

Create Product

The screenshot shows the Postman application interface. The left sidebar displays 'Jasmine's Workspace' with collections like 'My Collection' and 'ShoppingProducts'. Under 'ShoppingProducts', there are five requests: 'POST Create Product', 'GET Get Products', 'GET Get Product By ID', 'PUT Update Product', and 'DELETE Delete Product'. The 'POST Create Product' request is selected and expanded. The 'Body' tab shows a JSON payload:

```
1 {  
2   "name": "Red Hoodie",  
3   "description": "A warm hoodie",  
4   "price": 39.99,  
5   "category": "Clothes",  
6   "inStock": 20  
7 }  
8
```

The response tab shows a successful '201 Created' response with a response body:

```
1 {  
2   "success": true,  
3   "data": {  
4     "name": "Red Hoodie",  
5     "description": "A warm hoodie",  
6     "price": 39.99,  
7     "category": "Clothes",  
8     "imageUrl": "",  
9     "inStock": 20,  
10    "isActive": true,  
11    "_id": "69169d8a069141ab1182886d",  
12    "createdAt": "2025-11-14T03:10:02.889Z",  
13    "updatedAt": "2025-11-14T03:10:02.889Z",  
14    "__v": 0  
15  }  
16 }
```

Get Products

The screenshot shows the Postman application interface. On the left, the sidebar displays 'Jasmine's Workspace' with collections like 'My Collection' and 'ShoppingProducts'. Under 'ShoppingProducts', the 'GET Get Products' endpoint is selected. The main panel shows a 'GET' request to 'http://localhost:5000/api/products'. The response status is '200 OK' with a response time of '3.64 s'. The response body is a JSON object:

```
1 {  
2   "success": true,  
3   "page": 1,  
4   "limit": 10,  
5   "total": 1,  
6   "pages": 1,  
7   "data": [  
8     {  
9       "id": "69169d8a069141ab1182886d",  
10      "name": "Red Hoodie",  
11      "description": "A warm hoodie",  
12      "price": 39.99,  
13      "category": "Clothes",  
14      "imageUrl": "",  
15      "inStock": 20,  
16      "isActive": true,  
17      "createdAt": "2025-11-14T03:10:02.889Z",  
18      "updatedAt": "2025-11-14T03:10:02.889Z",  
19      "_v": 0  
20    }  
21  ]  
22 }
```

Get Product by ID

The screenshot shows the Postman application interface. On the left, the sidebar displays 'Jasmine's Workspace' with collections like 'My Collection' and 'ShoppingProducts'. Under 'ShoppingProducts', the 'GET Get Product By ID' request is selected. The main panel shows the request details: method 'GET', URL 'http://localhost:5000/api/products/69169d8a069141ab1182886d', and various tabs like Docs, Params, Headers, Body, Scripts, and Settings. The 'Params' tab is active, showing a table for query parameters. Below the table, the 'Body' tab shows the response in JSON format:

```
1 {  
2   "success": true,  
3   "data": {  
4     "_id": "69169d8a069141ab1182886d",  
5     "name": "Red Hoodie",  
6     "description": "A warm hoodie",  
7     "price": 39.99,  
8     "category": "Clothes",  
9     "imageUrl": "",  
10    "inStock": 20,  
11    "isActive": true,  
12    "createdAt": "2025-11-14T03:10:02.889Z",  
13    "updatedAt": "2025-11-14T03:10:02.889Z",  
14    "__v": 0  
15  }  
16 }
```

Update Product

The screenshot shows the Postman application interface. On the left, the sidebar displays 'Jasmine's Workspace' with collections like 'My Collection', 'ShoppingProducts', and 'PUT Update Product'. The main area shows a 'PUT' request for 'Update Product' with the URL `http://localhost:5000/api/products/69169d8a069141ab1182886d`. The request body is set to raw JSON:

```
1 {  
2   "price": 49.99  
3 }  
4
```

The response status is 200 OK, with a timestamp of 118 ms and a size of 547 B. The response body is displayed as JSON:

```
1 {  
2   "success": true,  
3   "data": {  
4     "_id": "69169d8a069141ab1182886d",  
5     "name": "Red Hoodie",  
6     "description": "Updated description",  
7     "price": 49.99,  
8     "category": "Clothes",  
9     "imageUrl": "",  
10    "inStock": 20,  
11    "isActive": true,  
12    "createdAt": "2025-11-14T03:10:02.889Z",  
13    "updatedAt": "2025-11-14T03:28:58.872Z",  
14    "__v": 0  
15  }  
16 }
```

Delete Product

Delete Product - Jasmine's Wor...

datajas-4389783.postman.co/workspace/86ea31a8-3b50-4cda-8936-068f139df6ab/request/50...

Home Workspaces API Network

Jasmine's Workspace

New Import

POST Create I | GET Get Prod | GET Get Prod | PUT Update F | DEL Delete Pr | + | No environment | Upgrade

HTTP ShoppingProducts / Delete Product

DELETE http://localhost:5000/api/products/69169d8a069141ab1182886d

Save Share

Docs Params Auth Headers (7) Body Scripts Settings Cookies

Query Params

Key	Value	Description	...	Bulk Edit
Key	Value	Description		

Body 200 OK 104 ms 575 B Save Response

{ } JSON Preview Visualize

```
1 {  
2   "success": true,  
3   "message": "Product deleted",  
4   "data": {  
5     "_id": "69169d8a069141ab1182886d",  
6     "name": "Red Hoodie",  
7     "description": "Updated description",  
8     "price": 49.99,  
9     "category": "Clothes",  
10    "imageUrl": "",  
11    "inStock": 20,  
12    "isActive": true,  
13    "createdAt": "2025-11-14T03:10:02.889Z",  
14    "updatedAt": "2025-11-14T03:28:58.872Z",  
15    "__v": 0  
16  }  
17 }
```

Console Import Complete

Runner Capture requests Desktop Agent Cookies Vault Trash

The screenshot shows the Postman application interface. On the left, the sidebar displays 'Jasmine's Workspace' with collections like 'My Collection', 'ShoppingProducts', and various API endpoints. The main area shows a DELETE request to 'http://localhost:5000/api/products/69169d8a069141ab1182886d'. The response body is a JSON object indicating success, message, and the deleted product details.