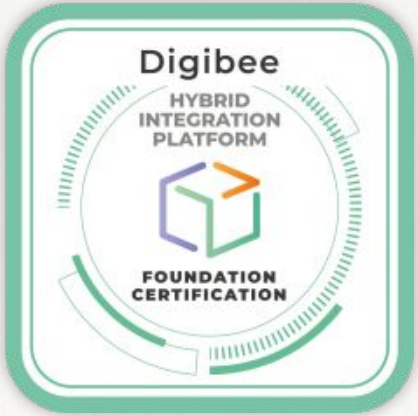
digibee Academy

Foundations

digibee Academy

This initiative aims to help professionals build knowledge in integration contexts.

We have applied our methodology, **Simplexity Mindset,** to enable these professionals to have a different view of integrations and take control of their data flows, which are their most important assets.



Foundation Certification

Foundations is Digibee's first certification. It presents basic concepts that allow you to start creating your integrations.

Agenda

Digibee **01**
- The integration challenge
- Offer
- Benefits

Demo Case **02**
- Business Case
- Demo and Navigation

General Concepts **03**
- JSON
- Connectors
- Functions

Hands-on Practice **04**
- Practical Case for building an integration
- Step by Step
- Case Examples

Closing **05**
- Satisfaction survey

01 Digibee

The Integration Challenge

System Integration and API exposures are complex tasks that are heavily supported by people, IT processes and tools to connect on-prem and cloud systems.

Traditional Business

On-Premise

SAP ORACLE IBM

Proprietary Protocols

Built-In Security

INTEGRATION GAP

Analysts – Project Managers - Network - Security - Developers - QA - DevOps - Operations - Support - Experts - 3rd Parties - etc...

Functional | Connectivity | Acess | Coding | Debug | Testing | Code Sec | Deploy | Run | Monitoring | Operation | Troubleshoot | Evolution | Rebuild

Digital Ecosystems

Cloud / Mobile

Open Protocols

Internet Security

Unique Approach to Solve the Problem

An end-to-end integration **platform** and a **model** to connect businesses and integrate systems and services using a modern, simple and digital approach.

Traditional Business

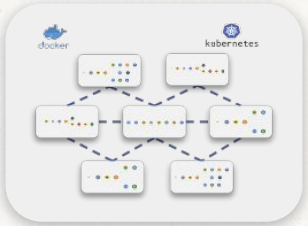
On-Premise

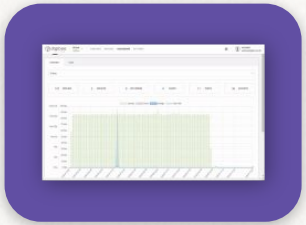
SAP ORACLE IBM

Proprietary Protocols

Built-In Security

Integration Model





BUILD EXECUTE OPERATE

Digibee HIP - Hybrid Integration Platform

Digital Ecosystems

Cloud / Mobile

servicenow intuit salesforce

Open Protocols

Internet Security

Digibee

What is Digibee?
A Hybrid Integration Platform (HIP)

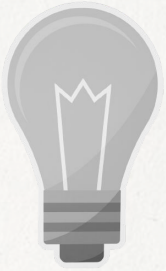
Hybrid Integration Platform (HIP)
It integrates systems hosted anywhere with resiliency, performance, scalability and security

DIGIBEE vision
The main issue in digital enterprises is **INTEGRATION**

Offer
Digibee Connect businesses and integrate systems and services using a modern, digital and simple approach

Benefits
Teams have more time to focus on what really matters, adding value to the business

It’s very hard to develop and manage integrations!



Not everyone wants to know how electricity is brought to a light bulb; we just want it lit!

Digibee does just that with data. It takes care of your integration process so that it runs smoothly and meets its purpose.

digibee

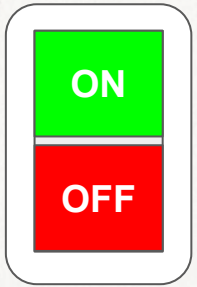
We need to know where the data is

We need to know why an integration has stopped

We need to monitor data

We need to know if the destination (consumer) received the data

We need to know where it stopped; with whom it stopped



Digibee's Offer

VALIDATE → ENRICH → TRANSFORM → DELIVER

VALIDATE	ENRICH	TRANSFORM	DELIVER
Disruptive and innovative, it simplifies processes and brings a new reality to the IT team	Reduces the build time 10 to 15 times	Cuts costs: several integration pieces are substituted	Teams have more time to focus on what really matters, adding value to the business

Add value to the BUSINESS

This is exactly our concept for an INTEGRATION PLATFORM

Invitation

From: A business that understands, breathes and lives system INTEGRATION

To: Professionals that will work with system integration

Integration is the next booming market for IT professionals!

Gartner: “Large enterprises will implement HIP in the next few years”

Data, once siloed, now can be easily, more effectively and directly available, generating more business.

Companies need to integrate data from internal, on-premise or cloud-based information with modern and specialized systems.

End users expect to have their IT experience in real time.

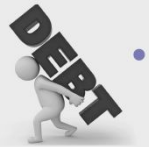
It is a great opportunity for the system integration professional.

A great opportunity for the system integration professional

- A platform like Digibee will leverage professional performance.
- The integration professional will be just like the old DBA (essential owner of the data).
- Managing and controlling data streams, as well as what you can do with it, is what markets value now.
- Thus, it is a great opportunity to become a system integration professional!

$$$ In this context, the integration professional will have more value.

The tech professional will not be in debt anymore...



... and will become the **ENABLER** of new services, products and business

02 Demo Case

Scenario: Customer Registration Flow

Imagine you are launching an APP that needs to register new users. The interface is ready and now the backend team needs to process application requests and save new-user information.

In this demo case we will practice:



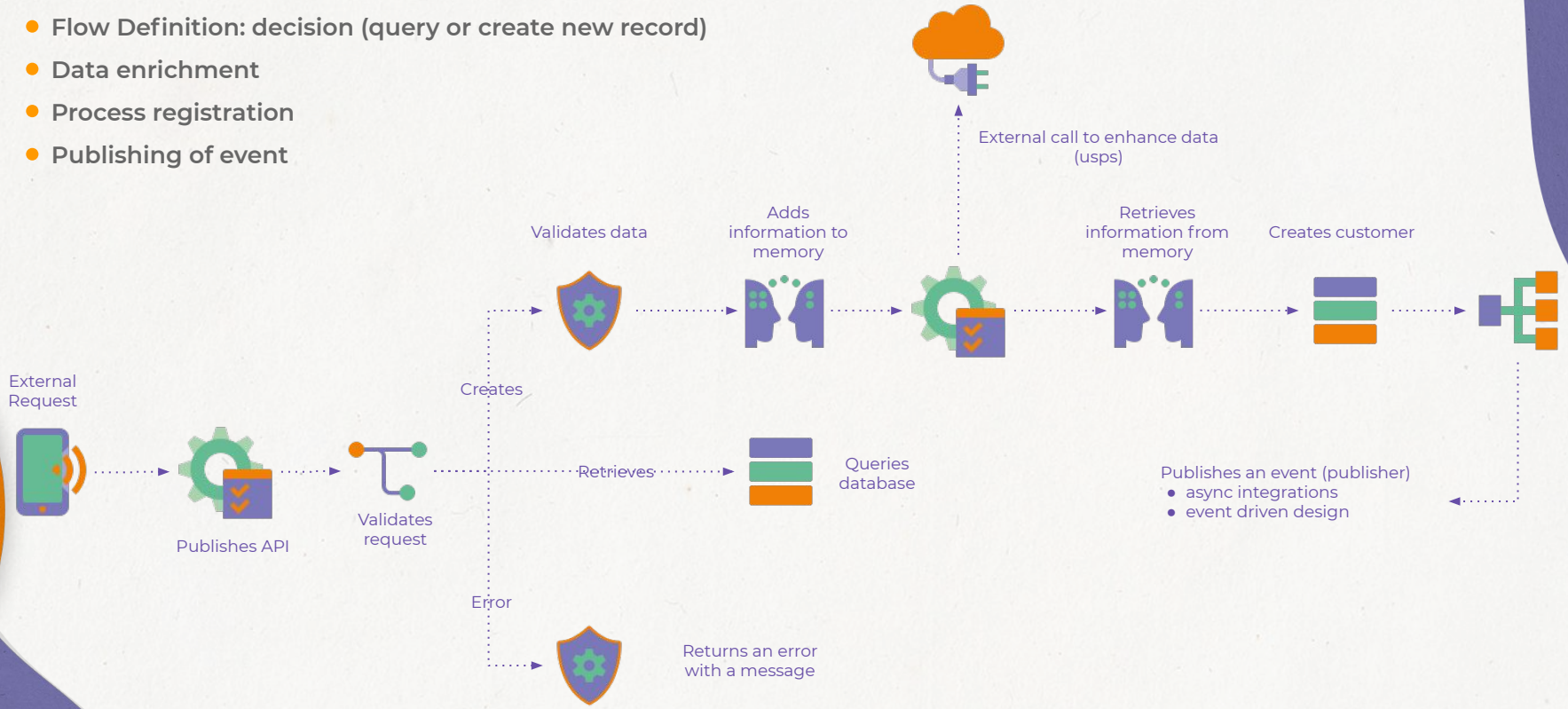




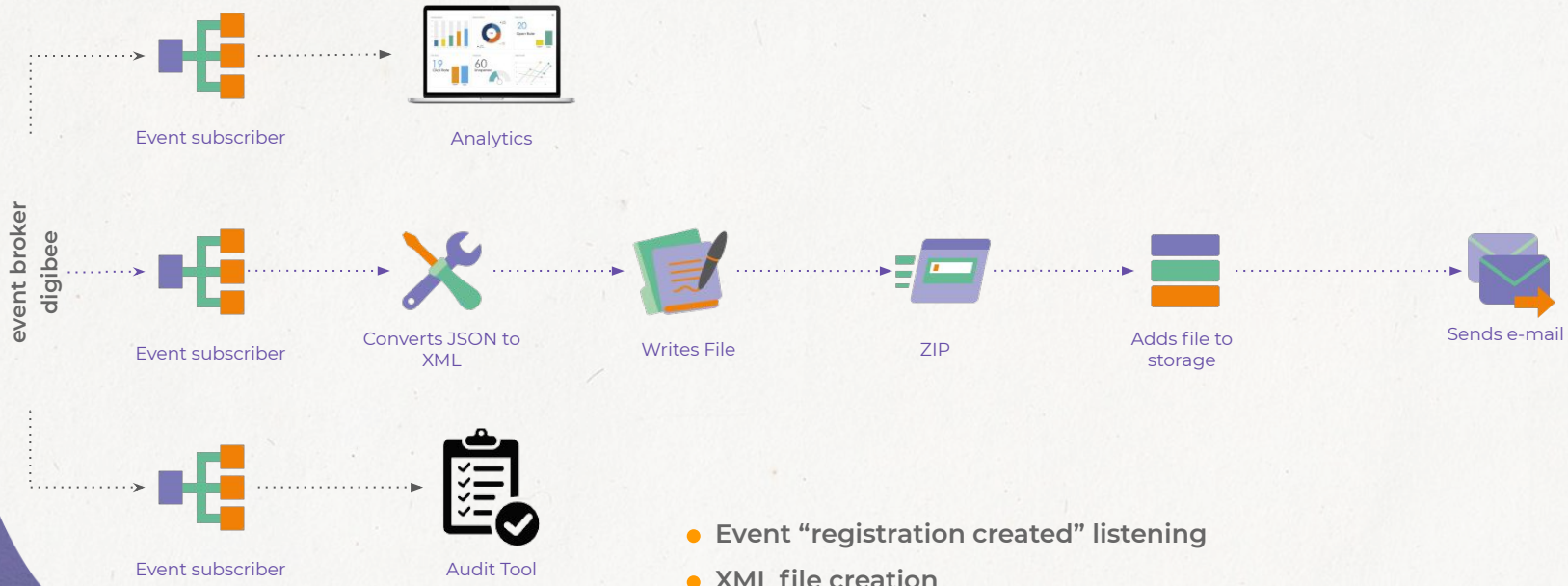
- Exposure of services through APIs
- Conditional flow branching
- Data validation
- Data enrichment
- Event publishing
- Event subscription
- Error and Exception Handling

Scenario: Customer Registration Flow

- Receiving of an external request from a published API
- Flow Definition: decision (query or create new record)
- Data enrichment
- Process registration
- Publishing of event

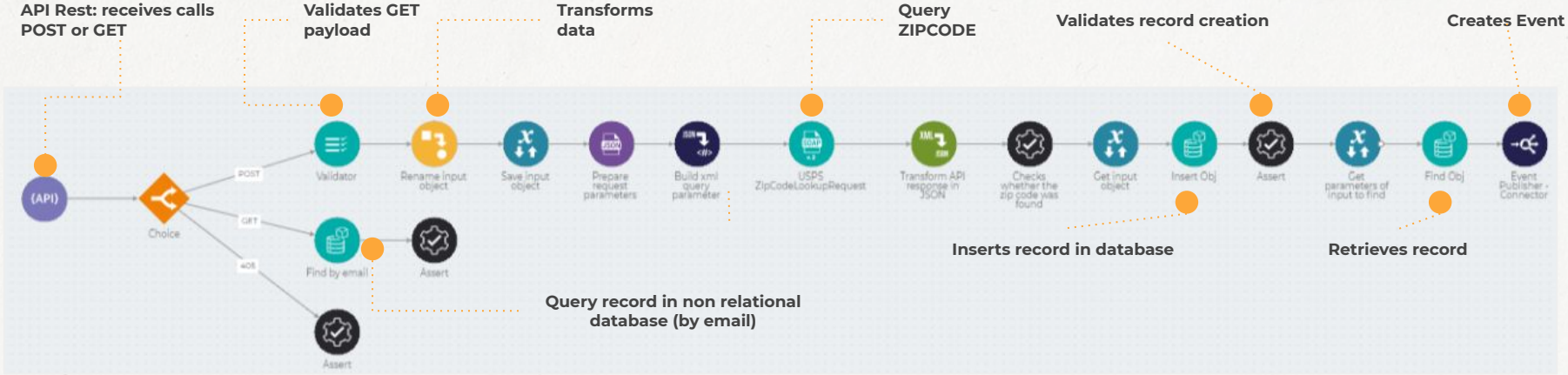


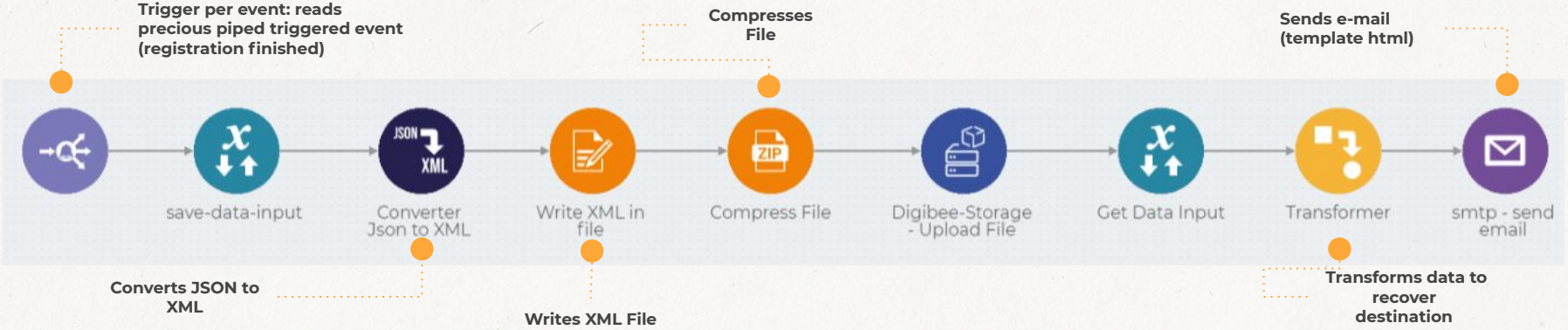
Scenario: Customer Registration Flow - Event Subscription



- Event “registration created” listening
- XML file creation
- Email sending and storage recording(database)

The Pipeline and its Connectors

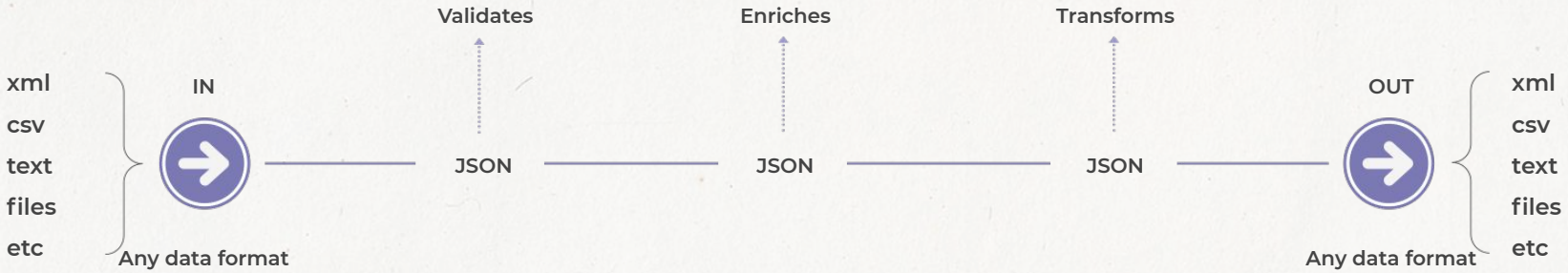




03 General Concepts

How does DIGIBEE accelerate development?

- It is a drag-and-drop, low code model for creating flows
- It abstracts and encapsulates market technologies in connectors
  e.g. SOAP, REST, DB, SAP RFC, encryption, transformation, file manipulation, etc.
- Digibee connectors transform the data into JSON and, after transforming and enriching it, deliver it with the expected format at the destination.



JSON Format

```
{
    "employees":[
        {
            "id": 1,
            "name": "Darth Vader"
        },
        {
            "id": 2,
            "name": "Luke Skywalker"
        },
        {
            "id": 3,
            "name": "Darth Vader"
        }
    ]
}
```

- A collection of name/value pairs
  - structure between braces, also called **OBJECT**
- A list of objects
  - bracketed structure, also called **LIST** or **ARRAY**

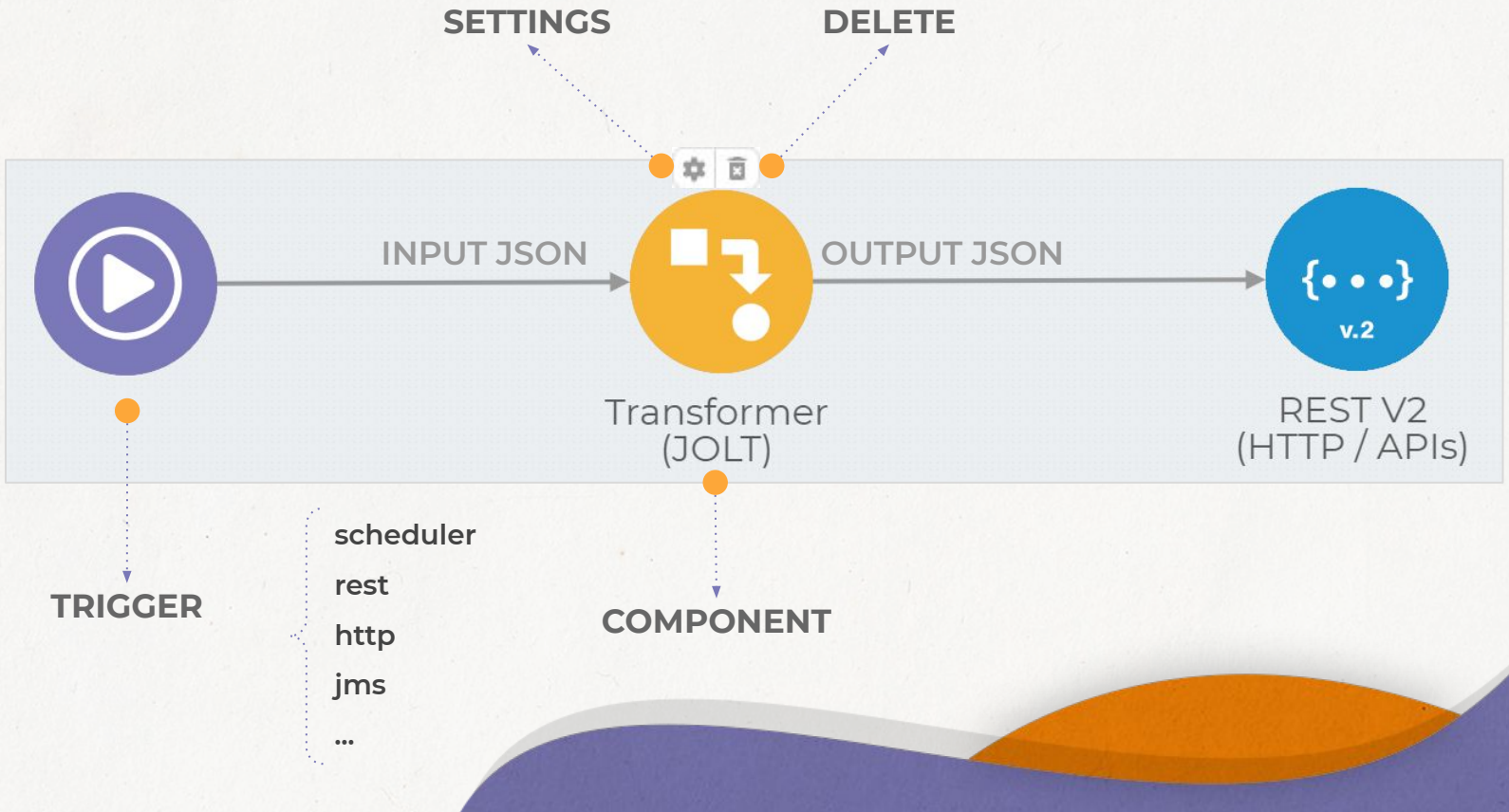
":" separates name/value pairs

"," separates objects and name/value pairs

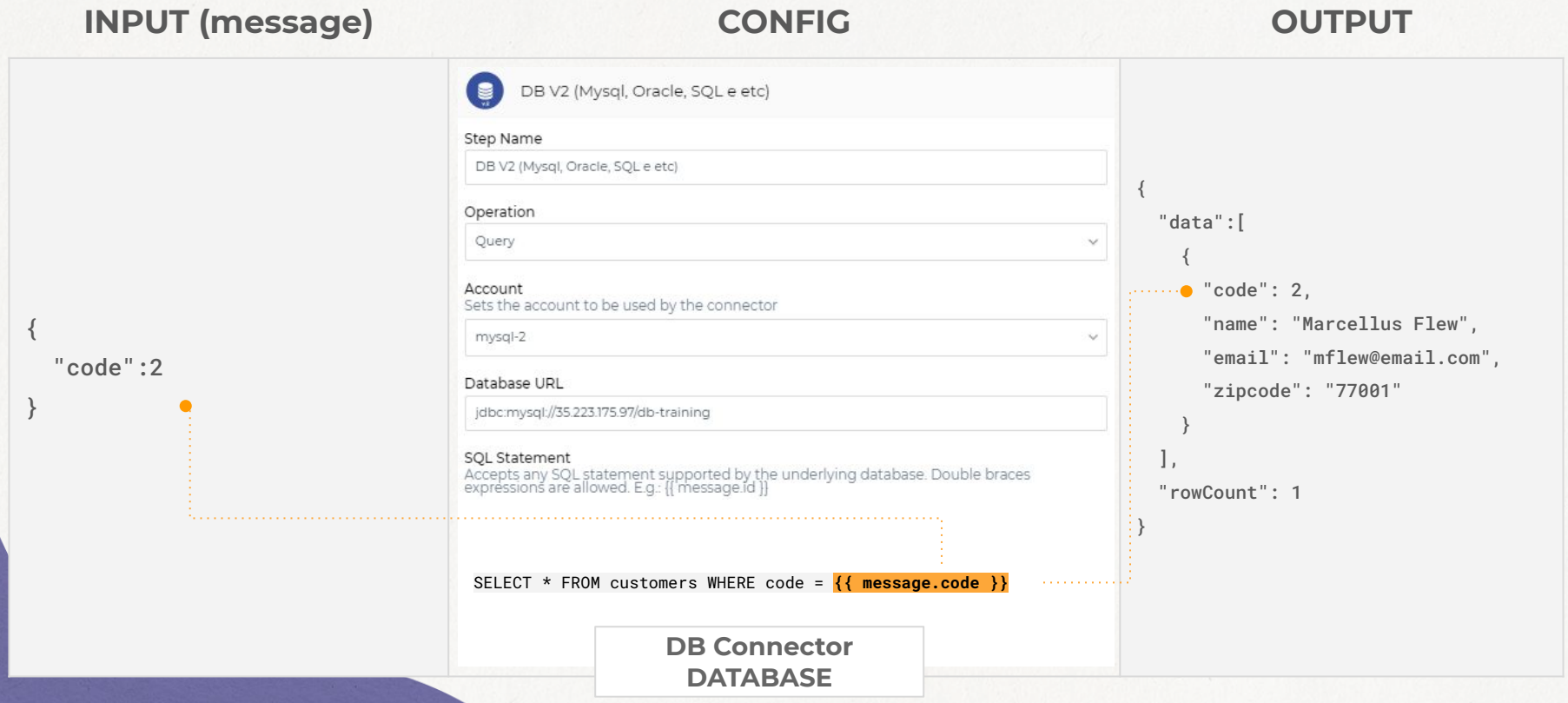
"{" and "}" for objects

"[" and "]" for lists

Components and Connectors - Settings and Usage



Components and Connectors - Variables

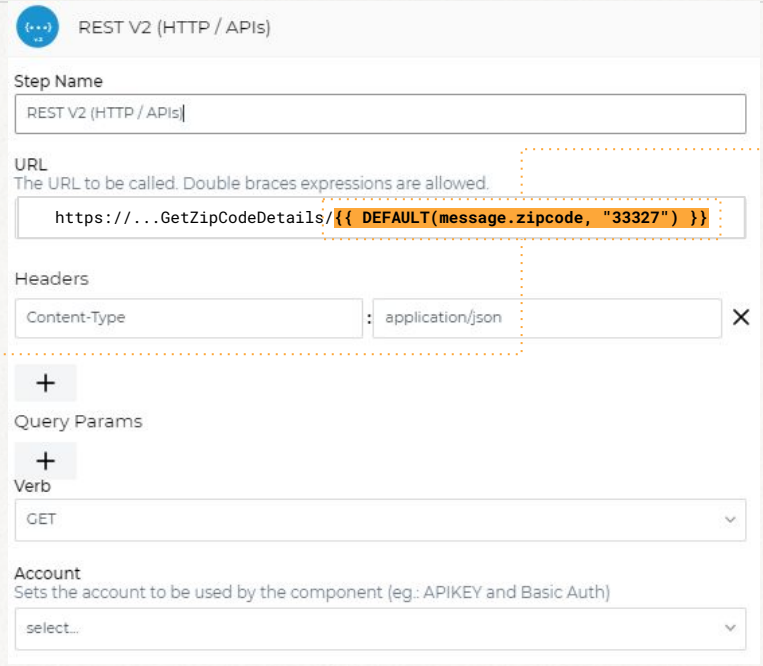


Components and Connectors - Functions

INPUT (message)

```
{
  "zipcode": null
}
```

CONFIG



REST Connector
REST API Request

OUTPUT

```
{
  "status": 200,
  "body":{
    "City": "FORT LAUDERDALE",
    "State": "FL",
    "Latitude": "26.119065",
    "Longitude": "-80.415078",
    "ZipCode": "33327",
    "County": "BROWARD"
  },
  "headers": {
    ...
    "Access-Control-Allow-Origin": "*",
    "Access-Control-Max-Age": "86400"
    ...
  }
}
```

04 Hands-on Practice

Hands-on Practice: Publishing a REST Service

Imagine that you are releasing an APP that needs to have the ability to retrieve customers and their respective addresses from a database. However, there is no query service available for that.

Given a database with a table that stores customer records, our goal is:

1. Give a code, retrieve the respective customer from the database;
2. Return a warning message to the user if the given code does not return any customers;
3. Retrieve the customer location from a ZIP Code using `https://api.godigibee.io/pipeline/digibee/v1/postalcode` query API;
4. Publish this service as a REST service (API), which returns the response in the following format:

```
{
    "customer": {
      "name": "<customer name>",
      "email": "<customer email>",
      "location": {
        "state": "<customer state>",
        "county": "<customer county>"
        ...other location info...
      }
  }
```

Hands-on Practice: Publishing a REST Service

In this scenario we will practice:

Displaying Services as a REST API

Conditional flow branching

Logs Usage

Data enrichment

Data storage in runtime session

Error Handling

Passing variables to components

Navigation of JSON objects with Double Braces

Pipeline Deployments

Using API Key for Security

Pipeline operations: logs, runs, and reruns

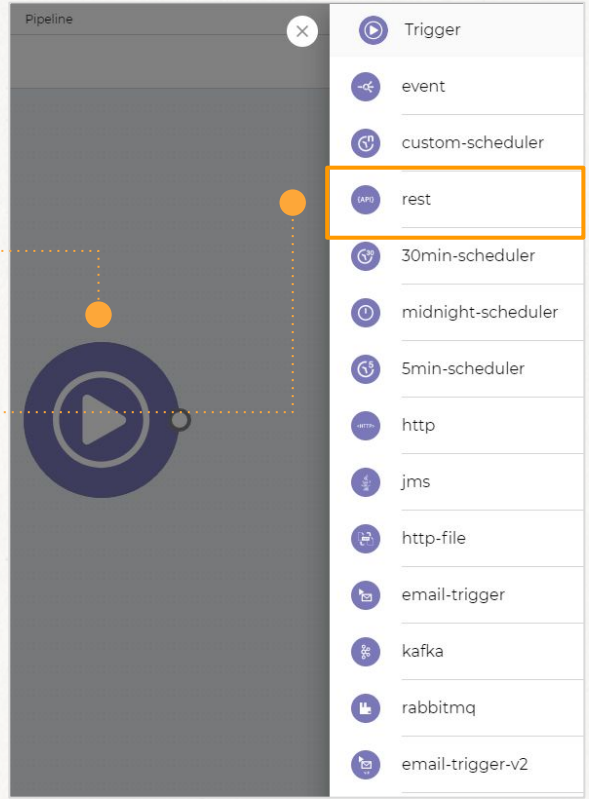
Use of accounts and password vault

Use of global variables

How to consume a deployed service

Hands-on Practice: Publishing a REST Service

Setting the trigger as a REST API, as shown in the image:



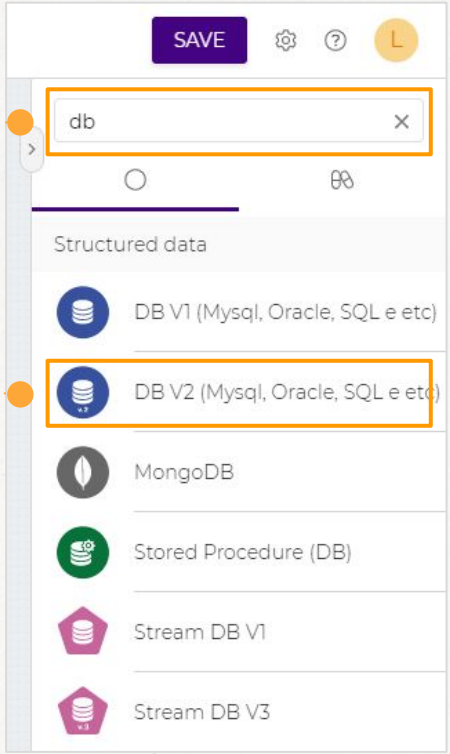
- The trigger is what makes the pipeline start running. In our example, after the pipeline is deployed, the trigger will be a REST API request.

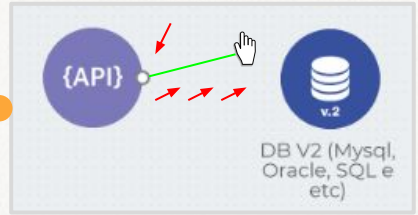
- Since we are still developing, we simulate the receipt of these parameters in test mode.

Hands-on Practice: Publishing a REST Service

Querying a database:

- **Search for DB** in the connector scan palette.
- Drag the **DB-V2** component to the canvas.
- **Connect** the connector to the trigger.





Hands-on Practice: Publishing a REST Service

Querying a database:

- Click twice on the **DB** connector to open the **configuration screen.**

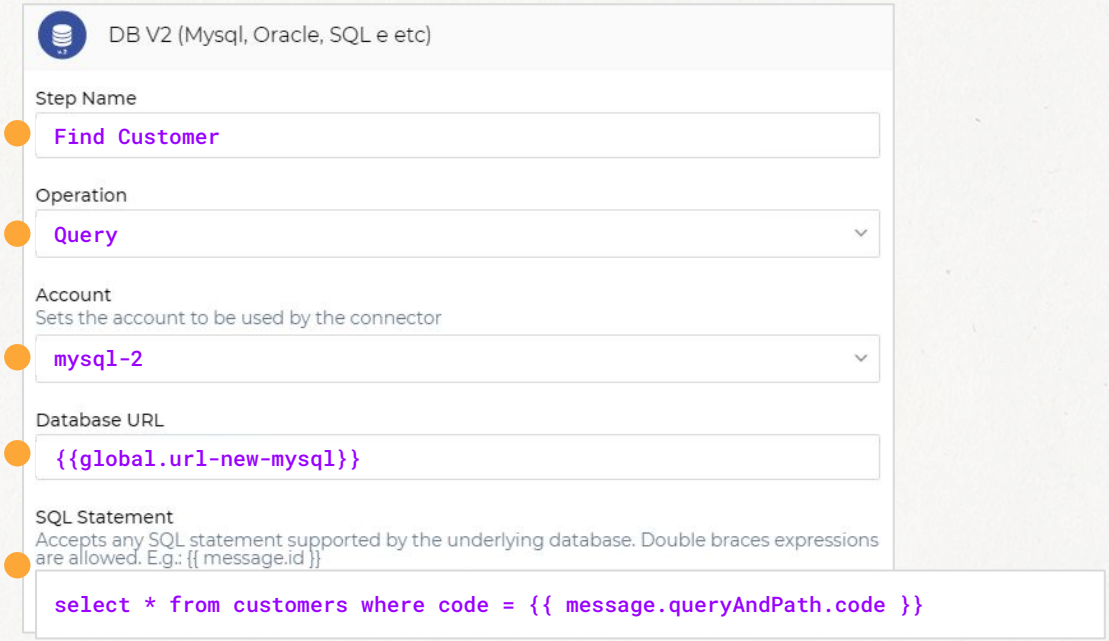
In **Step Name,** type **Find Customer**

Choose **Query** for **Operation**

Configure DB auth credentials in **Account**

Use a global variable at **URL**

Configure the **Query**



Hands-on Practice: Publishing a REST Service

Performing the first test:

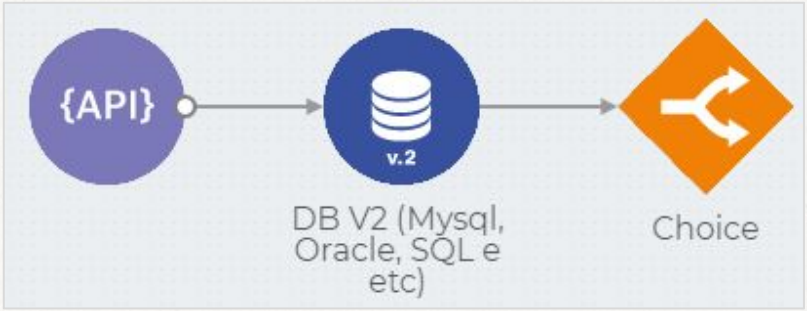
- Click **Test** mode at the bottom to simulate the request data that the API will receive.
- Fill the input field (left column at test mode console)

- Press **Ctrl+Enter** or click PLAY **to run.**
- The output should return content similar to this (right side of the sample test mode console).

Hands-on Practice: Publishing a REST Service

Flow deviation / Condition:

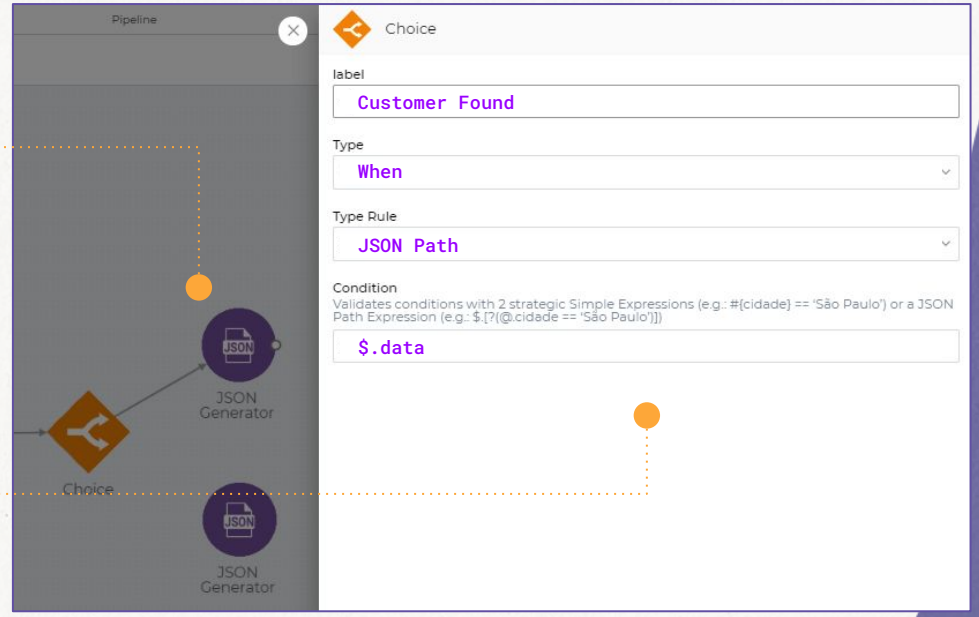
- In the connector palette, we search for **Choice** and drag it to the canvas. This connector alone does not require configuration.
- The moment we connect it to the next connector, the configuration form is displayed and we define the conditional deviation rule.
- This connector is often used to take a different path in integration, whether or not it meets certain conditions.



Hands-on Practice: Publishing a REST Service

Flow deviation:

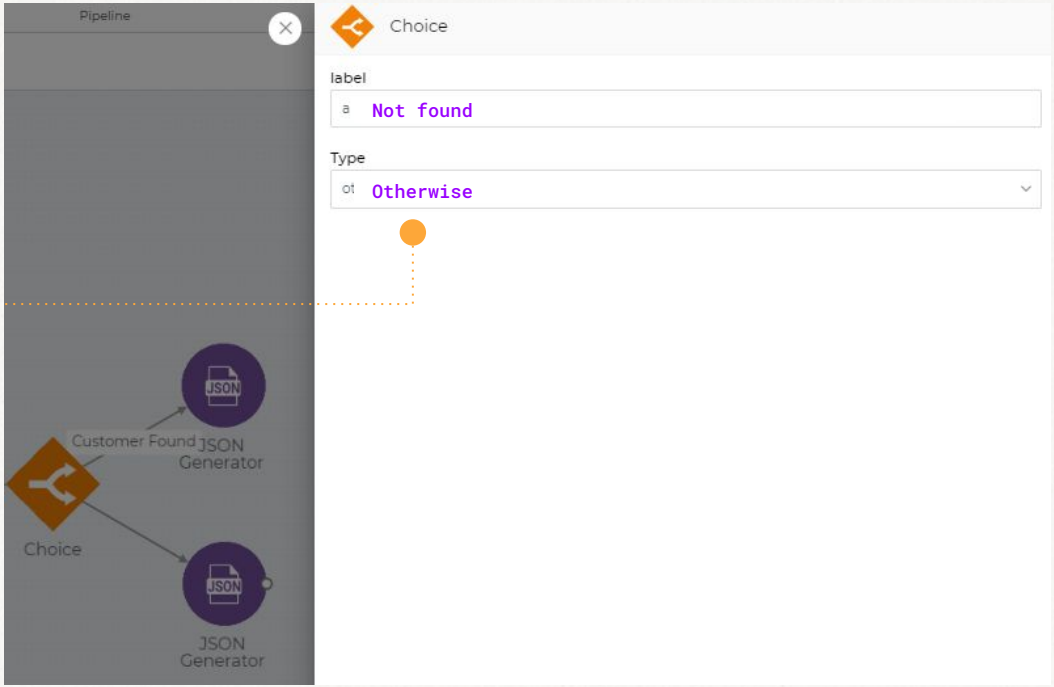
- Search and drag two **JSON Generator** connectors to the canvas.
- In **StepName**, name them **Transform Customer** and **Error Message**.
- Connect **Choice** to **Transform Customer**. The configuration screen for this flow deviation will be displayed (condition).
- Fill the **Label** field with **Customer Found** and the **Condition** field with **$.data.**
- The pipeline will only follow this path if there is an item in the data object.



Hands-on Practice: Publishing a REST Service

Flow deviation:

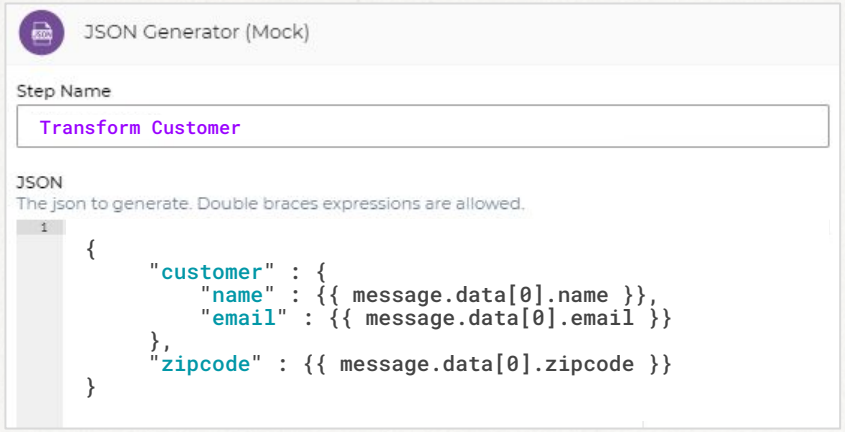
- Connect **Choice** connector to the other **JSON Generator**.
- Fill the **Label** field with **not found**.
- Change **Type** to **otherwise.**
- Specify a path from several different options, each with its own condition.
- When there is a **Choice** connector, indicate an "otherwise" path for the case in which none of the other conditions are met.



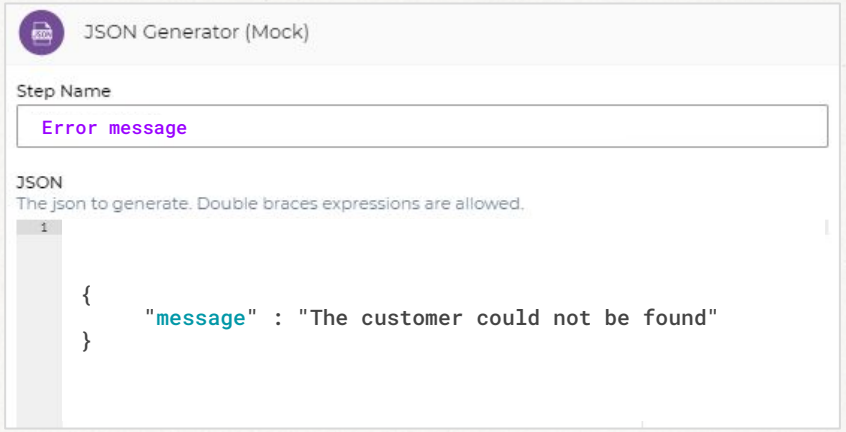
Hands-on Practice: Publishing a REST Service

JSON Generator configuration:

Transform Customer



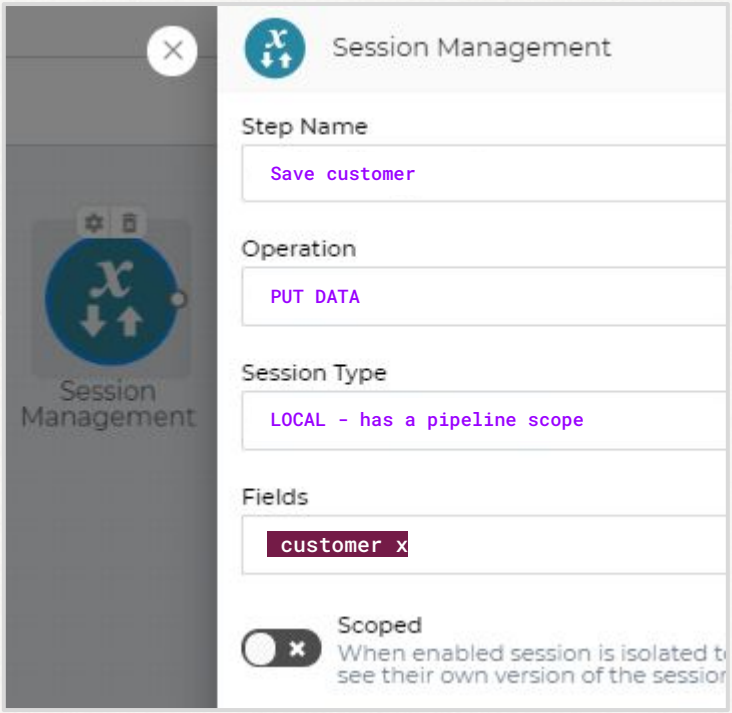
Error Message



Hands-on Practice: Publishing a REST Service

Session Management:

- In the connector palette, search for the **Session Management Component** and drag it to the canvas.
- Connect it to **Transform Customer** connector and open its settings form.
- Change **Operation** to **PUT**.
- In **Fields**, delete the default fields and add **customer** .
- We use this connector to add a JSON object with customer data to memory. After a few steps, we need to recover this content again.



Hands-on Practice: Publishing a REST Service

QUERY external service:

- In the connector palette, search for **REST** and drag **REST V2** to the canvas.
- Set up the **URL** field with the following data:

`https://api.godigibee.io/pipeline/digibee/v1/postalcode`

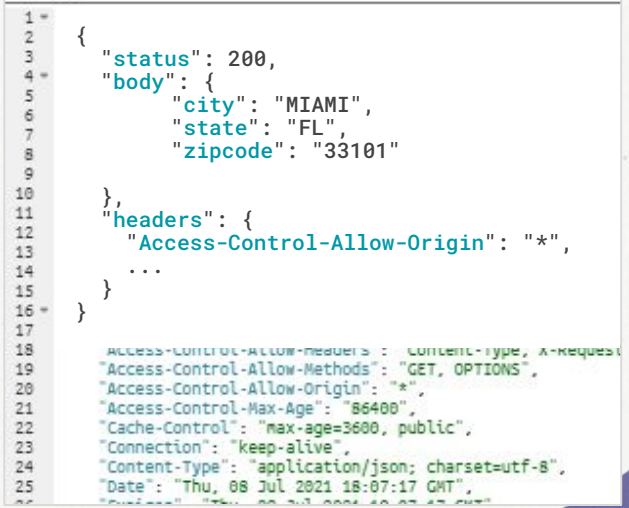
- Add the **API key** to the **headers**

`apikey`	`kHnQ48zInuVEwcuGbY6n4cO9VIX9Feqk`

- Add this **query parameter** to the **query params**

`postalCode`	`{{ message.zipcode }}`

- Connect this connector to the session connector from the previous step.
- Run the pipeline test once again (Ctrl + enter). The expected output is the response from that service.
- Note that the connector delivers all the information present in the actual response: HTTP Status, Message Content, and all response headers.



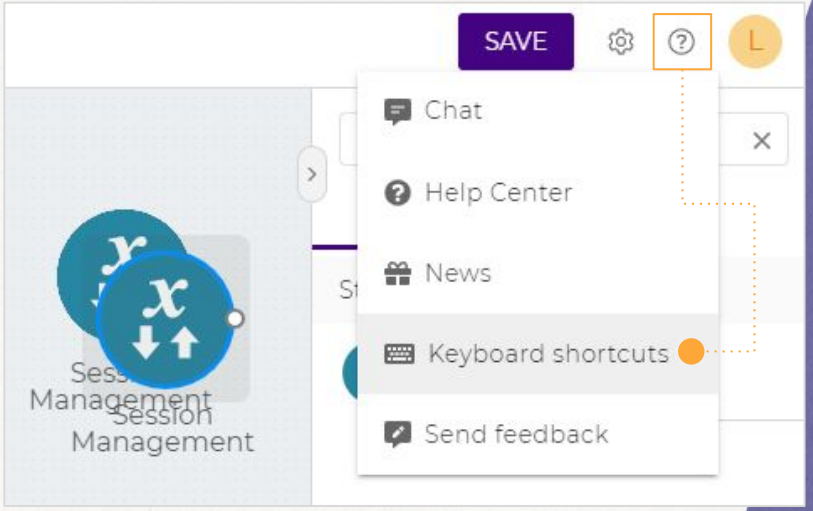
```
{
  "status": 200,
  "body": {
        "city": "MIAMI",
        "state": "FL",
        "zipcode": "33101"

  },
  "headers": {
    "Access-Control-Allow-Origin": "*",
    ...
  }
}
```

```
"Access-Control-Allow-Methods": "GET, OPTIONS",
"Access-Control-Allow-Origin": "*",
"Access-Control-Max-Age": "86400",
"Cache-Control": "max-age=3600, public",
"Connection": "keep-alive",
"Content-Type": "application/json; charset=utf-8",
"Date": "Thu, 08 Jul 2021 18:07:17 GMT",
```

Hands-on Practice: Publishing a REST Service

Copying:

- At this point, we need to retrieve the customer object we saved to the runtime memory.
- Use copy and paste to take advantage of the settings from the previous **Session** connector. To do this, click on the connector, press ctrl + c and + ctrl + v.
- A copy of the connector is added to the canvas. Access its settings, adjust the step name, and change the operation to **GET** this time.
- Connect it to the **REST** connector.
- Other shortcuts can be viewed in the menu on the upper right corner of the platform, under **Keyboard shortcuts**.



Hands-on Practice: Publishing a REST Service

Building the final JSON response:

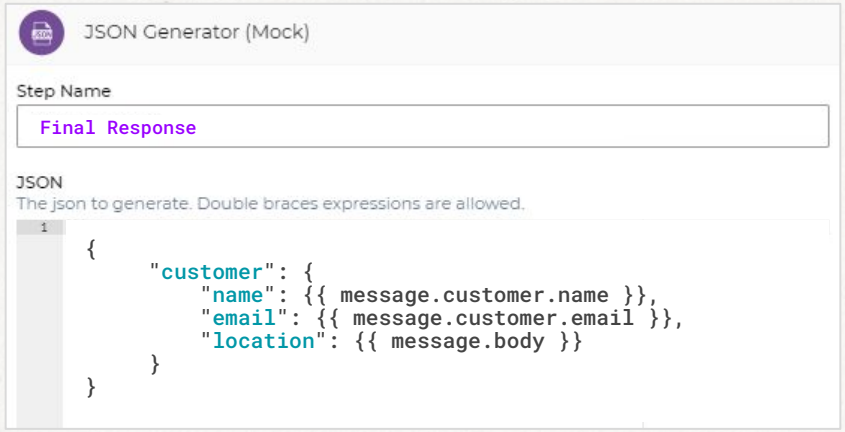
- After retrieving the customer with the session, perform a test and make sure that the response object was added in the response JSON at the end.

- If everything is alright so far, the output will be as shown on the right.

Hands-on Practice: Publishing a REST Service

Building the final JSON response:

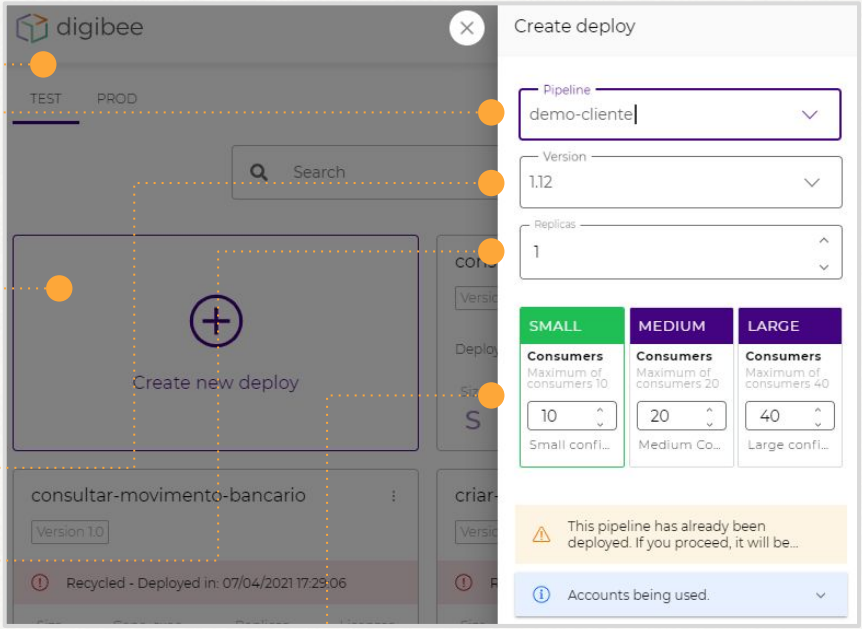
- Use the **JSON Generator** component to set a friendly response for the final content.
- Test once again and then save the pipeline.



Hands-on Practice: Publishing a REST Service

Runtime:

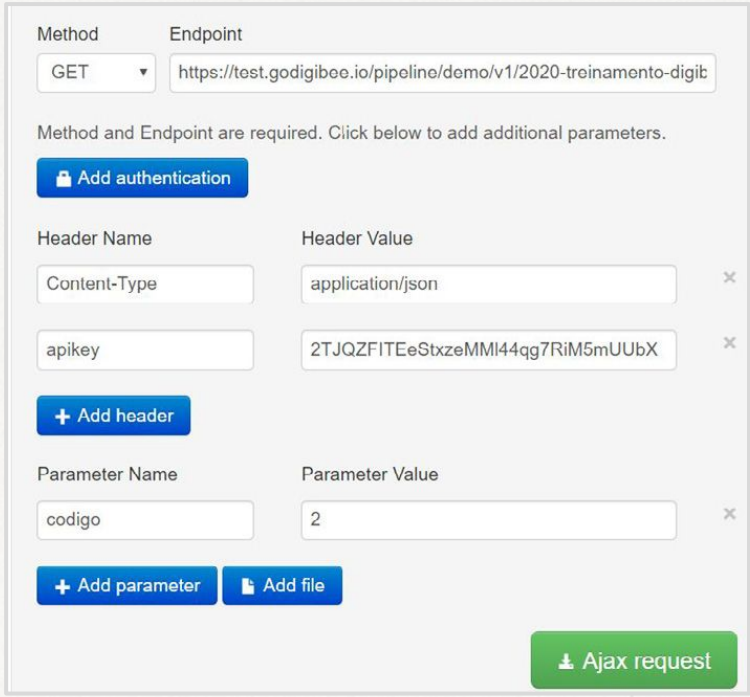
- Click the Digibee logo to access the main menu
- Click **RUNTIME**
- Select **TEST** environment
- Click on **Create new deploy**
- Select your pipeline
- Select the version you want to deploy
- Select how many copies (replicas) you want (scale horizontally).
- Set the size to **SMALL** (scale vertically) and confirm



Hands-on Practice: Publishing a REST Service

Request test:

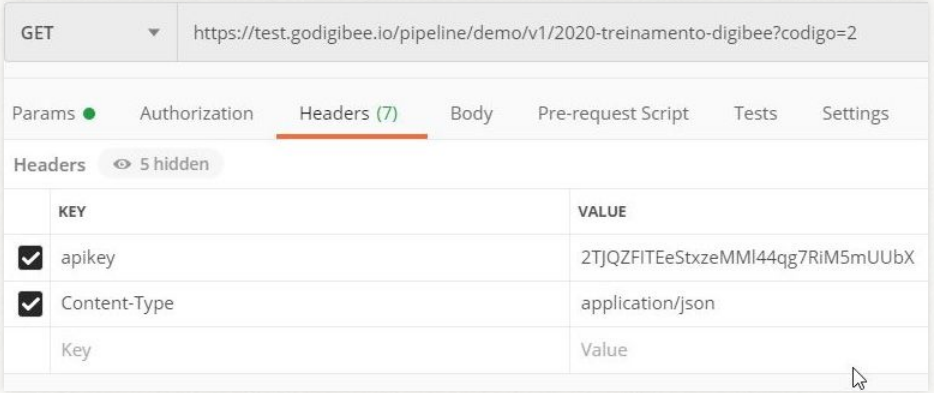
- Use any REST API client to do a test call to your deployed pipeline (API)
- Test in https://resttesttest.com/, after filling in the information as shown on the right.
- Try another online tester: https://reqbin.com/



Hands-on Practice: Publishing a REST Service

Request test:

- To test in **Postman**, fill in the information as shown in the example on the right.



- Don't forget to configure the API key (you can create one at settings/consumers) and also the Content-type as application/json.

Main Components

Transformations, Validations and Data Processing

- JSON Generator
- Transformer
- XML to Json
- Template
- CSV to Json
- JSON to CSV
- Validator
- Json Path

Security

- Digital-Signature
- Cryptography
- JWT

Data Persistence

- Relation
- Object Store

Reading and writing data with legacies

- DB - Banco de dados
- File Reader (Xml,
- Csv,posicional, excel)
- SAP (RFC e IDOC)
- Mongo
- SOAP/REST

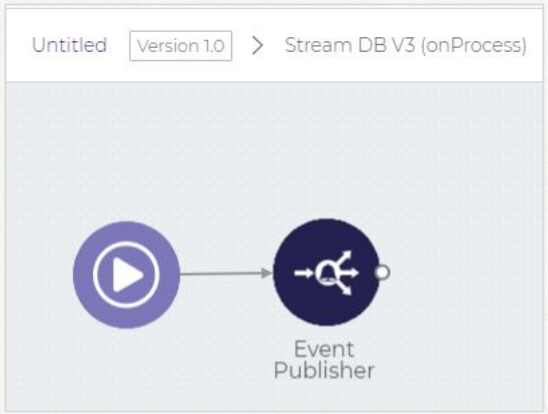
Main Components

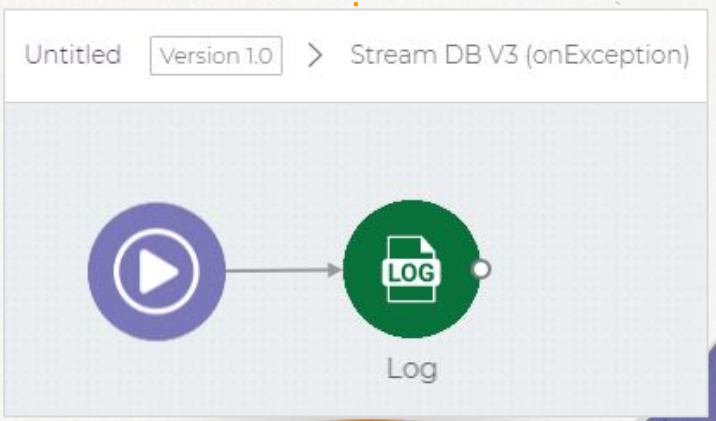
Components with sub-flows	Files	Message Queues
• For-Each • Stream (DB, Files) • Do While • Retry	• File Write • Google Storage • FTP/SFTP • S3 • One Drive	• Event • JMS (Oracle, Tibco...) • Kafka • SQS

Other Concepts - Sub Flows

SUB FLOW (on process)
Logic executed for each of the records

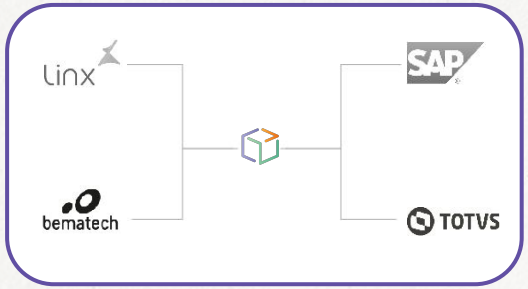
EXCEPTION HANDLING (on exception)
Logic that runs when there is error overflow in the sub-stream



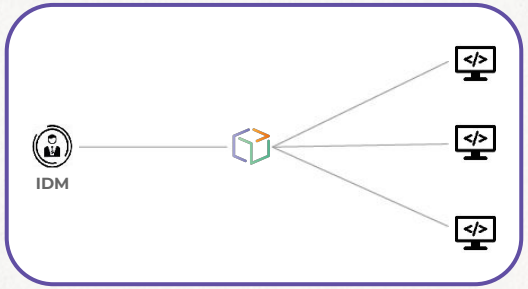


Use Cases - Integrations between Systems

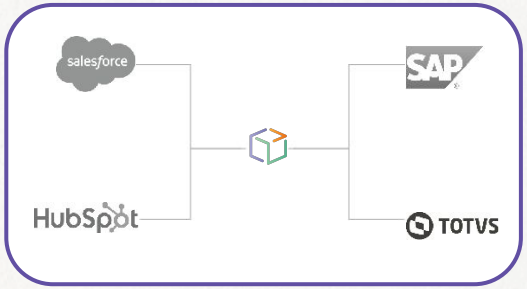
PDV <> ERP



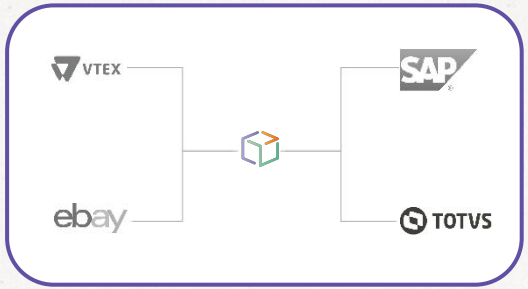
IDM <> Legados



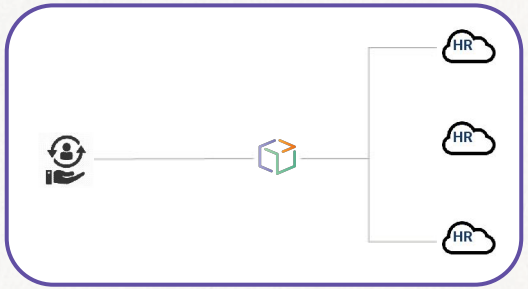
CRM <> ERP



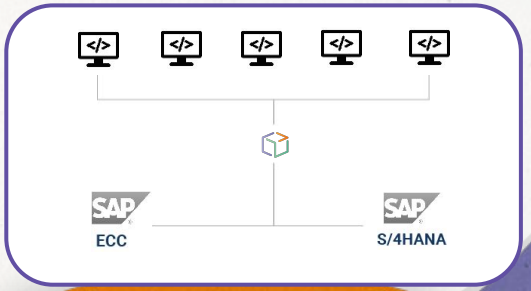
eCommerce <> ERP



HCM <> HR-Techs



Zero downtime migration



digibee

driving simplexity

100 STARTUPS TO WATCH 2019

cubo coworking Itaú

GAA INVESTMENTS

iDEXO by TOTVS

PROGRAMA SCALE UP ENDEAVOR