

The screenshot shows the Postman application interface. On the left is a sidebar with icons for Overview, New Collection, No environment, Collections, Environments, History, Flows, and Files (BETA). The main area has tabs for Overview, New Collection, and a selected tab showing a GET request to `http://localhost:8080/products/sorted`. The request details show a GET method, URL, and various tabs like Docs, Params, Headers, Body, Scripts, Settings, and Cookies. Below this is a table for Query Params. The Body tab displays a JSON response with three items:

```
1 [  
2 {  
3   "name": "WEBCAM",  
4   "category": "ACCESORIES",  
5   "price": 2000.0  
6 },  
7 {  
8   "name": "SMARTWATCH",  
9   "category": "FASHION",  
10  "price": 4000.0  
11 },  
12 {  
13   "name": "SPEAKERS",  
14   "category": "ACCESORIES",  
15   "price": 4000.0  
16 }]
```

The response status is 200 OK with 27 ms and 659 B. To the right, there's a documentation panel titled "Generate documentation for collection". It shows a GET request to `http://localhost:8080` and a note about generating documentation for existing collections. A modal window is open, asking for approval to create a new notebook titled "New Collection Documentation" and switch to it. The modal also includes "Approve" and "Reject" buttons. At the bottom, there's a message "Waiting for approval..." and a text input field for describing needs.

Pretty-print

```
[{"name": "WEBCAM", "category": "ACCESORIES", "price": 2000.0}, {"name": "SMARTWATCH", "category": "FASHION", "price": 4000.0}, {"name": "SPEAKERS", "category": "ACCESORIES", "price": 4000.0}, {"name": "POCO", "category": "MOBILES", "price": 15000.0}, {"name": "SAMSUNG", "category": "MOBILES", "price": 20000.0}, {"name": "ACER", "category": "LAPTOPS", "price": 60000.0}, {"name": "HP", "category": "LAPTOPS", "price": 67000.0}, {"name": "DELL", "category": "LAPTOPS", "price": 78000.0}, {"name": "POCO", "category": "MOBILES", "price": 150000.0}]
```

A screenshot of a web browser window. The address bar shows the URL `http://localhost:8080/products/filter?min=10000&max=70000`. Below the address bar is a toolbar with icons for back, forward, refresh, and other browser functions. The main content area displays a JSON array of product objects:

```
[{"name": "SAMSUNG", "category": "MOBILES", "price": 20000.0}, {"name": "ACER", "category": "LAPTOPS", "price": 60000.0}, {"name": "HP", "category": "LAPTOPS", "price": 67000.0}, {"name": "POCO", "category": "MOBILES", "price": 15000.0}]
```



```
{"name": "SAMSUNG", "category": "MOBILES", "price": 20000.0}, {"name": "POCO", "category": "MOBILES", "price": 150000.0}, {"name": "POCO", "category": "MOBILES", "price": 15000.0}]
```

! We're updating our plans and pricing on March 1. See [our blog](#) for more details.

The screenshot shows the Postman application interface. On the left, there's a sidebar with icons for Overview, New Collection, No environment, Collections, Environments, History, Flows, and Files (BETA). The main area displays a collection overview for 'New Collection' with an endpoint 'GET http://localhost:8080/products/expensive/10000'. Below this, a detailed view of the endpoint shows methods (GET), URL, and various tabs like Docs, Params, Authorization, Headers, Body, Scripts, Settings, Cookies, and Query Params. A table for Query Params is present. Below the endpoint details, there are tabs for Body, Cookies, Headers (5), Test Results, and a preview section showing JSON response data. The response body is a list of products:

```
1 [  
2   {  
3     "name": "SAMSUNG",  
4     "category": "MOBILES",  
5     "price": 20000.0  
6   },  
7   {  
8     "name": "ACER",  
9     "category": "LAPTOPS",  
10    "price": 60000.0  
11  },  
12  {  
13    "name": "HP",  
14    "category": "LAPTOPS",  
15    "price": 67000.0  
16  },
```

To the right, a modal window titled 'Generate documentation for collection' is open, showing a 'GET http://localhost:8...' request and a note about generating documentation for an existing collection. It also includes a 'New Collection Documentation' section with 'Approve' and 'Reject' buttons, and a message about creating a new notebook. At the bottom right, there's an 'AI' button and a status bar with 'Runner', 'Start Proxy', 'Cookies', 'Vault', 'Trash', and a help icon.

! We're updating our plans and pricing on March 1. See [our blog](#) for more details.

X

Overview | New Collection | GET http://localhost:8080/pr | + | No environment | ↴

Collections | Environments | History | Flows | Files | BETA

HTTP http://localhost:8080/products/filter

GET http://localhost:8080/products/filter?min=10000&max=70000

Send | Save | ↴

Docs Params Authorization Headers (7) Body Scripts Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
min	10000		
max	70000		
Key	Value	Description	

Body Cookies Headers (5) Test Results | ↴

200 OK • 11 ms • 378 B • ⌂ • ⌂

{ } JSON ▾ ▶ Preview | Visualize | ↴

```
1 [  
2 {  
3   "name": "SAMSUNG",  
4   "category": "MOBILES",  
5   "price": 20000.0  
6 },  
7 {  
8   "name": "ACER",  
9   "category": "LAPTOPS",  
10  "price": 60000.0  
11 },  
12 {  
13   "name": "HP",  
14   "category": "LAPTOPS",  
15   "price": 67000.0  
16 },
```

Waiting for approval...

GET http://localhost:8...
Describe what you need. Press @ for context, / for Skills.
Auto ▾

Cloud View | Find and replace | Console | Terminal | Runner | Start Proxy | Cookies | Vault | Trash | ↴

Overview | New Collection | GET http://localhost:8080/pr | + | No environment | Save | Generate documentation for collection | + | ⌂ | X

HTTP http://localhost:8080/products/category/FASHION

GET http://localhost:8080/products/category/FASHION Send

Docs Params Authorization Headers (7) Body Scripts Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

Body Cookies Headers (5) Test Results

200 OK • 13 ms • 223 B • [View Response](#)

{ } JSON ▾ Preview Visualize

```
1 [  
2 {  
3   "name": "SMARTWATCH",  
4   "category": "FASHION",  
5   "price": 4000.0  
6 }]  
7 ]
```

Waiting for approval...

GET http://localhost:8... Describe what you need. Press @ for context, / for Skills.

Auto

Cloud View Find and replace Console Terminal Runner Start Proxy Cookies Vault Trash