

Postman Setup Guide (Temp Class Help)

1. Environment Setup

- Create an Environment called Local-Dev.
- Variables: base_url = http://127.0.0.1:5000, token = (blank).

2. Collection Structure

- Auth (login, register)
- Customers (CRUD)
- Mechanics (login, CRUD, stats)
- Inventory (parts management)
- Service Tickets (assign mechanics, add parts)
- Utilities (health, docs)

3. Auth Endpoints

- POST /auth/register → Create user
- POST /auth/login → Login (Test tab saves token).
Script: let data = pm.response.json(); pm.environment.set('token', data.access_token);

4. Customers Endpoints

- POST /customers → Create
- GET /customers → All
- GET /customers?email= → Search
- PUT /customers/:id → Update
- DELETE /customers/:id → Delete

5. Mechanics Endpoints

- GET /mechanics/ping → Health
- POST /mechanics/login → Login mechanic
- GET /mechanics → List all
- PUT /mechanics/:id → Update
- DELETE /mechanics/:id → Delete
- GET /mechanics/my-tickets → Assigned tickets
- GET /mechanics/top → Top mechanic
- GET /mechanics/ticket-count → Ticket counts

6. Inventory Endpoints

- POST /inventory → Add part
- GET /inventory → All parts
- GET /inventory/:id → By ID
- PUT /inventory/:id → Update
- DELETE /inventory/:id → Delete

7. Service Tickets Endpoints

- POST /service_tickets → Create ticket
- GET /service_tickets → All tickets
- PUT /service_tickets/:id → Update ticket

- DELETE /service_tickets/:id → Delete ticket
- POST /service_tickets/:id/assign/:mechanic_id → Assign mechanic
- POST /service_tickets/:id/parts → Add part

8. Utilities

- GET /health → API status
- GET /docs → Swagger (if included).

9. Token Use

- Protected routes: Authorization → Bearer Token → {{token}}.
- Send login first, then other requests use the token automatically.

10. Export

- Export the Collection to .json for sharing or VS Code usage.