Best Practices for RESTfully Enabling Your IMS

(OpenAPI2 and OpenAPI3)

Central Canada
Db2 Users Group

Session IMS-05

Suzie Wendler
IBM – Washington Systems Center
wendler@us.ibm.com

Topics

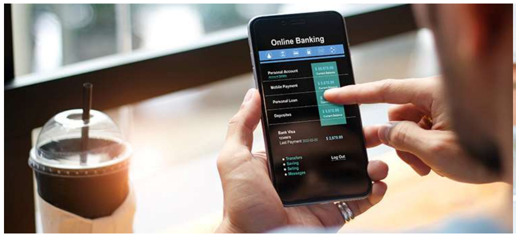
- The Open API Specification (OAS)
  - A level set
  - OpenAPI 2 versus Open API 3
- The implications for IMS
  - Transaction access
    - Considerations

IMS has supported OpenAPI 2.0. for several releases but recently announced additional support for OpenAPI 3.0

The implications of REST API support for IMS transactions, however, remains the same regardless of the OpenAPI version.

Level Set

- **Digital transformation evolution**
  - Includes many use cases drive data and functionality requirements to assets that are hosted on IBM Z
  - For Example:



Account queries on mobile devices using open APIs

Evolution of online apps
To hyrid cloud

Raise credt card issues

- **A key piece of digital transformation → *APIs***
  - They enable the increase of speed in which the business can operate
    - Connect applications through clear protocols and architectures

"More than 90 % of financial institutions use or plan to use APIs to generate additional revenue among existing customers"

Mckinsey & Co, 2021

https://www.mckinsey.com/industries/financial-services/our-insights/from-tech-tool-to-business-asset-how-banks-are-using-b2b-apis-to-fuel-growth

Level Set... APIs

- API architecture - framework of rules and structures for creating software interfaces
  - The rules determine how to provide server functionality to users

- **Popular types of API architectures**
  - SOAP (**S**imple **O**bject **A**ccess **P**rotocol)
  - GraphQL
  - Apache Kafka
  - AsyncAPI
  - RPC (Remote Procedure call)
  - **REST** (**RE**presentational **S**tate **T**ransfer) API
    - With **Open API S**pecifications **(OAS)**

    Provides a standardized format to describe the rest api interface

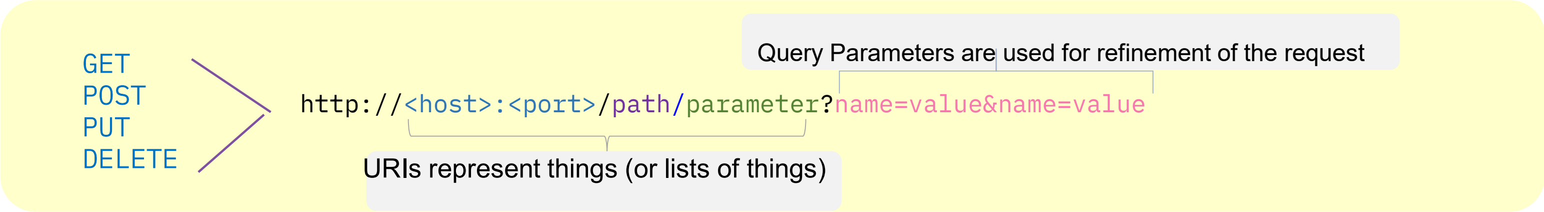
What we will be discuss in this presentation and referencing as API

Level Set… APIs

- Open API Specification (OAS)
  - Defines a specification language (standardized format) for HTTP APIs
  - OpenAPI ***definition file*** (yaml or json)
    - Describes what an api or service can do – readable by both humans and machines
    - Describes the rest api’s resources (properties or data types), endpoints, operations, parameters and the rest api’s authorization
    - Provides guidance - allows someone referencing the definition to understand or actually use the API
    - Allows extending the REST API with tooling
    - Supports API doc generation

Level Set…

- RESTful APIs with the Open API Specification
  - A type of web API with a **defined set (open standard)** of definitions and protocols
    - Enable different applications across multiple platforms to communicate with each other
  - A ***client*** application that invokes a RESTful API provides:
    - An identifier for the target server resource This is the URL for the resource, also known as the **endpoint**
    - The operation you want the server to perform on that resource, in the form of an **HTTP method**, or **verb**, *e.g,*



GET
POST
PUT
DELETE

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

Query Parameters are used for refinement of the request

URIs represent things (or lists of things)

- The ***server*** responds by *transferring a representation* of the *state* of the requested resource
  - Typically in a **JSON** (Java Script Object Notation) format - open standard and commonly used for file and data interchange
    - Uses human-readable text to store and transmit data objects consisting of attribute/name –value pairs and arrays

(Note: information transferred via HTTP can be in JSON, HTML, XML, or plain text. JSON is the most commonly used file format for REST APIs. JSON is language-agnostic and readable by humans.)

Level Set…

- For Example:

  - GET http://ESYSMVS1.wsclab.washington.ibm.com:23628/phonecontacts/LAST1

```
RESPONSE: HTTP 200 OK
BODY {“OUTPUT_AREA: {
     "OUT_LAST_NAME1": "LAST1",
     "OUT_EXTENSION1": "8-111-1111",
     "OUT_ZIP_CODE1": "D01/R01",
     "OUT_FIRST_NAME1": "FIRST1"   }
    }
```

  - POST http://ESYSMVS1.wsclab.washington.ibm.com:23628/phonecontacts/Butler

```
RESPONSE: HTTP 200 OK
BODY } "OUTPUT_AREA": {
     "OUT_ZIP_CODE": "77789",
     "OUT_FIRST_NAME": "SANDRA",
     "OUT_EXTENSION": "8/123-4567",
     "OUT_MESSAGE": "ENTRY WAS ADDED",
     "OUT_LAST_NAME": "BUTLER" }
  }
```

Request/Response Body is used to represent the data object

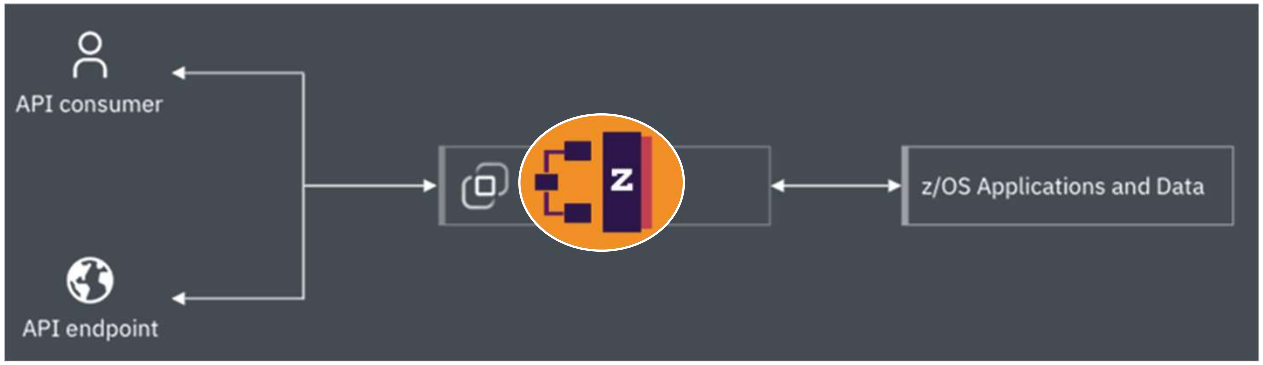
IMS

IVTNO transaction

How?

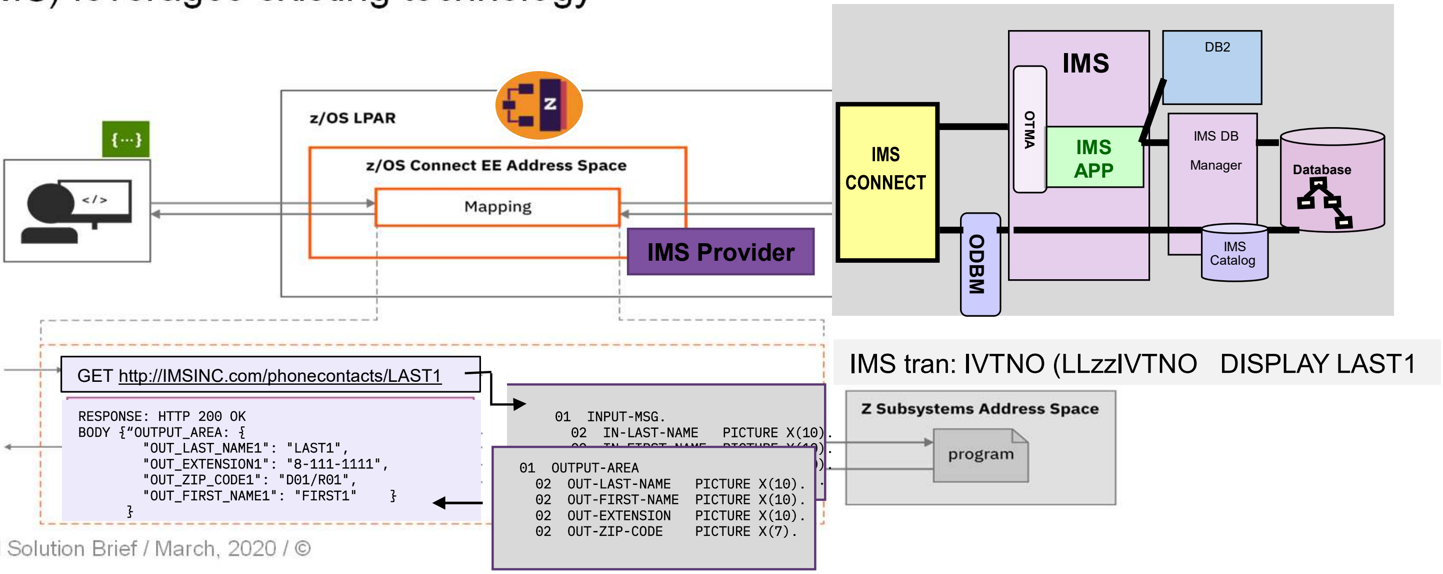
Level Set … z/OS Connect

- **z/OS Connect allows IBM Z to capitalize on the value of the API economy**
  - Provides ***seamless integration*** with the Z subsystems as truly RESTful APIs
    - Implements ***secure and robust business AP***Is by leveraging the recognized secure methods of IBM Z
      - Fast, secure, reliable connectors to reach any z/OS asset
    - Supports the creation of consumable APIs in minutes to make Z applications/data central to the hybrid cloud strategy
    - Provides Z applications with the capability of calling APIs to enhance them with the power of cloud native functions



z/OS Connect

- Provides standard access to z/OS assets (IMS, CICS, DB2,…) using REST technology
  - z/OS Connect configuration – specifies access to the assets
    - Relieves client applications (cloud, mobile, …) from having to understand the details about how to reach them and how to convert payloads to and from the formats that the applications require
  - z/OS asset (IMS) leverages existing technology



IMS tran: IVTNO (LLzzIVTNO   DISPLAY LAST1

GET http://IMSINC.com/phonecontacts/LAST1

```
RESPONSE: HTTP 200 OK
BODY {“OUTPUT_AREA: {
        "OUT_LAST_NAME1": "LAST1",
        "OUT_EXTENSION1": "8-111-1111",
        "OUT_ZIP_CODE1": "D01/R01",
        "OUT_FIRST_NAME1": "FIRST1"    }
     }
```

```
01  INPUT-MSG.
  02  IN-LAST-NAME   PICTURE X(10).
  02  IN-FIRST-NAME  PICTURE X(10).
```

```
01  OUTPUT-AREA
  02  OUT-LAST-NAME   PICTURE X(10).
  02  OUT-FIRST-NAME  PICTURE X(10).
  02  OUT-EXTENSION   PICTURE X(10).
  02  OUT-ZIP-CODE    PICTURE X(7).
```

z/OS Connect and the OpenAPI Specification

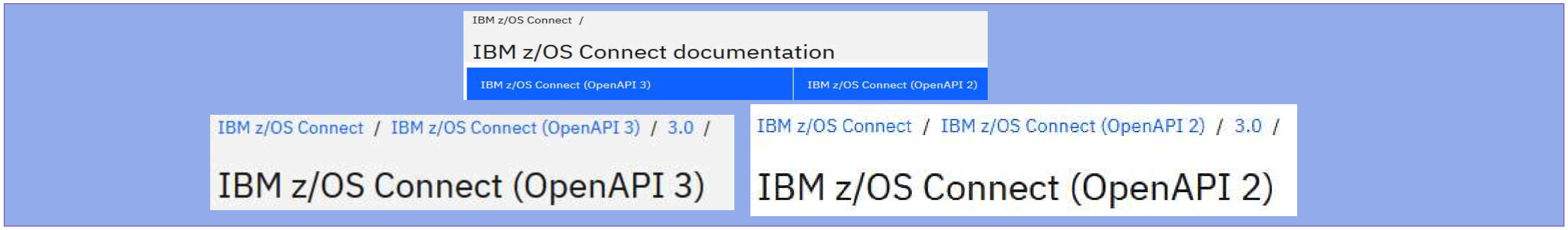
- z/OS Connect supports two OpenAPI standards
  - One product, one deliverable
    - OpenAPI 2 and OpenAPI 3

OpenAPI 2 --> Open API 3 is not intended as a migration path
Rather, it is a choice based on the use case

- Knowledge Center home page:
  - https://www.ibm.com/docs/en/zos-connect
- Each supported OpenAPI specification has own z/OS Connect Knowledge Center

https://www.ibm.com/docs/en/zos-connect/zos-connect/3.0

https://www.ibm.com/docs/en/zos-connect/zosconnect/3.0



IBM z/OS Connect /

IBM z/OS Connect documentation

IBM z/OS Connect (OpenAPI 3) | IBM z/OS Connect (OpenAPI 2)

IBM z/OS Connect / IBM z/OS Connect (OpenAPI 3) / 3.0 /

IBM z/OS Connect (OpenAPI 3)

IBM z/OS Connect / IBM z/OS Connect (OpenAPI 2) / 3.0 /

IBM z/OS Connect (OpenAPI 2)

Note:

V3.0.67 (APAR PH52418) z/OS Connect server update (March, 2023)

•**Enhancements** z/OS Connect is enhanced to support IMS COBOL applications, mapping from OpenAPI 3.0 operations to message segments or Large Data Structures.

z/OS Connect Designer (tooling)

•**Enhancements** You can now develop API projects in z/OS Connect Designer to map OpenAPI 3.0 operations to IMS COBOL applications.

z/OS Connect Documentation

•**Enhancements** 'How to' instructions to create an IMS z/OS Asset in z/OS Connect Designer, configure z/OS Connect connections to IMS and secure with basic authentication and AT-TLS. For more information see Creating an IMS z/OS Asset, Configuring connections to IMS and Configuring security for IMS connections.

•A new tutorial to create an API project to call the example IMS Phonebook program. Then, test your z/OS Connect API project with the built-in Liberty REST client. For more information, see Getting Started tutorial: Creating an IMS z/OS Connect API.

•The API requester tutorial project is available on GitHub and can be downloaded from here https://github.com/zosconnect/sample-oas3-requester or by using the link in the API requester tutorial. For more information, see Downloading the tutorial structure, prerequisites and downloading the tutorial.

OpenAPI2 Vs OpenAPI3

- OpenAPI2
  - ***Bottom-up*** development
    - Starts with the target environment, e.g., IMS application and COBOL copybook to build a service and API to drive it
    - Exposes the z/OS resource (transaction/data) to a standard language-agnostic interface to HTTP/HTTPS REST API callers
    - Allows the z/OS application to use the same functionality to call external APIs

- OpenAPI3
  - ***Meet-in-the-middle or*** API First development
    - Provides the ability to define the APIs based on the business requirements or those defined by governance boards
      - Development of public APIs that must adhere to external standards bodies or governments that may weigh in on API designs and standards to ensure interoperability across the industry

OpenAPI2 Vs OpenAPI3 – Comparing JSON Output

- z/OS Connect: **OpenAPI 2**
  - JSON returned based on structure in copy book
  - Cannot remove “header” field
  - Customers ask:
    - Can we manipulate copybook fields ?
    - Can we change JSON structure ?
    - Can we combine copybook fields ?
    - **Answer: NO**

```
{
 "DFH0XCP3": {
   "CA_RETURN_CODE": 0,
   "CA_RESPONSE_MESSAGE": "+15 ITEMS RETURNED",
   "CA_INQUIRE_REQUEST": {
     "CA_LIST_START_REF": 10,
     "CA_LAST_ITEM_REF": 150,
     "CA_ITEM_COUNT": 15,
     "CA_CAT_ITEM": [
      {
        "CA_ITEM_REF": 10,
        "item-description": "Ball Pens Black 24pk",
        "CA_COST": "002.90",
        "IN_STOCK": 170
      },
```

- z/OS Connect: **OpenAPI 3**
  - Meet in the middle approach
  - JSON returned up to API designer
  - Independent of copy book structure
  - Lot of capability to manipulate data returned from SOR

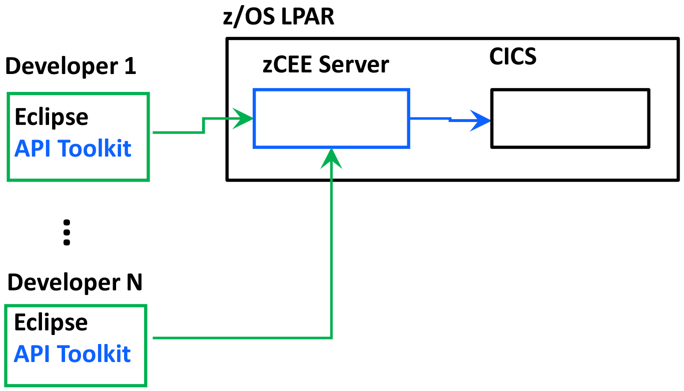
```
  "totalItems": 15,
  "items": [
   {
     "summary": {
       "stock": "Department 10 has 170 'Ball Pens Black 24pk' in stock.",
       "orders": "0 'Ball Pens Black 24pk' on order at unit price of $2.9.
Total order value: $0"
     },
     "information": {
       "itemReference": 10,
       "description": "Ball Pens Black 24pk",
       "department": 10,
       "stock": 170,
       "cost": "2.9",
       "onOrder": 0
```

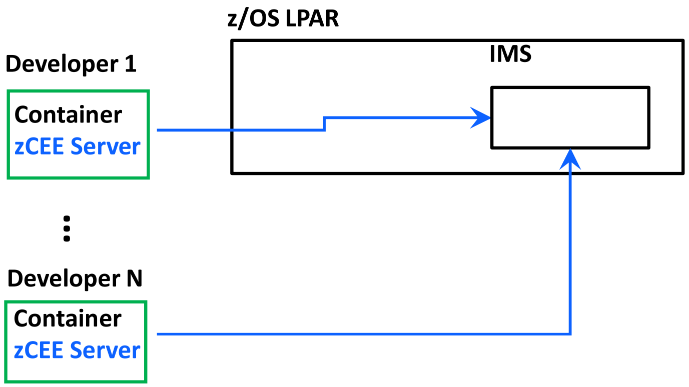
OpenAPI2 Vs OpenAPI3 – The Developer Experience

Click to edit Master title style

- z/OS Connect: **OpenAPI 2**
  - Each developer uses API Toolkit in Eclipse on their PC
- Click to edit Master text styles
  - Second level
    - Third level
      - Fourth level
        - Fifth level

- z/OS Connect: **OpenAPI 3**
  - Each Developer uses the new Designer in a container
    - There is a z/OS Connect server running in the container
    - Each developer has their own zCEE server
  - Container could be running in OpenShift on x86 or IBM Z Systems (zVM, z/OS)

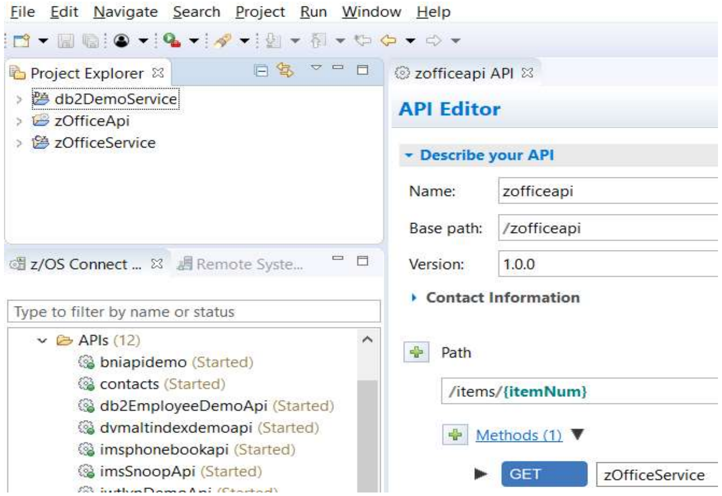




OpenAPI2 Vs OpenAPI3 – Development Compared

- z/OS Connect: OpenAPI 2
  - API Toolkit
  - Eclipse based, installed on your PC
  - **Bottom up development**
    - Use COBOL copybook to build a service and then an API

- z/OS Connect: OpenAPI 3
  - Designer
  - Runs in container on s390x and x86
  - Access via a browser
  - Input is yaml file that describes the API
  - **Meet in the middle development**



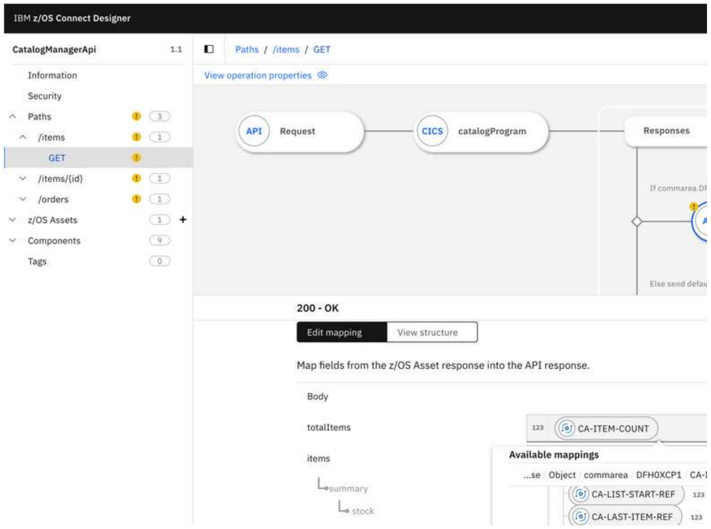
Name: zofficeapi

Base path: /zofficeapi

Version: 1.0.0

/items/{itemNum}

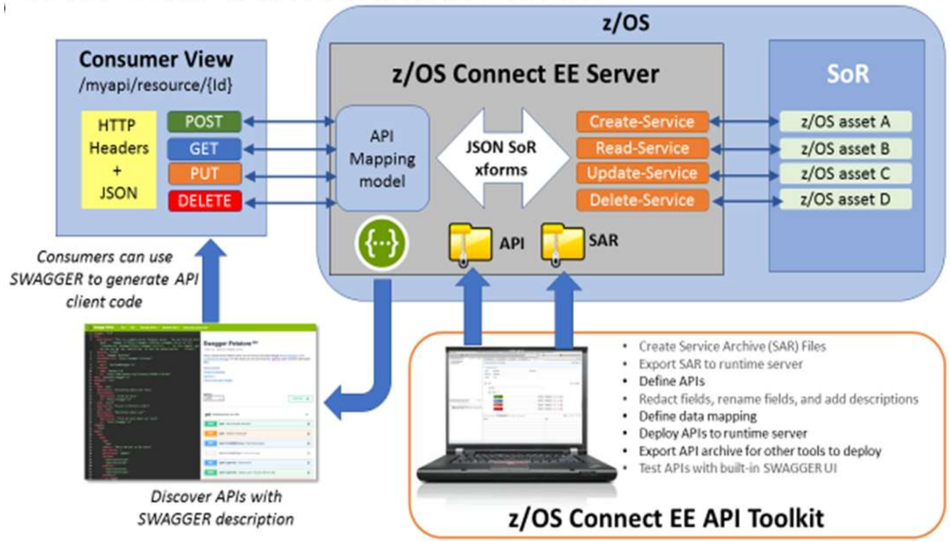
GET zOfficeService



Map fields from the z/OS Asset response into the API response.

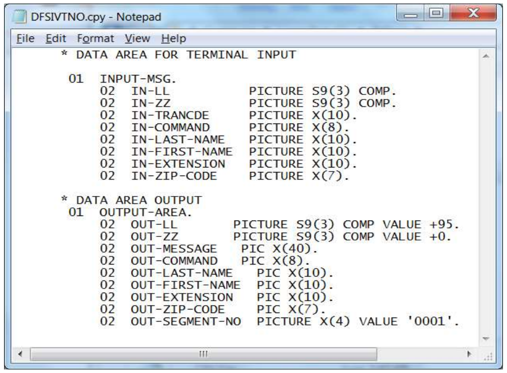
OpenAPI2 and OpenAPI3 Tooling

Cli



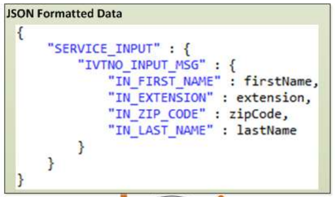
- Create Service Archive (SAR) Files
- Export SAR to runtime server
- Define APIs
- Redact fields, rename fields, and add descriptions
- Define data mapping
- Deploy APIs to runtime server
- Export API archive for other tools to deploy
- Test APIs with built-in SWAGGER UI

Starte with the IMS Copybooks



```
      * DATA AREA FOR TERMINAL INPUT

       01  INPUT-MSG.
           02  IN-LL          PICTURE S9(3) COMP.
           02  IN-ZZ          PICTURE S9(3) COMP.
           02  IN-TRANCDE     PICTURE X(10).
           02  IN-COMMAND     PICTURE X(8).
           02  IN-LAST-NAME   PICTURE X(10).
           02  IN-FIRST-NAME  PICTURE X(10).
           02  IN-EXTENSION   PICTURE X(10).
           02  IN-ZIP-CODE    PICTURE X(7).

      * DATA AREA OUTPUT
       01  OUTPUT-AREA.
           02  OUT-LL       PICTURE S9(3) COMP VALUE +95.
           02  OUT-ZZ       PICTURE S9(3) COMP VALUE +0.
           02  OUT-MESSAGE   PIC X(40).
           02  OUT-COMMAND   PIC X(8).
           02  OUT-LAST-NAME   PIC X(10).
           02  OUT-FIRST-NAME  PIC X(10).
           02  OUT-EXTENSION   PIC X(10).
           02  OUT-ZIP-CODE    PIC X(7).
           02  OUT-SEGMENT-NO  PICTURE X(4) VALUE '0001'.
```

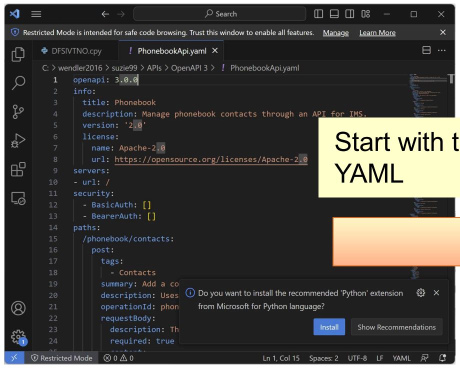
OpenAPI 2.0



JSON Formatted Data

```
{
    "SERVICE_INPUT" : {
        "IVTNO_INPUT_MSG" : {
            "IN_FIRST_NAME" : firstName,
            "IN_EXTENSION" : extension,
            "IN_ZIP_CODE" : zipCode,
            "IN_LAST_NAME" : lastName
        }
    }
}
```

JSON
to
IMS Binary Message
Transformation

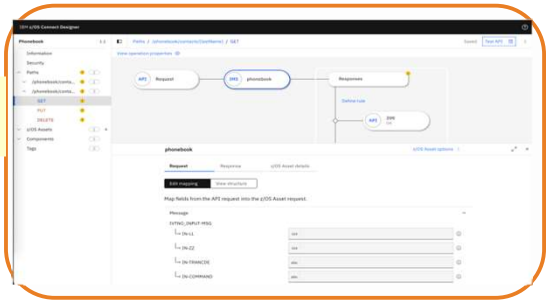


Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

```yaml
openapi: 3.0.0
info:
  title: Phonebook
  description: Manage phonebook contacts through an API for IMS.
  version: '2.0'
  license:
    name: Apache-2.0
    url: https://opensource.org/licenses/Apache-2.0
servers:
- url: /
security:
  - BasicAuth: []
  - BearerAuth: []
paths:
  /phonebook/contacts:
    post:
      tags:
        - Contacts
      summary: Add a co
      description: Uses
      operationId: phon
      requestBody:
        description: Th
        required: true
```

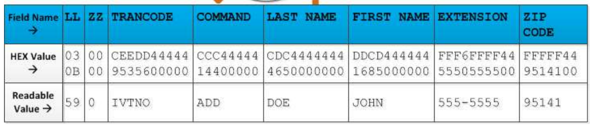
Do you want to install the recommended 'Python' extension from Microsoft for Python language?

Start with the API YAML



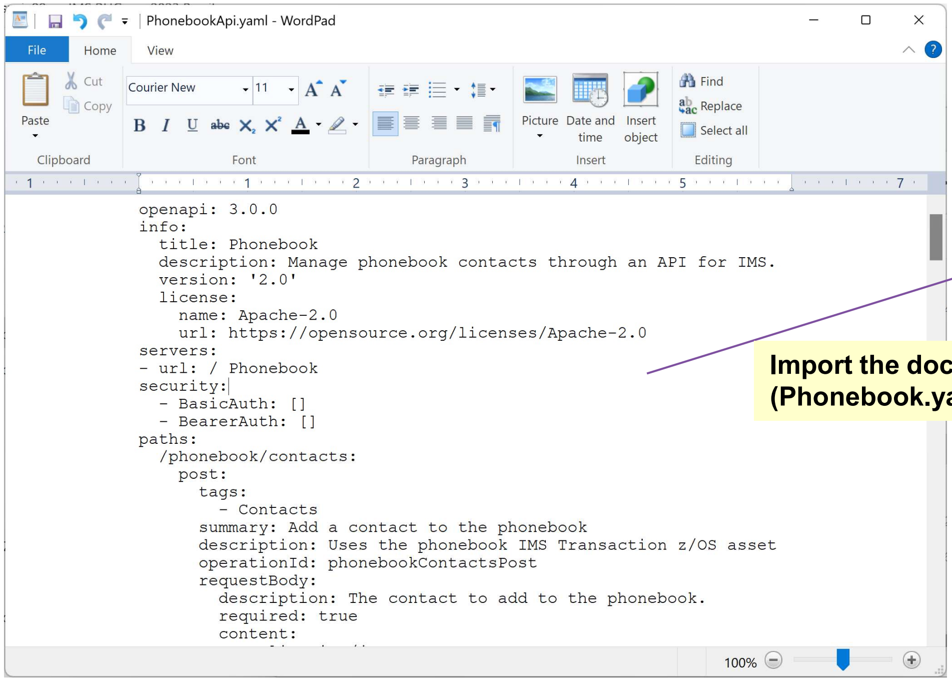
z/OS Connect Designer

OpenAPI 3.0



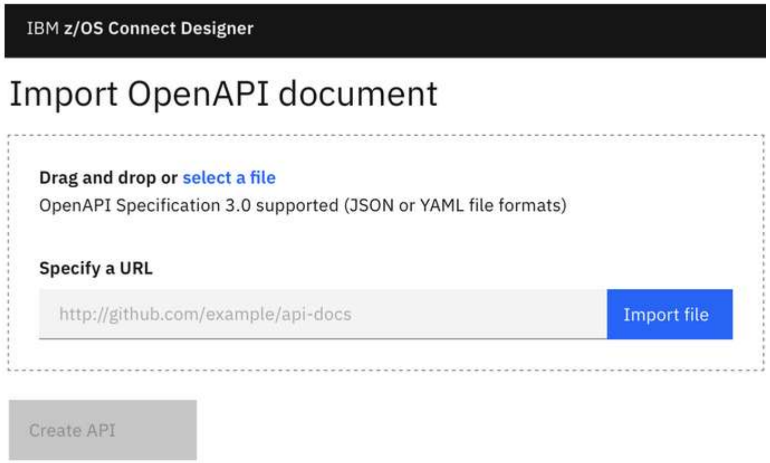
Field Name →	LL	ZZ	TRANCODE	COMMAND	LAST NAME	FIRST NAME	EXTENSION	ZIP CODE
HEX Value →	03 0B	00 00	CEEDD44444 9535600000	CCC44444 14400000	CDC4444444 4650000000	DDCD444444 1685000000	FFF6FFFF44 5550555500	FFFFF44 9514100
Readable Value →	59	0	IVTNO	ADD	DOE	JOHN	555-5555	95141

OpenAPI 3 and the Designer - IMS example



```
openapi: 3.0.0
info:
  title: Phonebook
  description: Manage phonebook contacts through an API for IMS.
  version: '2.0'
  license:
    name: Apache-2.0
    url: https://opensource.org/licenses/Apache-2.0
servers:
- url: / Phonebook
security:
  - BasicAuth: []
  - BearerAuth: []
paths:
  /phonebook/contacts:
    post:
      tags:
        - Contacts
      summary: Add a contact to the phonebook
      description: Uses the phonebook IMS Transaction z/OS asset
      operationId: phonebookContactsPost
      requestBody:
        description: The contact to add to the phonebook.
        required: true
        content:
```

Import the document (Phonebook.yaml)



Import OpenAPI document

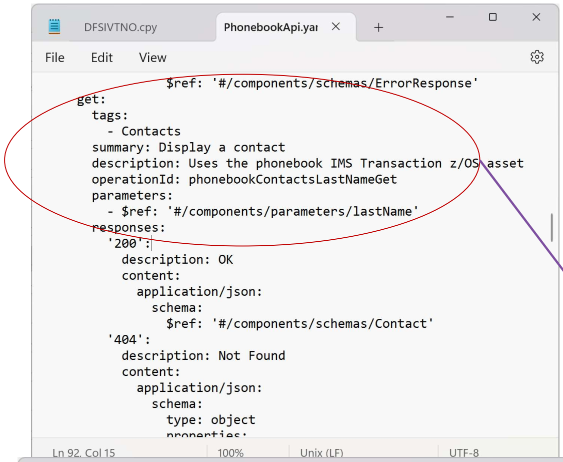
Drag and drop or select a file

OpenAPI Specification 3.0 supported (JSON or YAML file formats)

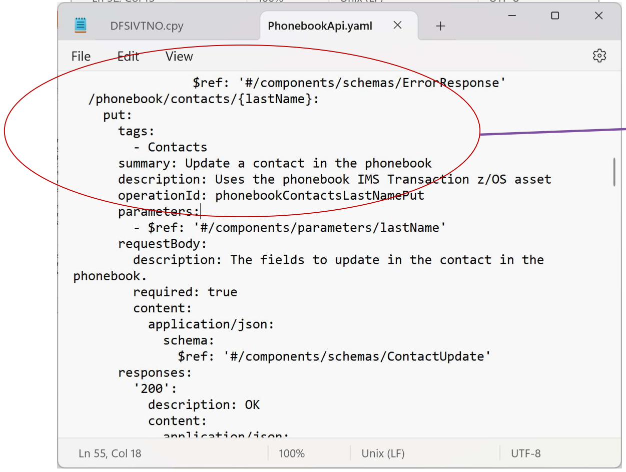
Specify a URL

OpenAPI 3 and the Designer - IMS example…

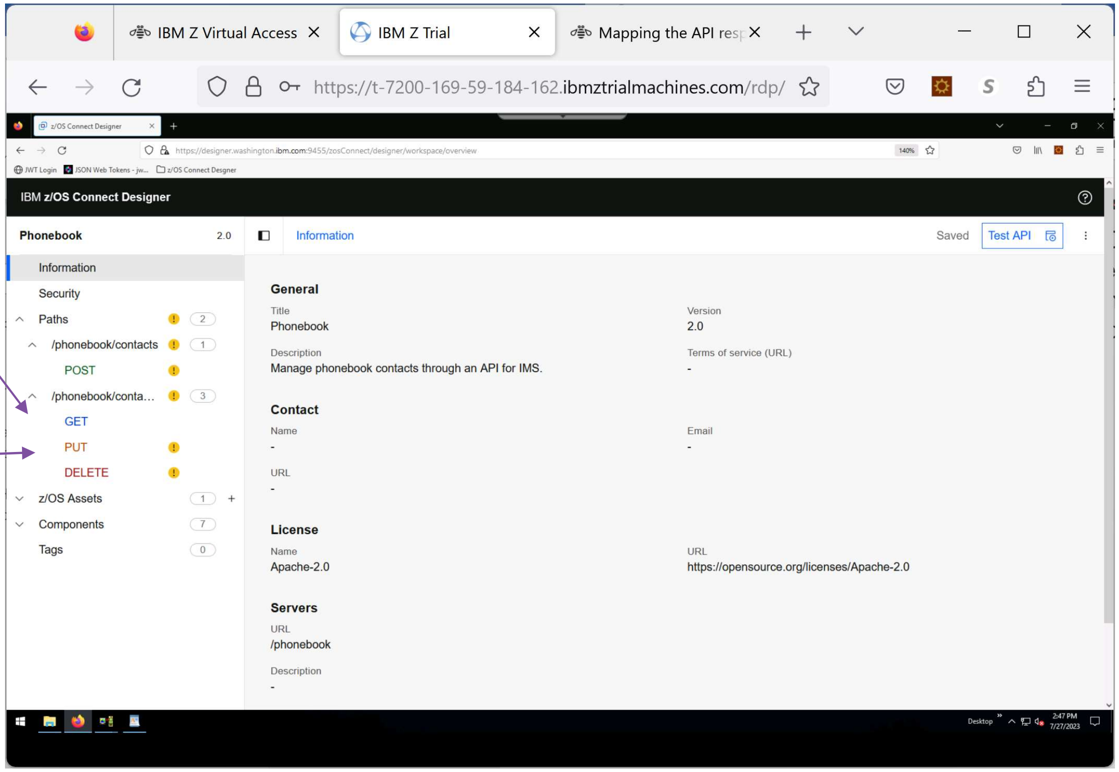
After importing the document (Phonebook.yaml), the Designer displays the pathsm methods, etc that can be defined in the API



```
              $ref: '#/components/schemas/ErrorResponse'
    get:
      tags:
        - Contacts
      summary: Display a contact
      description: Uses the phonebook IMS Transaction z/OS asset
      operationId: phonebookContactsLastNameGet
      parameters:
        - $ref: '#/components/parameters/lastName'
      responses:
        '200':
          description: OK
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/Contact'
        '404':
          description: Not Found
          content:
            application/json:
              schema:
                type: object
                properties:
```



```
              $ref: '#/components/schemas/ErrorResponse'
  /phonebook/contacts/{lastName}:
    put:
      tags:
        - Contacts
      summary: Update a contact in the phonebook
      description: Uses the phonebook IMS Transaction z/OS asset
      operationId: phonebookContactsLastNamePut
      parameters:
        - $ref: '#/components/parameters/lastName'
      requestBody:
        description: The fields to update in the contact in the
phonebook.
        required: true
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/ContactUpdate'
      responses:
        '200':
          description: OK
          content:
            application/json:
```



General

Title
Phonebook

Version
2.0

Description
Manage phonebook contacts through an API for IMS.

Terms of service (URL)
-

Contact

Name
-

Email
-

URL
-

License

Name
Apache-2.0

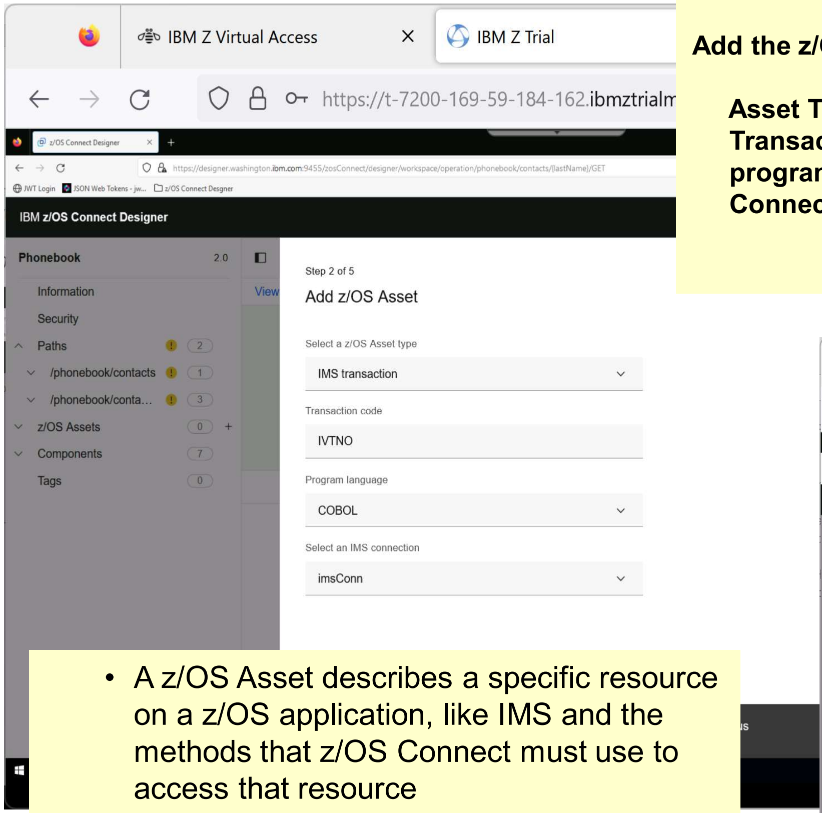
URL
https://opensource.org/licenses/Apache-2.0

Servers

URL
/phonebook

Description
-

OpenAPI 3 and the Designer - IMS example...



Step 2 of 5

Add z/OS Asset

Select a z/OS Asset type: IMS transaction

Transaction code: IVTNO

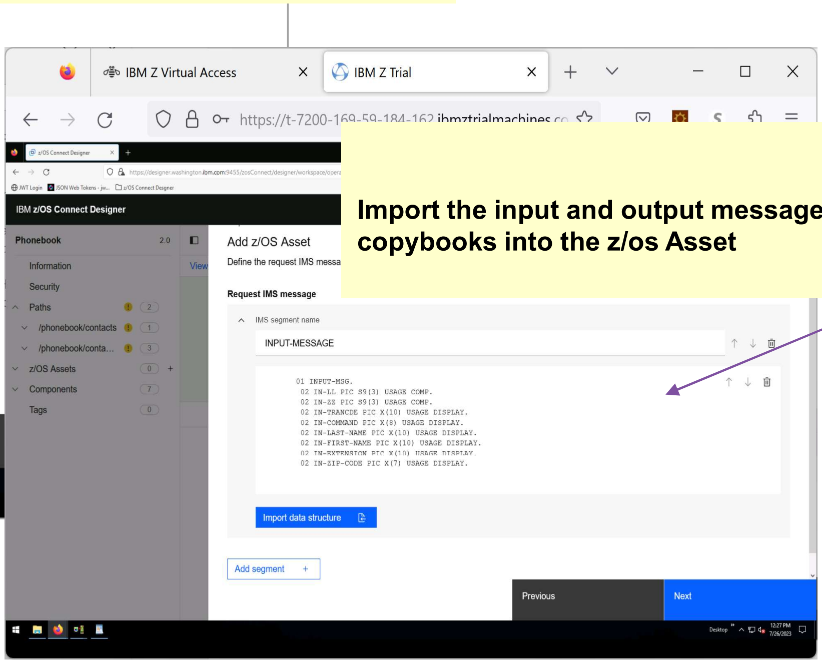
Program language: COBOL

Select an IMS connection: imsConn

Add the z/OS Asset

Asset Type - IMS transaction
Transaction code - IVTNO
program language – COBOL
Connection information (pre-defined)

- A z/OS Asset describes a specific resource on a z/OS application, like IMS and the methods that z/OS Connect must use to access that resource
- For IMS Assets
  - Import COBOL copybook



Add z/OS Asset

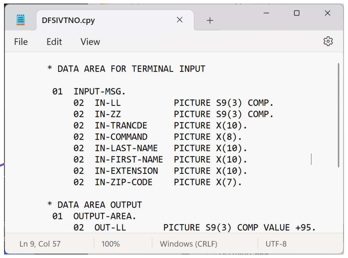
Define the request IMS messa

Request IMS message

IMS segment name: INPUT-MESSAGE

```
01 INPUT-MSG.
 02 IN-LL PIC S9(3) USAGE COMP.
 02 IN-ZZ PIC S9(3) USAGE COMP.
 02 IN-TRANCDE PIC X(10) USAGE DISPLAY.
 02 IN-COMMAND PIC X(8) USAGE DISPLAY.
 02 IN-LAST-NAME PIC X(10) USAGE DISPLAY.
 02 IN-FIRST-NAME PIC X(10) USAGE DISPLAY.
 02 IN-EXTENSION PIC X(10) USAGE DISPLAY.
 02 IN-ZIP-CODE PIC X(7) USAGE DISPLAY.
```

Import the input and output message copybooks into the z/os Asset



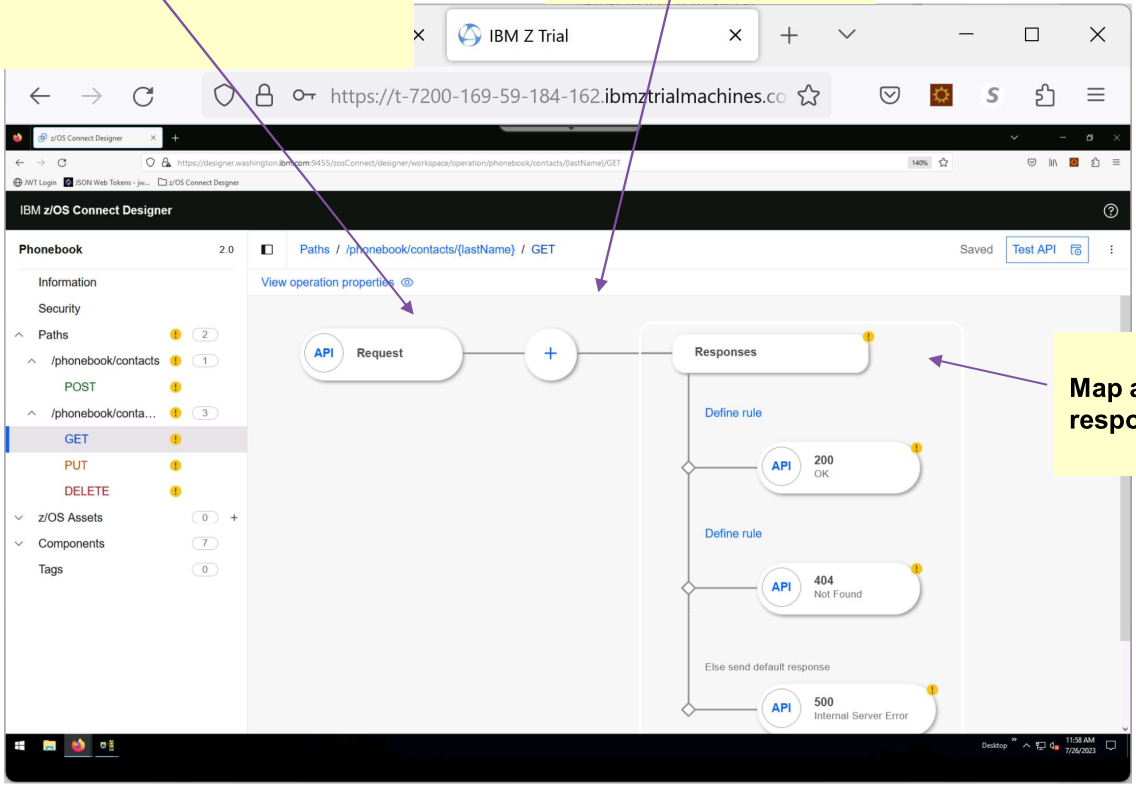
```
* DATA AREA FOR TERMINAL INPUT

 01  INPUT-MSG.
     02  IN-LL          PICTURE S9(3) COMP.
     02  IN-ZZ          PICTURE S9(3) COMP.
     02  IN-TRANCDE     PICTURE X(10).
     02  IN-COMMAND     PICTURE X(8).
     02  IN-LAST-NAME   PICTURE X(10).
     02  IN-FIRST-NAME  PICTURE X(10).
     02  IN-EXTENSION   PICTURE X(10).
     02  IN-ZIP-CODE    PICTURE X(7).

* DATA AREA OUTPUT
 01  OUTPUT-AREA.
     02  OUT-LL       PICTURE S9(3) COMP VALUE +95.
```

OpenAPI 3 and the Designer - IMS example...

Operations Flow Diagram

Map the api request to the asset

Specify the z/OS Asset you just created

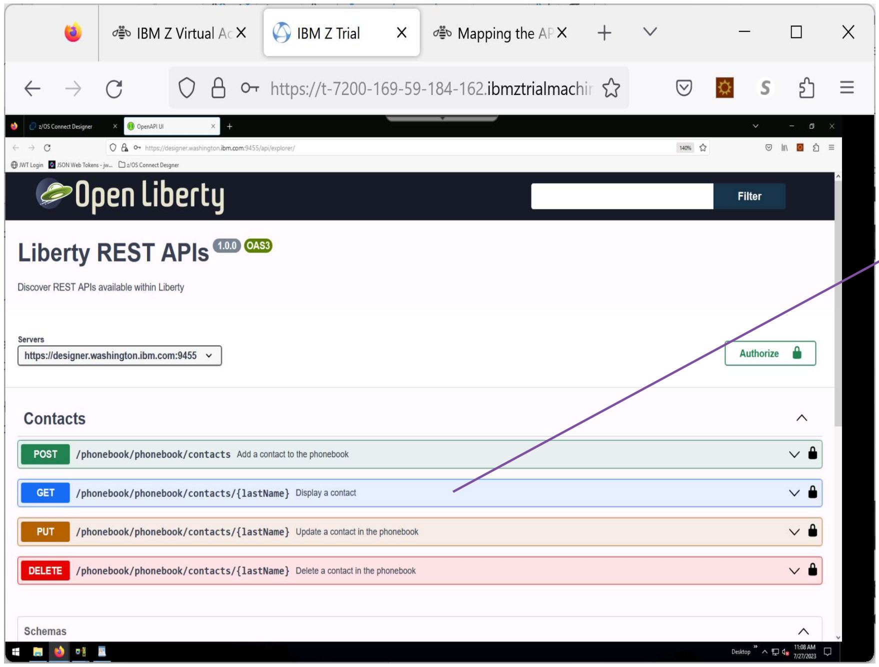


Methods to be defined

Map and customize responses

OpenAPI 3 and the Designer - IMS example…

Testing the API in the tool



IBM Z Virtual Ac
IBM Z Trial
Mapping the AF
https://t-7200-169-59-184-162.ibmztrialmachi
z/OS Connect Designer
OpenAPI UI
https://designer.washington.ibm.com:9455/api/explorer/
Open Liberty
Filter

Liberty REST APIs 1.0.0 OAS3

Discover REST APIs available within Liberty

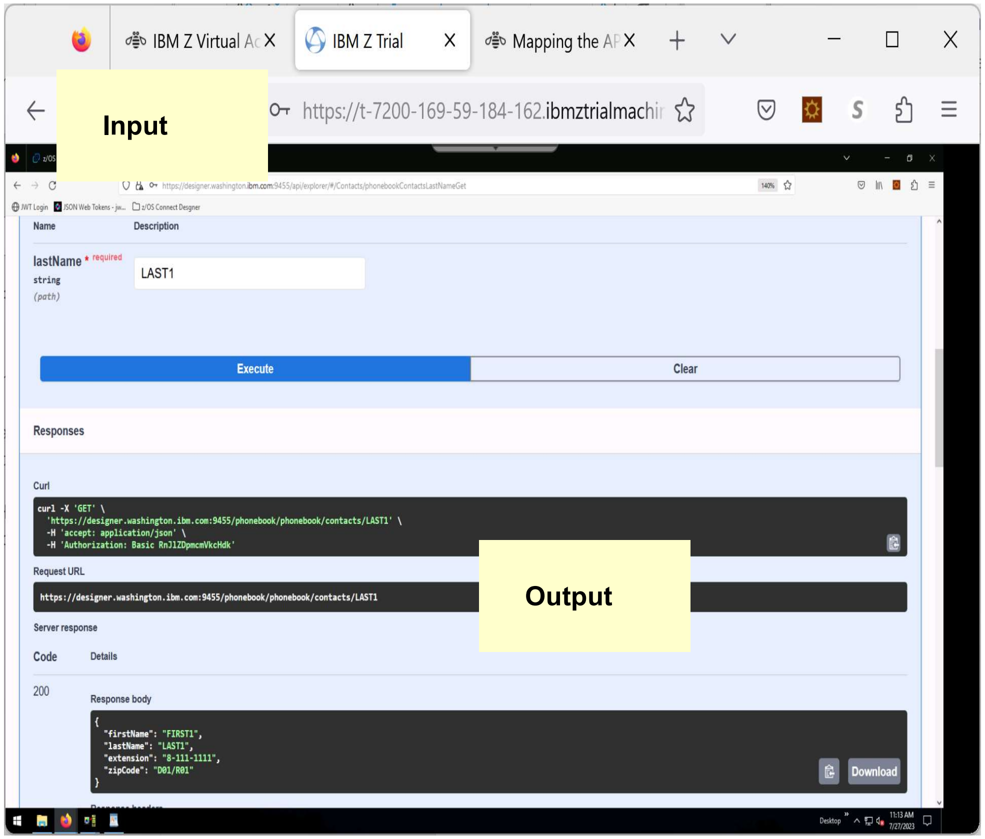
Servers
https://designer.washington.ibm.com:9455

Authorize

Contacts

- POST `/phonebook/phonebook/contacts` Add a contact to the phonebook
- GET `/phonebook/phonebook/contacts/{lastName}` Display a contact
- PUT `/phonebook/phonebook/contacts/{lastName}` Update a contact in the phonebook
- DELETE `/phonebook/phonebook/contacts/{lastName}` Delete a contact in the phonebook

Schemas



IBM Z Virtual Ac
IBM Z Trial
Mapping the AF
https://t-7200-169-59-184-162.ibmztrialmachi
https://designer.washington.ibm.com:9455/api/explorer/#/Contacts/phonebookContactsLastNameGet

Input

Name	Description
lastName * required string (path)	LAST1

Execute | Clear

Responses

Curl

```
curl -X 'GET' \
  'https://designer.washington.ibm.com:9455/phonebook/phonebook/contacts/LAST1' \
  -H 'accept: application/json' \
  -H 'Authorization: Basic RnJlZDpmcmVkcHdk'
```

Request URL

```
https://designer.washington.ibm.com:9455/phonebook/phonebook/contacts/LAST1
```

Output

Server response

Code	Details
200	Response body

```
{
  "firstName": "FIRST1",
  "lastName": "LAST1",
  "extension": "8-111-1111",
  "zipCode": "D01/R01"
}
```

Download

APIs and z/OS Connect Support for IMS

Click to edit Master title style

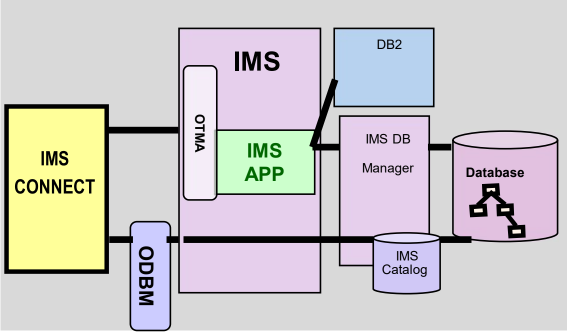
- Click to edit Master text styles
  - Second level
    - Third level
      - Fourth level
        - Fifth level

- OpenAPI 2 – z/OS Connect
  - ***IMS Service Provider***
    - Access to IMS transactions
    - Access to IMS large messages
  - ***IMS DB Service Provider***
    - Access to IMS databases
  - ***API Requester***
    - Access from IMS transactions

- OpenAPI 3: z/OS Connect
  - ***IMS Provider***
    - Access to IMS transactions
    - Access to IMS large messages

Today's focus

- IMS existing infrastructure functionality supports access from both specifications
  - ***IMS Connect*** (IMS TCP/IP gateway)
    - Configured to access transactions and/or databases
  - ***IMS***
    - *OTMA for transactions*
    - *ODBM for databases*
    - *Common Service layer*
      - SCI (structured call interface)
      - OM (Operations manager)



With the Ability to Control Connections and Interactions with IMS Connect

The **IMS connection profile** contains information about the IMS hostname, port number, and other security-related properties for connection to the specified IMS host system

id	Specify a unique ID for this connection profile.
connectionFactoryRef	Set this value to the ID of the connectionFactory element.
id	Specify an ID for the connection factory.
containerAuthDataRef	Set this value to the ID of the authData element. If the user ID from the originating client is to be sent to IMS Connect
hostName	Specify the hostname or IP address of the data store server (IMS Connect).
portNumber	Specify the port number that is used to connect to IMS Connect.
SSLEnabled	Indicates whether SSL is enabled for this connection factory.
SSLEncryptionType	Specifies the type of cipher suite to be used for encryption.
SSLProtocol	The SSL protocol to be used for encryption.
SSLKeyStoreName	Name (full path) of SSL keystore for client certificates and private keys..
SSLKeyStorePassword	Password of SSL keystore for client certificates and private keys.
SSLTrustStoreName	Name (full path) of SSL keystore for trusted certificates.
SSLTrustStorePassword	Password of SSL keystore for trusted certificates.

Auth data element:

id	Specify an ID for this authData element.
user	Specify the username to use to connect to IMS Connect.
password	Specify the encrypted password for the specified user. Use a tool such as securityUtility in Liberty to encrypt plain text.

- Click to edit Master text styles
  - Second level
    - Third level
      - Fourth level
        - Fifth level

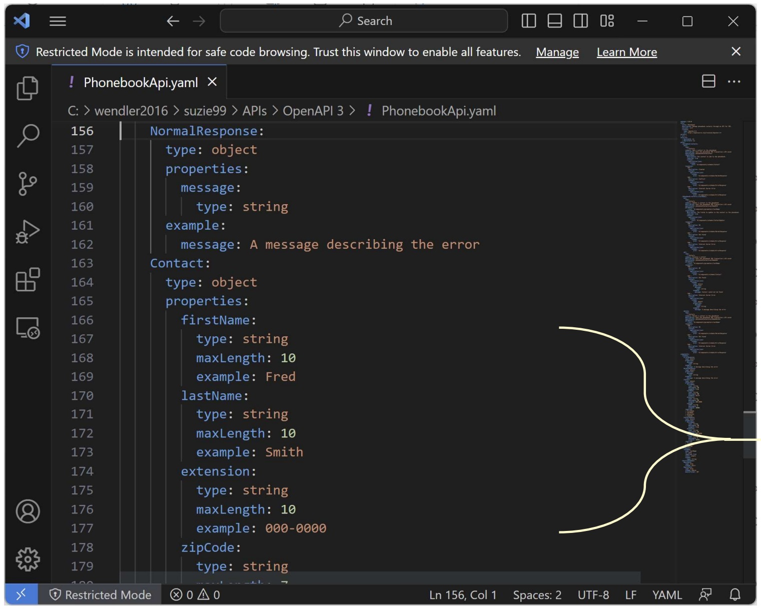
The **IMS interaction profile** specifies how each IMS transaction service interacts with IMS, such as the commit mode and sync level.

id	Specify the ID for this interaction profile. This is the value to specify in the z/OS® Connect API toolkit as the interaction profile name when configuring the service properties.
commitMode	Specify the commit mode. A value of 0 means commit-then-send (CM0); 1 means send-then-commit (CM1).
imsConnectTimeout	Specify the time in milliseconds to wait for a reply after sending a message to IMS Connect. Starting V3.0.2, the default is 30000, which means to wait for 30 seconds. Prior to V3.0.2, the default is 0, which means to wait indefinitely.
tranExpiration	Sets the TMRA IMSInteractionSpec property transExpiration. Accepted values for this attribute are “true” or “false”.To learn what these properties control, see the TMRA section of the *IMS documentation*.
propagateNetworkSecurityCred	Specify whether to propagate the network security credential if the IMS Connect is V15 or later. The default is true. The credential consists of the user ID and the network session ID (the realm) that are registered in the basic registry or SAF registry.
syncLevel	Specify the sync level. A value of 0 means None; 1 means Confirm. A commitMode value of 0 (CM0, Commit-then-send) is invalid with sync level 0 (None).
imsConnectCodepage	Specify the code page to use for character string conversion with IMS Connect. The default is Cp1047.
ltermOverrideName	Optional. Specify a LTERM name to override the value in the LTERM field of the IMS application program's I/O PCB.

Recommendation: always analyze the target IMS application environment for anomalies that might need to be considered

- From the API creation perspective, OpenAPI3 development requires the YAML request/response be matched to the input/ouput message Cobol copybook layouts

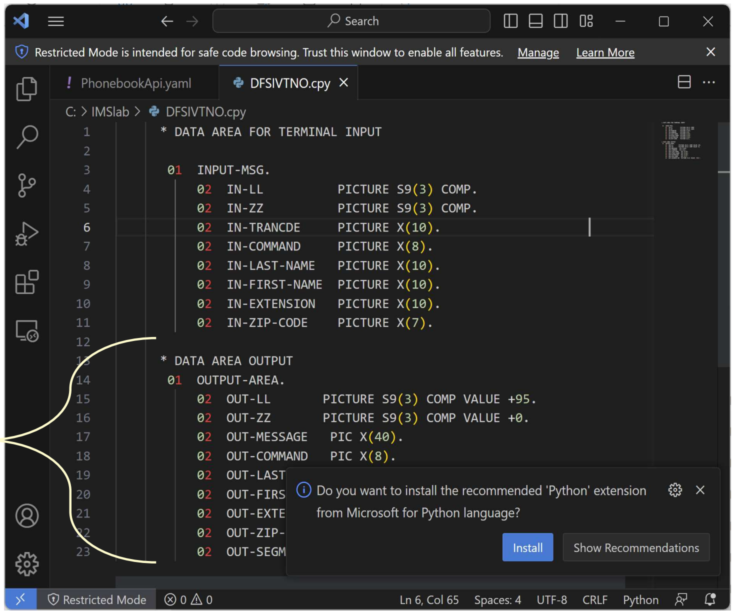
- From the IMS application perspective, which transaction(s) should be invoked for a valid interaction



Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

```yaml
    NormalResponse:
      type: object
      properties:
        message:
          type: string
      example:
        message: A message describing the error
    Contact:
      type: object
      properties:
        firstName:
          type: string
          maxLength: 10
          example: Fred
        lastName:
          type: string
          maxLength: 10
          example: Smith
        extension:
          type: string
          maxLength: 10
          example: 000-0000
        zipCode:
          type: string
```

Meet-in-the-middle



Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

```cobol
      * DATA AREA FOR TERMINAL INPUT

       01  INPUT-MSG.
           02  IN-LL          PICTURE S9(3) COMP.
           02  IN-ZZ          PICTURE S9(3) COMP.
           02  IN-TRANCDE     PICTURE X(10).
           02  IN-COMMAND     PICTURE X(8).
           02  IN-LAST-NAME   PICTURE X(10).
           02  IN-FIRST-NAME  PICTURE X(10).
           02  IN-EXTENSION   PICTURE X(10).
           02  IN-ZIP-CODE    PICTURE X(7).

      * DATA AREA OUTPUT
       01  OUTPUT-AREA.
           02  OUT-LL       PICTURE S9(3) COMP VALUE +95.
           02  OUT-ZZ       PICTURE S9(3) COMP VALUE +0.
           02  OUT-MESSAGE   PIC X(40).
           02  OUT-COMMAND   PIC X(8).
           02  OUT-LAST
           02  OUT-FIRS
           02  OUT-EXTE
           02  OUT-ZIP-
           02  OUT-SEGM
```

Do you want to install the recommended 'Python' extension from Microsoft for Python language?

How Simple or Complex is the IMS transaction interaction?

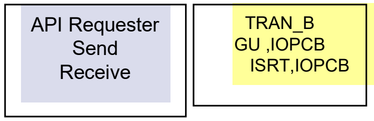
Ensure that someone who knows the architecture of the application is involved in the process

Otherwise, you can run into problems that could/should have been avoided

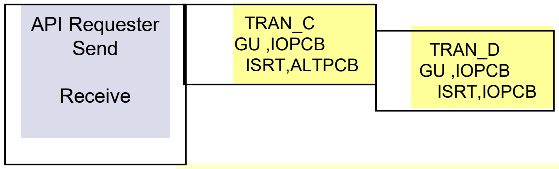
Simple

— Considerations – The IMS Provider uses CM1 Send-Commit (tpipe is the port number)
Can use CM0 Commit-send

- **Simple interactions**
  - Straightforward API implementation using the API Toolkit



Input and output layout for the Tool are from TRAN_B



Input layout for the tool is for TRAN_C
And output layout is from TRAN_D

- **AND:**
  - *IMS transaction is non-conversational*
  - Cobol programs provide the input/output message layouts (copybooks or include files)
  - Any program-to-program switches are done with
    - Non-express ALTPCBs
    - Final switched-to transaction responds to the IOPCB
    - No spawning of multiple transactions – switching is simply from one transaction to another

Complex – Example 1

- **What could cause complexity in this environment?**
  - No available copybooks to provide the toolkit to build the services?

    Analysis will need to be done of the input/output IMS messages to ***create the layouts***

  - What about having to determine the flow of the application business logic when multiple transactions are involved in ***program-program*** switches?

    The exposure to REST API implementation means that the ***input layout needs to be tied to the final output*** *to determine which output message from a switched-to transaction matches the original input message*

    - IMS application architects should be involved to determine this flow

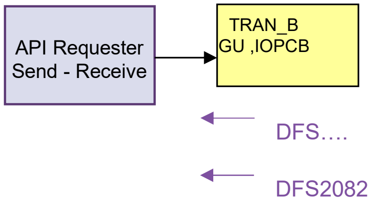
- Possible resolutions:
  - Execute the transaction flow and analyze
    - 01/03 log records

      Run DFSERA10 / DFSERA30 and assemble the IMS Log record DSECTs

    - Or: take a trace and analyze the trace records
    - Or use tooling

Complex – Example 2

- If there are problems in the IMS environment, or the IMS transaction abends, or the transaction does not reply to the IOPCB

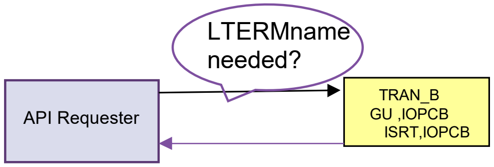


- The REST API may receive an unexpected response
  - If the target transaction does not ISRT to the IOPCB or ALTPCB
    The API requester will get a DFS2082
  - Or, If the target transaction experiences an error/problem on the IMS side
    The API requester will get a DFS... message

- Solution
  - Design the REST API with a catch-all error response code, e.g., 500, and return the output message

Complex – Example 3

— The IMS transaction environment has application security or needs an LTERMname



- LTERMnames may be required by the application(based on older 3270 environments)
  - Determine if a generic LTERMname can be used
    - *Define it in the connection profile in z/OS Connect*

- Otherwise, some possible solutions:
  - Define the value needed for LTERMname in the REST API input message (e.g., LLZZ trancode data ***overridename***)

Specify a dummy value in the LTERMname value of the connection profile

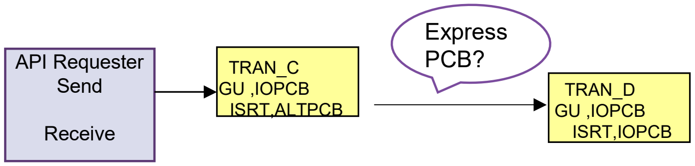
Write an IMS OTMA exit (DFSYIOE) or IMS Connect exit (HWSJAVA0) to recognize the dummy value from the connection profile and:

- *Remove the overridename from the input message*
- *Use the value to pass it in as an LTERMname*

Other possibilities would depend on the specific customer environment

Complex – Example 4

- The IMS transaction program-program switch uses an Express PCB



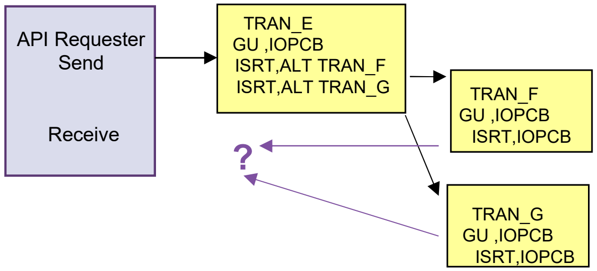
  - The REST API request will hang

- Solution:

  - Change the Express PCB to Non-express PCB

Complex – Example 5

— The target transaction spawns multiple transactions - which should respond back to the API requester?



API Requester
Send

Receive

TRAN_E
GU ,IOPCB
ISRT,ALT TRAN_F
ISRT,ALT TRAN_G

TRAN_F
GU ,IOPCB
ISRT,IOPCB

TRAN_G
GU ,IOPCB
ISRT,IOPCB

?

ETC.,.....

- If the target transaction spawns multiple transactions, only one IOPCB response can be sent to the API requester

- Solution
  - **Determine which transaction IOPCB message should reply to the request**
    - **OTMAASY = Y | N | S** parameter

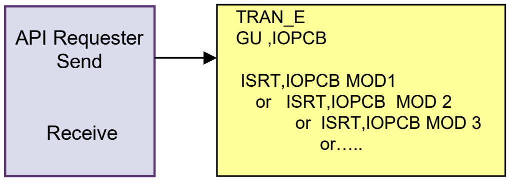
Controls which transactions is switched to synchronously (eligible to reply to the request)

Y – First RESPONSE program scheduled after the switching transaction ends is scheduled synchronously
N – First program (RESPONSE or NONRESPONSE) scheduled after the switching tran ends is scheduled synchronously
S – The first message ISRTed to a non-Express ALTPCB will be scheduled synchronously

➢Note: the other IOPCB messages will become asynchronous output and held

Complex – Example 6

— The target transaction replies using one of many possible output layouts



API Requester
Send

Receive

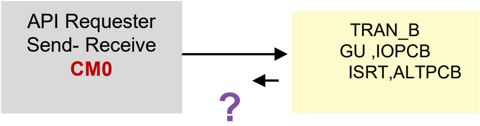
```
TRAN_E
GU ,IOPCB

ISRT,IOPCB MOD1
   or   ISRT,IOPCB  MOD 2
          or  ISRT,IOPCB MOD 3
                or.....
```

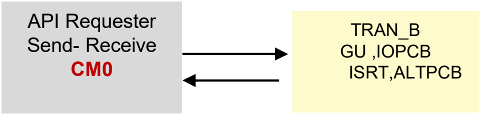
- ○ Existing 3270-based transactions may specify extended attributes in the program
  Extended attributes are 2-bytes per message field, e.g., color, highlight, etc. that affect the output message layout
- ○ Note that if the program specifies these attributes, they will be included as part of the application output message and will require the API requester to recognize them as such

- Possible actions
  - ○ Review the architecture with program architects
  - ○ Determine the possible paths/layouts that can be used
  - ○ Explore the ability to create a copybook for the API toolkit which uses Redefines
  - ○ Determine if a more generic output message layout to include all the attributes can be used

Something New - SENDALTP

— The target transaction replies only to the ALTPCB (not the IOPCB) and the API expects the reply





PH39434 – Provides SENDALTP support which allows the output to be sent back on the inputting TPIPE

- Considerations for PH39434
  - No message-level support in the IMS Service Provider – RFE would be needed
  - Alternatively – SENDALTP can be specified in the HWS or DATASTORE statement of IMS Connect HWSCFGxx

```
Example:  DATASTORE = (ID= IMS1, GROUP=XCFGROUP, MEMBER=HWS1, TMEMBER=IMS1, APPL=APPLID1...)
          DATASTORE = (ID= IMSX, GROUP=XCFGROUP, MEMBER=HWS1, TMEMBER=IMS1, APPL=APPLID1.., SENDALTP=Y)
```

> Either modify an existing DATASTORE statement, or create a new statement with SENDALTP so that only requests using this path will be affected

> OTMA detects that the message coming in through IMS Connect using the DATASTORE statement will have the specific ALTPCB output in the above scenario sent back through the inputting TPIPE

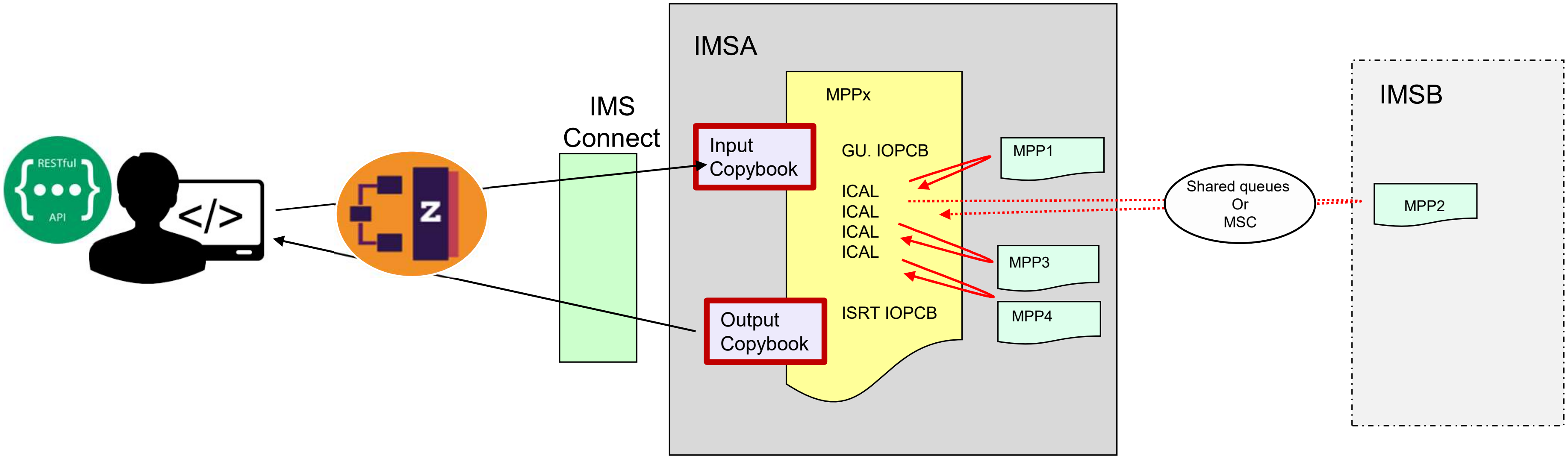
- Destination descriptor or DRU exit specifies SENDALTP

  - Reference the following for description and restrictions: https://ibm.biz/BdPncC

And Even Consider ...

— Creating a front-end transaction using the IMS synchronous program switch (DL/I ICAL) to access existing transactions

- Provides an internal service flow of IMS transactions to synchronously participate in a business process
  - In the same IMS or even across different IMS systems (Shared queues or MSC)
- Enhances the environment in lieu of re-coding



Overall Recommendations

- On the IMS side, ensure that the application architecture is understood
  - Determine if the application is ‘simple’
  - Anticipate complexity or potential show-stoppers
    - Determine if it would be easier to write/modify the IMS applications

Possibility: Orchestration transaction that uses IMS synchronous Program Switch support to invoke existing transactions

- For problems,
  - Log records, traces, or tools

Tooling

IMS Connect / IMS / Application

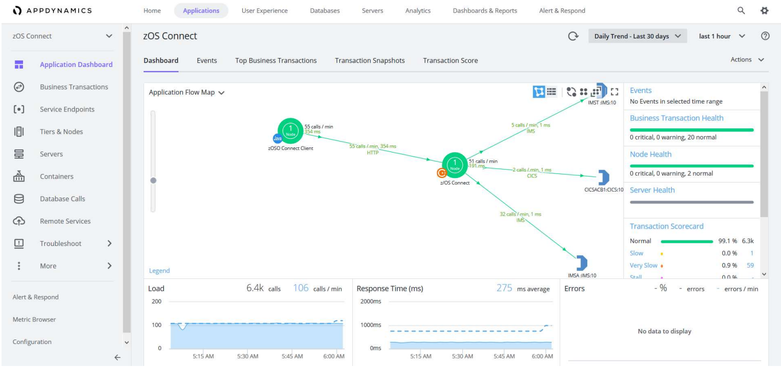
Journaling and control: IMS Connect Extensions (IMS CEX)
Problem determination: IMS Problem Investigator (IMS PI)
Performance: IMS Performance Analyzer (IMS PA)
Omegamon for IMS
Application workflow: ADDI

z/OS Connect

Omegamon for JVM – can monitor z/OS Connect API utilization

End-to-End tracking

IBM z APM Connect - Can be used to determine/ target problem areas



Events
No Events in selected time range

Business Transaction Health
0 critical, 0 warning, 20 normal

Node Health
0 critical, 0 warning, 2 normal

Server Health

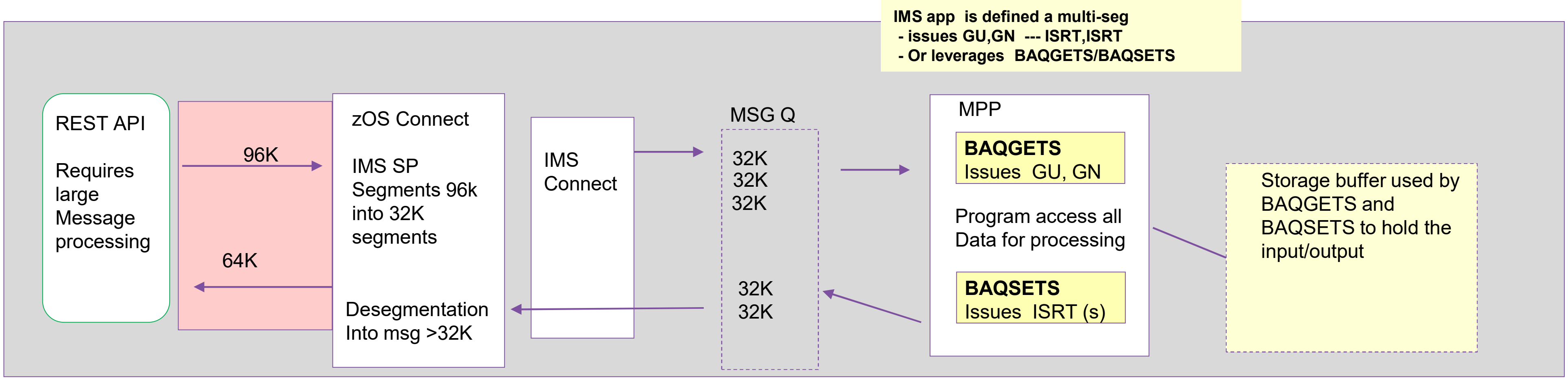
Transaction Scorecard

Normal	99.1 %	6.3k
Slow	0.0 %	1
Very Slow	0.9 %	59
Stall	0.0 %	-

No data to display

Support for Large Data Structures

- zOS Connect
  - Supports large data structures (>32K IMS standard segmentation)
    - IMS applications can invoke the supplied sample COBOL utilities BAQGETS and BAQSETS instead of interacting with the IMS message queue directly (GU, GN inbound and ISRT, ISRT outbound)
    - Sample segmentation utilities that are bound with the Cobol application



IMS app is defined a multi-seg
- issues GU,GN --- ISRT,ISRT
- Or leverages BAQGETS/BAQSETS

REST API
Requires large Message processing
96K
64K
zOS Connect
IMS SP Segments 96k into 32K segments
Desegmentation Into msg >32K
IMS Connect
MSG Q
32K
32K
32K
32K
32K
MPP
BAQGETS
Issues GU, GN
Program access all Data for processing
BAQSETS
Issues ISRT (s)
Storage buffer used by BAQGETS and BAQSETS to hold the input/output

Large IMS data structure support is automatically enabled when the IMS provider detects that an API message created by the Designer is based on an input or output transaction message that consists of a single segment whose data structure has a maximum length that exceeds 32 KB.

Large Data Structures - Recommendations

- Review the IMS application environment to determine the value proposition
  - If the remote program sending the API request simply need to sends a large unsegmented message (larger than 32K) or retrieve a large unsegmented message
    - If the IMS application is already multi-segment, then the use of this support could allow the IMS application program to remain unchanged

- Ensure the IMS application is multi-segment
  - E.g., Modify the program if it is currently single segment to issue GU, GN calls
  - Or, replace the IMS message calls with the BAQGETS/BAQSETS

- Understand the impact of BAQGETS/BAQSETS
  - ***Only use these if:*** the 'large' message is >32K
    - Large data structures sent in through the API must not begin with an LLZZ or LLZZ<*TRANCODE*> prefix. The BAQGETS utility generates segment prefixes with the first segment prefix set to the transaction code that is specified in the API toolkit

So, In Summary…

OpenAPI 2 --> Open API 3 is not intended to be a migration path
Rather, it is a choice based on the use case

- Remember:
  - **Your environment may have a need for both specifications**
- Some useful links:
  - **Learn about YAML**
    - > *https://www.tutorialspoint.com/yaml/yaml_basics.htm*
  - **Learn more about z/OS Connect**
    - > *https://www.ibm.com/products/zos-connect*
  - **Learn more about the z/OS Connect Designer**
    - > *https://www.ibm.com/docs/en/zos-connect/zos-connect/3.0?topic=developing-apis-zos-connect-designer*
  - **Convert a Swagger (Open API 2) document to Open API 3**
    - *https://converter.swagger.io/#/*
  - **Validate and/or view an Open API 3 document**
    - *https://jsonformatter.org/yaml-viewer*
  - **IMS tutorial**
    - *https://www.ibm.com/docs/en/zos-connect/zos-connect/3.0?topic=tutorials-creating-ims-zos-connect-api*