**HU BASIC TRACK – IE**

**Major Assignment 2**

The whole Assignment is split into Milestones which basically are the tasks you need to complete.

Submission Guidelines

|  |  |  |
| --- | --- | --- |
| 1. | Create a Word Document where you need to attach the screenshots of Part A and Part B. Document’s name should be “IE-Major-Asg-Your\_name”. Include the screenshots for each milestones of Part A& B (RAML, mocking responses, flow outputs) | 5 points |
| 2. | Push the RAML code(.raml file), Flow Designer Code(.jar file), Postman collections to a **GitHub repo**. If pushing to an existing repo, the code should be in a separate branch. | 5 points |

**Note-** We will freeze the solutions in 3 hours from the time it commenced. Please follow the timelines strictly.

**Part A- Integration**

|  |  |  |
| --- | --- | --- |
| **Sl.No** | **Milestones** | **Marks** |
| **1** | Create a Postman Collection with this Name “Integration-Workspace-YourName” which should contain all the below 3 milestone with request and response | 5 |
| **2** | Get the list of all the users whose gender is female, and status is inactive.  **Method:** Get  **Endpoint:** <http://hu-major-assignment.us-e2.cloudhub.io/users>  **Sample Response Body:**  [  {  "id": 2482,  "name": "Kartik Kocchar",  "email": "[kartik\_kocchar@rutherford.io](mailto:kartik_kocchar@rutherford.io)",  "gender": "female",  "status": "inactive"  }  ]    Dataweave script :    %dw 2.0  output application/json  ---  payload filter ((item, index) -> item.gender== "female" and item.status== "inactive" )  **Part 1:** Create a proper flow for the above-mentioned scenario in Flow designer with name Milestone1-” Your Name”    **Part 2:** Create endpoint in RAML with endpoint path(resource name) as /milestone1 for the above mentioned scenario | 6 |
| **3.** | Create an endpoint with a post method which will accept the post\_id in the body and get the details of all the users who belong to that post\_id.(post\_id should be passed in the body from postman)  **Method**: POST  **Endpoint:** <http://hu-major-assignment.us-e2.cloudhub.io/userdetails>    **Request Body:**  {  "post\_id": 2003  }    **Sample Response Body**:    {  "Userdetails": [  {  "post\_id": 2003,  "id": 1978,  "name": "Bhaaswar Pothuvaal III"  }  ]  }  **Dataweave Script:**    %dw 2.0  output application/json  var userdetails = payload filter ((item, index) -> item.post\_id == vars.postId)  ---  {  Userdetails: userdetails map (item, index) -> {    post\_id: item.post\_id,  id: item.id,  name: item.name  }    }  **Note:** You have to create a variable with the name “**postId**”.  **Part 1:** Create a proper flow for the above mentioned scenario in Flow designer with name Milestone2-” Your Name”    **Part 2:** Create endpoint in RAML with endpoint path(resource name) as /milestone2 for the above mentioned scenario | 12 |
| **4.** | Create an endpoint with a GET method. And write a Dataweave script to give the sample response as mentioned below:  **Method**: GET  **Endpoint**: <http://hu-major-assignment.us-e2.cloudhub.io/userid>    **Sample Response Body**:  {  "Name": "Gorakhanatha Chaturvedi",  "Status": "active",  "Gender": "female"  }  **Part 1:** Create a proper flow for the above-mentioned scenario in Flow designer with name Milestone3-” Your Name”    **Part 2:** Create endpoint in RAML with endpoint path (resource name) as /milestone3 for the above-mentioned scenario | 12 |

**Part B- API Testing**

|  |  |  |
| --- | --- | --- |
| SR No. | MILESTONE | Marks |
| 1. | Create a collection “API-Collection-Your Name” in Postman. All the endpoints added to this collection should have proper name. | 3 |
| 2. | Create an Environment “API -Env-Your Name” and load the commonly used values there. | 2 |
| 3. | GET : <https://jsonplaceholder.typicode.com/posts>   * Response payload: (1 point)   [  {  "userId": 1,  "id": 1,  "title": "sunt aut facere repellat provident”  "body": "quia et suscipit\nsuscipit”  },  . . .  ]     * Verify the following(2 points):   a. User with 'id' 40 has 'userId'=4  b. Every 'title' field in the response has the type is String. | 3 |
| 4. | POST: <https://reqres.in/api/users>   * Request payload: (1 point)   {    "email": "[amith.k@reqres.in](mailto:amith.k@reqres.in)",  "first\_name": "Amit",  "last\_name": "K",  "avatar": ""  }     * Set the 'email' and 'first\_name’ values in Environment before sending the request. (2 points) * Verify the “email’ and ‘first\_name’ values match with response.(2 point) | 5 |
| 5. | PUT: <https://reqres.in/api/users/>   * Request payload:   {  "email": "[amith.kumar@reqres.in](mailto:amith.kumar@reqres.in)",  "first\_name": "Amith",  "last\_name": "Kumar",  "avatar": "",  "id": "664",  "createdAt": "2022-02-15T15:18:16.933Z"  }     * Response payload:   {  "email": "[amith.kumar@reqres.in](mailto:amith.kumar@reqres.in)",  "first\_name": "Amith",  "last\_name": "Kumar",  "avatar": "",  "id": "664",  "createdAt": "2022-02-15T15:18:16.933Z",  "updatedAt": "2022-02-16T05:42:41.832Z"  }     * Verify “first\_name”,‘last\_name’ and ‘email’ are updated with correct values. These changed values should be reflected in the environment as well. (4 points) * Check whether the field “updatedAt” is present in the response body. (1 point) | 5 |
| 6. | Also perform the common validations listed , on each API response. (4 points each)   * HTTP status code * Content type * Response time is less than 1000ms | 12 |

