How to Test Your POST API view\_deal\_sheet.php  
  
view\_deal\_sheet.php  
  
<http://localhost/my-api-project/deals_sheet/view_deal_sheet.php>  
  
Headers:  
Content-Type: application/json  
  
{

"page": 1,

"per\_page": 5,

"sort\_data": "latest"

}  
  
  
Basic Fetch, default parameters  
{

"page": 2,

"per\_page": 2

}

Filtering by ID (valid)  
  
{

"id": 3

}

Filtering by title (partial match)  
  
{

"title": "NASDAQ"

}

Filtering by owner (user\_id)  
  
{

"owner": 27

}

Sorting options  
  
{

"sort\_data": "ascending"

}

# post\_deal\_sheet.php Different Scenarios for Testing post\_deal\_sheet.php API in Postman

| **Scenario No.** | **Description** | **Request Body Example** | **Expected Result** |
| --- | --- | --- | --- |
| **1** | **Successful creation** | { "title": "Deal 1", "descriptions": "<p>desc</p>", "user\_id": 27 } | Status: true, message: "Deal sheet created successfully.", data contains created record |
| **2** | Missing **title** field | { "descriptions": "<p>desc</p>", "user\_id": 27 } | Status: false, message about missing parameters |
| **3** | Empty **title** value | { "title": "", "descriptions": "<p>desc</p>", "user\_id": 27 } | Status: false, message about missing parameters |
| **4** | Missing **descriptions** field | { "title": "Deal 2", "user\_id": 27 } | Status: false, message about missing parameters |
| **5** | Empty **descriptions** value | { "title": "Deal 2", "descriptions": "", "user\_id": 27 } | Status: false, message about missing parameters |
| **6** | Missing **user\_id** field | { "title": "Deal 3", "descriptions": "<p>desc</p>" } | Status: false, message about missing parameters |
| **7** | Invalid **user\_id** (non-numeric) | { "title": "Deal 3", "descriptions": "<p>desc</p>", "user\_id": "abc" } | Status: false, message about missing parameters |
| **8** | Invalid JSON body | { title: "Deal 4", descriptions: "<p>desc</p>", user\_id: 27 } (missing quotes) | Status: false, message "Invalid JSON input." |
| **9** | Empty request body | (Empty body) | Status: false, message "Invalid JSON input." |
| **10** | Using GET instead of POST | Method: GET; Body: any | Status: false, message "Invalid request method. Use POST only." |
| **11** | SQL Injection attempt (security) | { "title": "Test'; DROP TABLE users;--", "descriptions": "<p>desc</p>", "user\_id": 27 } | Should safely fail or sanitize; no injection; status false or true if sanitized |
| **12** | Very long strings for title and description | Large string input (e.g., 5000 chars) | Status: true if DB supports it; otherwise error message (test max limits) |

edit\_deal\_sheet.php  
  
  
GET deal sheet by ID (view only)  
**Method:** GET  
  
http://localhost/my-api-project/deals\_sheet/edit\_deal\_sheet.php?id=4

POST update all fields successfully  
**Method:** POST  
Body (raw JSON):  
  
{

"id": 4,

"title": "Updated Title",

"descriptions": "<p>Updated description.</p>",

"user\_id": 27

}

3. **POST update title only  
  
{**

**"id": 4,**

**"title": "New Title Only"**

**}**

5. **POST update user\_id only  
  
{**

**"id": 4,**

**"user\_id": 28**

**}**

7. **POST with user\_id that doesn't exist  
{**

**"id": 4,**

**"user\_id": 9999**

**}**

9. **POST with invalid id (non-numeric or missing)  
{**

**"id": "abc",**

**"title": "Test Title"**

**}**