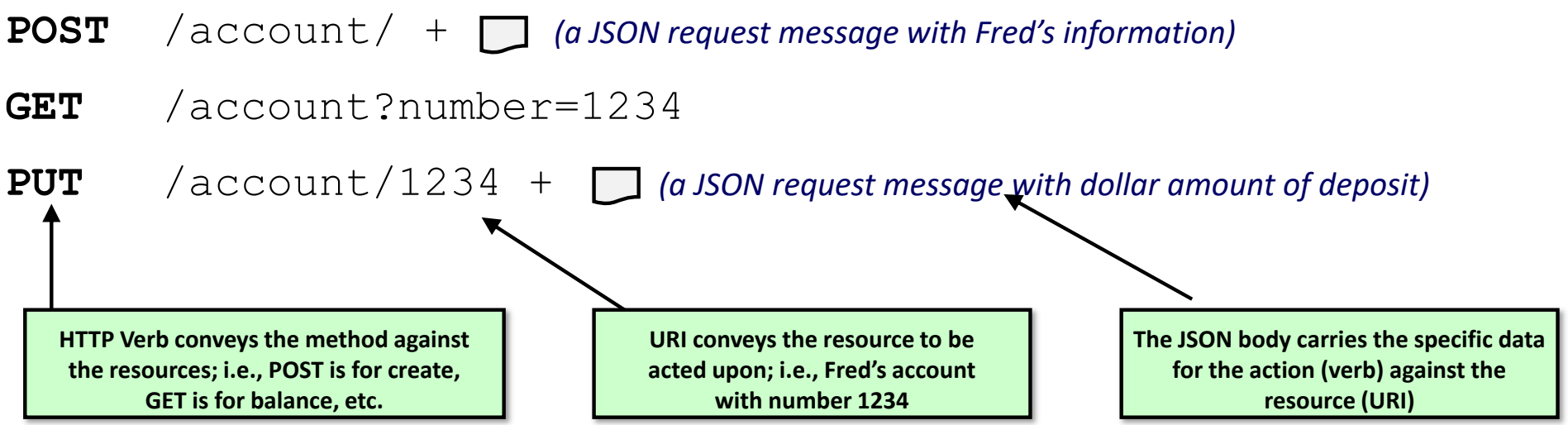
- It should be stateless (transaction management should be managed by the client)
- It should access and/or identify all server resources using only a single URI
- For performing CRUD operations, it should use HTTP verbs such as get, post, put and delete
- It should return the result only in the form of consistent and simple JSON

When an API follows these basic principles, it is considered a RESTful API, whereas a REST API only follows some but not all the above principles

- Remember - Not all REST APIs are RESTful APIs
- The key is consistency, RESTful APIs are consistent with these basic principles, REST APIs are not

mitchj@us.ibm.com

RESTful Examples



REST APIs are increasingly popular as an integration pattern because it is stateless, relatively lightweight, is relatively easy to program

https://martinfowler.com/articles/richardsonMaturityModel.html

mitchj@us.ibm.com

Not every REST API is a RESTful API

- (How to know if an API is not RESTful)

1. Different URIs with the same method for operations on the same object

```
POST http://www.acme.com/customers/GetCustomerDetails/12345
POST http://www.acme.com/customers/UpdateCustomerAddress/12345?address=
```

2. Different representations of the same objects between request and response messages

```
POST http://www.acme.com/customers
BODY { "firstName": "Joe",
       "lastName" : "Bloggs",
       "addr"     : "10 Old Street",
       "phoneNo"  : "01234 0123456" }
```



```
RESPONSE HTTP 201 CREATED
BODY { "id"      : "12345",
       "name"    : "Joe Bloggs",
       "address": "10 New Street"
       "tel"     : "01234 0123456"}
```

3. Operational data (update, etc.) embedded in the request body

```
POST http://www.acme.com/customers/12345
BODY { "updateField": "address",
       "newValue"   : "10 New Street"}
```



```
RESPONSE HTTP 200 OK
BODY { "id"      : "12345",
       "name"    : "Joe Bloggs",
       "address": "10 New Street"
       "tel"     : "01234 123456" }
```

mitchj@us.ibm.com

Strive to design API to use RESTful properties

1. Use the same URIs for the same resource with the appropriate method for operations

```
GET  http://www.acme.com/customers/12345
PUT  http://www.acme.com/customers/12345?address=10%20New%20Street
```

2. Use the same JSON property names between request and response messages

```
POST http://www.acme.com/customers/12345
BODY { "name": "Joe Bloggs",
       "address": "10 Old Street",
       "phoneNo": "01234 0123456" }
```



```
RESPONSE HTTP 201
BODY { "id"      : "12345",
       "name"    : "Joe Bloggs",
       "address": "10 New Street"
       "phoneNo": "01234 0123456"}
```

3. Use JSON name/value pairs

```
PUT http://www.acme.com/customers/12345
BODY { "address"    : "10 New Street"}
```



```
RESPONSE HTTP 200 OK
```

mitchj@us.ibm.com

Why is REST popular?

Ubiquitous Foundation	It's based on HTTP, which operates on TCP/IP, which is a ubiquitous networking topology.
Relatively Lightweight	Compared to other technologies (for example, SOAP/WSDL), the REST/JSON pattern is relatively light protocol and data model, which maps well to resource-limited devices.
Relatively Easy Development	Since the REST interface is so simple, developing the client involves very few things: an understanding of the URI requirements (path, parameters) and any JSON data schema.
Increasingly Common	REST/JSON is becoming more and more a de facto "standard" for exposing APIs and Microservices. As more adopt the integration pattern, the more others become interested.
Stateless	REST is by definition a stateless protocol, which implies greater simplicity in topology design. There's no need to maintain, replicate or route based on state.

mitchj@us.ibm.com

How do we describe a REST API?

mitchj@us.ibm.com

/oai/open_api_initiative

The industry standard framework for describing REST APIs

The OpenAPI Initiative (OAI) was created by a consortium of forward-looking industry experts who recognize the immense value of standardizing on how APIs are described. As an open governance structure under the Linux Foundation, the OAI is focused on creating, evolving and promoting a vendor neutral description format. The OpenAPI Specification was originally based on the Swagger Specification, donated by SmartBear Software.

mitchj@us.ibm.com

Why use OPENAPI?

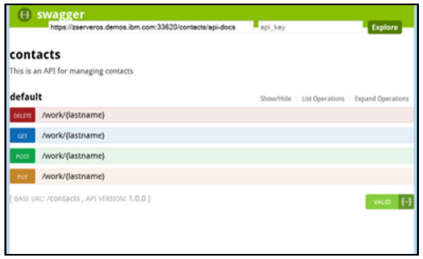
- It is more than just an API framework

OPEN API INITIATIVE

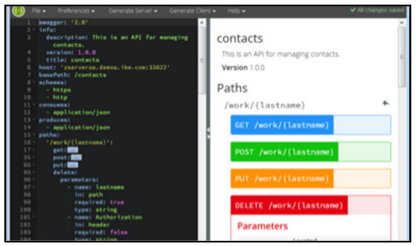
There are a number of tools available to aid consumption:

Code Generation* - create stub code to consume APIs from various languages

Test UIs - allows API consumers to easily browse and try APIs based on an OpenAPI document.



Editors - allows API developers to design their OpenAPI documents.



* z/OS Connect API Requester

+z/OS Connect, MQ REST support, Zowe

Important - You may have used or heard of the term Swagger with the use of APIs. As the use of APIs has grown this term has become in some respects misleading. To be more precise, OpenAPI refers to the API specifications (OpenAPI 2 and OpenAPI3) where Swagger refers to the tooling used to implement the specifcations.

mitchj@us.ibm.com

An OPENAPI 2 versus an OPENAPI 3 specification document

JSON format

cscvinc.json - Notepad

File Edit Format View Help

```
{
    "swagger": "2.0",
    "info": {
        "description": "",
        "version": "1.0.0",
        "title": "cscvincapi"
    },
    "basePath": "/cscvincapi",
    "schemes": [
        "https",
        "http"
    ],
    "consumes": [
        "application/json"
    ],
    "produces": [
        "application/json"
    ],
    "paths": {
        "/employee/{employee}": {
            "get": {
                "tags": [
                    "cscvincapi"
                ],
                "operationId": "getCscvincSelectService",
                "parameters": [
                    {
                        "name": "Authorization",
                        "in": "header",
                        "required": false,
                        "type": "string"
                    },
                    {
                        "name": "employee",
                        "in": "path",
                        "required": true,
                        "type": "string",
                        "maxLength": 6
                    }
                ],
                "responses": {
                    "200": {
                        "description": "OK",
                        "schema": {
                            "$ref": "#/definitions/getCscvincSelectService_response_200"
                        }
                    },
                    "404": {
                        "description": "Not Found",
```

Ln 18, Col 7 | 100% | Windows (CRLF) | UTF-8

cscvinc.yaml - Notepad

File Edit Format View Help

```
openapi: 3.0.1
info:
  title: cscvinc
  description: ""
  version: 1.0.0
servers:
- url: /cscvinc
x-ibm-zcon-roles-allowed:
- Manager
paths:
  /employee:
    post:
      tags:
      - cscvinc
      operationId: postCscvincInsertService
      x-ibm-zcon-roles-allowed:
        - Staff
      parameters:
      - name: Authorization
        in: header
        schema:
          type: string
      requestBody:
        description: request body
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/postCscvincInsertService_request'
        required: true
      responses:
        200:
          description: OK
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/postCscvincInsertService_response_200'
      x-codegen-request-body-name: postCscvincInsertService_request
  /employee/{employee}:
    get:
      tags:
      - cscvinc
      operationId: getCscvincSelectService
      x-ibm-zcon-roles-allowed:
        - Staff
      parameters:
      - name: Authorization
        in: header
        schema:
          type: string
```

Ln 44, Col 16 | 100% | Unix (LF) | UTF-8

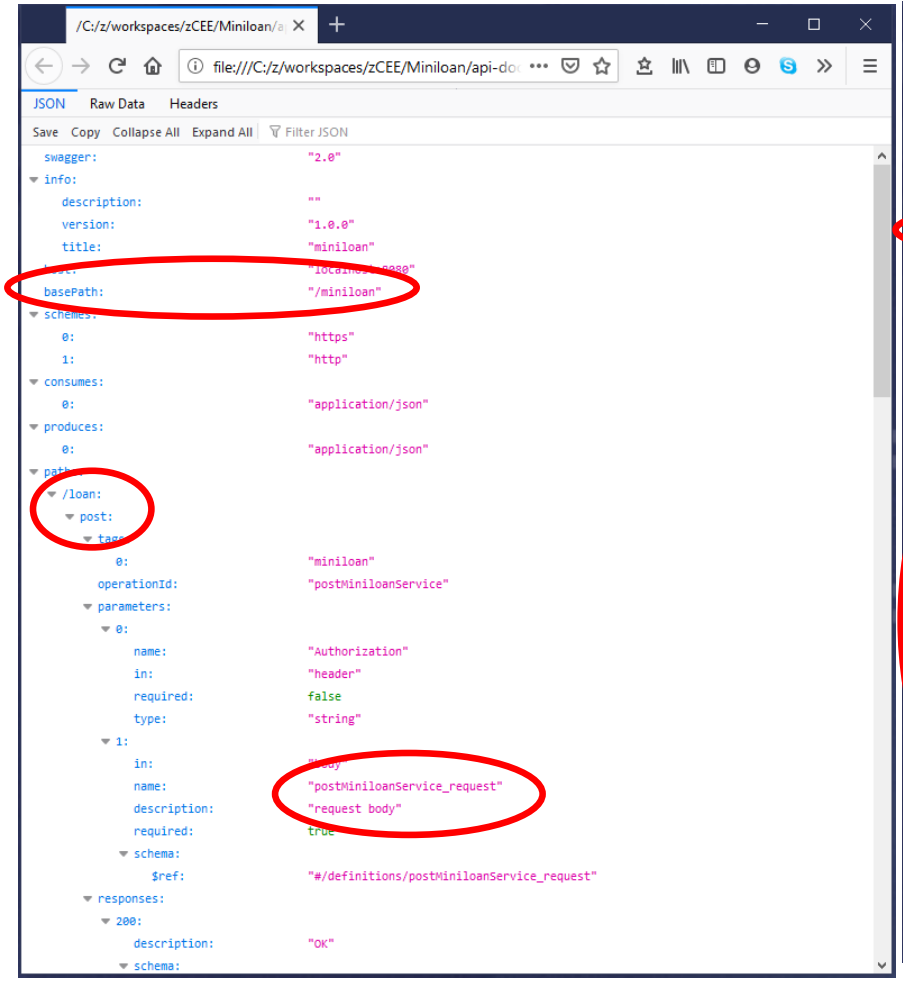
YAML Format*

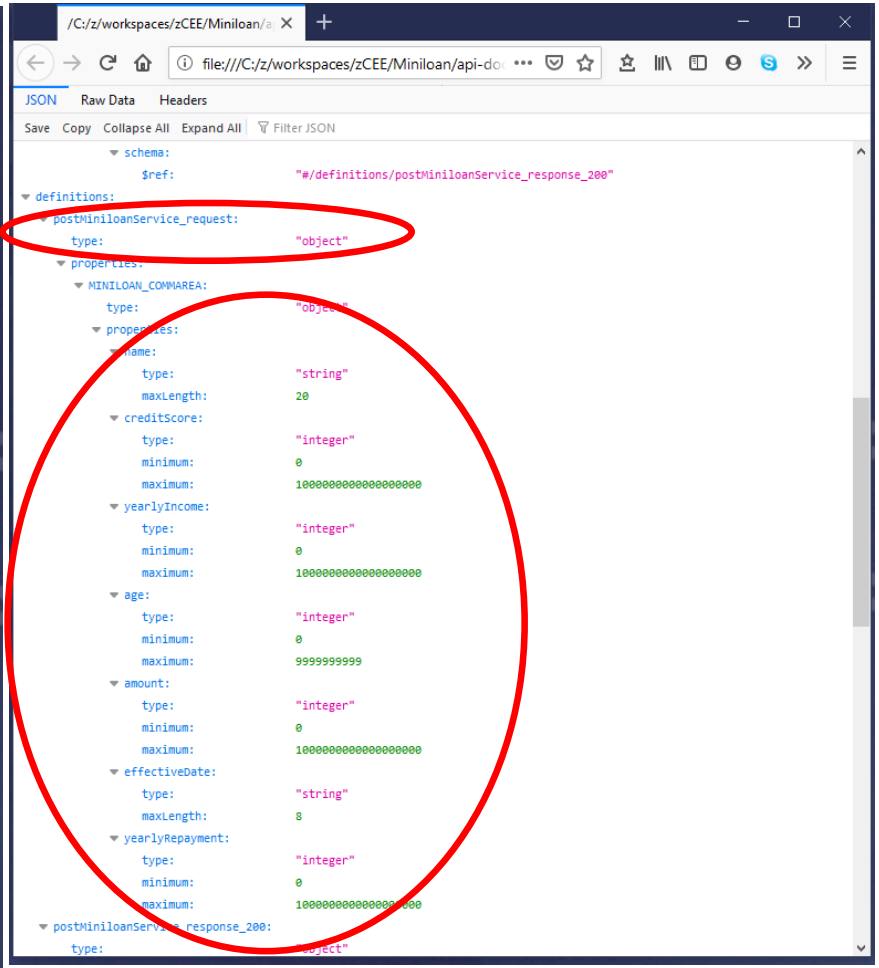
mitchj@us.ibm.com

https://swagger.io/blog/api-strategy/difference-between-swagger-and-openapi/

*Yet Another Markup Language

Tech-Tip: Swagger Example

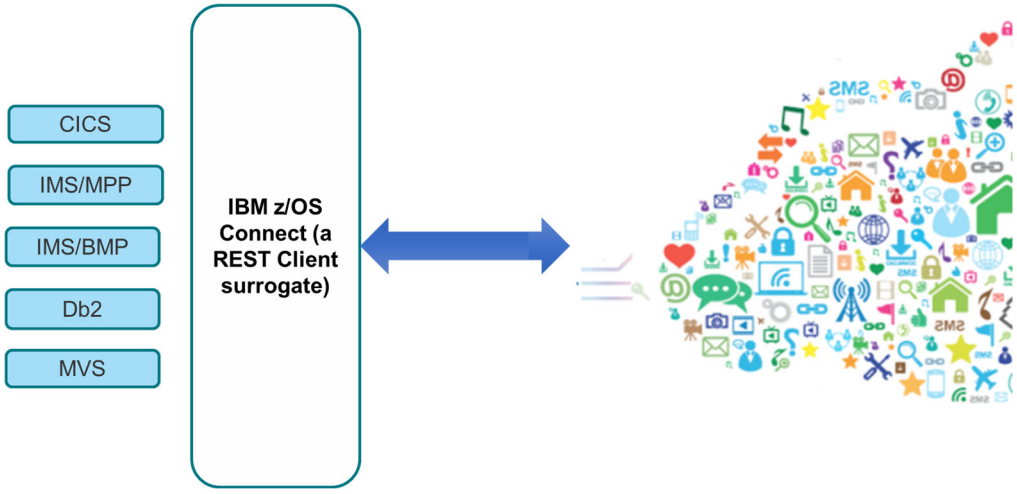




mitchj@us.ibm.com

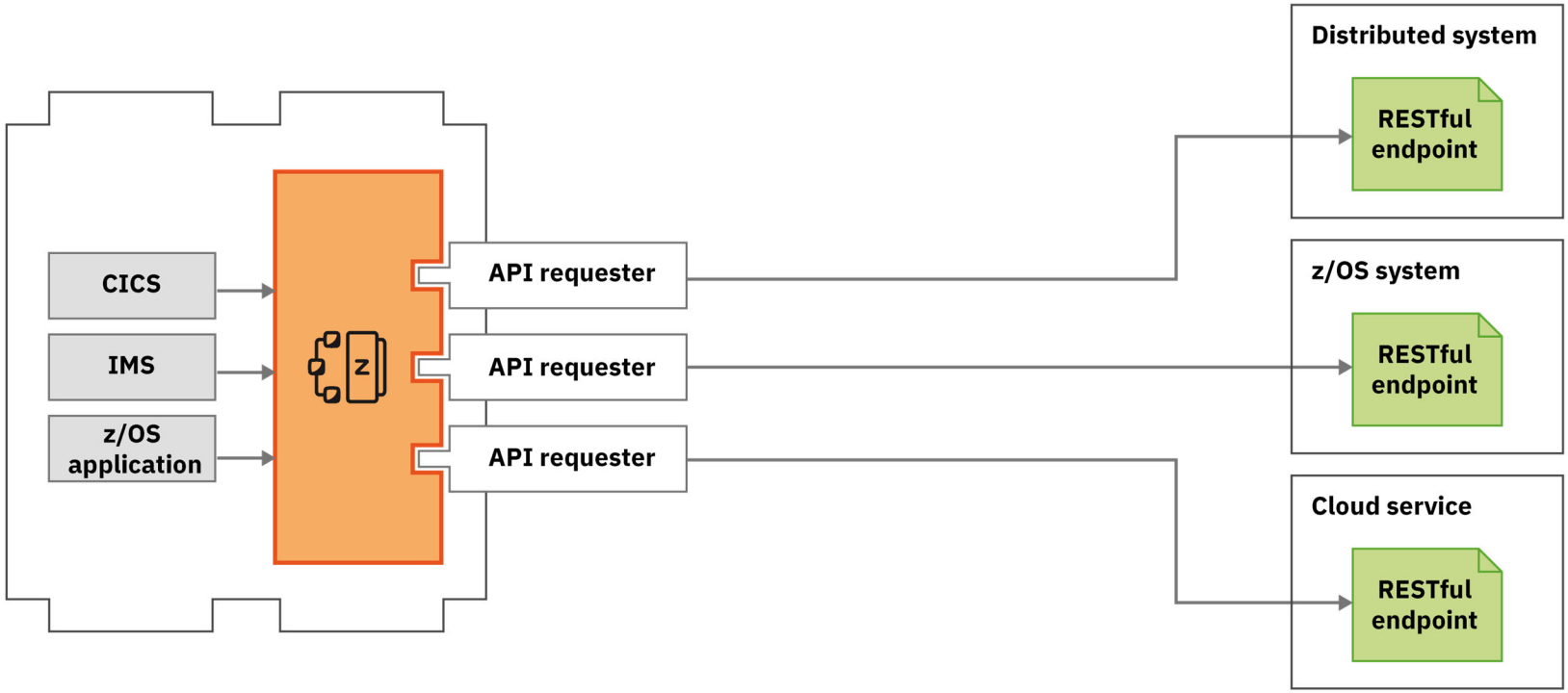
/api_toolkit/apiRequesters

Quick and easy **API mapping.**



mitchj@us.ibm.com

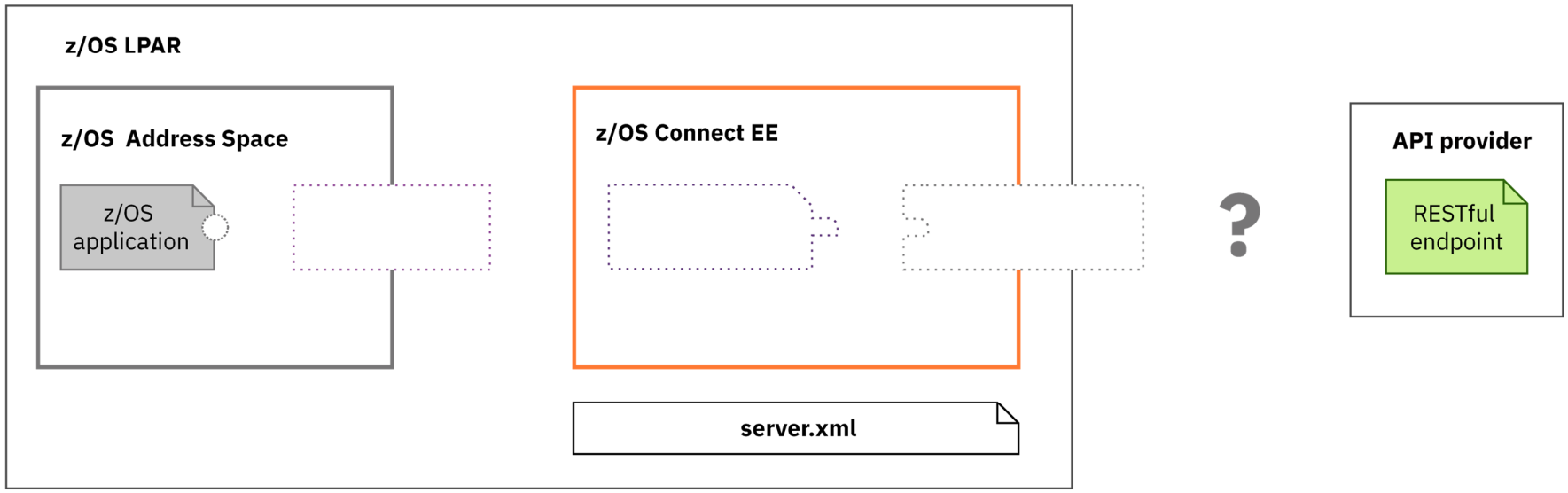
Use API requester to call external APIs from z/OS assets



mitchj@us.ibm.com

Steps to calling an external API

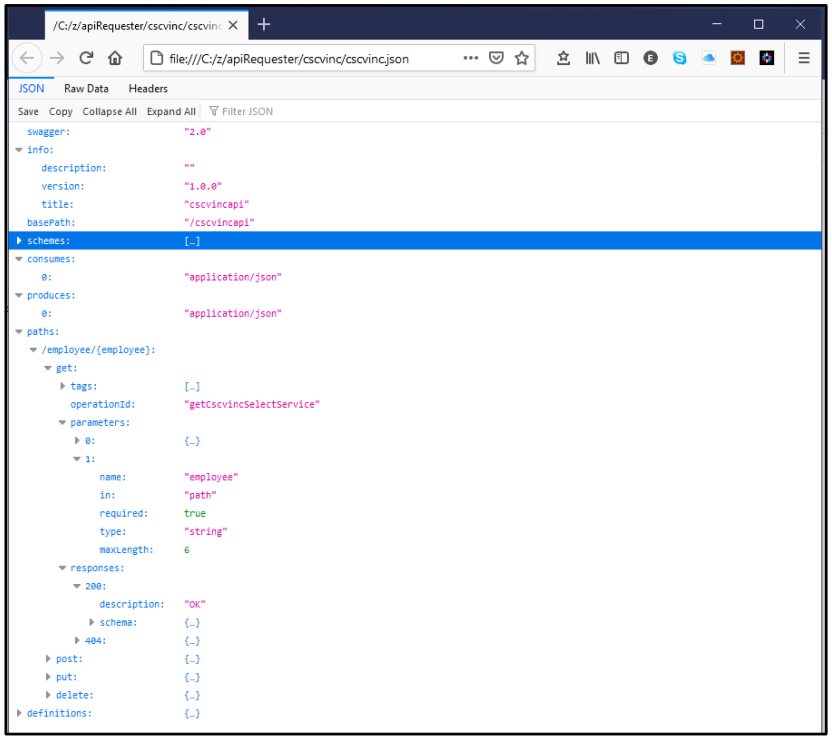
Starting point

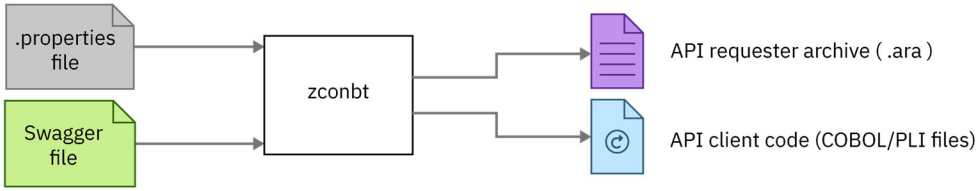


mitchj@us.ibm.com

Steps to calling an external API

Start with the API's specification and generate API requester archive file and API client code





properties file#

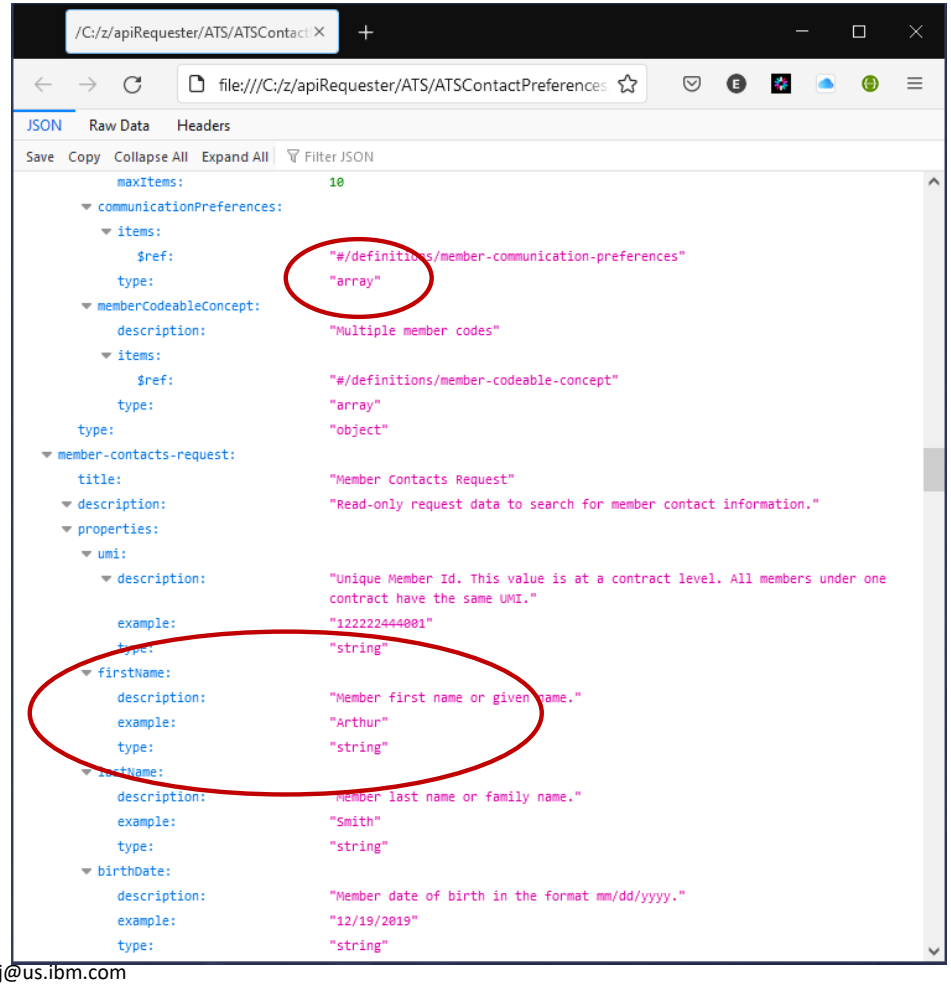
```
apiDescriptionFile=./cscvinc.json
dataStructuresLocation=./syslib
apiInfoFileLocation=./syslib
logFileDirectory=./logs
language=COBOL
connectionRef=cscvincAPI
requesterPrefix=csc
```

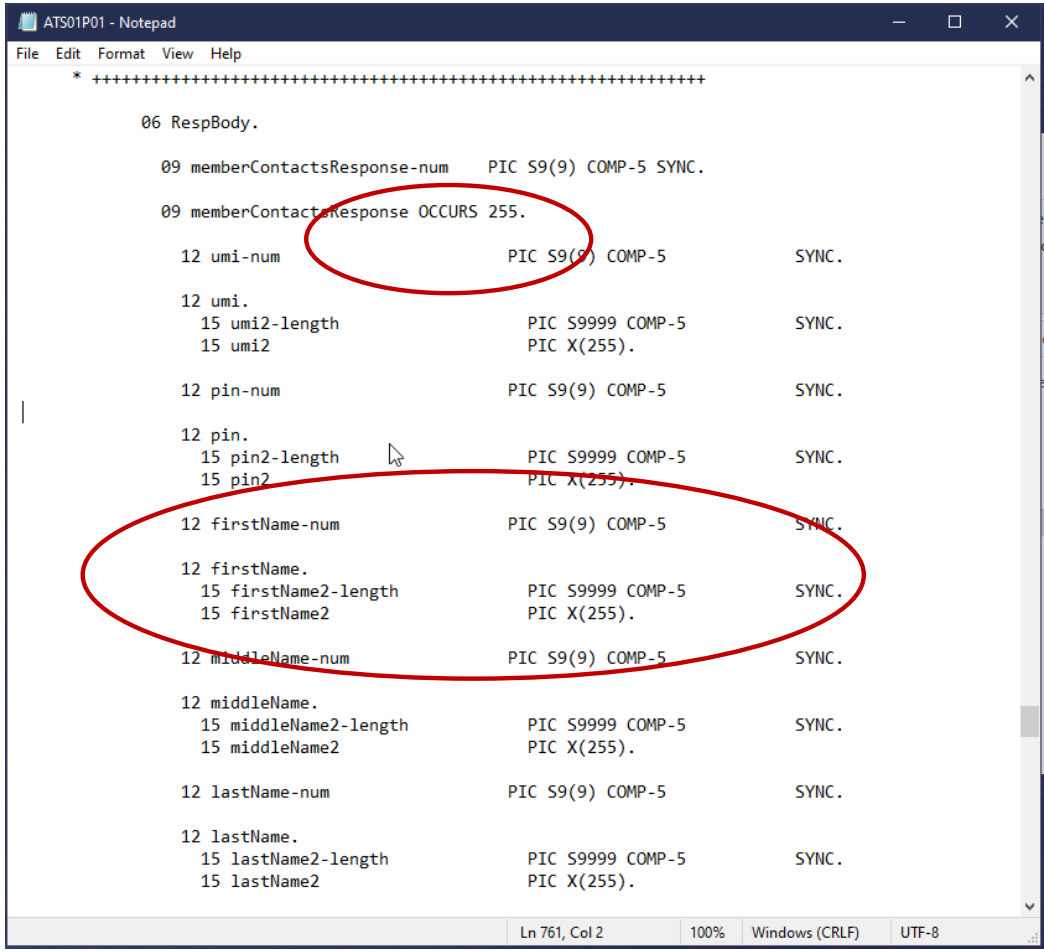
#Additional property file attributes, e.g., *defaultCharacterMaxLength*, *defaultArrayMaxItems*, etc. are described at **The build toolkit properties file** article at URL https://www.ibm.com/docs/en/zosconnect/3.0?topic=toolkit-build-properties-file

mitchj@us.ibm.com

COBOL Storage Considerations

Properties are usually not constrained, this can lead to excessive working storage consumption

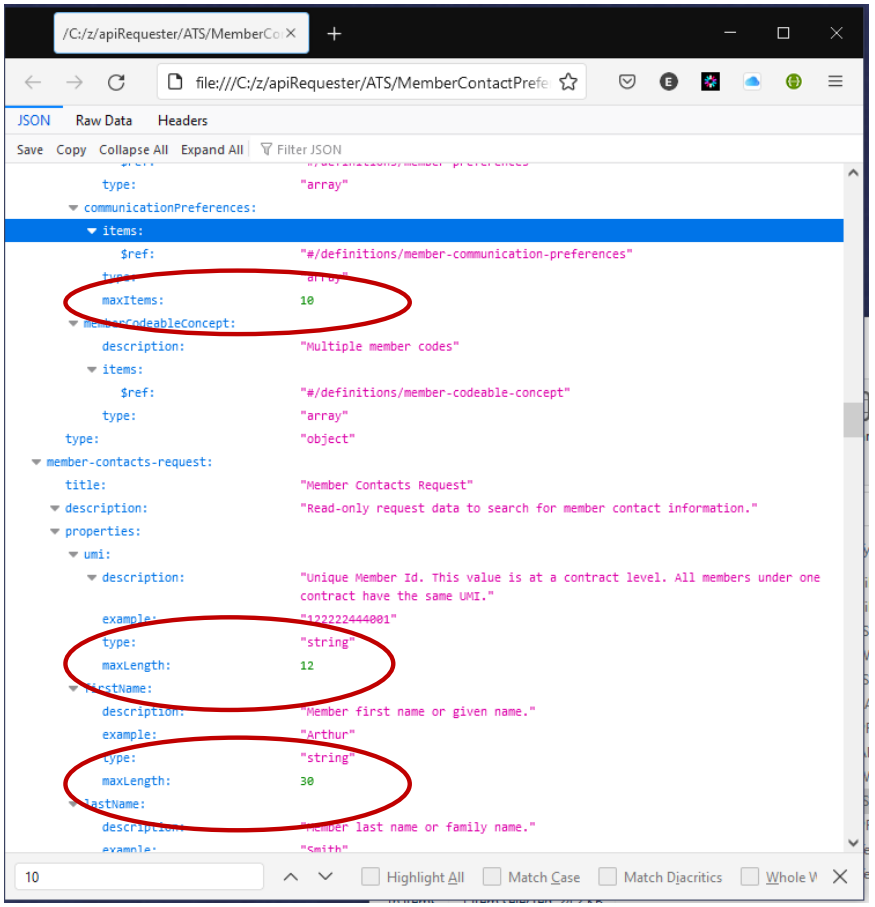


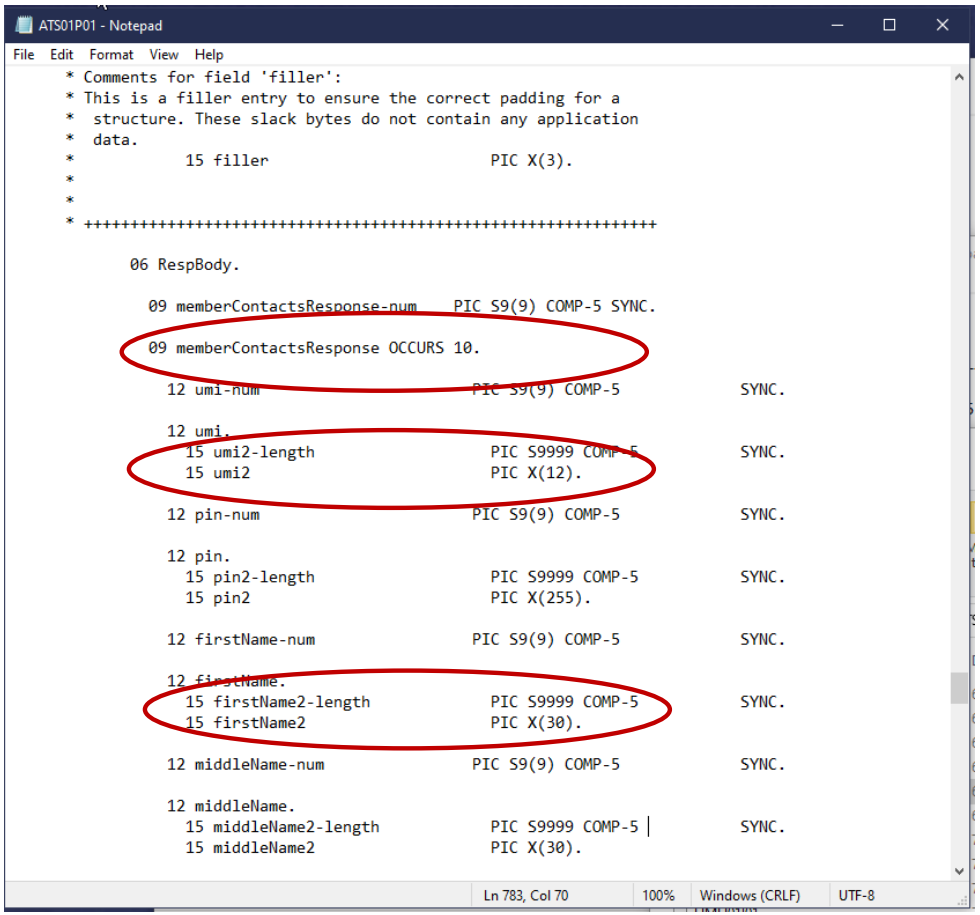


mitchj@us.ibm.com

Add constraints to the properties in the specification

Use the *maxItems* and *maxLength* attributes to set realistic maximum array and field sizes





mitchj@us.ibm.com

There are also API Requester properties available to help

Use these generation properties to set default array size and string field sizes

defaultArrayMaxItems - Specify the maximum array boundary to apply when no maximum occurrence information (maxItems) is implied in the Swagger. The value of this parameter can be a positive integer in the range 1 - 32767. By default, **defaultArrayMaxItems** is set to **255.**

defaultCharacterMaxLength - Specify the default array length of character data in characters for mappings where no length is implied in the JSON schema document. When **characterVarying** is set to YES, the value of this parameter can be a positive integer in the range of 1 to 32767. When **characterVarying** is set to NO or NULL the value of this parameter can be a positive integer in the range of 1 to 16777214 By default, **defaultCharacterMaxLength** is set to **255**.

characterVarying - Specifies how variable-length character data is mapped to the language structure.

- NO - Variable-length character data is mapped as fixed-length strings.
- NULL - Variable-length character data is mapped to null-terminated strings (defaultCharacterMaxLength + 1)
- YES - Variable-length character data is mapped to a CHAR VARYING data type in PL/I. In COBOL variable-length character data is mapped to an equivalent representation that consists of two related elements: the **data-length** and the **data**. By default, **characterVarying** is set to **YES**.

```
12 firstName-num                PIC S9(9) COMP-5            SYNC.

12 firstName.
  15 firstName2-length            PIC S9999 COMP-5          SYNC.
  15 firstName2                   PIC X(30).
```

```
12 firstName-num                PIC S9(9) COMP-5      SYNC.

12 firstName                    PIC X(31).
```

mitchj@us.ibm.com

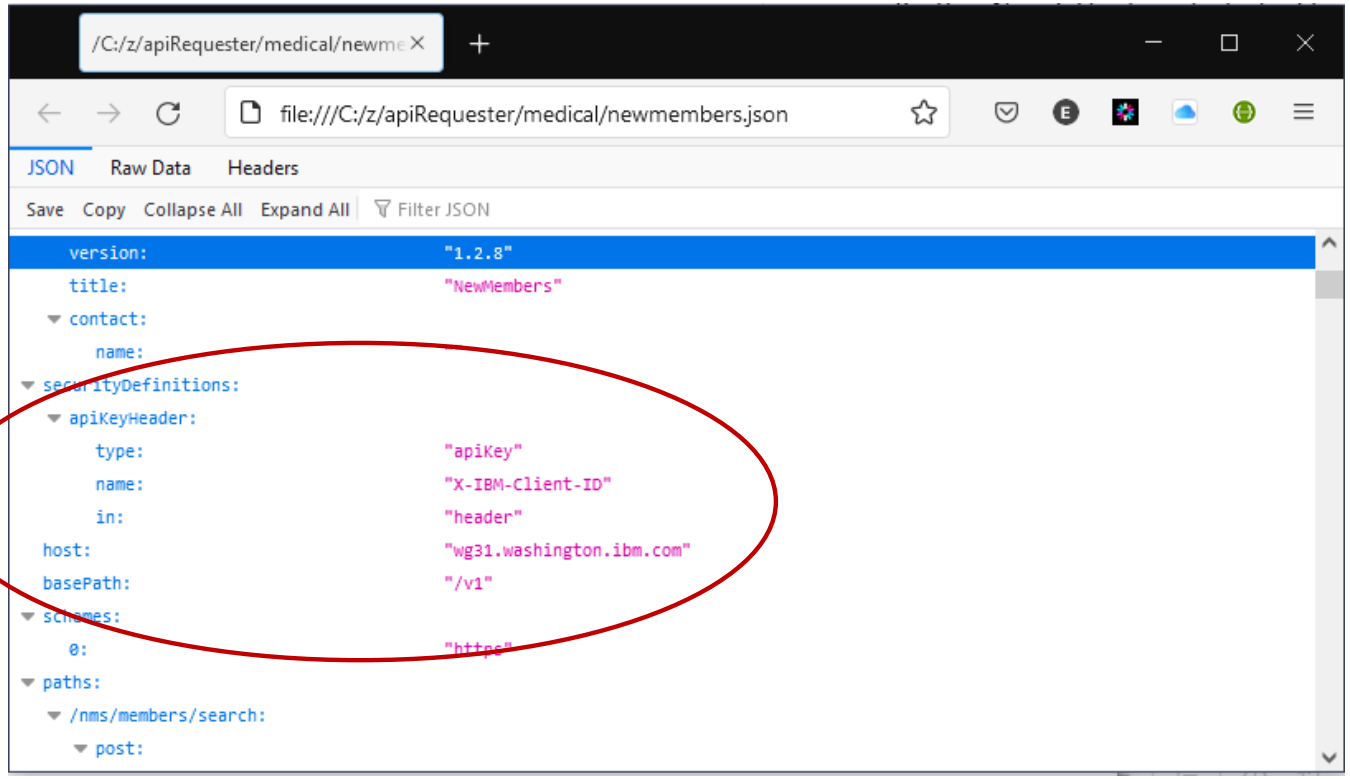
An API key to the request and the application

The application can provide the authentication credentials required by the API.

Via a HTTP header
GET /something HTTP/1.1
X-API-Key: abcdef12345

Or via a query parameter
GET /something?**api_key=abcdef12345**

When provided in the specification document as shown below or . . .



mitchj@us.ibm.com

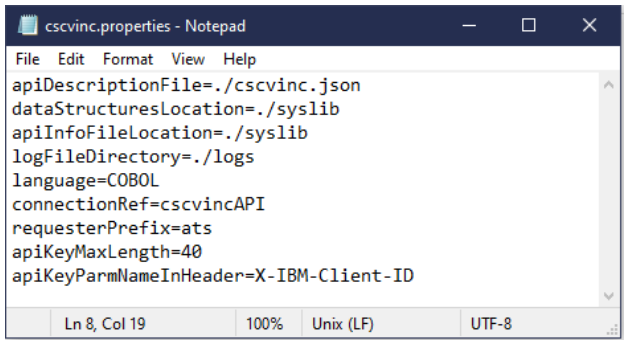
Or by using generation properties related to API keys

Use these generation properties to add API key information to the request message when not defined in specification document

apiKeyMaxLength - Specify the maximum length of the values set for API keys. The value of this parameter can be a positive integer in the range 1 - 32767. By default, **apiKeyMaxLength** is set to 255.

apiKeyParmNameInHeader - Specify the name of an API key that is sent as a request header. The value of this parameter can be set in a comma separated list of a combination of client ID and client secret. For example, you can set **apiKeyParmNameInHeader** =header-IBM-Client-ID, header-IBM-Client-secret when a client ID and a client secret are used to protect an API.

apiKeyParmNameInQuery - Specify the name of an API key that is sent in a query string. The value of this parameter can be set in a comma separated list of a combination of client ID and client secret. For example, you can set apiKeyParmNameInQuery=query-IBM-Client-ID, query-IBM-Client-secret when a client ID and a client secret are used to protect an API.



mitchj@us.ibm.com

Support for an application to add an API key to the request

Either way adds code to the request copy book which can be set by the application

request.cpy - Notepad

```
File  Edit  Format  View  Help
      *          12 dob2-length                  PIC S9999 COMP-5
      *  SYNC.
      *          12 dob2                         PIC X(255).
      *
      *
      * ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

           06 ReqHeaders.
              09 X-IBM-Client-ID-length         PIC S9999 COMP-5 SYNC.
              09 X-IBM-Client-ID                PIC X(255).
              09 X-HZN-ClientName-length        PIC S9999 COMP-5 SYNC.
              09 X-HZN-ClientName               PIC X(255).
              09 X-HZN-ClientSubmitDateTime     PIC S9(15) COMP-3.

              09 X-HZN-ClientTransactio-num     PIC S9(9) COMP-5 SYNC.

              09 X-HZN-ClientTransactionId.
                 12 X-HZN-ClientTransact-length PIC S9999 COMP-5
           SYNC.
                 12 X-HZN-ClientTransactionId2  PIC X(255).

              09 X-HZN-ClientSessionId-num      PIC S9(9) COMP-5 SYNC.

              09 X-HZN-ClientSessionId.
                 12 X-HZN-ClientSessionI-length PIC S9999 COMP-5
           SYNC.
                 12 X-HZN-ClientSessionId2      PIC X(255).

              09 X-HZN-UserRole-num             PIC S9(9) COMP-5 SYNC.

              09 X-HZN-UserRole.
                 12 X-HZN-UserRole2-length      PIC S9999 COMP-5
           SYNC.
                 12 X-HZN-UserRole2             PIC X(255).

              09 X-HZN-UserAssociationI-num     PIC S9(9) COMP-5 SYNC.

                                                  Ln 1, Col 1   100%   Windows (CRLF)   UTF-8
```

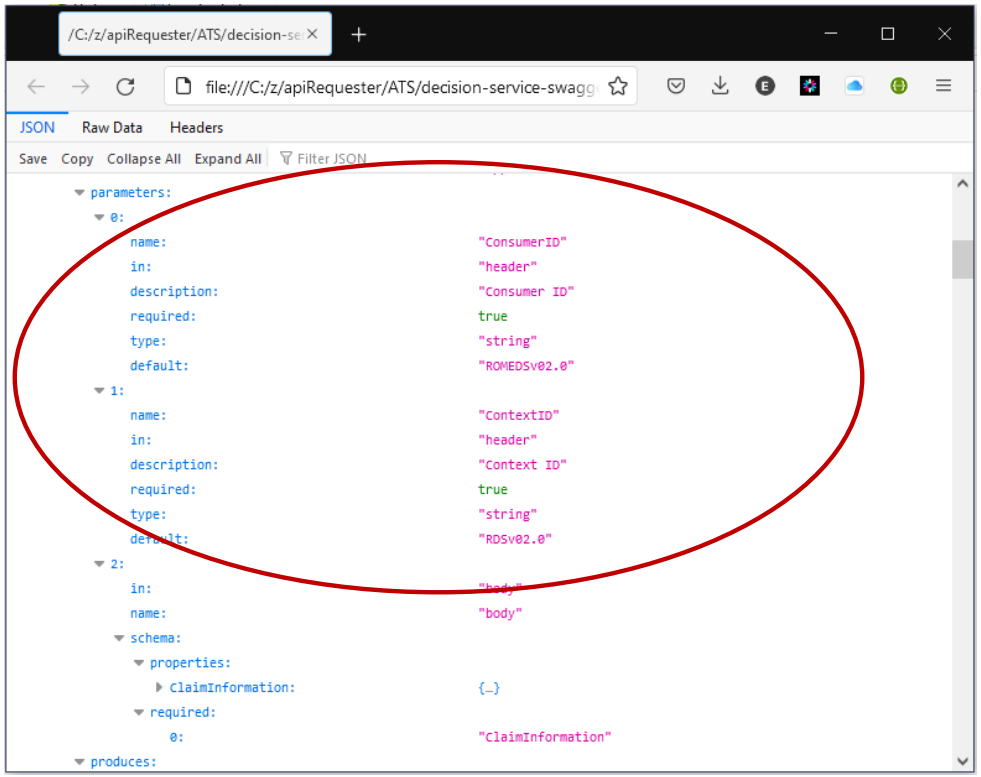
mpz3

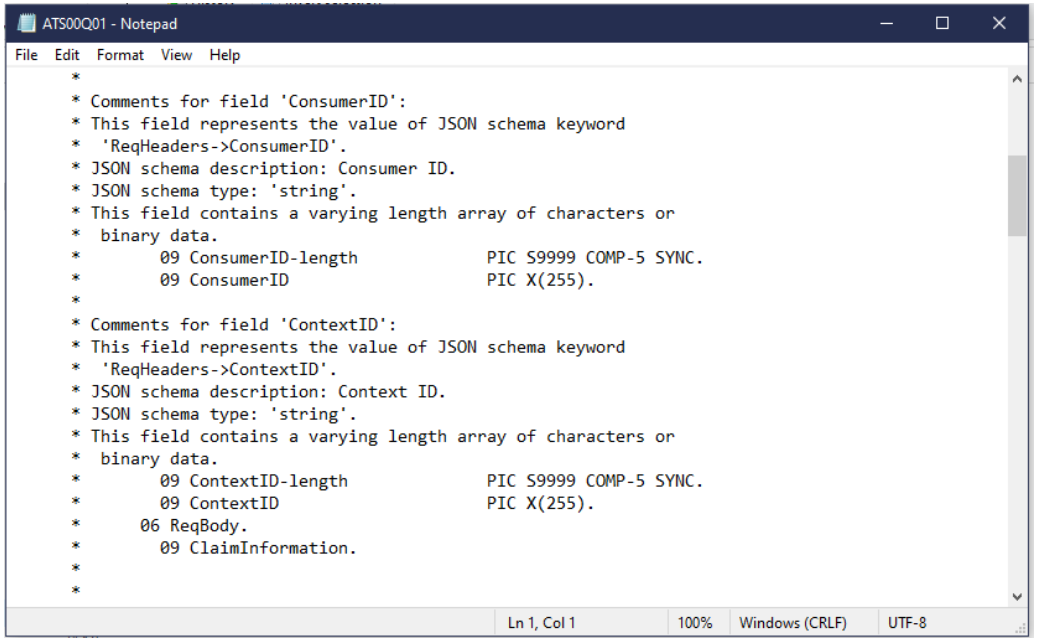
```
File  Edit  Settings  View  Communication  Actions  Window  Help
  File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help

EDIT       USER1.ZCEE.SOURCE(GETAPIEN) - 01.01            Columns 00001 00072
Command ===>                                               Scroll ===> PAGE
000081        *-------------------------------------------------------------*
000082        * Common code                                                 *
000083        *-------------------------------------------------------------*
000084        * initialize working storage variables
000085             INITIALIZE GET-REQUEST.
000086             INITIALIZE GET-RESPONSE.
000087             MOVE "abcdef12345" to X-IBM-Client-ID
000088             MOVE 11 to X-IBM-Client-ID-length
000089
000090        *-------------------------------------------------------------*
000091        * Set up the data for the API Requester call                  *
000092        *-------------------------------------------------------------*
000093             MOVE employee of PARM-DATA TO employee IN GET-REQUEST.
000094             MOVE LENGTH of employee in GET-REQUEST to
000095                 employee-length IN GET-REQUEST.
000096
000097        *-------------------------------------------------------------*
000098        * Initialize API Requester PTRs & LENs                        *
000099        *-------------------------------------------------------------*
000100        * Use pointer and length to specify the location of
000101        *  request and response segment.
000102        * This procedure is general and necessary.
000103             SET BAQ-REQUEST-PTR TO ADDRESS OF GET-REQUEST.
000104             MOVE LENGTH OF GET-REQUEST TO BAQ-REQUEST-LEN.
000105             SET BAQ-RESPONSE-PTR TO ADDRESS OF GET-RESPONSE.
000106             MOVE LENGTH OF GET-RESPONSE TO BAQ-RESPONSE-LEN.
000107
000108        *-------------------------------------------------------------*
MA    A                                                          13/002
Connected to remote server/host mpz3 using lu/pool MPZ30019 and port 23
```

mitchj@us.ibm.com

Additional Swagger header properties

The application can set values for additional header properties required by the API





mitchj@us.ibm.com

Steps to calling an external API

Using *zconbt* to generate API requester archive and API client code from Swagger

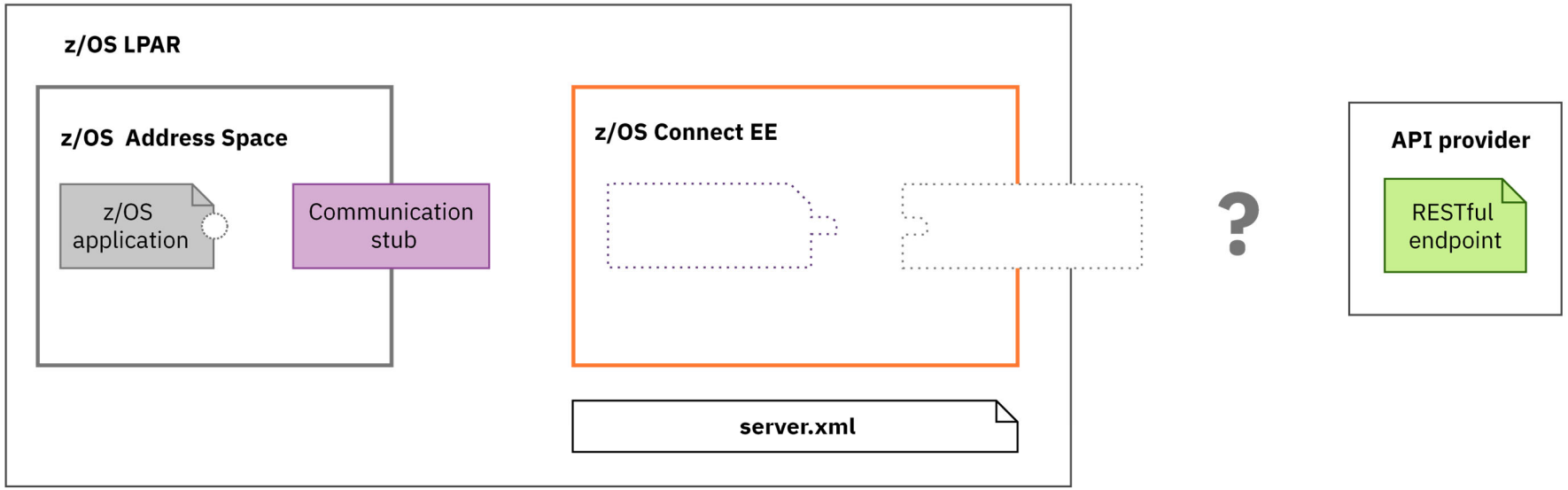
```
zconbt.bat -p=./cscvinc.properties -f=./cscvinc.ara
BAQB0000I: z/OS Connect Enterprise Edition 3.0 Build Toolkit Version 1.5 (20210816-0926).
BAQB0008I: Creating API requester archive from configuration file ./cscvinc.properties.
BAQB0040I: The generated API requester is automatically named cscvincapi_1.0.0 based on the title cscvincapi and version 1.0.0 of the API to be
called.
. . . .  .
Total 4 operation(s) (success: 4, ignored: 0) defined in api description file: ./cscvinc.json
------ Successfully processed operation(s) ------
operationId: getCscvincSelectService, basePath: /cscvincapi, relativePath: /employee/{employee}, method: GET
- request data structure  : CSC00Q01
- response data structure : CSC00P01
- api info file           : CSC00I01

operationId: putCscvincUpdateService, basePath: /cscvincapi, relativePath: /employee/{employee}, method: PUT
- request data structure  : CSC01Q01
- response data structure : CSC01P01
- api info file           : CSC01I01

operationId: postCscvincInsertService, basePath: /cscvincapi, relativePath: /employee/{employee}, method: POST
- request data structure  : CSC02Q01
- response data structure : CSC02P01
- api info file           : CSC02I01

operationId: deleteCscvincDeleteService, basePath: /cscvincapi, relativePath: /employee/{employee}, method: DELETE
- request data structure  : CSC03Q01
- response data structure : CSC03P01
- api info file           : CSC03I01


BAQB0009I: Successfully created API requester archive file ./cscvinc.ara.
```

mitchj@us.ibm.com

https://www.ibm.com/docs/en/zos-connect/zosconnect/3.0?topic=installing-build-toolkit

Steps to calling an external API

Update the application by adding the copy books and a call to communication stub



Configure a communication stub.

- For CICS region systems using URIMAP resources
- For non CICS client the configuration is done via environment variables

ibm.biz/zosconnect-configure-comms-stub

mitchj@us.ibm.com

Steps to calling an external API

Include the generated copy books in a COBOL program

GETAPI

```
      * Error message structure
       01  ERROR-MSG.
           03 EM-ORIGIN                PIC X(8)  VALUE SPACES.
           03 EM-CODE                  PIC S9(9) COMP-5 SYNC VALUE 0.
           03 EM-DETAIL                PIC X(1024) VALUE SPACES.

      * Copy API Requester required copybook
       COPY BAQRINFO.

      * Request and Response
       01 API-REQUEST.
           COPY CSC02Q01.
       01 API_RESPONSE.
           COPY CSC02P01.
      * Structure with the API information
       01 API-INFO-OPER1.
           COPY CSC02I01.

      * Request and Response segment  used to store request and
```

API-REQUEST

CSC00I01 | CSC00Q01

```
      * JSON schema keyword 'minLength' value: '0'.
      * JSON schema keyword 'maxLength' value: '6'.
      * This field contains a varying length array of characters or
      *  binary data.
      *       09 employee-length              PIC S9999 COMP-5 SYNC.
      *       09 employee                     PIC X(6).
      *
      *
      * ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

           06 ReqPathParameters.
             09 employee-length              PIC S9999 COMP-5 SYNC.
             09 employee                     PIC X(6).
```

API-RESPONSE

CSC00I01 | CSC00Q01 | CSC00P01

```
      *
      * ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

           06 RespBody.

             09 cscvincSelectServiceOp-num    PIC S9(9) COMP-5 SYNC.

             09 cscvincSelectServiceOperatio.
               12 Container1.

                 15 RESPONSE-CONTAINER2-num       PIC S9(9) COMP-5
           SYNC.
```

API-INFO-OPER1

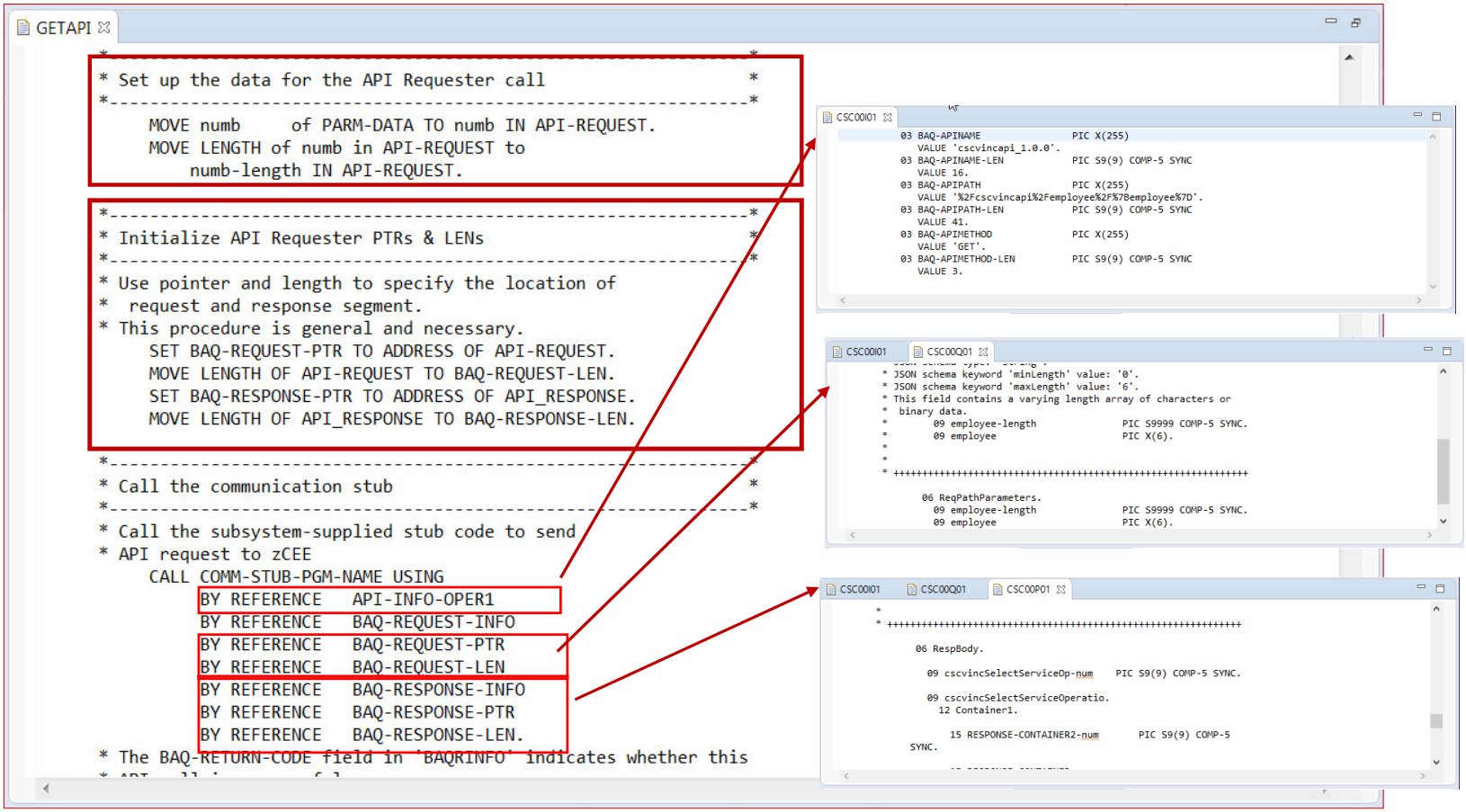
CSC00I01

```
           03 BAQ-APINAME                PIC X(255)
              VALUE 'cscvincapi_1.0.0'.
           03 BAQ-APINAME-LEN            PIC S9(9) COMP-5 SYNC
              VALUE 16.
           03 BAQ-APIPATH                PIC X(255)
              VALUE '%2Fcscvincapi%2Femployee%2F%7Bemployee%7D'.
           03 BAQ-APIPATH-LEN            PIC S9(9) COMP-5 SYNC
              VALUE 41.
           03 BAQ-APIMETHOD              PIC X(255)
              VALUE 'GET'.
           03 BAQ-APIMETHOD-LEN          PIC S9(9) COMP-5 SYNC
              VALUE 3.
```

mitchj@us.ibm.com

Steps to calling an external API

Add a call to the communication stub passing pointers to working storage of the copy books



mitchj@us.ibm.com

Steps to calling an external API

Access the results

GETAPI

```
            BY REFERENCE   BAQ-RESPONSE-LEN.
      * The BAQ-RETURN-CODE field in 'BAQRINFO' indicates whether this
      * API call is successful.

      * When BAQ-RETURN-CODE is 'BAQ-SUCCESS', response is
      * successfully returned and fields in RESPONSE copybook
      * can be obtained. Display the translation result.
          IF BAQ-SUCCESS THEN
             DISPLAY "NUMB:    " numb2   of API_RESPONSE
             DISPLAY "NAME:    " name2   of API_RESPONSE
             DISPLAY "ADDRX:   " addrx2  of API_RESPONSE
             DISPLAY "PHONE:   " phone2  of API_RESPONSE
             DISPLAY "DATEX:   " datex2  of API_RESPONSE
             DISPLAY "AMOUNT: " amount2 of API_RESPONSE
             MOVE CEIBRESP of API_RESPONSE to EIBRESP
             MOVE CEIBRESP2 of API_RESPONSE to EIBRESP2
             DISPLAY "EIBRESP:     " EIBRESP
             DISPLAY "EIBRESP2:    " EIBRESP2
             DISPLAY "HTTP CODE:   " BAQ-STATUS-CODE

      * Otherwise, some error happened in API, z/OS Connect EE server
      * or communication stub. 'BAQ-STATUS-CODE' and
      * 'BAQ-STATUS-MESSAGE' contain the detailed information
      *  of this error.
          ELSE
             DISPLAY "Error code: " BAQ-STATUS-CODE
             DISPLAY "Error msg:" BAQ-STATUS-MESSAGE
             MOVE BAQ-STATUS-CODE TO EM-CODE
             MOVE BAQ-STATUS-MESSAGE TO EM-DETAIL
             EVALUATE TRUE
      * When error happens in API, BAQ-RETURN-CODE is BAQ-ERROR-IN-API.
      * BAQ-STATUS-CODE is the HTTP response code of API.
```

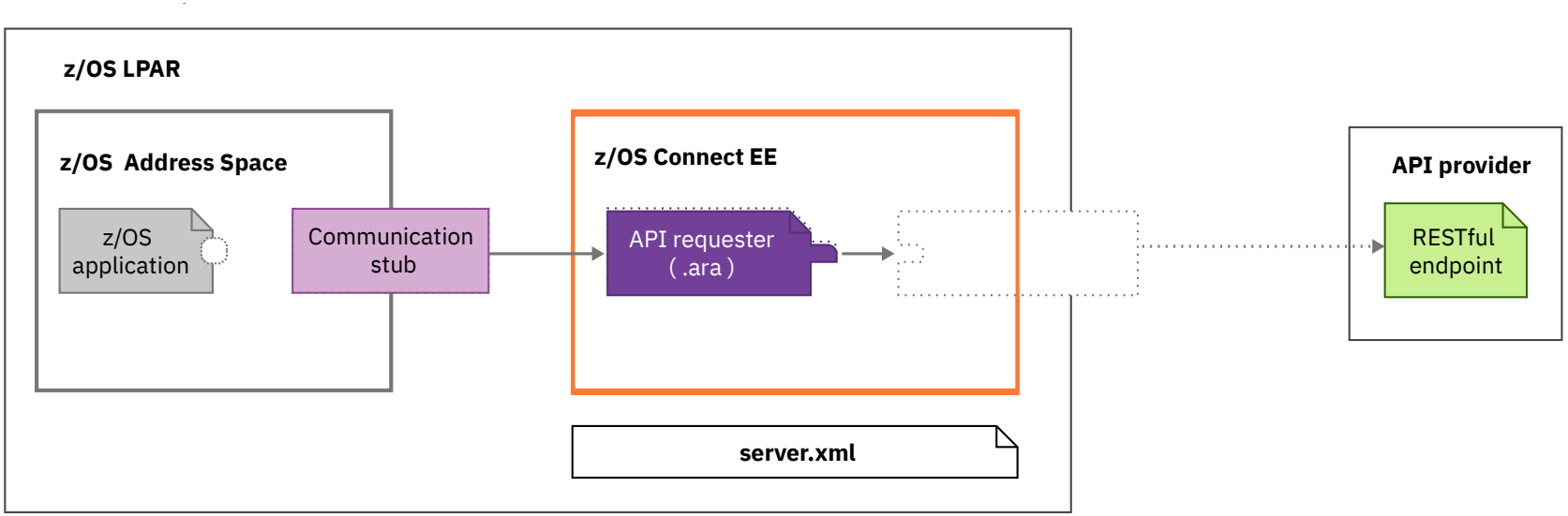
mpz3

```
File  Edit  Settings  View  Communication  Actions  Window  Help
  Menu  Utilities  Compilers  Help
BROWSE    ZCEE30.SBAQCOB(BAQRINFO)                Line 0000000066 Col 001 080
Command ===>                                        Scroll ===> PAGE

      01  BAQ-RESPONSE-INFO.
        03 BAQ-RESPONSE-INFO-COMP-LEVEL PIC S9(9) COMP-5 SYNC VALUE 0.
        03 BAQ-STUB-NAME                PIC X(8).
        03 BAQ-RETURN-CODE              PIC S9(9) COMP-5 SYNC.
           88 BAQ-SUCCESS                 VALUE 0.
           88 BAQ-ERROR-IN-API            VALUE 1.
           88 BAQ-ERROR-IN-ZCEE           VALUE 2.
           88 BAQ-ERROR-IN-STUB           VALUE 3.
           88 BAQ-ERROR-NO-RESPONSE       VALUE 4.
        03 BAQ-STATUS-CODE              PIC S9(9) COMP-5 SYNC.
        03 BAQ-STATUS-MESSAGE           PIC X(1024).
        03 BAQ-STATUS-MESSAGE-LEN       PIC S9(9) COMP-5 SYNC.
******************************** Bottom of Data ********************************
MA    B                                                          18/058
Connected to remote server/host mpz3 using lu/pool MPZ30021 and port 23
```

mitchj@us.ibm.com

Steps to calling an external API

Deploy API requester (.ara) archive

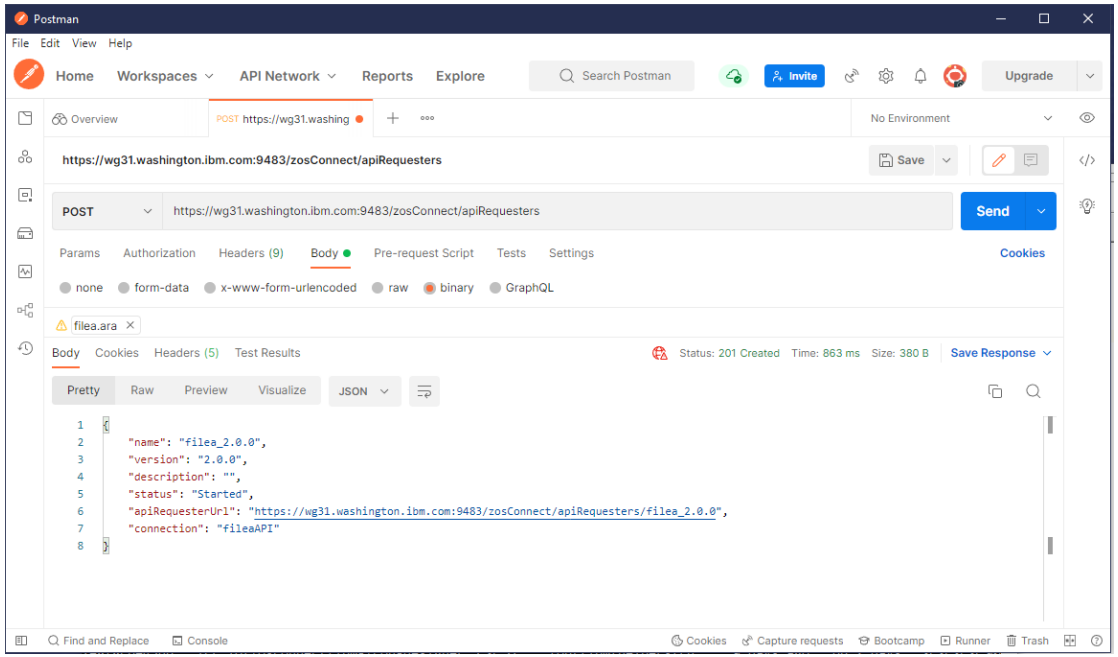


Deploy your API requester archive to the *apiRequesters* directory.

mitchj@us.ibm.com

Deploying API requester archive files

- Use API requester archive as request message and use HTTP POST
- Use URI path /zosConnect/apiRequesters
- Postman or cURL



Command:

```
curl --data-binary @filea.ara
--header "Content-Type: application/zip"
https://mpxm:9453/zosConnect/apiRequesters
```

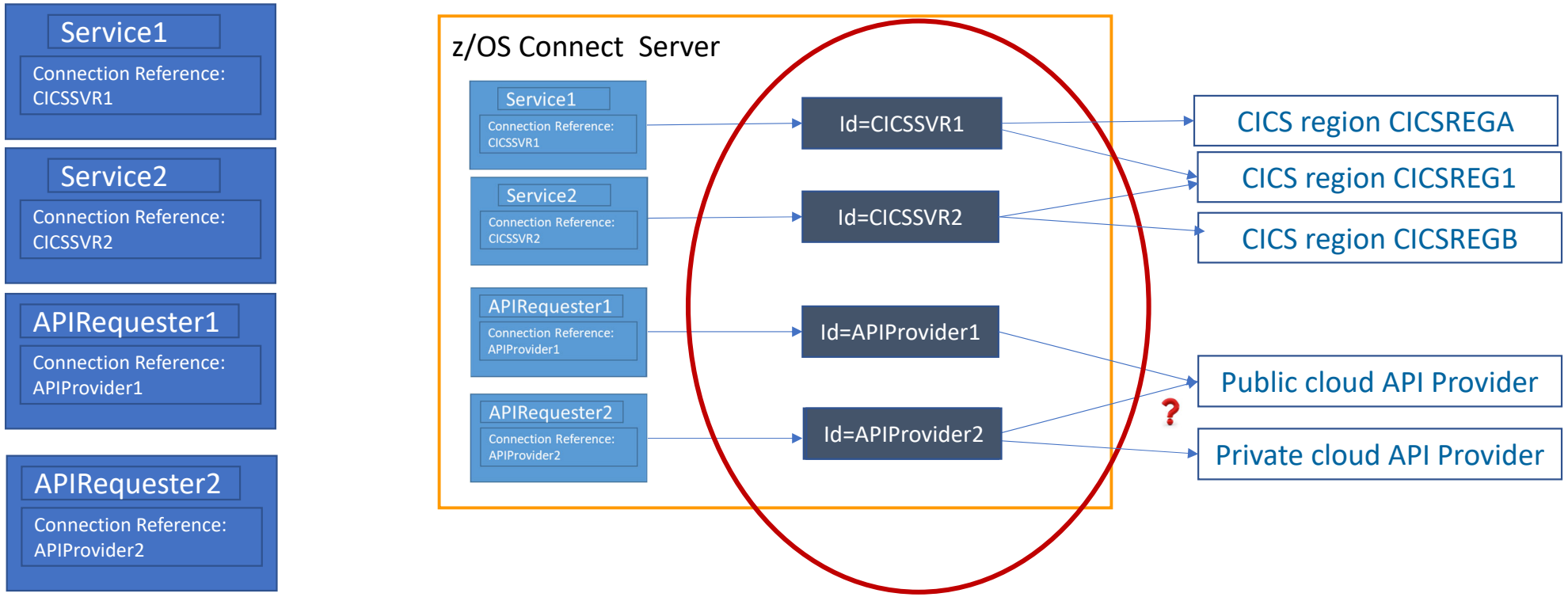
Results:


```
{"name":"filea_2.0.0","version":"2.0.0","description":"",
"status":"Started","apiRequesterUrl":"https:\/\/wg31.wash
ington.ibm.com:9483\/zosConnect\/apiRequesters\/filea_2.0
.0","connection":"fileaAPI"}
```


mitchj@us.ibm.com

Use naming conventions for connection references

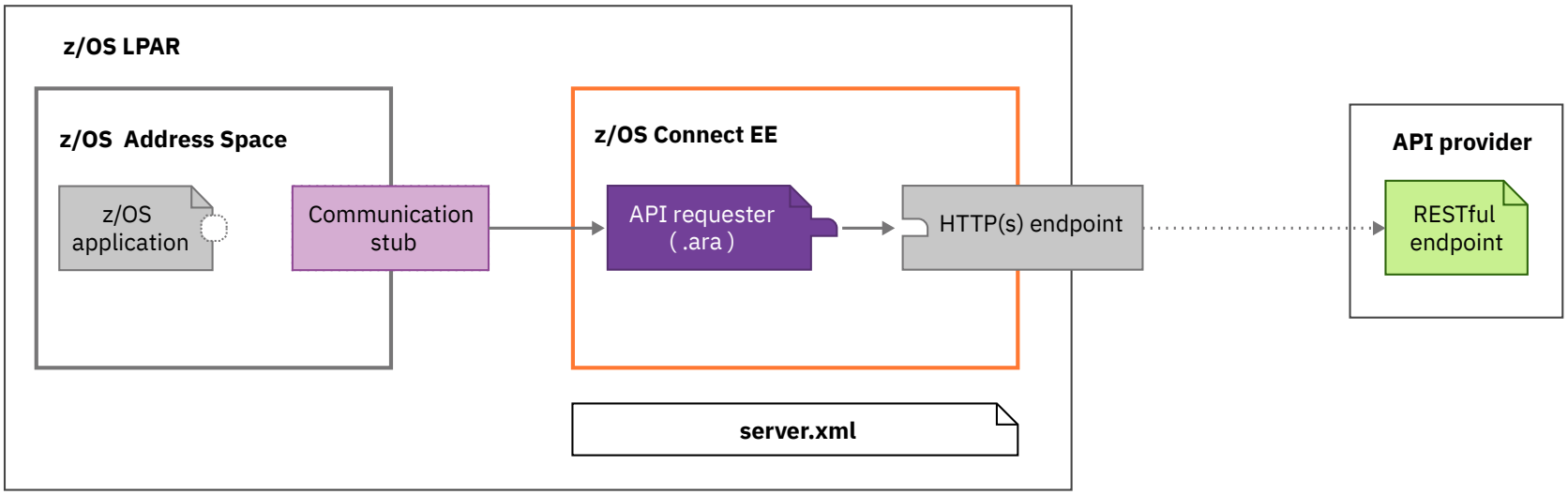
Use application meaningful names or an extendable convention for connection reference names



mitchj@us.ibm.com

Steps to calling an external API

Configure HTTP(S) endpoint configuration element



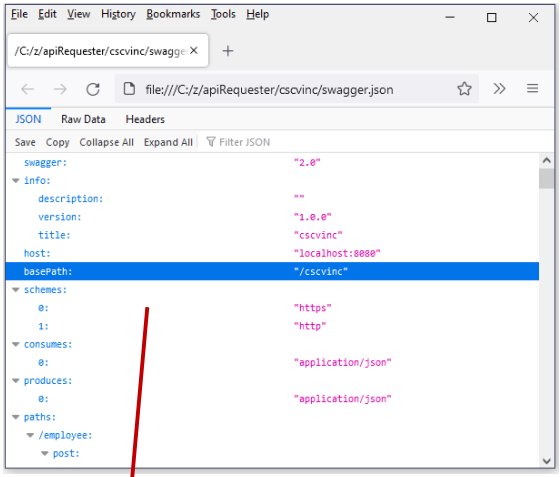
Configure the connection between z/OS Connect EE and the external API.

ibm.biz/zosconnect-configure-endpoint-connection

mitchj@us.ibm.com

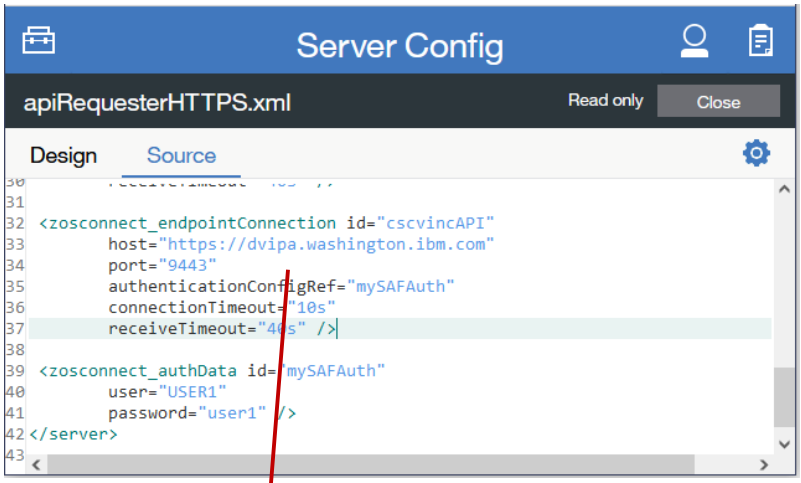
Steps to calling an external API

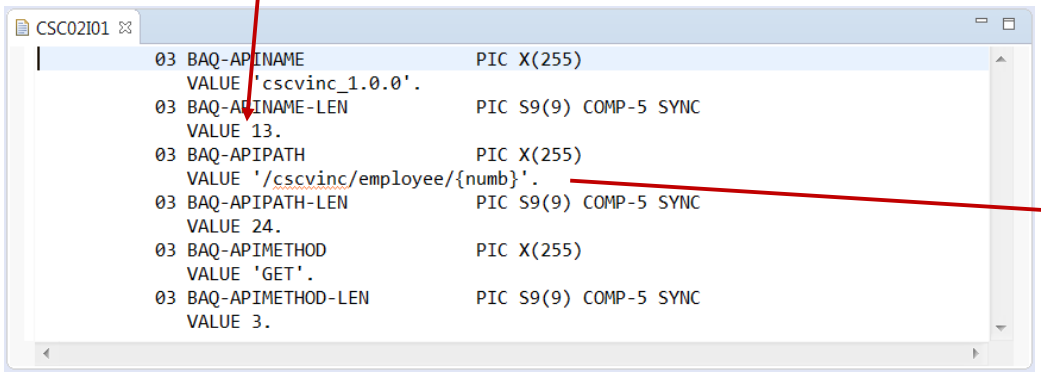
Update the server XML configuration for the endpoint



cscvinc.properties

```
connectionRef=cscvincAPI
```



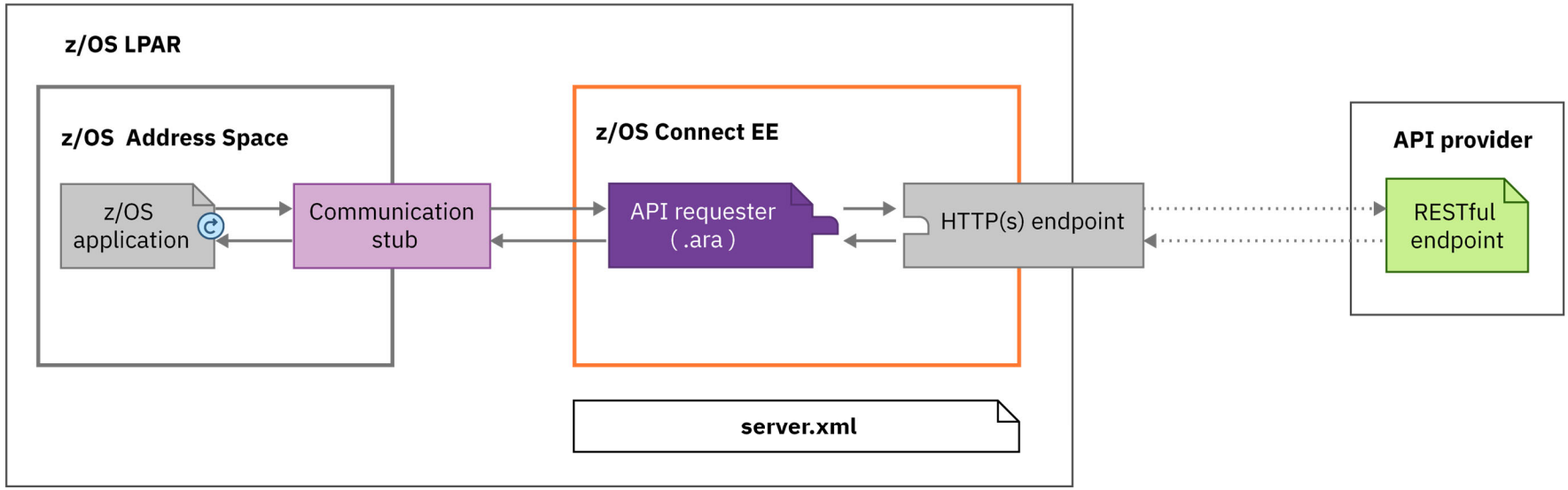


http://dvipa.washington.ibm.com:9443/cscvincapi/employee/{numb}

mitchj@us.ibm.com

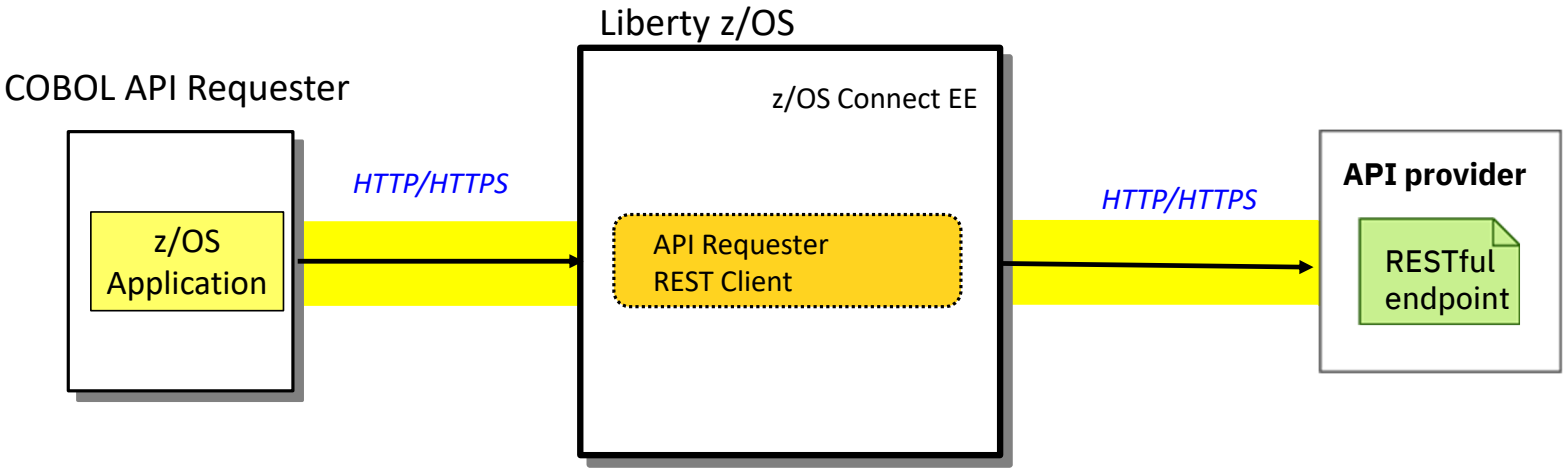
Steps to calling an external API

Done



mitchj@us.ibm.com

API requester to API Provider connection overview



MVS Batch, IMS HTTP and Db2 stored procedure connection details provided by:

- Environment Variables (BAQURI, BAQPORT)
  - Via JCL
  - LE Options (CEEROPTS)
  - Programmatically (CEEENV)
- HTTP or HTTPS

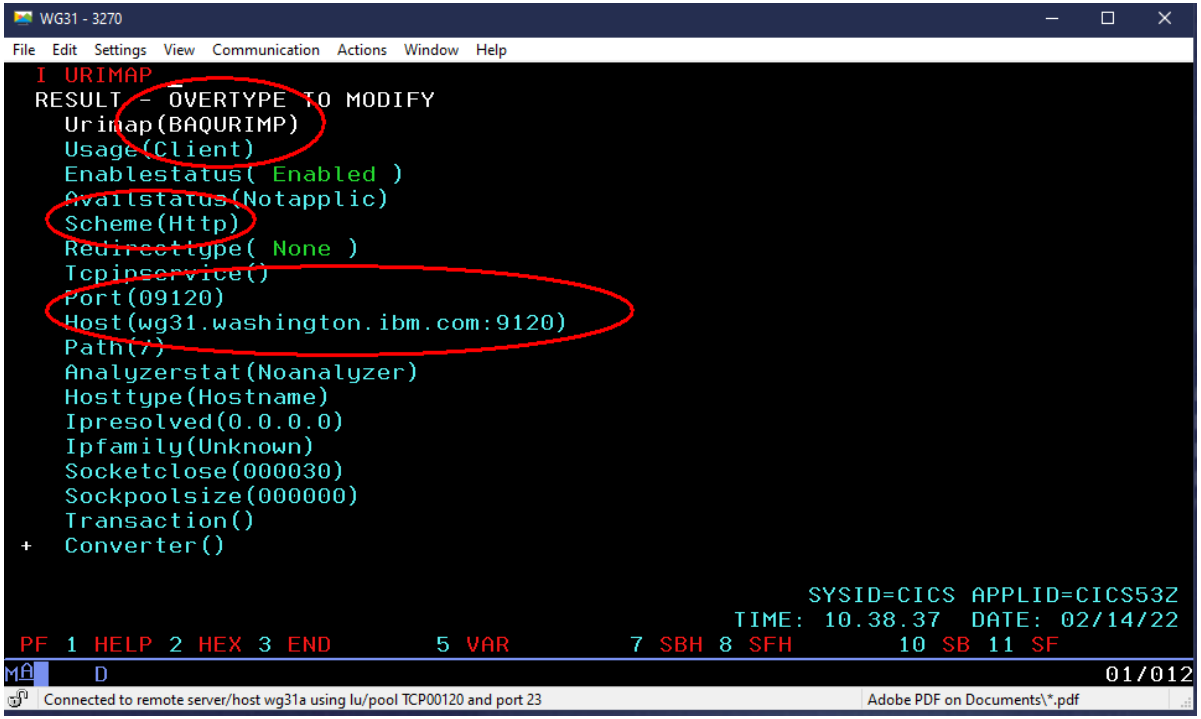
CICS HTTP connection details provided by:

- CICS URIMAP resource (default BAQURIMP)
  - HOST
  - PORT
  - SCHEME (HTTP/HTTPS)

mitchj@us.ibm.com

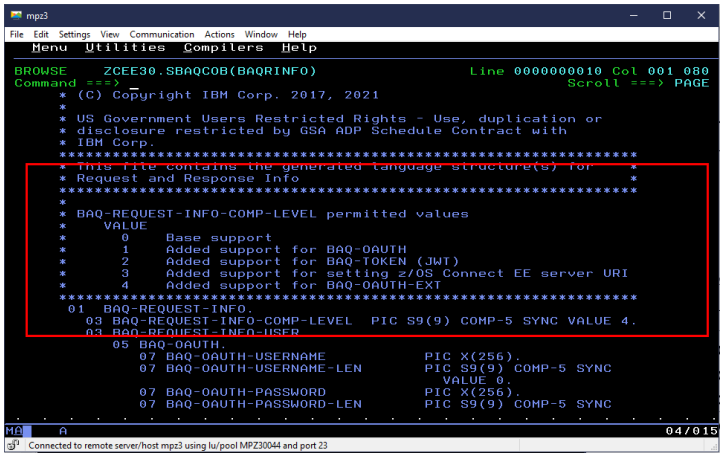
Configure connections to the z/OS API requester server

Default CICS URI MAP*



LE Environment Variables

```
//DELTAPI EXEC PGM=DELTAPI,PARM='323232'
//STEPLIB  DD
DISP=SHR,DSN=USER1.ZCEE.LOADLIB
//          DD DISP=SHR,DSN=ZCEE30.SBAQLIB
//SYSOUT    DD SYSOUT=*
//SYSPRINT DD SYSOUT=* //CEEOPTS  DD *
 POSIX(ON),
 ENVAR("BAQURI=wg31.washington.ibm.com",
 "BAQPORT=9120")
```



* V3.0.37 added support for a CICS application to specify or request a specific URIMAP resource the using BAQ-ZCON-SERVER-URI variable in BAQRINFO

mitchj@us.ibm.com

Runtime Environment variables

Use these runtime environment variables when connecting to a z/OS Connect server

BAQPASSWORD - Specifies the password, in clear text, for the specified BAQUSERNAME to be authenticated with the z/OS Connect server. The username and password that are used for basic authentication, when SSL mutual authentication is not enabled.

BAQPORT - Specifies the port number for the z/OS Connect server.

BAQTIMEOUT - An optional 4-byte integer to set a timeout value in seconds for waiting for an API response. Valid range is 1 - 2,678,400 seconds. The default timeout value is 10 seconds.

BAQURI - Specifies either an IPv4 or IPV6 address, or a hostname of the host where the z/OS Connect server resides.

BAQUSERNAME - Specifies the username for connections if basic authentication is used.

BAQVERBOSE - An optional value to turn on verbose messages to assist debugging of runtime and configuration issues. Valid values are **OFF**, **ON**, **ERROR**, **AUDIT** and **ALL**. See URL https://www.ibm.com/docs/en/zos-connect/zosconnect/3.0?topic=car-configuring-other-zos-applications-access-zos-connect-api-calls for more information.

mitchj@us.ibm.com

Basic authentication – COBOL API Requester

- A MVS batch, IMS or Db2 stored procedure requester application sends basic authentication information (identity and password) by using environment variables.
  - BAQUSERNAME
  - BAQPASSWORD

- The variables can be provided in JCL using CEEOPTS DD statement:

```
//CEEOPTS  DD *
     POSIX(ON),
     ENVAR("BAQURI=wg31.washington.ibm.com",
    "BAQPORT=9080",
    "BAQUSERNAME=USER1",
    "BAQPASSWORD=USER1")
```

- Or, provided by using a CEEROPT or CEEUOPT module:

```
CEEROPT CSECT
CEEROPT AMODE ANY
CEEROPT RMODE ANY
CEEXOPT POSIX=((ON),OVR),                                          +
               ENVAR=(('BAQURI=wg31.washington.ibm.com',           +
               'BAQPORT=9120',                                     +
               'BAQUSERNAME=USER1',                                +
               'BAQPASSWORD=USER1'),OVR),                          +
               RPTOPTS=((ON),OVR)
       END
```

Tech/Tip: This is good opportunity to use a pass ticket rather than a password

mitchj@us.ibm.com

Tech/Tip: A PassTicket provides an alternative to a password

- A PassTicket is generated by or for a client by using a secured sign-on key (whose value is masked or encrypted) to encrypt a valid *RACF identity* combined with the *application name* of the targeted resource. Also embedded in the PassTicket is a time stamp (based on the current Universal Coordinated Time (UCT)) which sets the time when the PassTicket will expire (usually 10 minutes).

- Access to PassTickets is managed using the RACF PTKTDATA class.

- For z/OS Connect, a RACF PassTicket can be used for basic authentication when connecting from any REST client on any platform to a z/OS Liberty server and for requests from a z/OS Connect server accessing IMS and Db2.

- ***PassTickets do not have to be generated on z/OS using RACF services.*** IBM has published the algorithm used to generate a PassTickets, see manual *z/OS Security Server RACF Macros and Interfaces, SA23-2288-40. Github has examples using Java, Python and other example are available on other sites*.

```
<safRegistry id="saf" />
   <safAuthorization racRouteLog="ASIS" />
   <safCredentials unauthenticatedUser="WSGUEST"
      profilePrefix="BBGZDFLT" />
```

mitchj@us.ibm.com

Tech/Tip: Generating PassTickets on z/OS

- On z/OS, a COBOL user application can generate a pass tickets by calling RACF service IRRSPK00:

```
 77 COMM-STUB-PGM-NAME          PIC X(8) VALUE 'BAQCSTUB'.
 77 PTKT-STUB-PGM-NAME          PIC X(8) VALUE 'ATSPTKTC'.

*-----------------------------------------------------------------*

*****************************************************************
*    L I N K A G E   S E C T I O N
*****************************************************************
 LINKAGE SECTION.
*****************************************************************
*    P R O C E D U R E S
*****************************************************************
 PROCEDURE DIVISION using PARM-BUFFER.

*-----------------------------------------------------------------*
 MAINLINE SECTION.

*-----------------------------------------------------------------*
* Common code                                                     *
*-----------------------------------------------------------------*
* initialize working storage variables
     INITIALIZE GET-REQUEST.
     INITIALIZE GET-RESPONSE.
     CALL PTKT-STUB-PGM-NAME.
```

JOHNSON.PASSTCKT.SOURCE(ATSPTKTC)

```
      *-----------------------------------------------------------------*
      * Build IRRSPK00 parameters                                       *
      *-----------------------------------------------------------------*
           MOVE 0 to ws-length
           MOVE LENGTH OF identity to identity-length.
           INSPECT FUNCTION REVERSE (identity)
                  TALLYING ws-length FOR ALL SPACES.
           SUBTRACT ws-length FROM identity-length.
           MOVE 0 to ws-length
           MOVE LENGTH OF applid to applid-length.
           INSPECT FUNCTION REVERSE (applid)
                  TALLYING ws-length FOR ALL SPACES.
           SUBTRACT ws-length FROM applid-length.
           MOVE 8 to passTicket-length.
           MOVE 'NOTICKET' to passTicket.
           MOVE X'0003' to irr-functionCode.
           MOVE X'00000001' to irr-ticketOptions.
           SET irr-ticketOptions-ptr to ADDRESS OF irr-ticketOptions.
      *-----------------------------------------------------------------*
      * Call RACF service IRRSPK00 to obtain a pass ticket based        *
      *      on identity and applid                                     *
      *-----------------------------------------------------------------*
           PERFORM CALL-RACF.
           IF irr-safrc NOT = zero then
              DISPLAY "SAF_return_code:     " irr-safrc
              DISPLAY "RACF_return_code:    " irr-racfrc
              DISPLAY "RACF_reason_code:    " irr-racfrsn
           End-if
. . . .
      *-----------------------------------------------------------------*
      * Call IRRSPK00 requesting a pass ticket                          *
      *-----------------------------------------------------------------*
       CALL-RACF.
            CALL W-IRRSPK00 USING irr-workarea,
                  IRR-ALET, irr-safrc,
                  IRR-ALET, irr-racfrc,
                  IRR-ALET, irr-racfrsn,
                  IRR-ALET, irr-functionCode,
                  irr-optionWord,
                  IRR-PASSTICKET,
                  irr-ticketOptions-ptr,
                  IRR-IDENTITY,
                  IRR-APPLID
```

mitchj@us.ibm.com

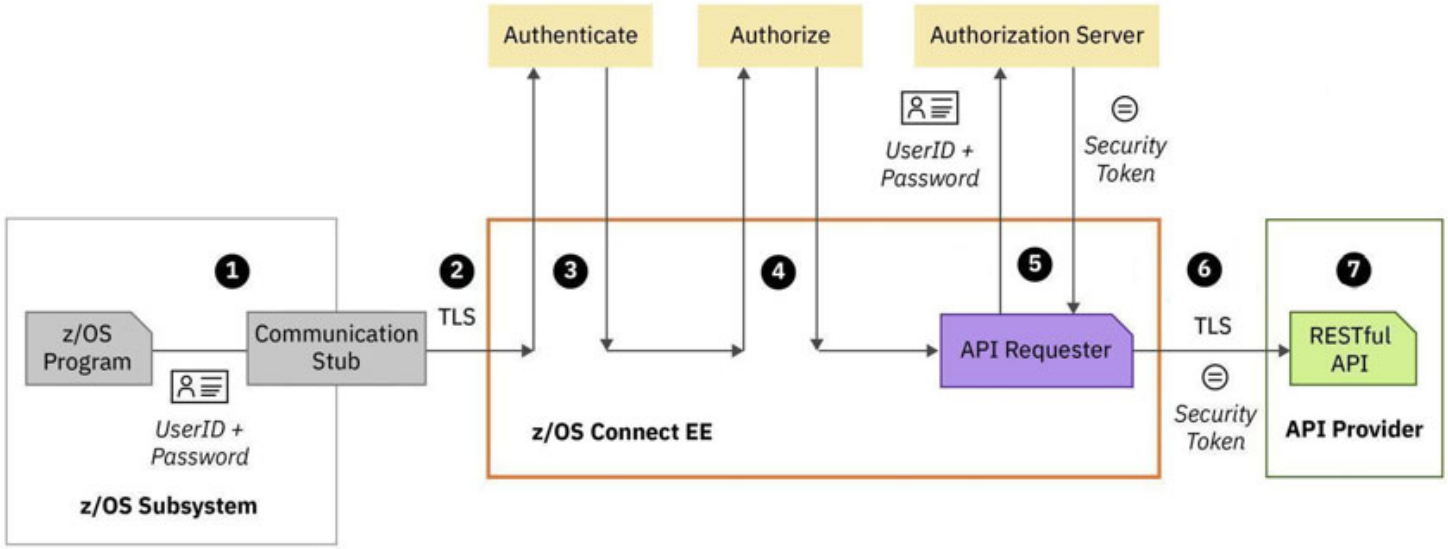
See the full source of the program at URL https://ibm.biz/BdPXny

/security

How is security implement?

mitchj@us.ibm.com

Typical z/OS Connect EE API Requester security flow



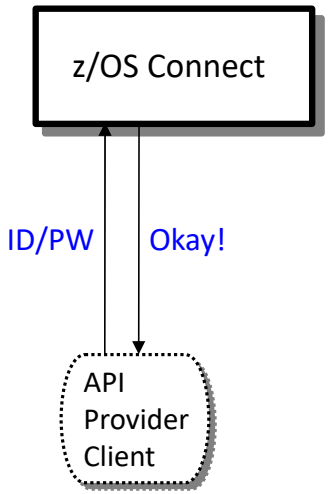
1. A user ID and password can be used for basic authentication by the z/OS Connect EE server
2. Connection between the CICS, IMS, or z/OS application and the z/OS Connect EE server can use TLS
3. Authenticate the CICS, IMS, or z/OS application.
4. Authorize the authenticated user ID to connect to z/OS Connect EE and to perform specific actions on z/OS Connect EE API requesters
5. Pass the user ID and password credentials to an authorization server to obtain a security token.
6. Secure the connection to the external API provider, and provide security credentials such as a security token to be used to invoke the RESTful API
7. The RESTful API runs in the external API provider

mitchj@us.ibm.com

z/OS Application to z/OS Connect API Requester

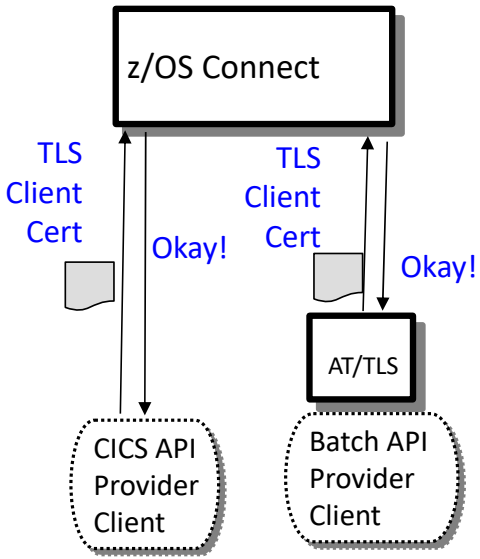
Two options for providing credentials for authentication

Basic Authentication



Application provides ID/PW or ID/PassTicket

Client Certificate

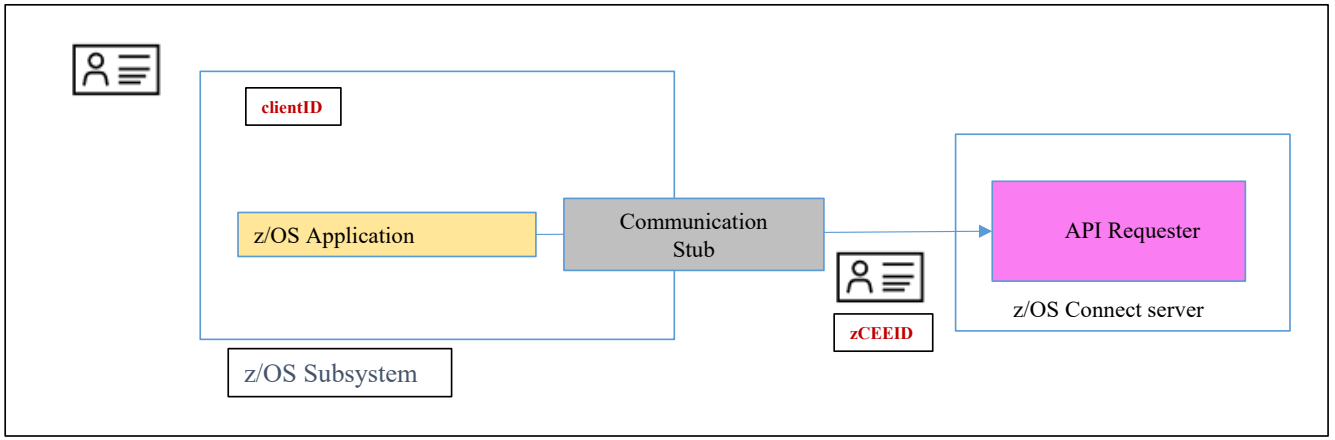


z/OS Connect requests a client certificate

CICS or AT/TLS supplies a client certificate

mitchj@us.ibm.com
mitchj@us.ibm.com

API Requester - basic authentication and identity assertion



zCEEID – The identity that is used for authenticating connectivity the z/OS subsystem to the zCEE server. It is configured using basic authentication or for CICS, TLS client authentication. For MVS batch, IMS and Db2 stored procedures, the *zCEEID* is the provided by the environment variable ***BAQUSERNAME***. For CICS, the value for *zCEEID* is usually provided by the identity mapped to the CICS client certificate.

clientID – the identity under which the z/OS application is executing.

- For CICS, the CICS task identity
- For IMS, the transaction owner
- For batch, the job card USERID

requireAuth	idAssertion	Actions performed by z/OS Connect
true	OFF	Identity assertion is disabled. The zCEE server authenticates *zCEEID* and checks whether *zCEEID* has the authority to invoke an API requester.
	ASSERT_SURROGATE	Identity assertion is enabled. The zCEE server authenticates *zCEEID* and checks whether *zCEEID* is a surrogate of *clientID*. If *zCEEID* is a surrogate of *clientID*, the server further checks whether *clientID* has the authority to invoke an API requester; otherwise, a BAQR7114E message occurs.
	ASSERT_ONLY	Identity assertion is enabled. The zCEE server authenticates *zCEEID* and directly checks whether *clientID* has the authority to invoke an API requester
false	OFF	Identity assertion is disabled. A BAQR0407W message occurs.
	ASSERT_SURROGATE	Identity assertion is enabled. The zCEE server checks whether *clientID* has the authority to invoke an API requester, and a warning message occurs to indicate that the ASSERT_ONLY value is used instead of the ASSERT_SURROGATE value.
	ASSERT_ONLY	Identity assertion is enabled. The zCEE server checks whether *clientID* has the authority to invoke an API requester

```
<zosconnect_apiRequesters idAssertion="OFF">

 <zosconnect_apiRequester name="cscvinc_1.0.0"
           idAssertion="ASSERT_ONLY"> *

 <zosconnect_apiRequester name="db2employee_1.0.0"
           idAssertion="ASSERT_SURROGATE"> *

</zosconnect_apiRequesters>
```

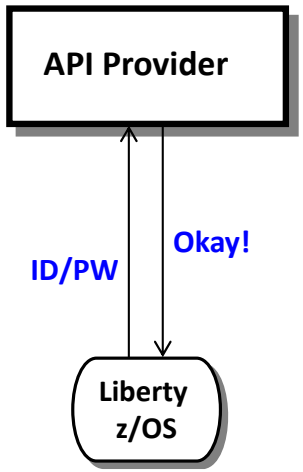
* Added in V3.0.45

mitchj@us.ibm.com

API Requester - API Provider Authentication

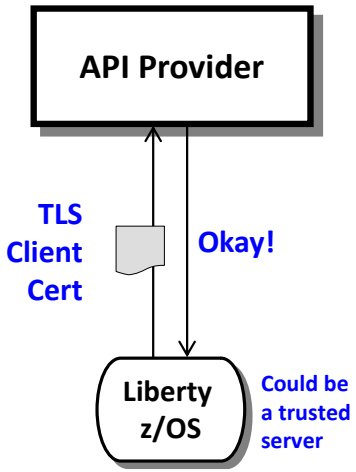
Several different ways this can be accomplished:

Basic Authentication



zCEE supplies ID/PW or ID/Passticket

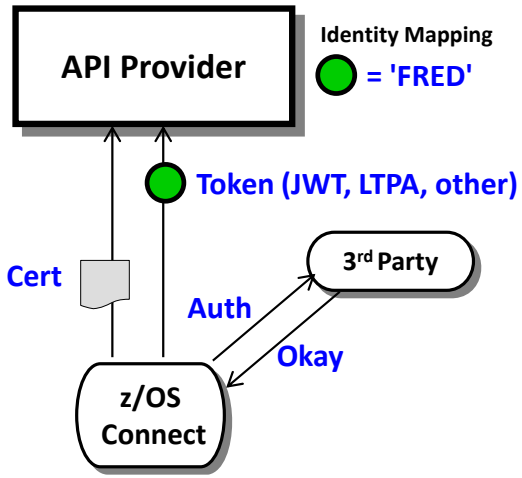
Client Certificate



Server prompts for certificate

zCEE supplies certificate

Third Party Authentication



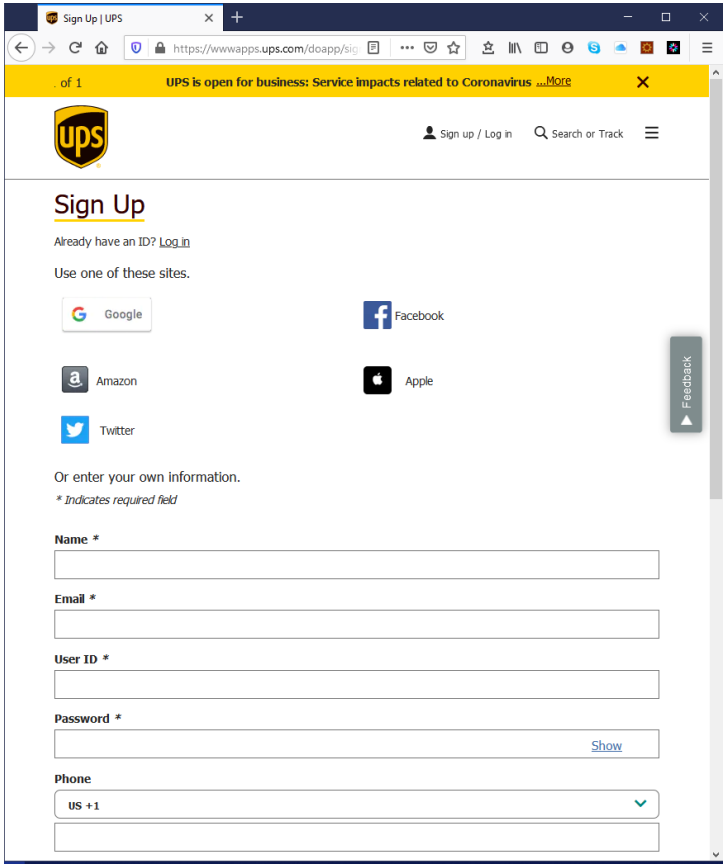
zCEE authenticates to 3rd party sever

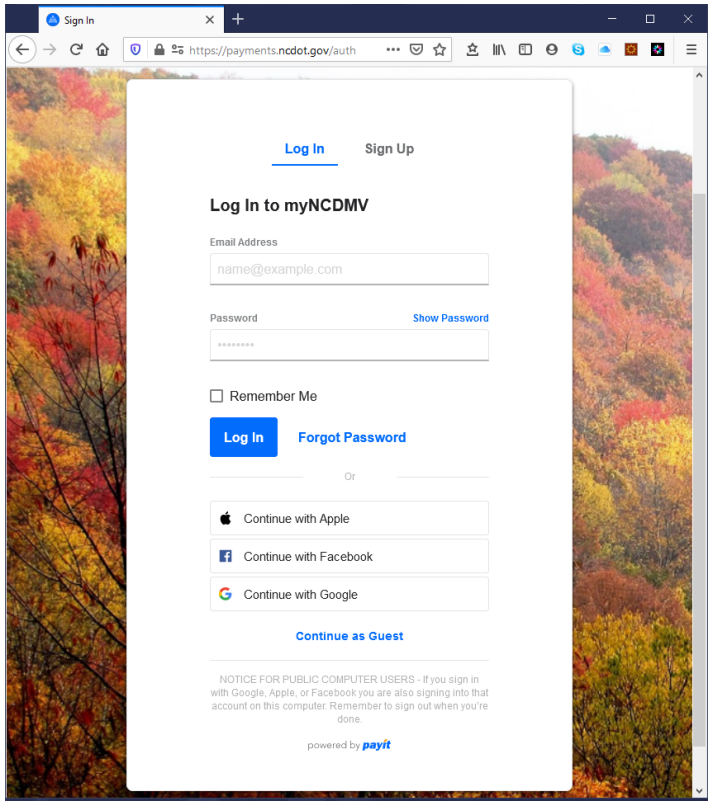
zCEE receives a trusted 3rd party token

Token flows to API Provider

mitchj@us.ibm.com

Third Party Authentication Examples

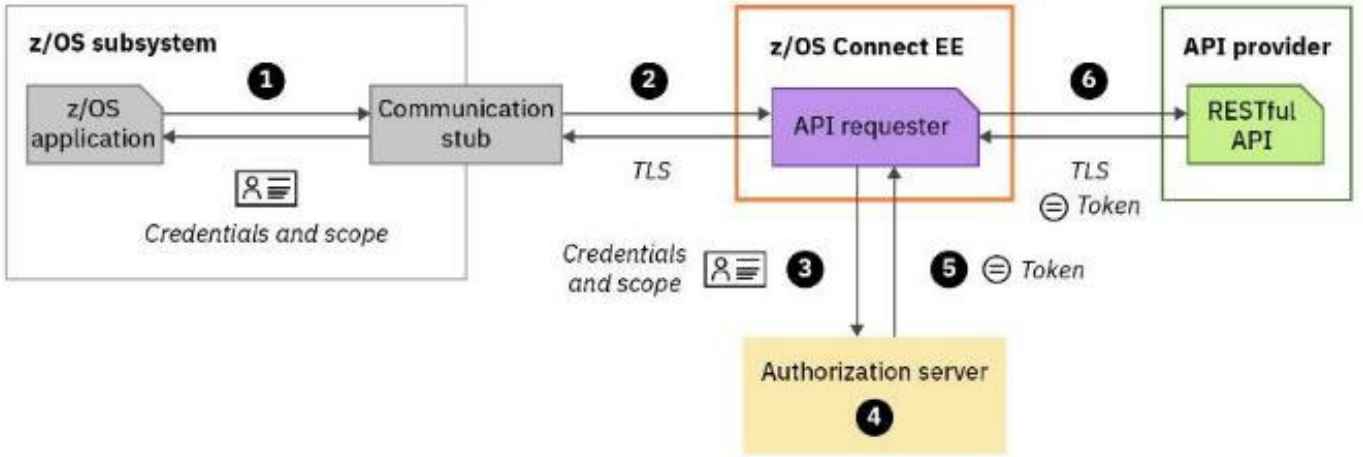




mitchj@us.ibm.com

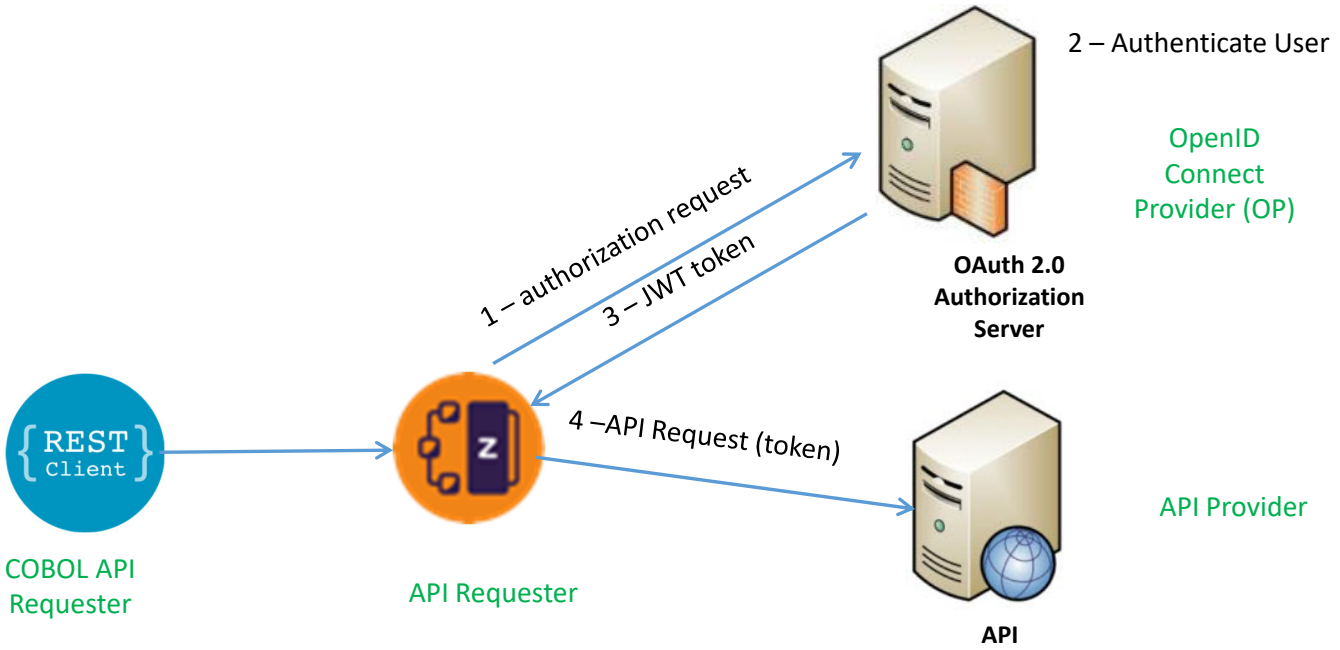
mitchj@us.ibm.com

Calling an API with OAuth 2.0 support



mitchj@us.ibm.com

z/OS Connect OAuth Flow for API requester



Grant Types:

- client_credentials
- password

mitchj@us.ibm.com

OAuth Grant Types Supported by z/OS Connect

client_credentials - *the identity associated with the combination of the CICS, IMS, or z/OS application, and the z/OS Connect EE server that calls the RESTful API on behalf of the CICS, IMS, or z/OS application When this grant type is used, the z/OS Connect EE server sends the client credentials and the access scope to the authorization server.*

```
<zosconnect_oAuthConfig id="myoAuthConfig"
      grantType="client_credentials"
      authServerRef="myoAuthServer"/>
```

password - *The identity of the user of the CICS, IMS, or z/OS application, or it might be another entity. When this grant type is used, the z/OS Connect EE server sends the resource owner's credentials, the client credentials, and the access scope to the authorization server.*

```
<zosconnect_oAuthConfig id="myoAuthConfig"
      grantType="password"
      authServerRef="myoAuthServer"/>
```

mitchj@us.ibm.com

Configuring OAuth support – BAQRINFO copy book

```
05 BAQ-OAUTH.
  07 BAQ-OAUTH-USERNAME            PIC X(256).
  07 BAQ-OAUTH-USERNAME-LEN        PIC S9(9) COMP-5 SYNC VALUE 0.
  07 BAQ-OAUTH-PASSWORD            PIC X(256).
  07 BAQ-OAUTH-PASSWORD-LEN        PIC S9(9) COMP-5 SYNC VALUE 0.
  07 BAQ-OAUTH-CLIENTID            PIC X(256).
  07 BAQ-OAUTH-CLIENTID-LEN        PIC S9(9) COMP-5 SYNC VALUE 0.
  07 BAQ-OAUTH-CLIENT-SECRET       PIC X(256).
  07 BAQ-OAUTH-CLIENT-SECRET-LEN   PIC S9(9) COMP-5 SYNC VALUE 0.
  07 BAQ-OAUTH-SCOPE-PTR           USAGE POINTER.
  07 BAQ-OAUTH-SCOPE-LEN           PIC S9(9) COMP-5 SYNC.
```

Grant Type: *client_credentials* - the identity associated with the combination of the CICS, IMS, or z/OS application, and the z/OS Connect EE server that calls the RESTful API on behalf of the CICS, IMS, or z/OS application

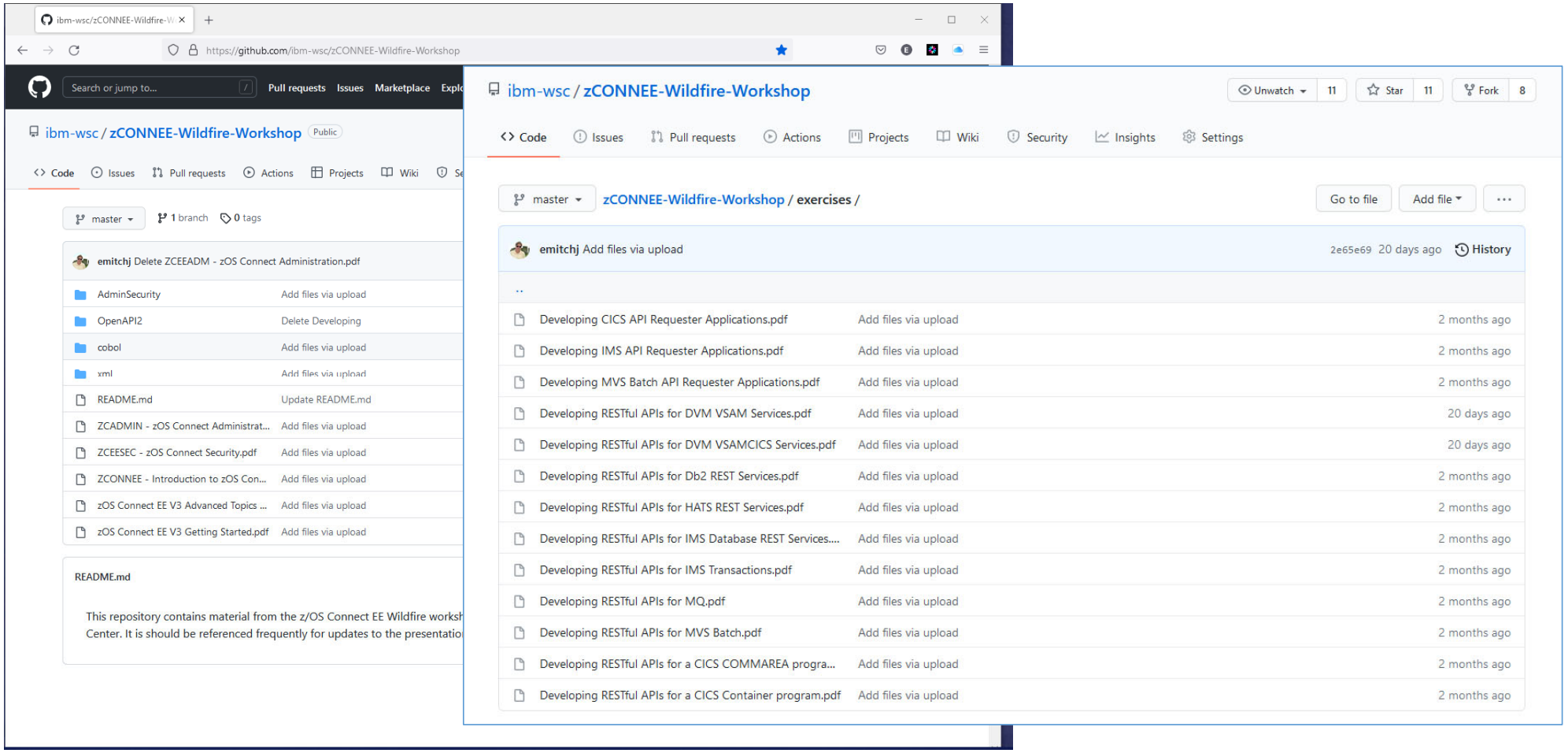
Grant Type: *password* - The identity of the user provided by the CICS, IMS, or z/OS application, or it might be another entity. Client_credentials can be supplied by the program or in the server XML configuration.

Scope is always required.

OAuth 2.0 specification entity	password	client_credentials	Where Set
Client ID	required	Required	server.xml or by application
Client Secret	optional	Required	server.xml or by application
Username	required	N/A	by application
Password	required	N/A	by application

mitchj@us.ibm.com

z/OS Connect Wildfire Github Site https://ibm.biz/BdPRGD



- **Contact your IBM representative to schedule access to these exercises**

mitchj@us.ibm.com

Thank you for listening and your questions.

mitchj@us.ibm.com