Employee Information API Documentation

# Overview:

This API is designed to retrieve employee information, tax details, and insurance information using the employeeId as the primary input. The process includes:

1. Calling the getEmployeeInfo subflow with the employeeId to fetch employee details.  
2. Extracting the driverLicenseId from the employee details.  
3. Using the driverLicenseId to fetch tax information via the getTaxInfo subflow.  
4. Using the driverLicenseId to fetch insurance information via the getInsuranceInfo subflow.

# API Endpoint Details

Base URL:

http://localhost:8081/api/v2

HTTP Method:

GET

Endpoint:

/employee-tax/{employeeId}

Description:

This endpoint takes an employeeId and returns the employee information along with associated tax and insurance data based on the driverLicenseId fetched from the employee's information.

# Request Example

Endpoint: GET /employee-tax/66b7662778db3e4db71f37c7

# Response Example

Success Response (200 OK):

{

"insuranceInfo": {

"\_id": "66f4c3cdad86920734402693",

"driverLicenseId": "123456",

"insurancePlan": "Premium Health Plan",

"expiryDate": "2025-12-31"

},

"salaryInfo": {

"bonus": 5000,

"employeeId": "66b7662778db3e4db71f37c7",

"salary": 75000

},

"employeeInfo": {

"driverLicenseId": "123456",

"email": "john.wick@example.com"

},

"taxInfo": {

"name": "John Wick",

"taxRecordId": "TR123456",

"taxFileNumber": "TFN987654321",

"remainingTaxDue": 5000,

"grossIncome": 75000,

"totalTaxPaid": 15000,

"driverLicenseId": "123456"

}

}

Error Response (400 Bad Request):

{  
 "error": “EmployeeId must be 24 characters long"  
}

# Subflows Description

## 1. getEmployeeInfo Subflow

Input: employeeId (String)

Output: Employee information including driverLicenseId, name, and email.

## 2. getTaxInfo Subflow

Input: driverLicenseId (String)

Output: Tax information including taxFileNumber, grossIncome, totalTaxPaid, and remainingTaxDue.

## 3. getInsuranceInfo Subflow

Input: driverLicenseId (String)

Output: Insurance details including insurancePlan, and expiryDate.

**4. getSalaryInfo Subflow**

Input: employeeId (String)

Output: Salary details including salary and bonus

# Mainflow Description

**Part 1: Initial Parallel Execution**

* **getEmployeeInfo**: Initiates a call to retrieve the employee's basic details using the provided employeeId. The response includes the driverLicenseId, which is used for subsequent API calls.
* **getSalaryInfo**: Simultaneously retrieves the employee's salary information using the employeeId.

**Part 2: Secondary Parallel Execution**

* **getTaxInfo**: Using the driverLicenseId obtained from the **getEmployeeInfo** flow, this flow fetches the employee's tax records.
* **getInsuranceInfo**: Runs in parallel with **getTaxInfo** to retrieve insurance information based on the driverLicenseId.

**Deployment**

* **Deployment Method**: The project is deployed on CloudHub for prod version and on mule runtime locally for test version
* **CI/CD Integration**:
  + A **GitHub Actions script** automates the deployment process:
* **Environment variable for the development version:**
  + PORT: "8081"
  + EMAIL\_PASSWORD: "majbsqfzgpgzbuaq"
  + EMAIL\_NAME: "baby.university09@gmail.com"
  + BASE\_URI\_TAX: "http://localhost:4000"
  + BASE\_URI\_EMPLOYEE: "http://localhost:8080"
  + BASE\_URI\_INSURANCE: "http://localhost:4001"
  + CLIENT\_EMAIL\_NAME: "theo.pham@alphaplussolution.com.au"
  + BUCKET\_NAME: "tavola-italiano-res"
  + BLOB\_NAME: "salary.csv"
* **CloudHub Configuration**:
  + **Runtime version**: 4.8.0
  + **Target**: Cloudhub-US-East-2
  + **Environment**: SANDBOX
  + **Properties**:
    - CLIENT\_EMAIL\_NAME=theo.pham@alphaplussolution.com.au
    - PORT=8081
    - BLOB\_NAME=salary.csv
    - BASE\_URI\_EMPLOYEE=http://54.80.15.151:8080
    - BASE\_URI\_TAX=http://54.80.15.151:4000
    - BASE\_URI\_INSURANCE=http://54.80.15.151:4001
    - EMAIL\_PASSWORD=majbsqfzgpgzbuaq
    - EMAIL\_NAME=baby.university09@gmail.com
    - BUCKET\_NAME=tavola-italiano-res

**Testing**

**MUnit Tests:**

* **Key Tests:**
  + **Test Flow**: integrationFlow1
  + **Endpoint**: api/v2/employee-tax/{employeeId}
  + **Port**: 8081
  + **Expected Response:** The returned payload aligns with the structure and data specified in the sample\_data/output.json file.
* **Error Scenarios**: MUnit tests also cover error handling for missing employeeId or invalid driverLicenseId.

**Sequence Diagram**

