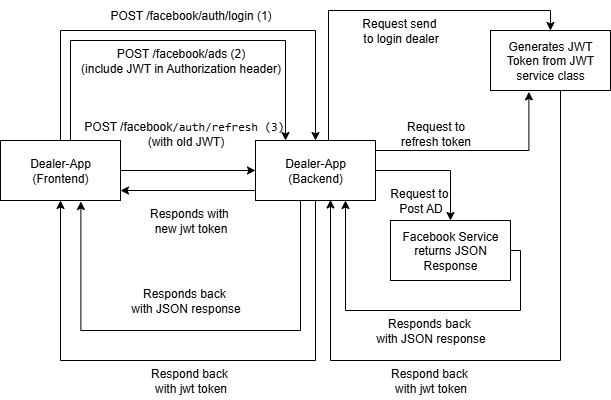
**Task 2: Facebook Ads Integration – System Design**

**Objective -** Design and implement backend APIs to enable dealers to authenticate with Facebook, post vehicle ads, and manage token lifecycle securely.

**1. Architecture Diagram**

****

**Description:**

* **Client (Dealer App)**: Sends requests to backend APIs
* **Backend (Spring Boot)**: Handles OAuth, posting ads, and token management
* **Facebook OAuth / Graph API**: Provides authentication and ad posting services

**Flow:**

1. Dealer logs in via Facebook OAuth → Backend receives authorization code
2. Backend exchanges code for access token → Returns success to client
3. Dealer posts vehicle ad → Backend sends POST request to Facebook Service
4. Backend receives response from Facebook API → Returns Response to Clinet
5. Token expiry handled automatically → Backend refreshes token if needed

**2. API Design**

| **Endpoint** | **Method** | **Request** | **Response** | **Notes** |
| --- | --- | --- | --- | --- |
| /facebook/auth/login | POST | { "facebookAuthCode": "213123" } | { "accessToken": "<JWT>", "facebookAccessToken": "<FB Token>", "expiresIn": 5184000 } | OAuth login with Facebook |
| /facebook/ads | POST | { "vehicle": {...}, "token": "<JWT>" } | { "adId": "mock\_ad\_123", "status": "posted", "facebookUrl": "<url>" } | Post vehicle ad to Marketplace/Page |
| /facebook/auth/refresh | POST | { "oldJwtToken": "<JWT>" } | { "accessToken": "<newJWT>", "facebookAccessToken": "<refreshed FB Token>", "expiresIn": 5184000 } | Refresh expired token |

**3. Pseudo-Code / Logic**

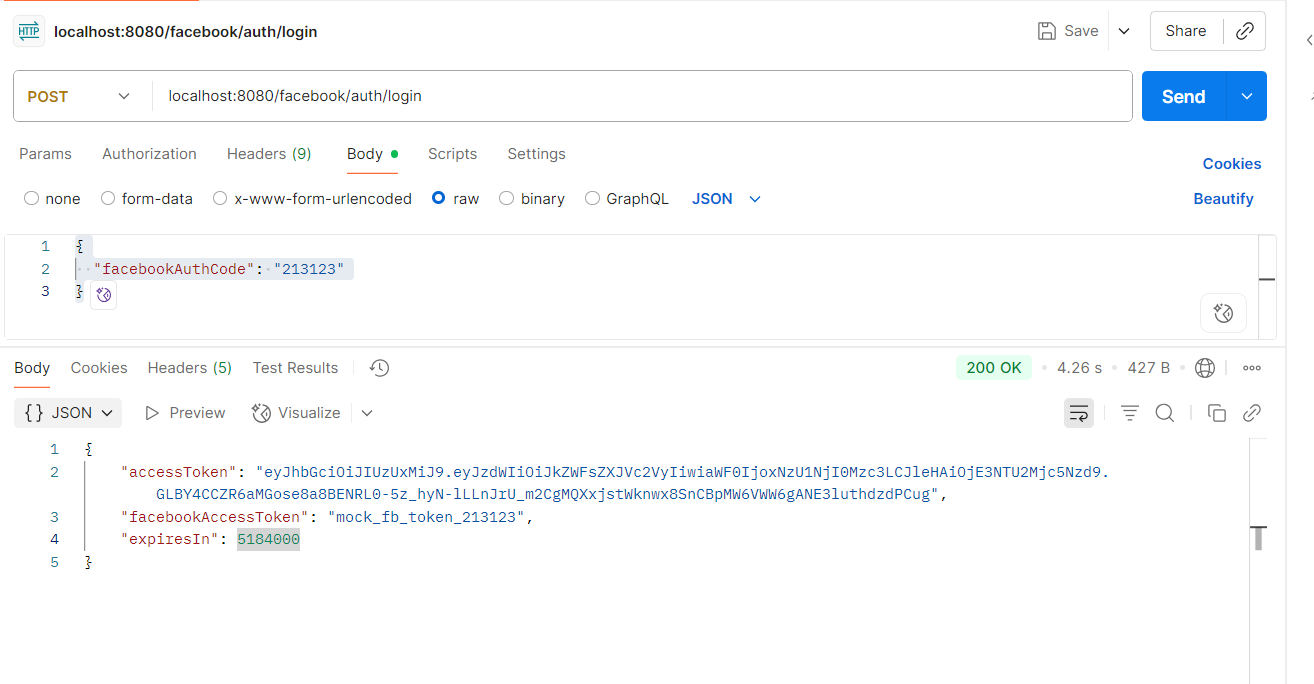
**//OAuth Login**

POST /facebook/auth/login

receive authCode

exchange authCode for accessToken via Facebook API

return accessToken to client



**// Post Vehicle Ad**

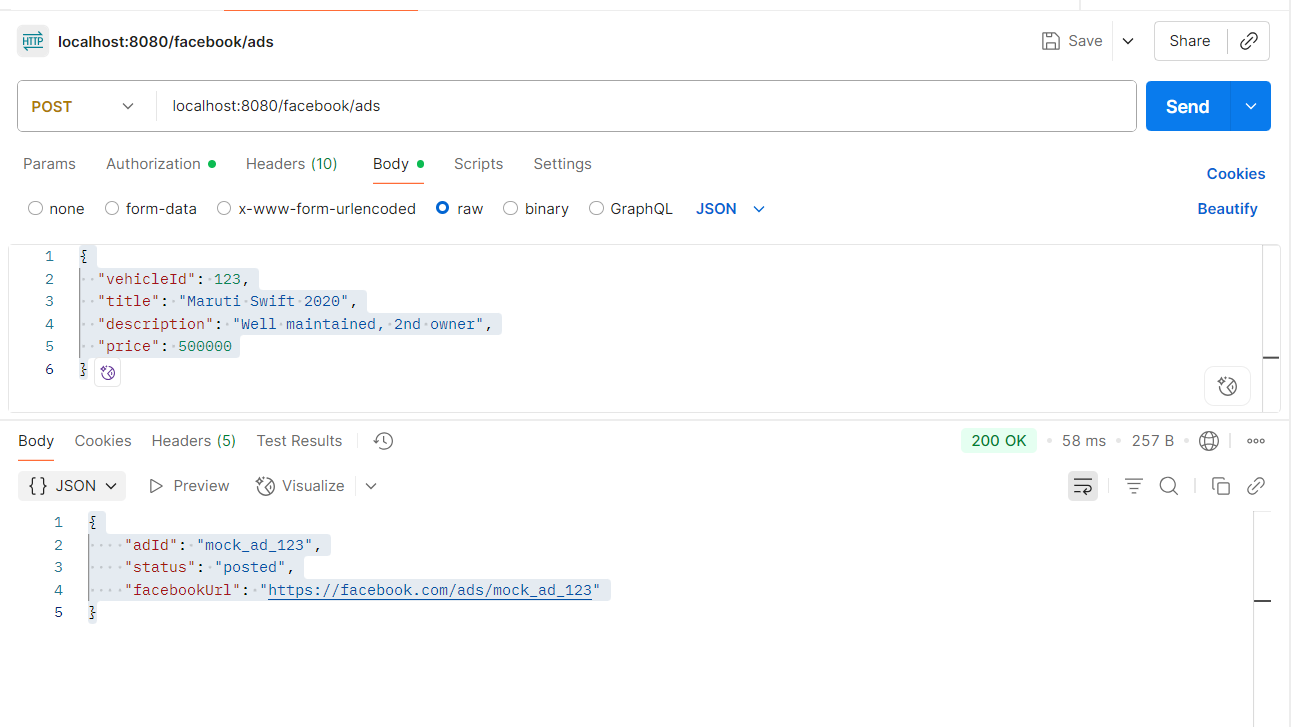
POST /facebook/ads

if accessToken expired:

refresh token

send vehicle ad JSON to Facebook Graph API

return response to client

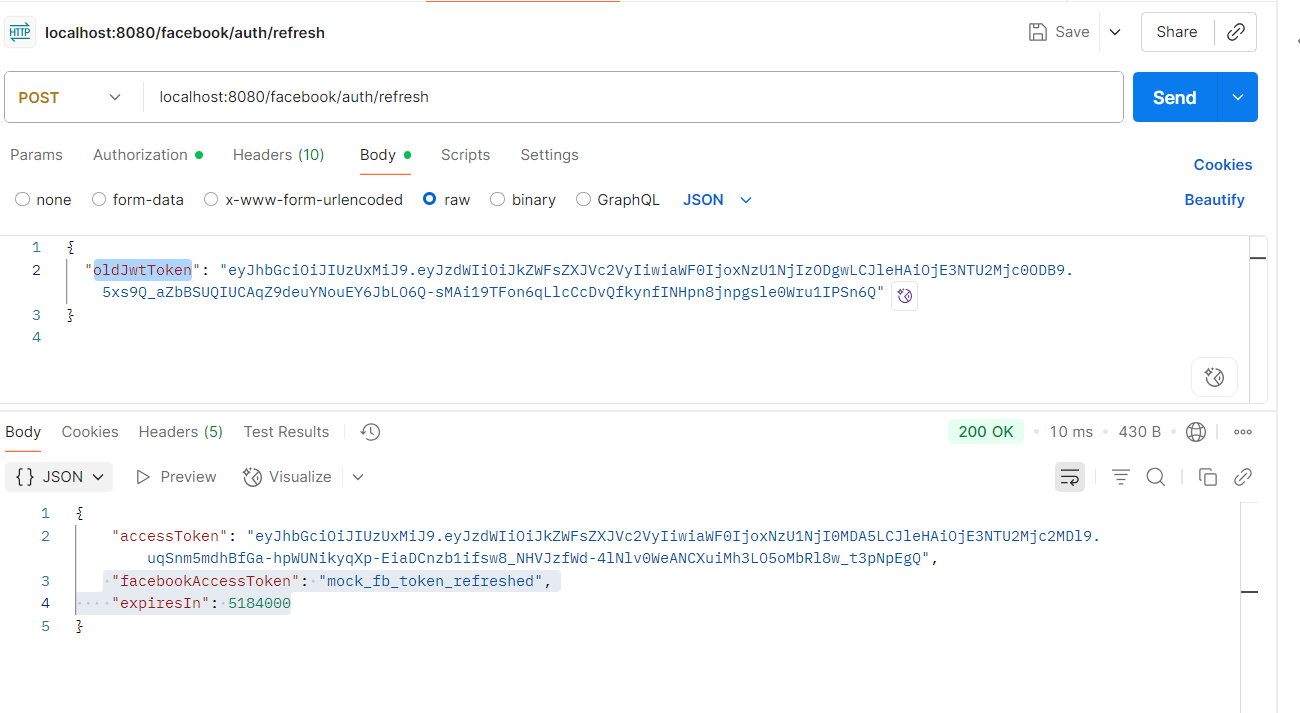


**// Refresh Token**

POST /facebook/auth/refresh

use refreshToken to get new accessToken from Facebook API

return new accessToken to client



**5. Error Handling**

* Invalid auth code → return 401 Unauthorized
* Expired token → call refresh endpoint
* Facebook API error → return 500 Internal Server Error with message

**6. Notes**

* All communication is HTTPS
* Tokens are stored securely in backend (not in client)
* Reusable service methods for OAuth, posting, and token refresh