



IBM z/OS Connect

API Requester Introduction and Overview

Mitch Johnson

mitchj@us.ibm.com

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 be 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 to 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.



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

The screenshot shows a GitHub repository page for 'ibm-wsc/zCONNEE-Wildfire-Workshop'. The 'Code' tab is selected. On the left sidebar, there is a list of files and folders:

- master (branch)
- Admin_Security
- OpenAPI2 (highlighted with a red oval)
- zlabel
- xml
- README.md
- ZADMIN - zOS Connect Administrat...
- ZCESEC - zOS Connect Security.pdf
- ZCONNEE - Introduction to zOS Con...
- zOS Connect EE V3 Advanced Topics ...
- zOS Connect EE V3 Getting Started.pdf

Below the sidebar, there is a 'README.md' section containing a note about the repository's purpose.

The main content area shows the 'exercises' directory. It contains a single commit by 'emitchj' titled 'Add files via upload' (commit hash: 2e65e69, 20 days ago). This commit includes several PDF files:

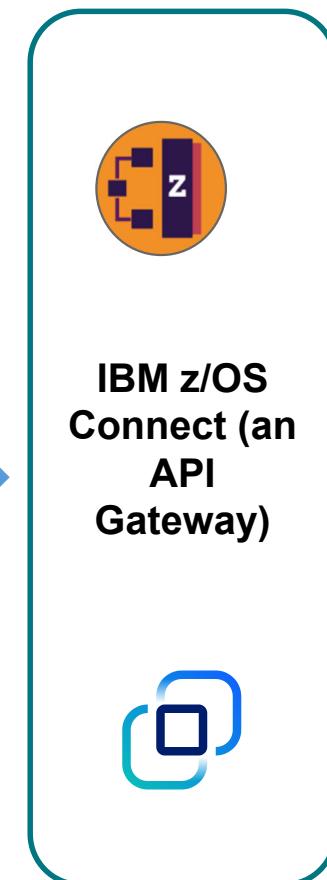
- Developing CICS API Requester Applications.pdf
- Developing IMS API Requester Applications.pdf
- Developing MVS Batch API Requester Applications.pdf
- Developing RESTful APIs for DVM VSAM/CICS Services.pdf
- Developing RESTful APIs for Db2 REST Services.pdf
- Developing RESTful APIs for HATS REST Services.pdf
- Developing RESTful APIs for IMS Database REST Services....
- Developing RESTful APIs for IMS Transactions.pdf
- Developing RESTful APIs for MQ.pdf
- Developing RESTful APIs for MVS Batch.pdf
- Developing RESTful APIs for a CICS COMMAREA progra...
- Developing RESTful APIs for a CICS Container program.pdf

- Contact your IBM representative to schedule access to these exercises

© 2018, 2022 IBM Corporation

mitchj@us.ibm.com

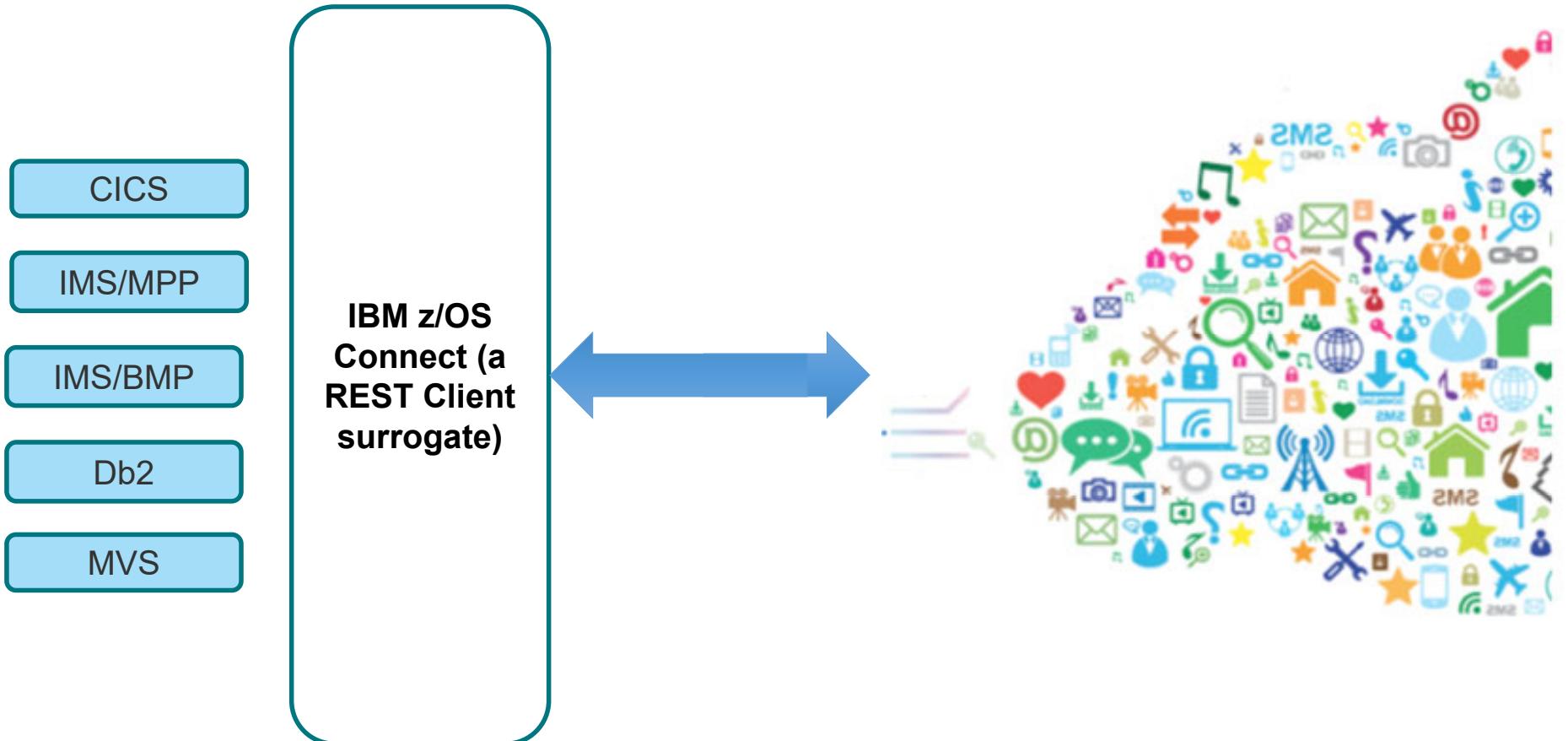
z/OS Connect EE exposes z/OS resources to the “cloud” via RESTful APIs



+ HCL and Rocket Software

*Other Vendors or your own implementation

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



/but_first, what_is_REST?

What makes an API “RESTful”?



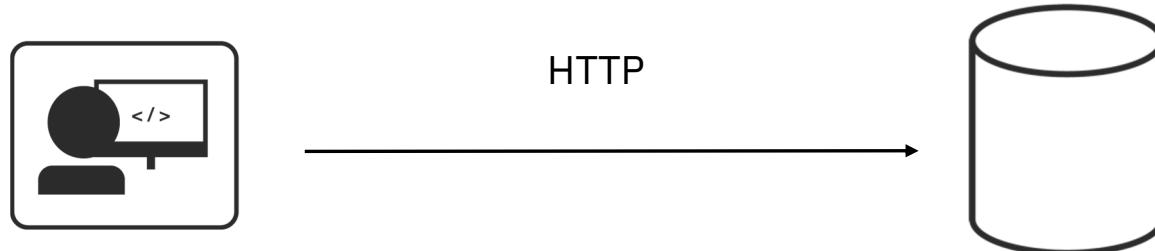
REST is architectural programming style

REST stands for **R**epresentational **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.



Key Principles of REST

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

POST
GET
PUT
DELETE

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

Use Path and Query parameters to refine the request

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" }
```



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



RESTful Examples

POST /account/ +  (*a JSON request message with Fred's information*)

GET /account?number=1234

PUT /account/1234 +  (*a JSON request message with dollar amount of deposit*)

HTTP Verb conveys the method against the resources; i.e., POST is for create, GET is for balance, etc.

URI conveys the resource to be acted upon; i.e., Fred's account with number 1234

The JSON body carries the specific data for the action (verb) against the resource (URI)

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>



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" }



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
```



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.

How do we describe a REST API?



/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.



Why use OPENAPI?

- It is more than just an API framework

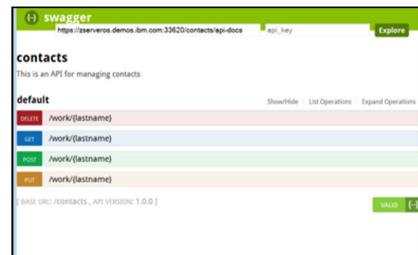


There are a number of tools available to aid consumption:

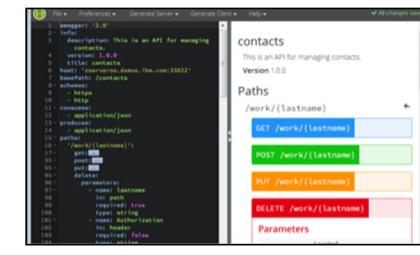
Code Generation* - create stub code to consume APIs from various languages



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 specifications.



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",
          }
        }
      }
    }
  }
}
```

```
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/{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

```

YAML
Format*

Tech-Tip: Swagger Example



The image displays two side-by-side browser windows, each showing a JSON-based Swagger API definition. The left window shows the full API structure, while the right window provides a detailed view of a specific request schema.

Left Window (Overall API Structure):

```
swagger: "2.0"
info:
  description: ""
  version: "1.0.0"
  title: "miniloan"
  basePath: "/miniloan"
schemes:
  0: "https"
  1: "http"
consumes:
  0: "application/json"
produces:
  0: "application/json"
path:
  /loan:
    post:
      tags:
        0:
          operationId: "postMiniloanService"
          parameters:
            0:
              name: "Authorization"
              in: "header"
              required: false
              type: "string"
            1:
              in: "body"
              name: "postMiniloanService_request"
              description: "request body"
              required: true
              schema:
                $ref: "#/definitions/postMiniloanService_request"
      responses:
        200:
          description: "OK"
          schema:
```

Right Window (Detailed Request Schema):

```
schema:
  $ref: "#/definitions/postMiniloanService_response_200"
definitions:
  postMiniloanService_request:
    type: "object"
    properties:
      MINILOAN_COMMAREA:
        type: "object"
        properties:
          name:
            type: "string"
            maxLength: 20
          creditscore:
            type: "integer"
            minimum: 0
            maximum: 10000000000000000000
          yearlyIncome:
            type: "integer"
            minimum: 0
            maximum: 10000000000000000000
          age:
            type: "integer"
            minimum: 0
            maximum: 9999999999
          amount:
            type: "integer"
            minimum: 0
            maximum: 10000000000000000000
          effectiveDate:
            type: "string"
            maxLength: 8
          yearlyRepayment:
            type: "integer"
            minimum: 0
            maximum: 10000000000000000000
      postMiniloanService_response_200:
        type: "object"
```

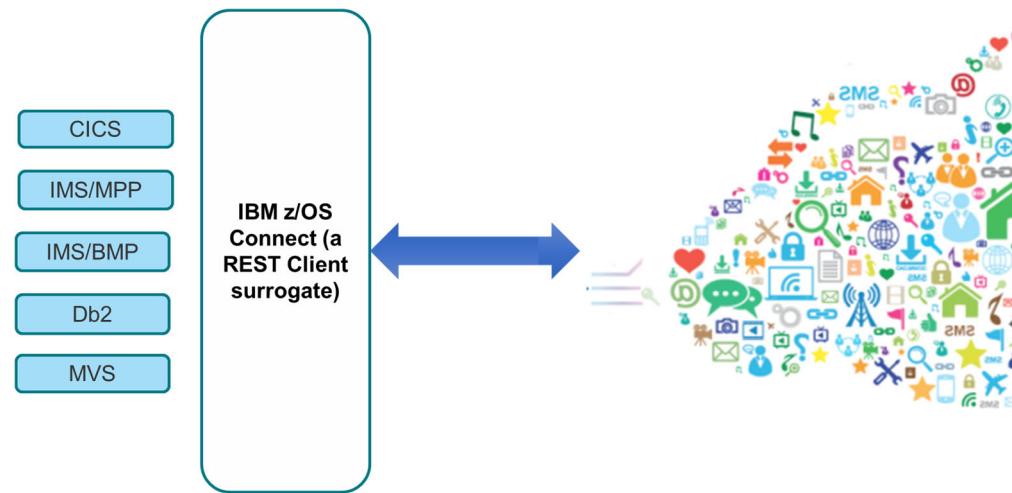
mitchj@us.ibm.com

© 2018, 2022 IBM Corporation



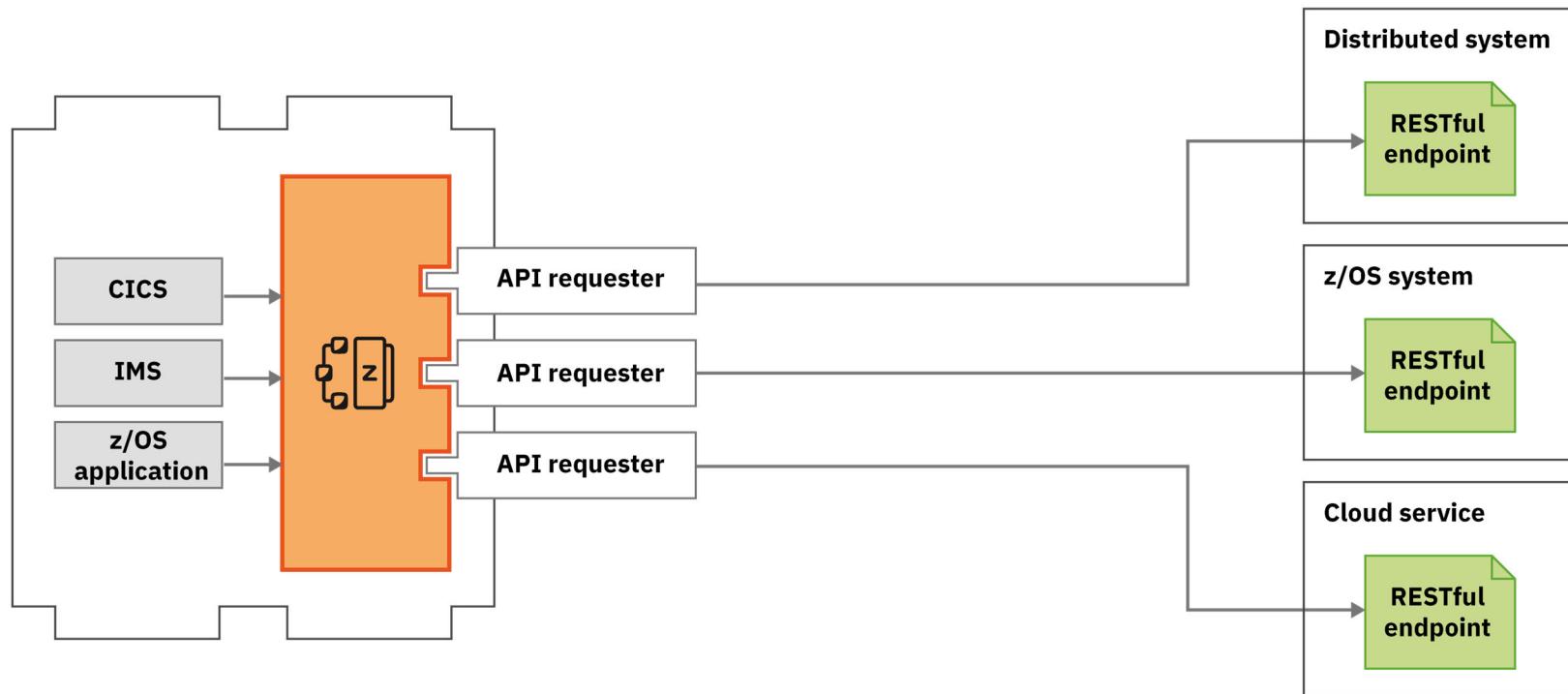
/api_toolkit/apiRequesters

Quick and easy API mapping.





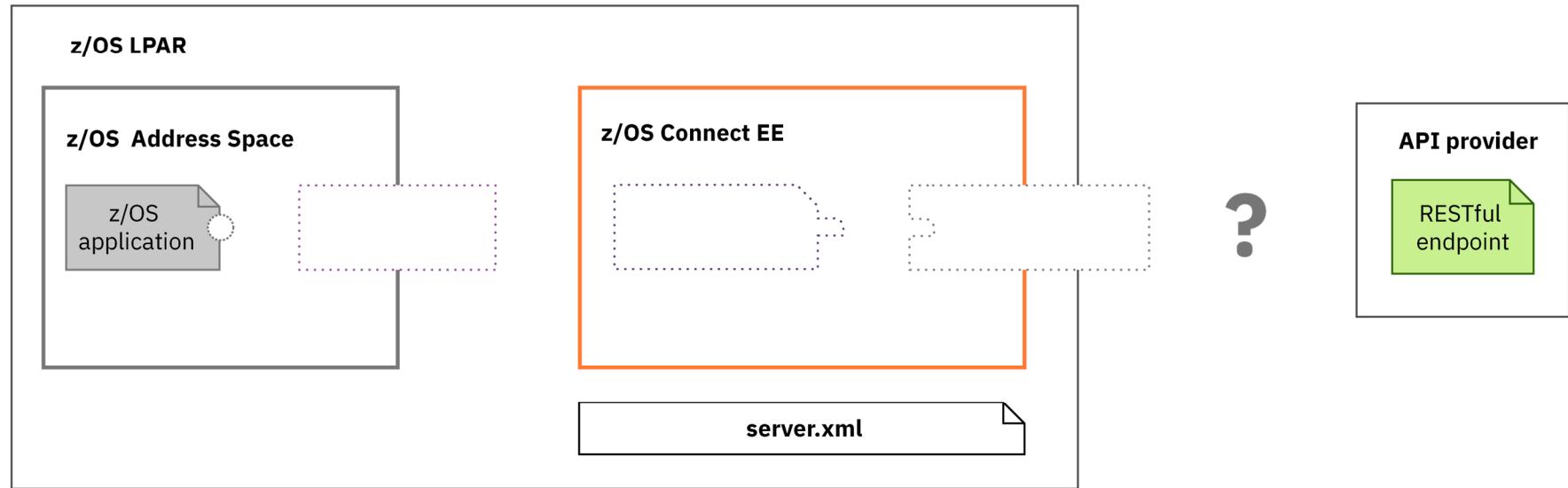
Use API requester to call external APIs from z/OS assets





Steps to calling an external API

Starting point



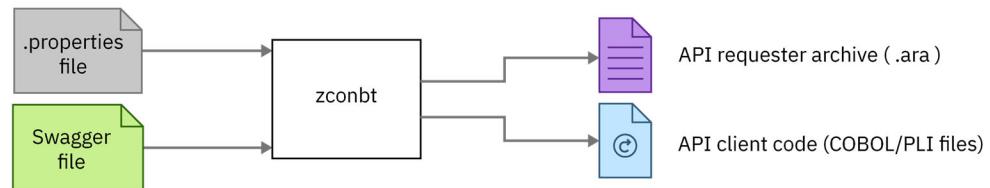


Steps to calling an external API

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

The screenshot shows a JSON editor window with the file path /C:/z/apiRequester/cscvinc/cscvinc.json. The JSON content is a Swagger specification for an API named 'cscvincapi'. It includes definitions for paths, parameters, and responses.

```
swagger: "2.0"
info:
  description: ""
  version: "1.0.0"
  title: "cscvincapi"
  basePath: "/"
schemes: []
consumes:
  0: "application/json"
produces:
  0: "application/json"
paths:
  /employee/{employee}:
    get:
      tags:
        0: []
      operationId: "getCscvincSelectService"
      parameters:
        0:
          name: "employee"
          in: "path"
          required: true
          type: "string"
          maxLength: 6
      responses:
        200:
          description: "OK"
          schema: {}
        404:
      post:
      put:
      delete:
    definitions: {}
```



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>



COBOL Storage Considerations

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

```
/C:/z/apiRequester/ATS/ATSContactX +  
file:///C:/z/apiRequester/ATS/ATSContactPreferences  
JSON Raw Data Headers  
Save Copy Collapse All Expand All Filter JSON  
maxItems: 10  
communicationPreferences:  
  items:  
    $ref: "#/definitions/member-communication-preferences"  
    type: "array"  
memberCodeableConcept:  
  description: "Multiple member codes"  
  items:  
    $ref: "#/definitions/member-codeable-concept"  
    type: "array"  
    type: "object"  
member-contacts-request:  
  title: "Member Contacts Request"  
  description: "Read-only request data to search for member contact information."  
  properties:  
    umi:  
      description: "Unique Member Id. This value is at a contract level. All members under one contract have the same UMI."  
      example: "122222444001"  
      type: "string"  
    firstName:  
      description: "Member first name or given name."  
      example: "Arthur"  
      type: "string"  
    lastName:  
      description: "Member last name or family name."  
      example: "Smith"  
      type: "string"  
    birthDate:  
      description: "Member date of birth in the format mm/dd/yyyy."  
      example: "12/19/2019"  
      type: "string"
```

mitchj@us.ibm.com

```
ATS01P01 - Notepad  
File Edit Format View Help  
* ++++++  
06 RespBody.  
09 memberContactsResponse-num PIC S9(9) COMP-5 SYNC.  
09 memberContactResponse OCCURS 255.  
12 umi-num PIC S9(9) COMP-5 SYNC.  
12 umi.  
 15 umi2-length  
 15 umi2 PIC X(255). SYNC.  
12 pin-num PIC S9(9) COMP-5 SYNC.  
12 pin.  
 15 pin2-length  
 15 pin2 PIC X(255). SYNC.  
12 firstName-num PIC S9(9) COMP-5 SYNC.  
12 firstName.  
 15 firstName2-length  
 15 firstName2 PIC X(255). SYNC.  
12 middleName-num PIC S9(9) COMP-5 SYNC.  
12 middleName.  
 15 middleName2-length  
 15 middleName2 PIC X(255). SYNC.  
12 lastName-num PIC S9(9) COMP-5 SYNC.  
12 lastName.  
 15 lastName2-length  
 15 lastName2 PIC X(255). SYNC.
```

© 2018, 2022 IBM Corporation

Page 23



Add constraints to the properties in the specification

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

```
JSON Raw Data Headers
Save Copy Collapse All Expand All Filter JSON
  type: "array"
    items:
      $ref: "#/definitions/member-communication-preferences"
      type: "array"
      maxItems: 10
      memberCodeableConcept:
        description: "Multiple member codes"
      items:
        $ref: "#/definitions/member-codeable-concept"
        type: "object"
      type: "object"
    member-contacts-request:
      title: "Member Contacts Request"
      description: "Read-only request data to search for member contact information."
      properties:
        umi:
          description: "Unique Member Id. This value is at a contract level. All members under one contract have the same UMI."
          example: "112222444001"
          type: "string"
          maxLength: 12
        firstName:
          description: "Member first name or given name."
          example: "Arthur"
          type: "string"
          maxLength: 30
        lastName:
          description: "Member last name or family name."
          example: "Smith"
          type: "string"
          maxLength: 30
```

```
ATS01P01 - Notepad
File Edit Format View Help
  * Comments for field 'filler':
  * This is a filler entry to ensure the correct padding for a
  * structure. These slack bytes do not contain any application
  * data.
  *      15 filler          PIC X(3).
  *
  *
  * ++++++
 06 RespBody.

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

 09 memberContactsResponse OCCURS 10.

 12 umi-num          PIC S9(9) COMP-5           SYNC.
 12 umi,
 15 umi2-length          PIC S9999 COMP-5           SYNC.
 15 umi2          PIC X(12).

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

 12 pin.
 15 pin2-length          PIC S9999 COMP-5           SYNC.
 15 pin2          PIC X(255).

 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 middleName-num          PIC S9(9) COMP-5           SYNC.

 12 middleName.
 15 middleName2-length          PIC S9999 COMP-5           SYNC.
 15 middleName2          PIC X(30).

Ln 783, Col 70 100% Windows (CRLF) UTF-8
```



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 15 firstName2	PIC S9999 COMP-5 PIC X(30).	

12 firstName-num	PIC S9(9) COMP-5	SYNC.
12 firstName	PIC X(31).	



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 . . .

```
version: "1.2.8"
title: "NewMembers"
contact:
  name:
securityDefinitions:
  apiKeyHeader:
    type: "apiKey"
    name: "X-IBM-Client-ID"
    in: "header"
  host: "wg31.washington.ibm.com"
  basePath: "/v1"
schemes:
  0: "https"
paths:
  /nms/members/search:
    post:
```



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.

```
cscvinc.properties - Notepad
File Edit Format View Help
apiKeyDescriptionFile=../cscvinc.json
dataStructuresLocation=../syslib
apiInfoFileLocation=../syslib
logFileDirectory=../logs
language=COBOL
connectionRef=cscvincAPI
requesterPrefix=ats
apiKeyMaxLength=40
apiKeyParmNameInHeader=X-IBM-Client-ID

Ln 8, Col 19 100% Unix (LF) UTF-8
```



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

The 'request.cpy' file contains the following COBOL definitions:

```
*      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 PIC S9999 COMP-5 SYNC.
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     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            PIC S9999 COMP-5 SYNC.
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.
09 X-HZN-UserAssociationI    PIC X(255).
```

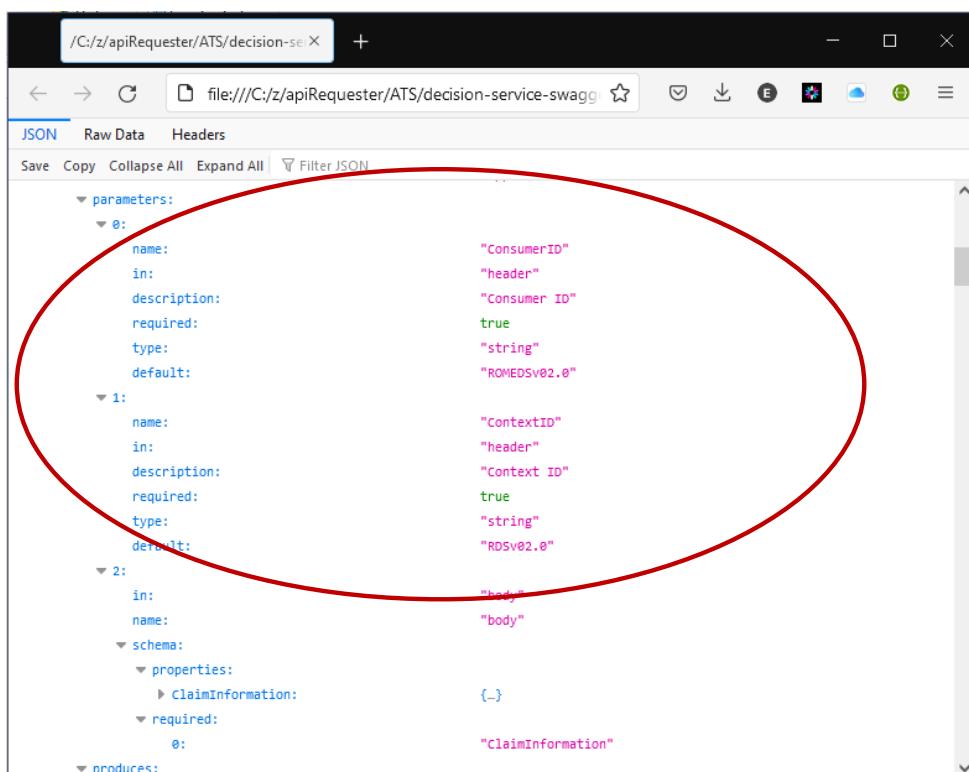
The 'mpz3' session shows the following AS/400 source code:

```
EDIT      USER1.ZCEE.SOURCE(GETAPIEN) - 01.01
Command ==> *-----
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
000092
000093
000094    MOVE employee of PARM-DATA TO employee IN GET-REQUEST.
000095    MOVE LENGTH of employee in GET-REQUEST to
000096      employee-length IN GET-REQUEST.
000097
000098
000099
000100
000101
000102
000103    *----- * Initialize API Requester PTRs & LENs
000104    *----- *
000105    * Use pointer and length to specify the location of
000106    * request and response segment.
000107    * This procedure is general and necessary.
000108    SET BAQ-REQUEST-PTR TO ADDRESS OF GET-REQUEST.
000109    MOVE LENGTH OF GET-REQUEST TO BAQ-REQUEST-LEN.
000110    SET BAQ-RESPONSE-PTR TO ADDRESS OF GET-RESPONSE.
000111    MOVE LENGTH OF GET-RESPONSE TO BAQ-RESPONSE-LEN.
```

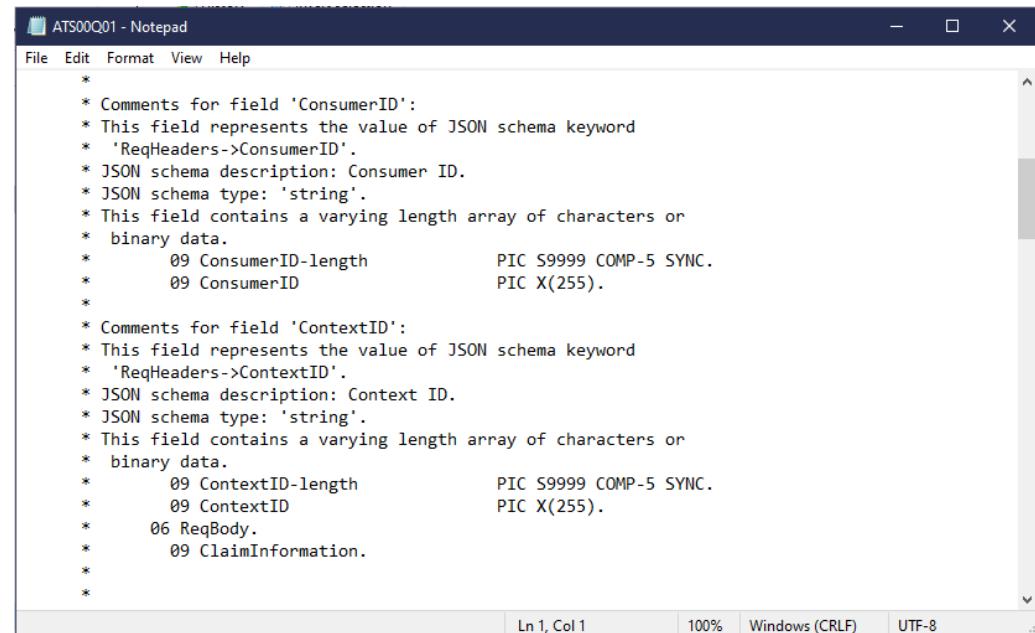


Additional Swagger header properties

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



```
/C:/apiRequester/ATS/decision-se X +  
file:///C:/apiRequester/ATS/decision-service-swagg  
JSON Raw Data Headers  
Save Copy Collapse All Expand All Filter JSON  
parameters:  
  0:  
    name: "ConsumerID"  
    in: "header"  
    description: "Consumer ID"  
    required: true  
    type: "string"  
    default: "ROMEDSv2.0"  
  1:  
    name: "ContextID"  
    in: "header"  
    description: "Context ID"  
    required: true  
    type: "string"  
    default: "RDSv02.0"  
  2:  
    in: "body"  
    name: "body"  
schema:  
  properties:  
    ClaimInformation: (...)  
  required:  
    0: "ClaimInformation"  
produces:
```



```
ATS00Q01 - Notepad  
File Edit Format View Help  
*  
* Comments for field 'ConsumerID':  
* This field represents the value of JSON schema keyword  
* 'ReqHeaders->ConsumerID'.  
* JSON schema description: Consumer ID.  
* JSON schema type: 'string'.  
* This field contains a varying length array of characters or  
* binary data.  
* 09 ConsumerID-length PIC S9999 COMP-5 SYNC.  
* 09 ConsumerID PIC X(255).  
*  
* Comments for field 'ContextID':  
* This field represents the value of JSON schema keyword  
* 'ReqHeaders->ContextID'.  
* JSON schema description: Context ID.  
* JSON schema type: 'string'.  
* This field contains a varying length array of characters or  
* binary data.  
* 09 ContextID-length PIC S9999 COMP-5 SYNC.  
* 09 ContextID PIC X(255).  
* 06 ReqBody.  
* 09 ClaimInformation.  
*  
*  
Ln 1, Col 1 100% Windows (CRLF) UTF-8
```



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: getCsvincSelectService, basePath: /cscvincapi, relativePath: /employee/{employee}, method: GET
- request data structure : CSC00Q01
- response data structure : CSC00P01
- api info file : CSC00I01

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

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

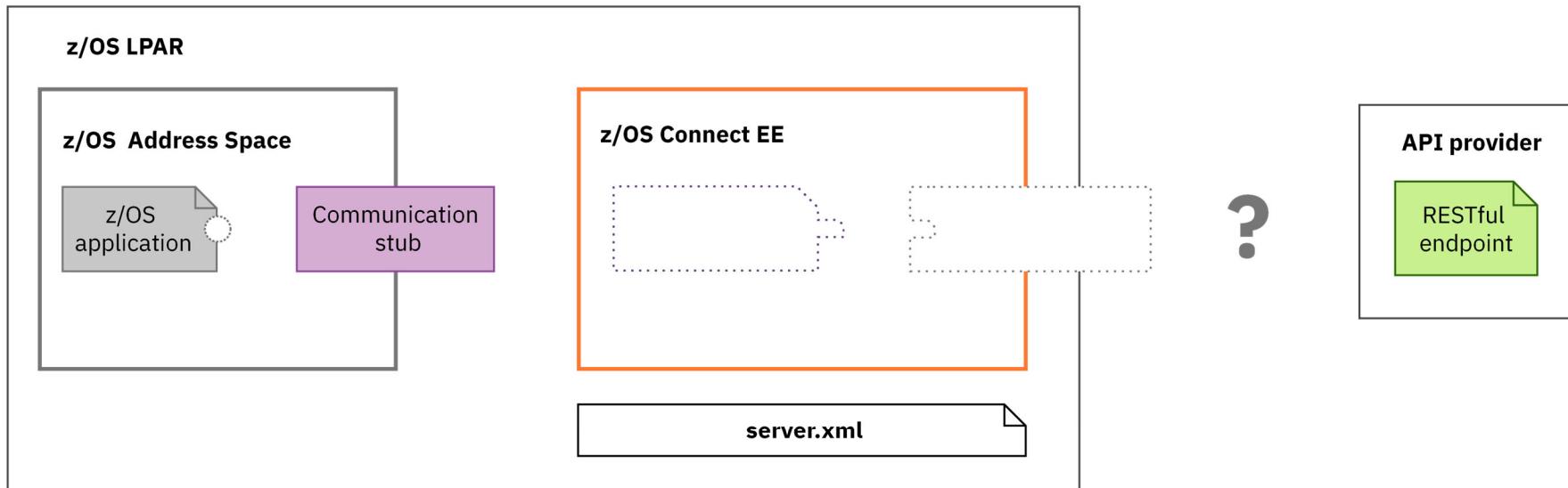
operationId: deleteCsvincDeleteService, basePath: /cscvincapi, relativePath: /employee/{employee}, method: DELETE
- request data structure : CSC03Q01
- response data structure : CSC03P01
- api info file : CSC03I01

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



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

i ibm.biz/zosconnect-configure-comms-stub

Steps to calling an external API



Include the generated copy books in a COBOL program

API-REQUEST

```
CSC00101 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

```
*  
* ++++++  
  
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.  
  
    17 CONTINUE-CONTAINER
```

API-INFO-OPER1

```
CSC00101 X
 03 BAQ-APINAME          PIC X(255)
    VALUE 'csvincapi_1.0.0'.
 03 BAQ-APINAME-LEN      PIC S9(9) COMP-5 SYNC
    VALUE 16.
 03 BAQ-APIPATH          PIC X(255)
    VALUE '%2fcsvincapi%2Femployee%2F%Employee%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.
```



Steps to calling an external API

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

The diagram illustrates the steps to calling an external API. It shows the GETAPI program (left) interacting with the CSC00101 copy book (right) through several windows:

- GETAPI**: The main program window showing assembly code. Red boxes highlight sections:
 - * Set up the data for the API Requester call
 - * Initialize API Requester PTRs & LENs
 - * Call the communication stub
 - * Call the subsystem-supplied stub code to send API request to zCEE
 - CALL COMM-STUB-PGM-NAME USING
 BY REFERENCE API-INFO-OPER1
 BY REFERENCE BAQ-REQUEST-INFO
 BY REFERENCE BAQ-REQUEST-PTR
 BY REFERENCE BAQ-REQUEST-LEN
 BY REFERENCE BAQ-RESPONSE-INFO
 BY REFERENCE BAQ-RESPONSE-PTR
 BY REFERENCE BAQ-RESPONSE-LEN.
- CSC00101**: The main copy book window showing the structure of the API call.
 - 03 BAQ-APINAME VALUE 'cscvincap1_1.0.0'. PIC X(255)
 - 03 BAQ-APINAME-LEN VALUE 16. PIC S9(9) COMP-5 SYNC
 - 03 BAQ-APIPATH VALUE 'S2fcsvincap1%2Femployee%7D'. PIC X(255)
 - 03 BAQ-APIPATH-LEN VALUE 41. PIC S9(9) COMP-5 SYNC
 - 03 BAQ-APIMETHOD VALUE 'GET'. PIC X(255)
 - 03 BAQ-APIMETHOD-LEN VALUE 3. PIC S9(9) COMP-5 SYNC
- CSC00Q01**: A sub-copy book window showing the JSON schema for the employee array.
 - * 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).
- CSC00P01**: Another sub-copy book window showing the response parameters.
 - 06 ReqPathParameters.
 - 09 employee-length PIC S9999 COMP-5 SYNC.
 - 09 employee PIC X(6).
- Bottom Window**: A fourth window showing the final response structure.
 - 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.



Steps to calling an external API

Access the results

```
GETAPI X
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

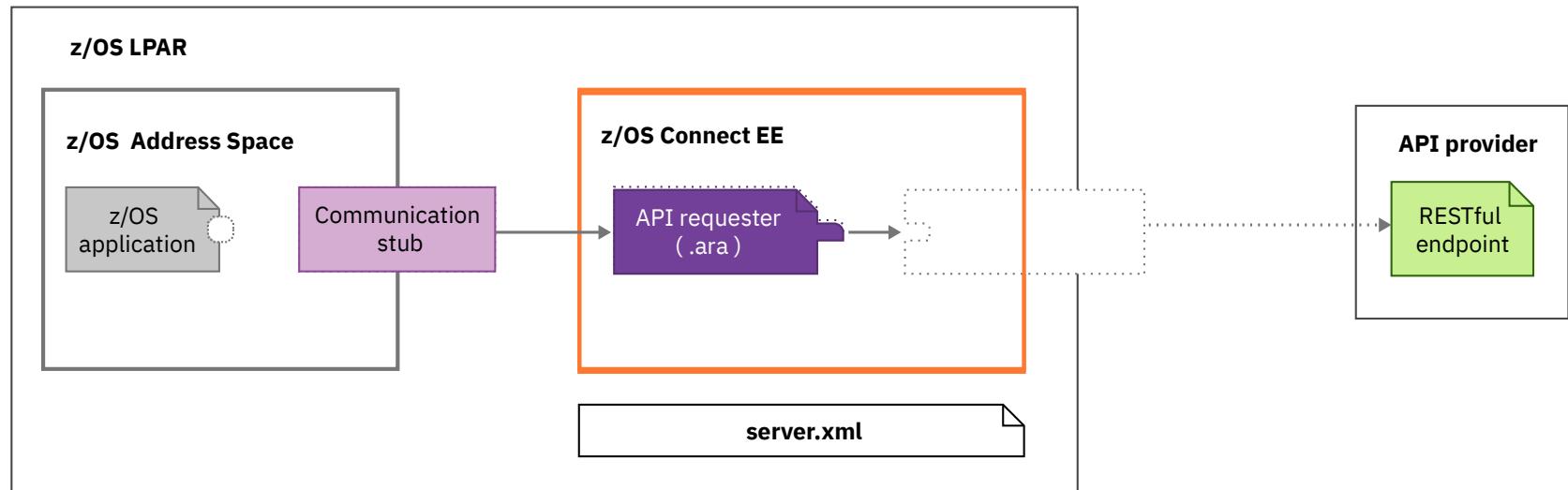
* Otherwise, some error happened in API, z/OS Connect EE server
* or communication stub. 'BAQ-STATUS-CODE' and
* 'BAQ-STATUS-MESSAGE' contain the detailed information
* of this error.
ELSE
    DISPLAY "Error code: " BAQ-STATUS-CODE
    DISPLAY "Error msg: " BAQ-STATUS-MESSAGE
    MOVE BAQ-STATUS-CODE 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.
    LINES BAQ-ERROR-IN-API
```

```
mpz3
File Edit Settings View Communication Actions Window Help
Menu Utilities Compilers Help
BROWSE ZCEE30.SBAQC0B(BAQRINFO)
Command ==>
Line 000000066 Col 001 080
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 ****
18/058
Connected to remote server/host mpz3 using lu/pool MPZ30021 and port 23
```



Steps to calling an external API

Deploy API requester (.ara) archive



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



Deploying API requester archive files

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

The screenshot shows the Postman application interface. A POST request is being made to the URL <https://wg31.washington.ibm.com:9483/zosConnect/apiRequesters>. The request body is set to 'binary' and contains the file 'filea.ara'. The response status is 201 Created, with a time of 863 ms and a size of 380 B. The response body is displayed in JSON format:

```
1
2   "name": "filea_2.0.0",
3   "version": "2.0.0",
4   "description": "",
5   "status": "Started",
6   "apiRequesterUrl": "https://wg31.washington.ibm.com:9483/zosConnect/apiRequesters/filea\_2.0.0",
7   "connection": "fileaAPI"
8 }
```

Command:

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

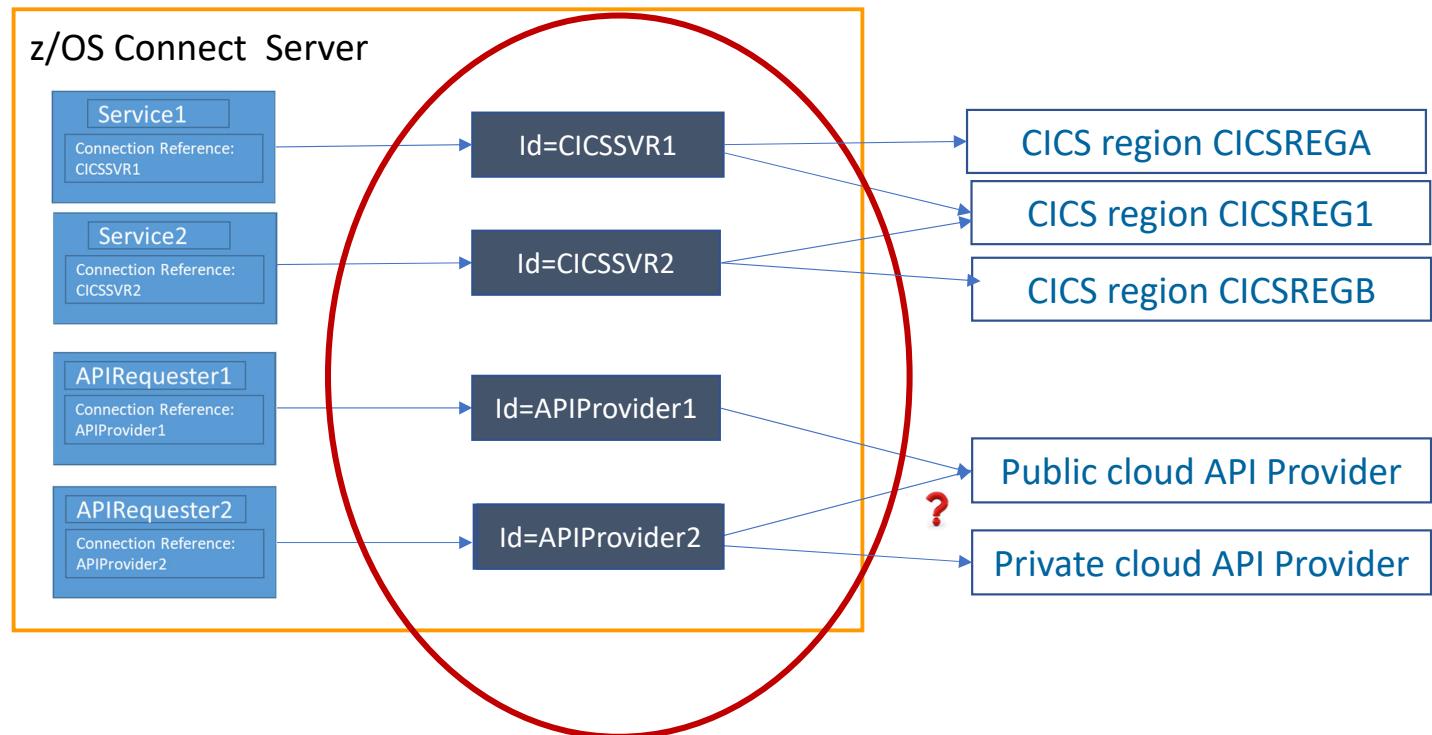
Results:

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



Use naming conventions for connection references

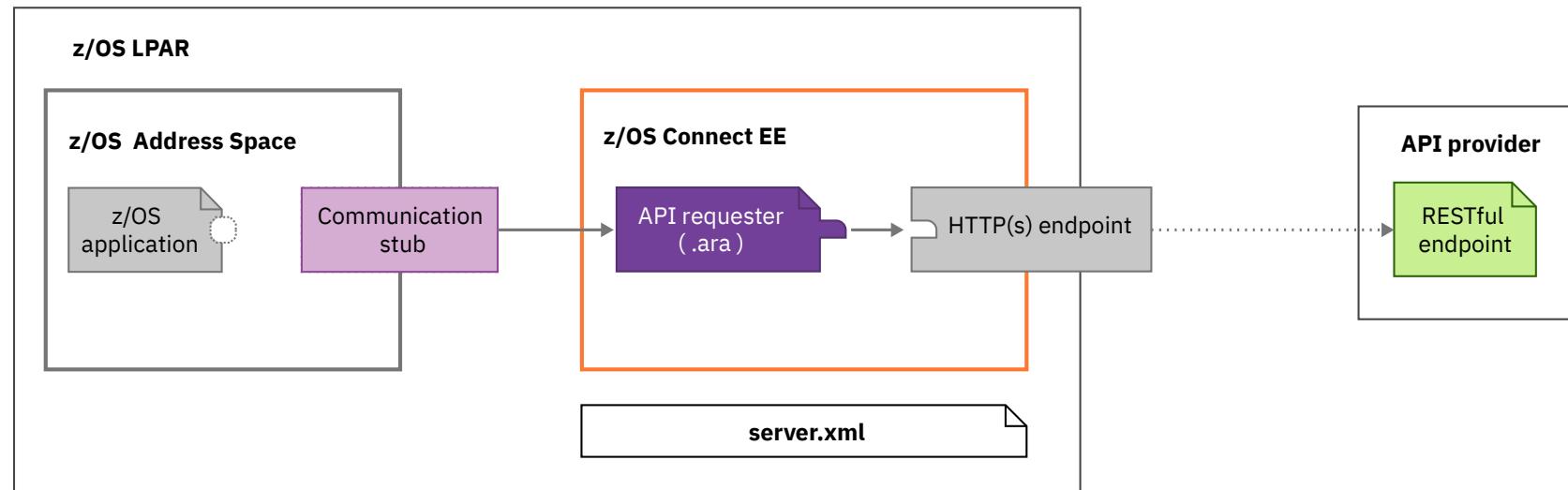
Use application meaningful names or an extendable convention for connection reference names





Steps to calling an external API

Configure HTTP(S) endpoint configuration element



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

i ibm.biz/zosconnect-configure-endpoint-connection

Steps to calling an external API



Update the server XML configuration for the endpoint

```
File Edit View History Bookmarks Tools Help
/C:/z/apiRequester/cscvinc/swagger + 
← → ⌂ file:///C:/z/apiRequester/cscvinc/swagger.json ⌂ ⌂ ⌂
JSON Raw Data Headers
Save Copy Collapse All Expand All Filter JSON
swagger: "2.0"
info:
  description: ""
  version: "1.0.0"
  title: "cscvinc"
  host: "localhost:8080"
  basePath: "/cscvinc"
schemes:
  @: "https"
  #: "http"
consumes:
  @: "application/json"
produces:
  @: "application/json"
paths:
  /employee:
    post:
```

```
| 03 BAQ-APINAME          PIC X(255)
|   VALUE 'cscvinc_1.0.0'.
| 03 BAQ-APINAME-LEN      PIC S9(9) COMP-5 SYNC
|   VALUE 13.
| 03 BAQ-APIPATH          PIC X(255)
|   VALUE '/cscvinc/employee/{numb}'.
| 03 BAQ-APIPATH-LEN      PIC S9(9) COMP-5 SYNC
|   VALUE 24.
| 03 BAQ-APIMETHOD        PIC X(255)
|   VALUE 'GET'.
| 03 BAQ-APIMETHOD-LEN    PIC S9(9) COMP-5 SYNC
|   VALUE 3.
```

cscvinc.properties
connectionRef=cscvincAPI

connectionRef=cscvincAPI

Server Config

apiRequesterHTTPS.xml Read only Close

Design Source

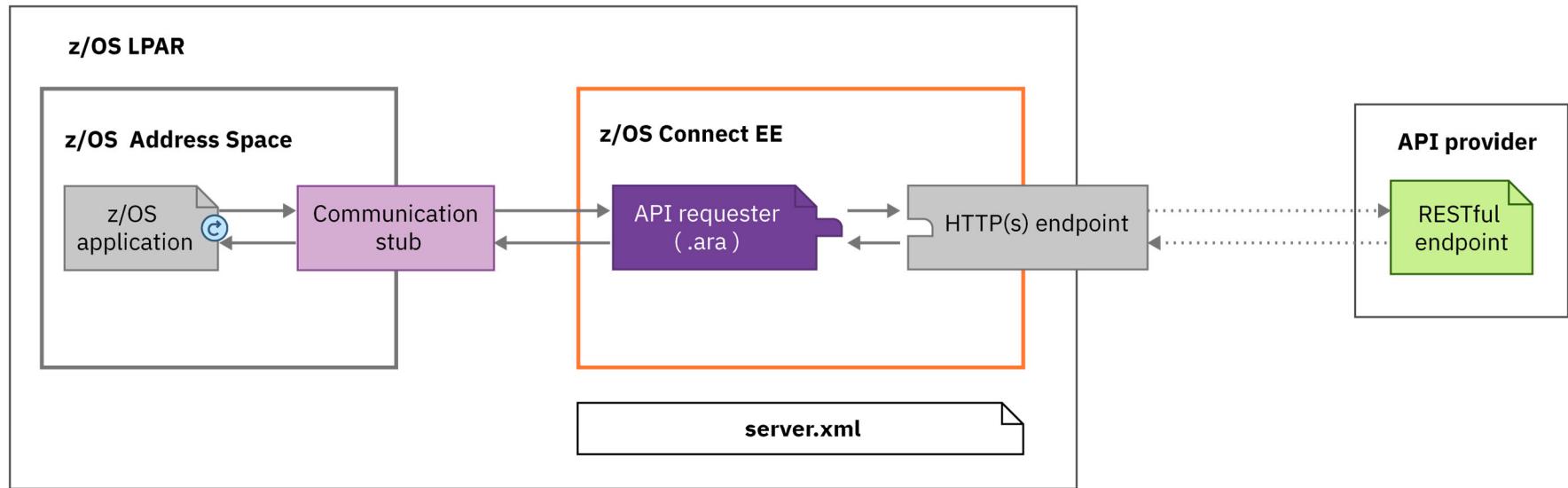
```
30 <!-->
31
32 <zosconnect_endpointConnection id="cscvincAPI"
33     host="https://dvipa.washington.ibm.com"
34     port="9443"
35     authenticationConfigRef="mySAFAuth"
36     connectionTimeout="10s"
37     receiveTimeout="40s" />
38
39 <zosconnect_authData id="mySAFAuth"
40     user="USER1"
41     password="user1" />
42</server>
43
```

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



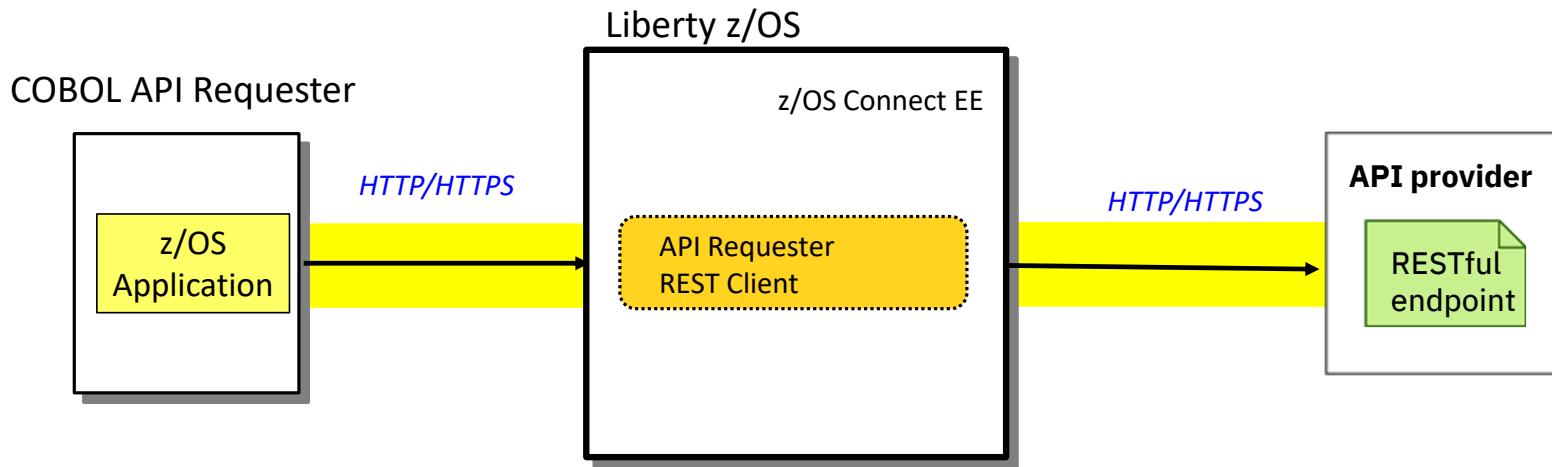
Steps to calling an external API

Done





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)



Configure connections to the z/OS API requester server

Default CICS URI MAP*

```

WG31 - 3270
File Edit Settings View Communication Actions Window Help
I URIMAP
RESULT - OVERTYPE TO MODIFY
Urimap(BAQURIMP)
Usage(Client)
Enablestatus(Enabled)
Availstatus(Notapplic)
Scheme(Http)
Redirecttype( None )
Tcpinservice()
Port(09120)
Host(wg31.washington.ibm.com:9120)
Path(/)
Analyzerstat(Noanalyzer)
Hosttype(Hostname)
Ipresolved(0.0.0.0)
Ipfamily(Unknown)
Socketclose(000030)
Sockpoolsize(000000)
Transaction()
+ Converter()

SYSID=CICS APPLID=CICS53Z
TIME: 10.38.37 DATE: 02/14/22
PF 1 HELP 2 HEX 3 END      5 VAR      7 SBH 8 SFH      10 SB 11 SF
01/012
M A D
Connected to remote server/host wg31a using lu/pool TCP00120 and port 23
Adobe PDF on Documents\*.pdf

```

* 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

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")

```

```

mpz3
File Edit Settings View Communication Actions Window Help
BROWSE ZCEE30.SBAQCOB(BAQRINFO) Line 000000010 Col 001 080
Command ==> Scroll ==> PAGE
* (C) Copyright IBM Corp. 2017, 2021
* US Government Users Restricted Rights - Use, duplication or
* disclosure restricted by GSA ADP Schedule Contract with
* IBM Corp
*****
* This file contains the generated language structure(s) for
* Request and Response Info
*****
* BAQ-REQUEST-INFO-COMP-LEVEL permitted values
* VALUE
* 0 Base support
* 1 Added support for BAQ-OAUTH
* 2 Added support for BAQ-TOKEN (JWT)
* 3 Added support for setting z/OS Connect EE server URI
* 4 Added support for BAQ-OAUTH-EXT
*****
01 BAQ-REQUEST-INFO.
03 BAQ-REQUEST-INFO-COMP-LEVEL PIC S9(9) COMP-5 SYNC VALUE 4.
03 BAQ-REQUEST-INFO-USER
05 BAQ-OAUTH.
07 BAQ-OAUTH-USERNAME PIC X(256).
07 BAQ-OAUTH-USERNAME-LEN PIC S9(9) COMP-5 SYNC
07 BAQ-OAUTH-PASSWORD VALUE 0.
07 BAQ-OAUTH-PASSWORD-LEN PIC X(256).
PIC S9(9) COMP-5 SYNC
04/015
Connected to remote server/host mpz3 using lu/pool MPZ30044 and port 23

```



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.



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:

```
//CEELOPTS 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

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" />
```



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 'ATSPKTTC'.
*-----
***** 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(ATSPKTTC)

```
*-----*
* 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
```

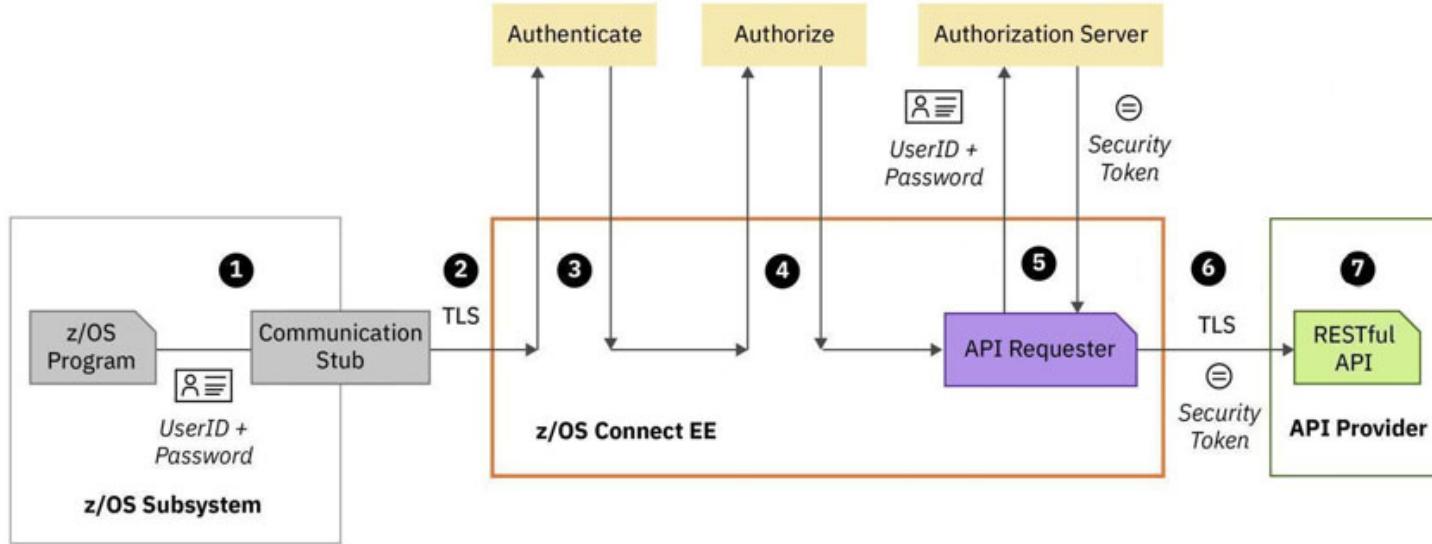


/security

How is security implement?



Typical z/OS Connect EE API Requester security flow



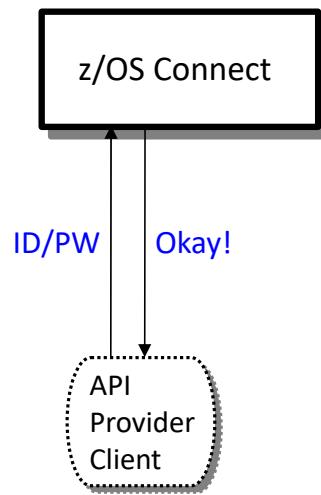
1. A user ID and password can be used for basic authentication by the z/OS Connect EE server
2. Connection between the CICS, IMS, or z/OS application and the z/OS Connect EE server can use TLS
3. Authenticate the CICS, IMS, or z/OS application.
4. Authorize the authenticated user ID to connect to z/OS 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



z/OS Application to z/OS Connect API Requester

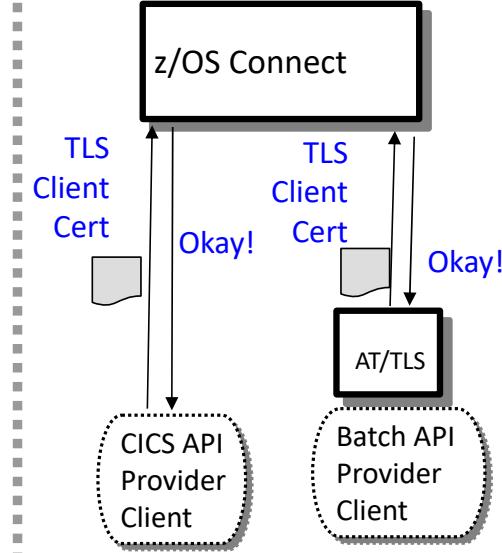
Two options for providing credentials for authentication

Basic Authentication



**Application provides
ID/PW or ID/PassTicket**

Client Certificate

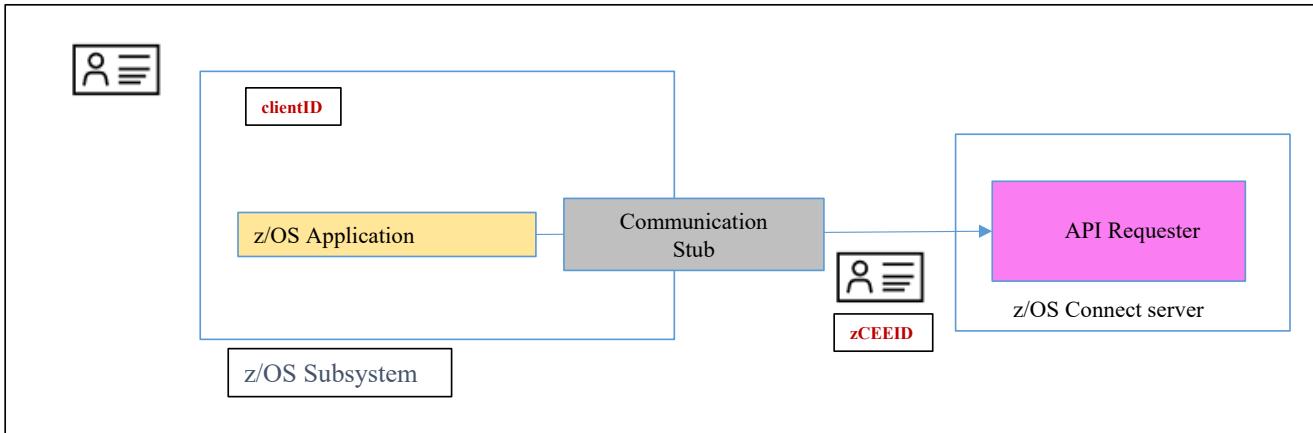


**z/OS Connect requests a
client certificate**

**CICS or AT/TLS supplies a
client certificate**



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 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 <i>zCEEID</i> and checks whether <i>zCEEID</i> has the authority to invoke an API requester.
	ASSERT_SURROGATE	Identity assertion is enabled. The zCEE server authenticates <i>zCEEID</i> and checks whether <i>zCEEID</i> is a surrogate of <i>clientID</i> . If <i>zCEEID</i> is a surrogate of <i>clientID</i> , the server further checks whether <i>clientID</i> has the authority to invoke an API requester; otherwise, a BAQR7114E message occurs.
	ASSERT_ONLY	Identity assertion is enabled. The zCEE server authenticates <i>zCEEID</i> and directly checks whether <i>clientID</i> 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 <i>clientID</i> 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 <i>clientID</i> 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>
```

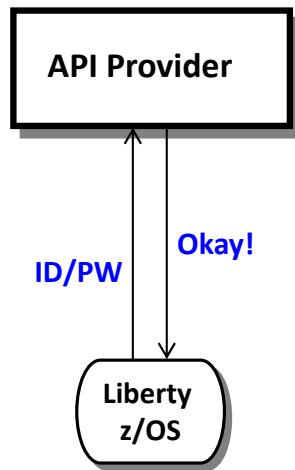
* Added in V3.0.45



API Requester - API Provider Authentication

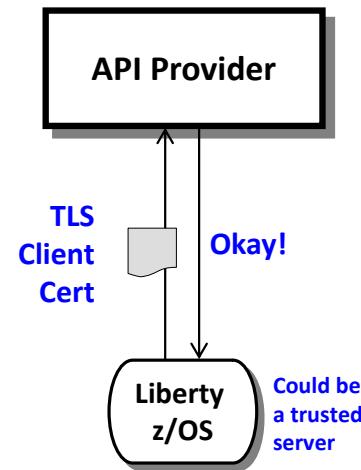
Several different ways this can be accomplished:

Basic Authentication



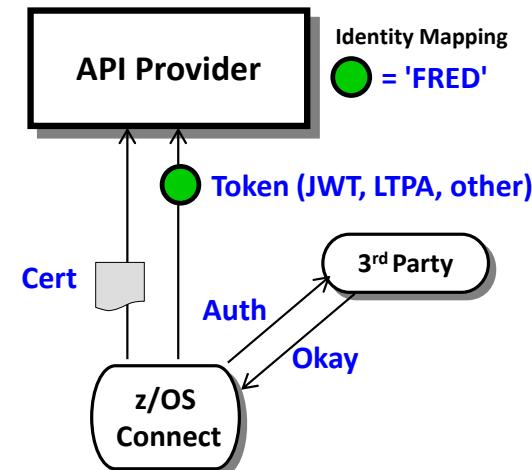
zCEE supplies ID/PW or
ID/Passticket

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



Third Party Authentication Examples

The image displays two side-by-side screenshots of web pages illustrating third-party authentication.

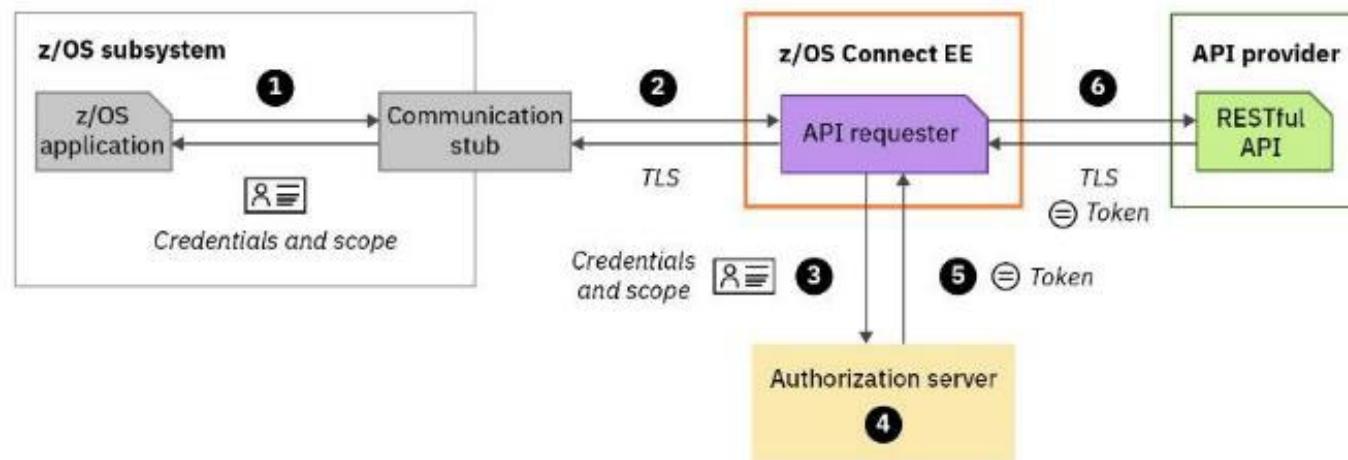
Left Screenshot: UPS Sign Up

This screenshot shows the UPS Sign Up page. At the top, there's a banner stating "UPS is open for business: Service impacts related to Coronavirus ...More". Below the banner, the UPS logo is displayed. A "Sign Up / Log in" link and a "Search or Track" input field are visible. The main section is titled "Sign Up" and includes a link for users who already have an ID. It provides several social media sign-in options: Google, Facebook, Amazon, Apple, and Twitter. Below these, there are fields for entering personal information: Name*, Email*, User ID*, Password*, and Phone. The "Password" field has a "Show" link next to it. A "Feedback" button is located on the right side of the form.

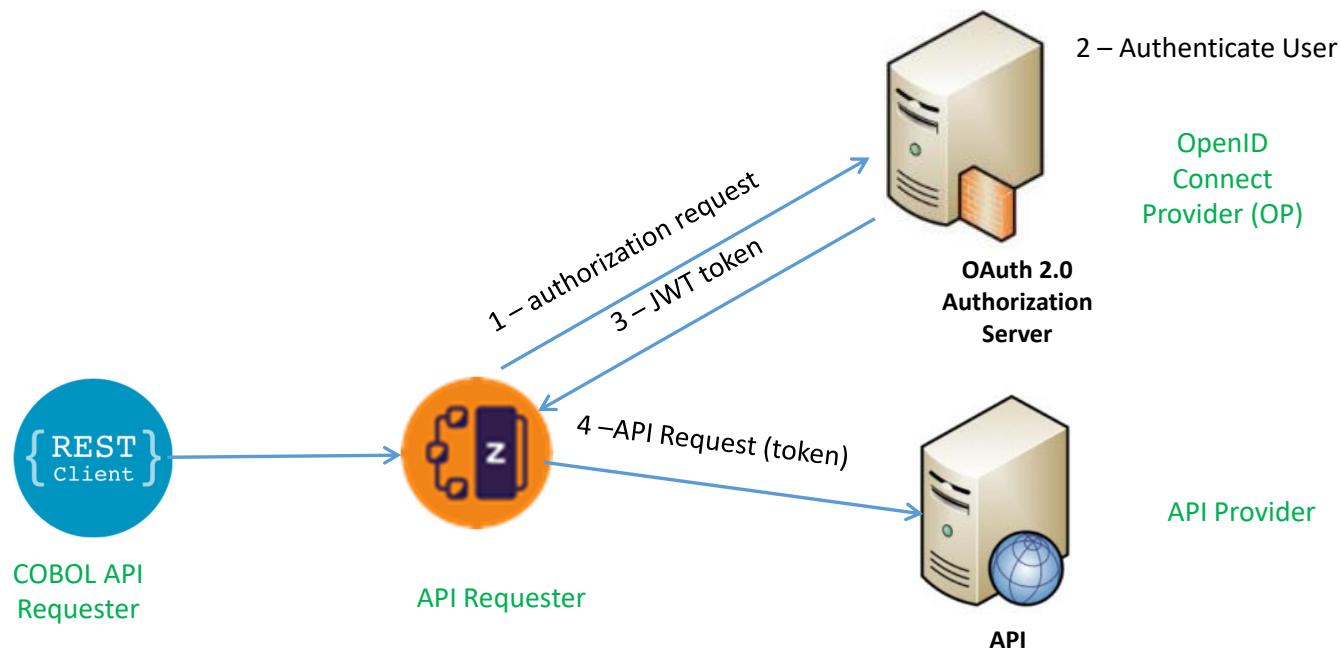
Right Screenshot: myNCDMV Log In

This screenshot shows the myNCDMV Log In page. The background features a scenic view of autumn foliage. The page has "Log In" and "Sign Up" tabs at the top. The "Log In" tab is selected. It contains fields for "Email Address" (with placeholder "name@example.com") and "Password" (with placeholder "*****"). There is a "Remember Me" checkbox and "Log In" and "Forgot Password" buttons. Below these, there are three social media sign-in options: "Continue with Apple", "Continue with Facebook", and "Continue with Google". A "Continue as Guest" link is also present. At the bottom, a notice for public computer users states: "NOTICE FOR PUBLIC COMPUTER USERS - If you sign in with Google, Apple, or Facebook you are also signing into that account on this computer. Remember to sign out when you're done." The page is powered by "payit".

Calling an API with OAuth 2.0 support



z/OS Connect OAuth Flow for API requester



Grant Types:

- client_credentials
- password



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"/>
```



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



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

The screenshot shows a GitHub repository page for 'ibm-wsc/zCONNEE-Wildfire-Workshop'. The repository has 11 stars and 8 forks. The 'Code' tab is selected. In the 'exercises' directory, there is a commit by 'emitchj' titled 'Add files via upload' made 20 days ago. This commit contains 14 files, all of which are PDFs related to developing APIs for various IBM z/OS Connect services. The files include:

- Developing CICS API Requester Applications.pdf
- Developing IMS API Requester Applications.pdf
- Developing MVS Batch API Requester Applications.pdf
- Developing RESTful APIs for DVM VSAM Services.pdf
- Developing RESTful APIs for DVM VSAMCICS Services.pdf
- Developing RESTful APIs for Db2 REST Services.pdf
- Developing RESTful APIs for HATS REST Services.pdf
- Developing RESTful APIs for IMS Database REST Services....
- Developing RESTful APIs for IMS Transactions.pdf
- Developing RESTful APIs for MQ.pdf
- Developing RESTful APIs for MVS Batch.pdf
- Developing RESTful APIs for a CICS COMMAREA program.pdf
- Developing RESTful APIs for a CICS Container program.pdf

mitchj@us.ibm.com

- Contact your IBM representative to schedule access to these exercises

© 2018, 2022 IBM Corporation

Page 58



Thank you for listening and your questions.