

Industry-Standard Postman Features Tutorial

1. Custom Headers

Goal: Simulate real-world header usage with custom values.

Steps:

1. Create a new GET request: `{{base_url}}/posts`
2. Go to the Headers tab.
3. Add:

Key: Content-Type -> Value: application/json

Key: Accept -> Value: application/xml

Key: User-Agent -> Value: KamalikaBot/1.0
4. Click Send.

Note: Even though JSONPlaceholder always returns JSON, you're mimicking client preference logic.

2. Dynamic Variables via Pre-request Script

Goal: Add a timestamp header dynamically.

Steps:

1. In the Pre-request Script tab, paste:


```
pm.variables.set("current_timestamp", new Date().toISOString());
```
2. In the Headers tab, add:

Key: X-Timestamp -> Value: `{{current_timestamp}}`

3. Chaining Requests

Goal: Use data from one request (POST) in the next (GET).

POST Request:

- Endpoint: POST {{base_url}}/posts

- Body:

```
{  
  "title": "Auto chain",  
  "body": "This post was created just now",  
  "userId": 1  
}
```

- Tests tab:

```
let jsonData = pm.response.json();
```

```
pm.environment.set("new_post_id", jsonData.id);
```

GET Request:

- Endpoint: GET {{base_url}}/posts/{{new_post_id}}

4. Data-Driven Testing (Collection Runner + CSV)

Goal: Run multiple test cases using CSV input.

Steps:

1. Create a CSV file:

title,body

First Post,Hello 1

Second Post,Hello 2

Third Post,Hello 3

2. In the POST request body:

```
{
```

```
"title": "{{title}}",  
"body": "{{body}}",  
"userId": 1  
}
```

3. Go to Collection Runner -> Select collection -> Import CSV -> Run.

5. Mock Server (Optional)

Goal: Simulate an API response before the backend is built.

Steps:

1. Go to Mock Servers in Postman.
2. Click Create Mock Server.
3. Choose a collection and set example responses.
4. Share mock URL with frontend/devs.

6. Monitor (Scheduled Runs)

Goal: Detect API issues proactively.

Steps:

1. Click on your collection -> Monitor -> Create Monitor.
2. Set schedule (hourly, daily).
3. Choose email alert settings.

7. Newman + CI/CD

Goal: Run Postman tests via CLI or automation pipelines.

Steps:

1. Export your collection and environment (Save as JSON).

2. Install Newman:

```
npm install -g newman
```

3. Run from terminal:

```
newman run collection.json -e environment.json -r cli,html
```

8. Postman Flows (Visual Chain)

Goal: Drag-n-drop API chaining visually.

Steps:

1. Go to Flows on the sidebar.
2. Drag requests and connect logic blocks.
3. Click Run Flow to visualize execution.