

Phoenix Project – Postman API Automation Assignment

Sprint Story: Inwarranty Flow Automation

You are part of the QA Automation team working on the Phoenix Application. In the current sprint, you have been assigned to automate the test cases for the 'Inwarranty Flow'.

Until now, API responses were manually verified by the team for each deployment. This manual validation is ***time-consuming, error-prone, and not scalable as the number of test cases increases.***

To solve this, your task is to write automated test scripts in Postman using the Chai.js assertion library.

These scripts should be able to validate key API responses and be integrated into a CI pipeline for automated execution.

API 1: Login API Request (Front Desk)

Method: POST

URL: {{base_url}}/login

Request Body

```
{  
  "username": "iamfd",  
  "password": "password"  
}
```

Positive Test Cases

- Status code should be 200
- Message should be 'Success'
- Token should be a valid JWT (3 parts)
- Validate JSON Schema for Response
- Response time should be under 150 ms
- Save token in collection variable for chaining

Negative Test Cases

- Invalid credentials should return 401 Unauthorized
- Blank username or password should return 400 Bad Request and proper error message should be shown like "Invalid username or password".

API 2: User Details API Request

Method: GET

URL: {{base_url}}/user/details

Header: Authorization: {{fd_token}}

Positive Test Cases

- Status code should be 200
- User data should be present in response
- Check for response time
- Validate JSON SCHEMA for the response.
- Response should contain fields like `id` and `email_id`

Negative Test Cases

- Missing token should return 401 Unauthorized
- Invalid or expired token should return 403 Forbidden

Other(Security Check):

Sensitive information like password is NOT returned

API 3: Count API Request

Purpose:

To fetch the count of jobs by category like 'Pending for Delivery', 'Created Today', and 'Pending for FST assignment'. Used to verify the effect of job creation and operational tracking.

Endpoint:

GET {{base_url}}/job/count

Authorization: Token (FD login token)

Positive Test Cases:

- Status code is 200
- Response contains keys: label, key, count
- Key values must match expected categories
- Count values must be numbers [Type Check]
- Response time under 150 ms

Negative Test Cases:

- Missing token returns 401 Unauthorized
- Invalid token returns 403 Forbidden

Note:

After a successful job creation, **the count for 'Created today' should increment by 1.**

API 4: Master API Request

Purpose:

To fetch static master data like models, OEMs, action status, warranty status, roles, and service locations. These are used as input reference values while creating a job.

Endpoint:

GET {{base_url}}/master

Authorization: Token (FD login token)

Positive Test Cases:

- Status code is 200
- Master keys: mst_oem, mst_model, mst_product, mst_problem should be present
- **Each key should return a non-empty array**
- Validate response schema
- Response time under 200 ms

Negative Test Cases:

- Missing token returns 401 Unauthorized
- Invalid token returns 403 Forbidden

API 5: Create Job API Request

Purpose:

To create a new inwarranty job entry in the Phoenix system by sending customer, address, product, and problem details.

Endpoint:

POST {{base_url}}/job/create

Authorization: Token (FD login token)

Sample Request Payload:

```
{  
    "mst_service_location_id": 0,  
    "mst_platform_id": 2,  
    "mst_warranty_status_id": 1,  
    "mst_oem_id": 1,  
    "customer": {  
        "first_name": "{{first_name}}",  
        "last_name": "{{last_name}}",  
        "mobile_number": "{{mobile_number}}",  
        "mobile_number_alt": "",  
        "email_id": "{{email_id}}",  
        "email_id_alt": ""  
    },  
    "customer_address": {  
        "flat_number": "c 304",  
        "apartment_name": "Jupiter",  
        "street_name": "MG road",  
        "landmark": "Bangur Nagar",  
        "area": "Goregaon West",  
        "pincode": "411039",  
        "country": "India",  
        "state": "Maharashtra"  
    },  
    "customer_product": {  
        "dop": "2025-04-06T18:30:00.000Z",  
        "serial_number": "{{imeiNumber}}",  
        "imei1": "{{imeiNumber}}",  
        "imei2": "{{imeiNumber}}",  
        "popurl": "2025-04-06T18:30:00.000Z",  
        "product_id": 1,  
    }  
}
```

```
"mst_model_id": 1
},
"problems": [
{
  "id": 1,
  "remark": "Battery Issue"
}
]
```

Positive Test Cases:

- Status code is 200
- Message is "Job created successfully."
- job_number starts with "JOB_"**
- Response body contains: id, tr_customer_id, tr_customer_product_id**
- Response time is < 200ms
- Save job_number to collection variable
- Validate response schema

Negative Test Cases:

- Missing customer or product_id should return 400 Bad Request
- Invalid imeiNumber format
- Empty problems array
- Missing Authorization Token → 401 Unauthorized
- Invalid Token → 403 Forbidden

Post-Check:

After job creation, re-hit the Count API and assert: "Created today": increased by 1

API 6: Dashboard Details API Request

Purpose:

To retrieve a list of job entries based on dashboard filters such as created_today, pending_for_delivery, etc.

Endpoint:

POST {{base_url}}/dashboard/details

Authorization:

Header → Authorization: Token {{fd_login_token}}

Payload Example:

```
{  
  "details_key": "created_today"  
}
```

Positive Test Cases

-Status code should be 200 OK

-Response message should be "Success"

-Response data should be a non-empty array

-Each object in the array should have:

 - job_number (starting with "JOB_")

 - Required fields like id, created_at, name, mobile_number, mst_action_status, etc.

-Response time should be below 200 ms

- ***The job number created in API 5 should be present in this response***

Negative Test Cases

- Missing token: Should return 401 Unauthorized
- Invalid token: Should return 403 Forbidden
- Invalid details_key: Should return 400 Bad Request

API 7: Job Search API

Purpose:

To retrieve job details based on a given job number by performing a search in the Phoenix system.

Endpoint:

POST {{BASE_URL}}/job/search

Authorization:

Authorization: Token {{fd_login_token}}

Request Payload:

```
{  
  "searchText": "{{jobNumber}}"  
}
```

Positive Test Cases

- Status code should be 200 OK
- Response message should be "Success"
- Response data should be an array with exactly 1 matching job object
- The job number in response should match searchText
- Validate essential fields in the object:
 - job_number (starts with "JOB_")
 - name, mobile_number, email_id, mst_platform_name, etc.
- Validate timestamps: created_at, modified_at are valid ISO date strings
- Response time should be under 200 ms

Negative Test Cases

- **Empty searchText** → Returns 400 Bad Request or error message
- **Invalid job number** → Returns 200 with an empty array
- **Missing token** → Returns 401 Unauthorized
- **Invalid token** → Returns 403 Forbidden

API 8: Get Job Details by Job Number

Purpose:

To fetch complete job details using a job_number by making a POST request.

Endpoint:

POST {{BASE_URL}}/job/details/{{jobNumber}}

Headers:

Authorization: Token {{fd_login_token}}

Positive Test Cases

- Status code is 200 OK
- Message is "Success"
- Job number matches the request
- Fields like name, mobile_number, email_id, mst_model_name, mst_oem_name are present
- **Details array contains history with proper status transitions**
- Problems array contains at least one problem with correct structure
- Quotations is an array (can be empty)
- Response time is under 200 ms

Negative Test Cases

1. Missing Authorization Token

- **Request:** Send the request without the Authorization header.
- **Expected Response:**
 - Status code: 401 Unauthorized
 - Message: "Unauthorized access" or equivalent

2. Invalid Authorization Token

- **Request:** Send the request with an invalid or expired token.
- **Expected Response:**

- Status code: 403 Forbidden
- Message: "Forbidden" or appropriate error

3. Invalid Job Number (Non-existent)

- **Request:** Provide a job number that does not exist in the system.
- **Expected Response:**
 - Status code: 404 Not Found or 200 OK with
 - Message: "No record found" or similar
 - data: null or empty object/array

4. Malformed Job Number Format

- **Request:** Use a string that doesn't follow "JOB_XXXX" format (e.g., "123" or "invalid_job").
- **Expected Response:**
 - Status code: 400 Bad Request
 - Message: "Invalid job number format"

5. Empty Job Number in Path

- **Request:** Send a request like POST /job/details/ without providing jobNumber.
- **Expected Response:**
 - Status code: 404 Not Found or 400 Bad Request

6. Invalid Method

- **Request:** Try to access the same endpoint with a GET instead of POST.
- **Expected Response:**
 - Status code: 405 Method Not Allowed

API 9: Supervisor Login API

Purpose:

Authenticate a supervisor and return a JWT token for authorized access to other APIs.

Endpoint:

GET {{base_url}}/v1/login

Payload:

```
{  
    "username": "iamsup",  
    "password": "password"  
}
```

Positive Test Cases

- Status code should be 200
- Response message should be 'Success'
- Token should be present and valid (JWT - 3 parts)

Negative Test Cases

- Invalid credentials should return 401 Unauthorized
- Blank username or password should return 400 Bad Request with appropriate error message

API 10: Get User Details API (Supervisor Role)

Purpose: To retrieve detailed user information for the currently authenticated Supervisor user in the Phoenix system.

Endpoint: GET {{BASE_URL}}/userdetails

Authorization: Token {{supervisor_login_token}}

Positive Test Cases

- Ensure the response status code is 200 OK.
- Verify that the response time is below 200 ms.
- Response should contain a message field with the value: "Success".
- The response data object should include the following keys:
 - id
 - first_name
 - last_name
 - login_id
 - mobile_number
 - email_id
 - password
 - lock_status
 - is_active
 - mst_role_id
 - mst_service_location_id
 - created_at
 - modified_at
 - role_name
 - service_location

- The field `role_name` should be equal to "Supervisor".
- Both `created_at` and `modified_at` should be valid ISO 8601 date strings.
- **mobile_number should be a 10-digit numeric string.**

Negative Test Cases

- Request without token should return 401 Unauthorized.
- Request with an incorrect token should return 403 Forbidden.

API 11: Count API Request (Dashboard Overview for Supervisor)

Purpose:

Provides a summary of job status counts relevant to the supervisor's dashboard..

Endpoint:

GET {{base_url}}/job/count

Authorization: Token (SUP login token)

Positive Test Cases

- - Status code should be 200
- Response message should be 'Success'
- Each label should have a valid count
- Check if all keys like 'pending_for_assignment', 'pending_for_repair',
'pending_for_delivery' exist

Negative Test Cases

- - Invalid or expired token should return 401 Unauthorized
- - Missing token should return 403 Forbidden

API 12: Pending For Job Assignment API Request by Supervisor

Purpose:

Allows the supervisor to fetch all jobs currently in 'Pending For Job Assignment' status.

Endpoint:

GET {{base_url}}/job/pending

Authorization: Token (SUP login token)

Positive Test Cases

- Status code should be 200
- Response should contain jobs in 'Pending For Job Assignment' status
- Each job object should include fields like 'job_number', 'serial_number', 'imei1', 'imei2'
- Created job should be present for pending for assignment state

Negative Test Cases

- Invalid filter parameter should return appropriate error
- Token not provided should result in 401 Unauthorized
- Wrong Token should give me 403 Forbidden

API 13: Engineer Mapped To Service Center API Request

Purpose:

Retrieves a list of engineers mapped to the supervisor's current service location.

Endpoint:

GET {{base_url}} /engineer/mapped

Authorization: Token (SUP login token)

Positive Test Cases

- Status code should be 200
- Response should return a list of engineers for the supervisor's service center
- Each engineer should include 'id', 'first_name', 'last_name', 'login_id'
- JSON SCHEMA VALIDATION

Negative Test Cases

- - Missing authorization should return 401 Unauthorized

API 14: Assign Job to Engineer API Request

Purpose:

Enables the supervisor to assign a specific job to a selected engineer..

Endpoint:

POST {{base_url}} /engineer/assign

Authorization: Token (SUP login token)

Request Payload: {

```
"job_id": {{jobId}},  
"engineer_id":{{engId}}  
}
```

Positive Test Cases

- - Status code should be 200
- - Message should be 'Engineer assigned successfully'
- - Job should now appear in engineer's job list
- - Created job should be present for pending for repair state

Negative Test Cases

- - Assigning with invalid job_id or engineer_id should return appropriate error
- - Missing job_id or engineer_id should return 400 Bad Request

Also the job should not be available in the pending for assignment state!

API 15: Login API Request for Engineer

Purpose:

Authenticate a supervisor and return a JWT token for authorized access to other APIs.

Endpoint:

POST {{base_url}}/v1/login

Payload:

```
{"username": "iameng",  
 "password": "password"}  
}
```

Positive Test Cases

- Status code should be 200
- Response message should be 'Success'
- Token should be present and valid (JWT - 3 parts)

Negative Test Cases

- Invalid credentials should return 401 Unauthorized
- Blank username or password should return 400 Bad Request with appropriate error message

API 16: Get User Details API (Engineer Role)

Purpose: To retrieve detailed user information for the currently authenticated Supervisor user in the Phoenix system.

Endpoint: GET {{BASE_URL}}/userdetails

Authorization: Token {{engineer_login_token}}

Positive Test Cases

- Ensure the response status code is 200 OK.
- Verify that the response time is below 200 ms.
- Response should contain a message field with the value: "Success".
- The response data object should include the following keys:
 - id
 - first_name
 - last_name
 - login_id
 - mobile_number
 - email_id
 - password
 - lock_status
 - is_active
 - mst_role_id
 - mst_service_location_id
 - created_at
 - modified_at
 - role_name
 - service_location

- The field `role_name` should be equal to "Engineer".
- Both `created_at` and `modified_at` should be valid ISO 8601 date strings.
- **mobile_number should be a 10-digit numeric string.**

Negative Test Cases

- Request without token should return 401 Unauthorized.
- -

API 17: Dashboard Count API for Engineer

Purpose:

To retrieve job status counts specific to the authenticated Engineer user in the Phoenix system. The API provides the number of jobs currently "Pending for Repair" and those "Repair Completed", enabling engineers to manage their workflow effectively.

Endpoint:

GET {{BASE_URL}}/dashboard/count

Authorization:

Token {{engineer_login_token}}

Positive Test Cases

- Ensure the response status code is **200 OK**.
- Verify that the response time is **below 200 ms**.
- The response should contain a message field with the value: "**Success**".
- The response should include a data array that is not empty.
- Each object in the data array should include the following keys:
 - **label**
 - **key**
 - **count**
- The key values should include:
 - **pending_for_repair**
 - **repair_completed**
- The count value should be a **non-negative integer**.
- The label should be a **string describing the job status**.
- There should be exactly **two objects** in the data array for Engineer users.
- **Optionally verify that count value for "repair_completed" increases after job completion.**

Negative Test Cases

- Request without token should return **401 Unauthorized**.
- Request with an incorrect/expired token should return **403 Forbidden**.
- If an invalid HTTP method is used (e.g. POST), the API should return **405 Method Not Allowed**.
- Tampered token (e.g. altered payload) should return an **error or access denied**.
- If the authenticated user is not an engineer, the response might return **empty data or restricted access**.

API 18: Pending For Repair Details API

Purpose:

To retrieve all jobs that are currently in "Pending For Repair" status for the logged-in Engineer in the Phoenix system. This data is used to show detailed job-level records on the engineer's dashboard.

Endpoint:

POST {{BASE_URL}}/dashboard/details

Authorization:

Token {{engineer_login_token}}

Positive Test Cases

- Ensure the response status code is **200 OK**.
- Verify that the **response time is under 200 ms**.
- Response should contain:
 - A top-level field: "message": "Success"
 - A non-empty "data" array.
- For each object inside the "data" array:
 - Validate that job_number starts with "JOB_".
 - Required fields should exist:
 - id
 - name
 - mobile_number
 - email_id
 - job_number
 - mst_action_status
 - mst_action_status_code
 - assigned_to

- assigned_by
 - created_at
 - modified_at
- created_at and modified_at should be valid **ISO 8601 date strings**.
 - mobile_number should be a **10-digit number** (or valid format).
 - mst_action_status should be equal to "**Pending For Repair**".
 - assigned_to should match the logged-in engineer's ID.

Negative Test Cases

- Request without authorization token should return **401 Unauthorized**.
- Request with an **invalid or expired token** should return **403 Forbidden**.
- If the engineer has **no pending jobs**, the "data" array should be **empty**, but the "message" should still be "Success".
- If the request method is incorrectly set to GET instead of POST, it should return **405 Method Not Allowed**.

API 19: Repair Completed Details API

Purpose:

To retrieve detailed job records that are marked as "**Repair Completed**" for the logged-in engineer in the Phoenix system. These records represent jobs where repair activities have been completed.

Endpoint:

POST {{BASE_URL}}/dashboard/details

Authorization:

Token {{engineer_login_token}}

Request Payload:

```
{  
    "details_key": "repair_completed"  
}
```

Positive Test Cases

- Ensure the response status code is **200 OK**.
- Response time should be **below 200 ms**.
- The response must contain:
 - "message": "Success"
 - "data" field which should be:
 - **A non-empty array** if jobs with mst_action_status as "Repair Completed" exist.
 - An empty array if no such jobs exist.
- For each object in the "data" array:
 - job_number should start with "JOB_".
 - mst_action_status should be equal to "**Repair Completed**".
 - Validate essential fields:

- id, job_number, name, mobile_number, email_id, created_at, modified_at, assigned_to, assigned_by, mst_action_status_code
 - created_at and modified_at should be valid **ISO 8601 date strings**.
 - assigned_to should match the logged-in engineer ID.
 - mobile_number should be in a valid format (typically 10 digits).
 -

Negative Test Cases

- **Missing Token:** Request without token should return **401 Unauthorized**.
- **Invalid Token:** Request with an incorrect or expired token should return **403 Forbidden**.
- **Invalid details_key:** Sending a wrong details_key like "repair_done" should return **400 Bad Request** or an appropriate error.
- **Empty Data Edge Case:** If "data" is empty, but repair-completed jobs exist for that engineer, then **it's a bug** — and should be reported.
- **Wrong Role Access:** If another role (e.g., Supervisor) hits this API, response should either be filtered or denied based on role-based access control logic.

API 20- My Jobs API – Engineer

Purpose:

To fetch all jobs that are currently assigned to the **logged-in engineer**, including job details, assigned status, and associated problems.

Endpoint:

GET {{BASE_URL}}/engineer/myjobs

Authorization:

Token {{engineer_login_token}}

Positive Test Cases

- Ensure the response status code is **200 OK**.
- Verify response time is **under 200 ms**.
- Response should include:
 - "message": "Success"
 - "data": an array of job objects assigned to the engineer.

For each job object in data:

- job_number should start with "JOB_".
- Required fields to validate:
 - id, job_number, name, mobile_number, email_id, serial_number
 - mst_model_name, mst_product_name, mst_action_status, mst_oem_name
 - mst_warranty_status, mst_platform_name, assigned_by, assigned_to
 - created_at, modified_at, assigned_on — should be valid **ISO 8601** dates.
 - mobile_number should be a valid 10-digit numeric string.
 - mst_action_status should be "Pending For Repair" (or a valid known status).
- Validate nested problems array:

- Should contain at least 1 problem (if added).
- Each problem should have id, name, remark, role_name, and created_at.

Negative Test Cases

- **Missing Token:** Should return **401 Unauthorized**.
- **Invalid Token:** Should return **403 Forbidden**.
- **Engineer not assigned to any jobs:** Should return "data": [] with message "Success".
- **Accessing as wrong role (e.g., Supervisor):** Should not return engineer jobs; return an empty array or 403 as per backend control.
- **Malformed Token Header:** e.g., Authorization: Bearer <token> instead of Token <token> should return 401.

API 21: Master API Request

Purpose:

To fetch static master data like models, OEMs, action status, warranty status, roles, and service locations and problems. Problems is data is needed by engineer to work on the job

Endpoint:

POST {{base_url}}/master

Authorization: Token (Engineer login token)

Positive Test Cases:

- Status code is 200
- Master keys: mst_oem, mst_model, mst_product, mst_problem should be present
- **Each key should return a non-empty array**
- Validate response schema
- Response time under 200 ms

Negative Test Cases:

- Missing token returns 401 Unauthorized
- Invalid token returns 403 Forbidden

API 22: Repair Complete API – Engineer

Purpose:

To mark a job as **repair completed** by the assigned engineer and update remarks for associated problems.

Endpoint:

POST {{BASE_URL}}/engineer/repaircomplete

Authorization:

Token {{engineer_login_token}}

Payload Example:

```
{  
  "job_id": {{jobId}},  
  "problems": [  
    {  
      "id": 2,  
      "remark": "Battery Replaced"  
    }  
  ]  
}
```

Positive Test Cases

- Ensure the response status code is **200 OK**.
- Verify response time is **under 200 ms**.
- The response should include:
 - "message": "Repair successful."
- Validate that the job's status is updated to "**Pending for QC**" in subsequent Search **JOB API Request**

Negative Test Cases

- **Missing Token:** Should return **401 Unauthorized**.
- **Invalid Token:** Should return **403 Forbidden**.
- **Missing job_id:** Should return **400 Bad Request** or an appropriate validation error.
- **Invalid job_id:** Should return an error such as "Job not found" or "Unauthorized to update this job".
- **Empty problems array or missing remark:** Should return **400** with validation message.
- **Non-assigned engineer attempting the repair:** Should return **403 Forbidden** or a similar error.
- **Invalid problem ID in the list:** Should return error if that ID is not linked to the given job.

API 23: QC Login API

Purpose:

Authenticate a QC and return a JWT token for authorized access to other APIs.

Endpoint:

POST {{base_url}}/v1/login

Payload:

```
{"username": "iamqc",  
 "password": "password"}  
}
```

Positive Test Cases

- Status code should be 200
- Response message should be 'Success'
- Token should be present and valid (JWT - 3 parts)
- JSON Schema Validation

Negative Test Cases

- Invalid credentials should return 401 Unauthorized
- Blank username or password should return 400 Bad Request with appropriate error message

API 24: Get User Details API (QC Role)

Purpose: To retrieve detailed user information for the currently authenticated QC user in the Phoenix system.

Endpoint: GET {{BASE_URL}}/userdetails

Authorization: Token {{qc_login_token}}

Positive Test Cases

- Ensure the response status code is 200 OK.
- Verify that the response time is below 200 ms.
- Response should contain a message field with the value: "Success".
- The response data object should include the following keys:
 - id
 - first_name
 - last_name
 - login_id
 - mobile_number
 - email_id
 - password
 - lock_status
 - is_active
 - mst_role_id
 - mst_service_location_id
 - created_at
 - modified_at
 - role_name
 - service_location

- The field role_name should be equal to "QC".
- Both created_at and modified_at should be valid ISO 8601 date strings.
- **mobile_number should be a 10-digit numeric string.**

Negative Test Cases

- Request without token should return 401 Unauthorized.
- Request with an incorrect token should return 403 Forbidden.

API 24: Count API Request (Dashboard Overview for QC)

Purpose:

Provides a summary of job status counts relevant to the QC dashboard..

Endpoint:

GET {{base_url}}/job/count

Authorization: Token (QC login token)

Positive Test Cases

- Status code should be 200
- Response message should be 'Success'
- Each label should have a valid count
- Check if all keys like 'Pending_for_QC', 'QC completed', 'QC Rejected' exist

Negative Test Cases

- Invalid or expired token should return 403 Forbidden
- Missing token should return 401 Unauthorized

API 25: Pending For QC Request

Purpose:

To retrieve all jobs that are currently in "Pending For QC" status for the logged-in QC in the Phoenix system.

Endpoint:

POST {{BASE_URL}}/dashboard/details

Authorization:

Token {{qc_login_token}}

Positive Test Cases

- Status code should be 200
- Response message should be "Success"
- Response data should be a non-empty array when jobs pending for QC exist
- Each job in the response should include fields like job_number, mst_model_name, mst_action_status, serial_number
- All jobs should have mst_action_status as "Pending For QC"
- JSON response should match the defined schema
- Response time should be within 2 seconds

Negative Test Cases

- Request without authorization token should return **401 Unauthorized**.
- Request with an **invalid or expired token** should return **403 Forbidden**.
- If the request method is incorrectly set to GET instead of POST, it should return **405 Method Not Allowed**.

Note: Currently, the API returns a 200 OK with an empty data array even though multiple jobs are actually pending for QC.

This is a bug and must be reported to the development team.

The backend should return all jobs with mst_action_status = "Pending For QC" for valid requests made by a QC role.

API 26: QC Completed Jobs – API Request

Purpose:

To retrieve a list of jobs marked as "**QC Completed**"

Endpoint:

POST {{BASE_URL}}/dashboard/details

Authorization:

Token {{qc_login_token}} (QC user's token must be used)

Payload:

```
{  
  "details_key": "qc_completed"  
}
```

Positive Test Cases:

- Status code should be 200
- Response message should be "Success"
- Response data should be a non-empty array (if jobs exist)
- Each job should contain fields like job_number, mst_model_name, mst_action_status, email_id
- mst_action_status should be "Pending for Delivery"
- Validate response JSON schema
- Validate performance: response time should be < 2 seconds
- Verify correct filtering of jobs that are QC completed

- **Negative Test Cases:**
- Missing token should return 401 Unauthorized
- Invalid or expired token should return 403 Forbidden
- Token of a non-QC role (e.g., engineer) should return empty data or 403 Forbidden
- Incorrect details_key (e.g., "qc_complete" or "QC_Completed") should return empty data or 400
- Empty body should return 400 Bad Request
- Incorrect HTTP method (e.g., GET instead of POST) should return 405 Method Not Allowed
- Malformed JSON body should return 400 Bad Request

API 27: QC Job List API Request

Purpose:

To retrieve a list of jobs assigned to the QC

Endpoint:

GET {{BASE_URL}}/qc

Authorization:

Token {{qc_login_token}} (QC user's token must be used)

Positive Test Cases

- Status code should be 200 OK.
- message field in the response should be "Success".
- data should be a non-empty array when jobs are available.
- Each job in the response should have:
 - mst_action_status equal to "Pending For QC"
 - Required fields like job_number, email_id, serial_number, mst_model_name, assigned_by
- Validate JSON schema for the response.
- Validate data types of fields:
e.g., id should be a number, email_id should be string, assigned_by should not be null.

Negative Test Cases:

- Missing token should return 401 Unauthorized.
- Invalid or expired token should return 403 Forbidden.
- If no jobs are assigned to the logged-in QC, the data array should return empty ([]), and this should be clearly documented.
- Tampering with the URL (e.g., /qc1 or /qc/) should return 404 Not Found.

API 28: Complete QC for Job API Request

Purpose:

Allows the QC to complete the quality check of the job.

Endpoint:

POST {{BASE_URL}}/qc/

Authorization:

Token {{qc_login_token}} (QC user's token must be used)

Payload:

```
{  
  "job_id": {{jobId}}  
}
```

Positive Test Cases

- Status code should be 200 OK.
- Response message should be "QC completed successfully".
- Job's status should be updated in the system (can be verified through job status APIs or dashboard).
- Ensure response time is within 200 ms.
- Validate request with a valid job_id assigned to the logged-in QC.

Negative Test Cases

- Missing Authorization header should return 401 Unauthorized.
- Invalid or expired token should return 403 Forbidden.
- Missing job_id in the payload should return 400 Bad Request.
- Providing a non-existent job_id should return an appropriate error like 404 Not Found or a custom error message.
- Trying to complete a QC for a job not assigned to the current QC should return a 403 Forbidden or relevant message.
- Repeating the same request for an already completed job should return an error like "QC already completed" or a relevant message (if idempotency is enforced).

API 29: Pending for Delivery API Request

Purpose:

Allows the FD to see which jobs are pending for delivery

Endpoint:

POST {{BASE_URL}}/dashboard/details

Authorization:

Token {{fd token}} (FD user's token must be used)

Payload:

```
{  
  "details_key": "pending_for_delivery"  
}
```

Positive Test Cases

- Status code should be 200 OK.
- Response message should be "Success".
- Response data array should not be empty if there are jobs pending for delivery.
- Each item in the data array should have essential fields like ***job_number, mst_action_status, mst_oem_name, serial_number,***
- Confirm that **mst_action_status** is "Pending for Delivery" for all returned jobs.
- Validate response time is within acceptable limits (e.g., < 2 seconds).
- Validate the JSON schema of the response matches expectations.
- Check if the QC Completed Job is present in the data array

Negative Test Cases

- Missing Authorization header should return 401 Unauthorized.
- Invalid or expired token should return 403 Forbidden.
- Sending an incorrect details_key (like "pending_delivery") should return an appropriate error or an empty list.
- Sending an empty payload should return 400 Bad Request or a relevant error.
- Sending non-JSON or malformed JSON in the payload should return 400 Bad Request.

API 30: Ready for Delivery API Request

Purpose:

Allows the FD to see retrieve all read for delivery jobs

Endpoint:

GET {{BASE_URL}}/fd/readyfordelivery

Authorization:

Token {{fd token}} (FD user's token must be used)

Positive Test Cases

- Status code should be 200 OK.
- Response message should be "Success".
- Response data array should contain jobs that are ready for delivery.
- Each job object should have key fields like job_number, mst_action_status, mst_oem_name, serial_number, and assigned_by.
- All mst_action_status values should be "Pending for Delivery" (PDL).
- Validate the response time is within the acceptable threshold (e.g., < 2 seconds).
- Validate the JSON schema of the response against expected fields and types.

Negative Test Cases

- Missing Authorization header should return 401 Unauthorized.
- Invalid or expired token should return 403 Forbidden.
- If no jobs are ready for delivery, the response data should be an empty array but still return "Success".
- Sending an unsupported HTTP method (e.g., POST or PUT) should return 405 Method Not Allowed.
- If there's a backend failure, the API should return 500 Internal Server Error or a proper error code.

API 31: Deliver to Customer API Request

Purpose:

Allows the FD to deliver the phone to the customer and the job will be closed

Endpoint:

POST {{BASE_URL}}/fd/delivertocustomer

Authorization:

Token {{fd token}} (FD user's token must be used)

Payload:

```
{  
  "job_id": {{jobId}}  
}
```

Positive Test Cases

- Status code should be 200 OK.
- Response message should be "Delivered to customer successfully".
- The specified job_id should be marked as delivered in the system.
- Empty Data Array Should not be present
- Verify that the job's mst_action_status updates to reflect the delivered state.
- Validate the response time is within the acceptable threshold (e.g., < 2 seconds).
- Validate the JSON schema of the response.
- Confirm that the delivery action is recorded in the delivery logs/audit trail.

Negative Test Cases

- Missing Authorization header should return 401 Unauthorized.
- Invalid or expired token should return 403 Forbidden.
- Sending an invalid job_id (e.g., nonexistent or already delivered) should return an appropriate error like 404 Not Found or 400 Bad Request.
- Missing job_id in the payload should return 400 Bad Request.

API 32: Job Search API Request

Purpose:

Allows the User to Get the Details of the job

Endpoint:

POST {{BASE_URL}}/fd/delivertocustomer

Authorization:

Token {{fd token}} (FD/SUP/ENG/QC user's token can be used)

Payload:

```
{  
  "searchText": "{{jobNumber}}"  
}
```

Positive Test Cases

- Status code should be 200 OK.
- ***Response should return job details matching the provided jobNumber. [Cross Check the details with the Create Job API Response details]***
- ***Repair Start and Repair End Date should be properly set.***

Negative Test Cases

- Missing Authorization header should return 401 Unauthorized.
- Invalid or expired token should return 403 Forbidden.

API 33: Job Details API Request

Endpoint:

POST {{BASE_URL}}/ /job/details/{{jobNumber}}

Authorization:

Token {{fd token}}

Positive Test Cases

- Status code should be 200 OK.
- All top-level job details (job_number, name, email_id, status, etc.) match the job queried.
- The details array contains correct status progression (JobCreation → Pending for Job Assignment → Pending for Repair → Pending for QC → Pending for Delivery → Delivered → Closed) [Order Needs to be maintained]
- Cross Check assigned by and assigned to data
- Check performance (response time under 200 miliseconds).

Negative Test Cases

- Missing or invalid Authorization token should return 401 Unauthorized.
- Supplying an empty or malformed jobNumber (like symbols or blank) should return 400 Bad Request
- Using GET instead of POST should return 405 Method Not Allowed.