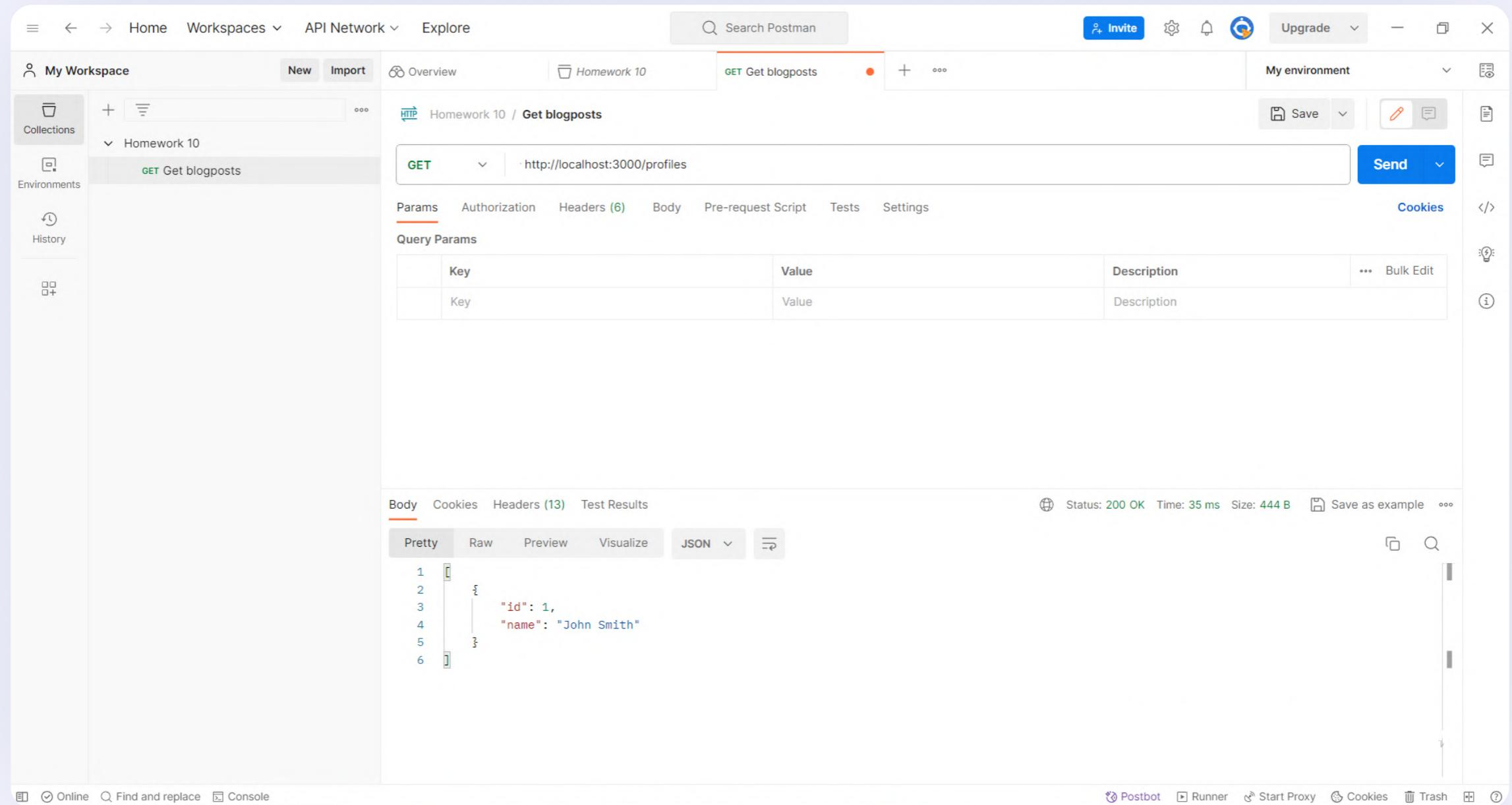




POSTMAN

# GET requests

# GET all profiles



The screenshot shows the Postman application interface. The top navigation bar includes 'Home', 'Workspaces', 'API Network', and 'Explore' buttons, along with a search bar and user account options. The left sidebar features 'My Workspace' with 'Collections' (Homework 10), 'Environments', and 'History' sections. The main workspace displays a collection named 'Homework 10' with a 'Get blogposts' request. The request details show a 'GET' method, URL 'http://localhost:3000/profiles', and a table for 'Query Params' with one row. The 'Body' tab is selected, showing a JSON response with a single profile object:

```
1
2  {
3    "id": 1,
4    "name": "John Smith"
5  }
6
```

The status bar at the bottom shows 'Status: 200 OK', 'Time: 35 ms', and 'Size: 444 B'. The bottom navigation bar includes 'Postbot', 'Runner', 'Start Proxy', 'Cookies', 'Trash', and a help icon.

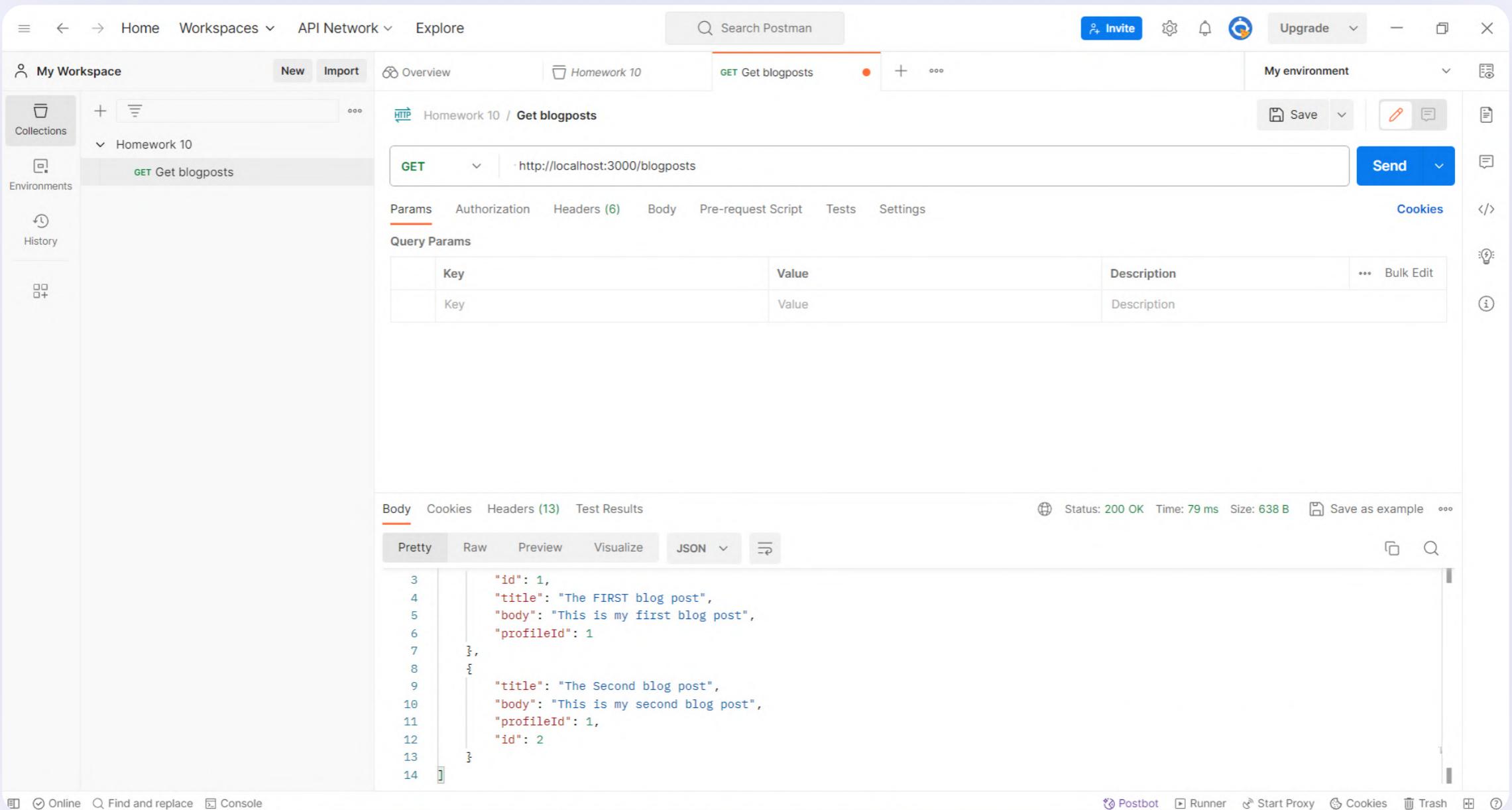
# GET one profile

The screenshot shows the Postman application interface. The top navigation bar includes 'Home', 'Workspaces', 'API Network', and 'Explore' buttons, along with a search bar and user account options. The left sidebar features 'My Workspace' with 'Collections' (Homework 10), 'Environments', and 'History' sections. The main workspace displays a collection named 'Homework 10' with a single request named 'Get blogposts'. The request method is 'GET' and the URL is 'http://localhost:3000/profiles/1'. The 'Params' tab is selected, showing a table with one row: 'Key' (empty) and 'Value' (empty). The 'Headers' tab shows 6 items. The 'Body' tab is selected, showing the response body in 'Pretty' format:

```
1  "id": 1,  
2  "name": "John Smith"  
3  
4
```

The response status is '200 OK' with a time of '46 ms' and a size of '432 B'. The bottom navigation bar includes 'Online', 'Find and replace', 'Console', 'Postbot', 'Runner', 'Start Proxy', 'Cookies', 'Trash', and a help icon.

# GET all blogposts



The screenshot shows the Postman application interface. The top navigation bar includes Home, Workspaces, API Network, and Explore. A search bar and various tool buttons are also present. The left sidebar shows 'My Workspace' with a 'Collections' section containing 'Homework 10' and a 'GET Get blogposts' item. The main workspace displays an 'HTTP Homework 10 / Get blogposts' request. The method is set to 'GET' and the URL is 'http://localhost:3000/blogposts'. The 'Params' tab is selected, showing a table for 'Query Params' with one row containing 'Key' and 'Value'. The 'Headers' tab shows 6 items. The 'Body' tab is selected, showing the response body in 'Pretty' format. The response is a JSON array of two blogposts:

```
3 |   "id": 1,
4 |   "title": "The FIRST blog post",
5 |   "body": "This is my first blog post",
6 |   "profileId": 1
7 | },
8 | {
9 |   "title": "The Second blog post",
10 |  "body": "This is my second blog post",
11 |  "profileId": 1,
12 |  "id": 2
13 | }
14 | ]
```

The status bar at the bottom indicates the response was successful (200 OK), took 79 ms, and was 638 B in size. Other options like 'Save as example' are also visible.

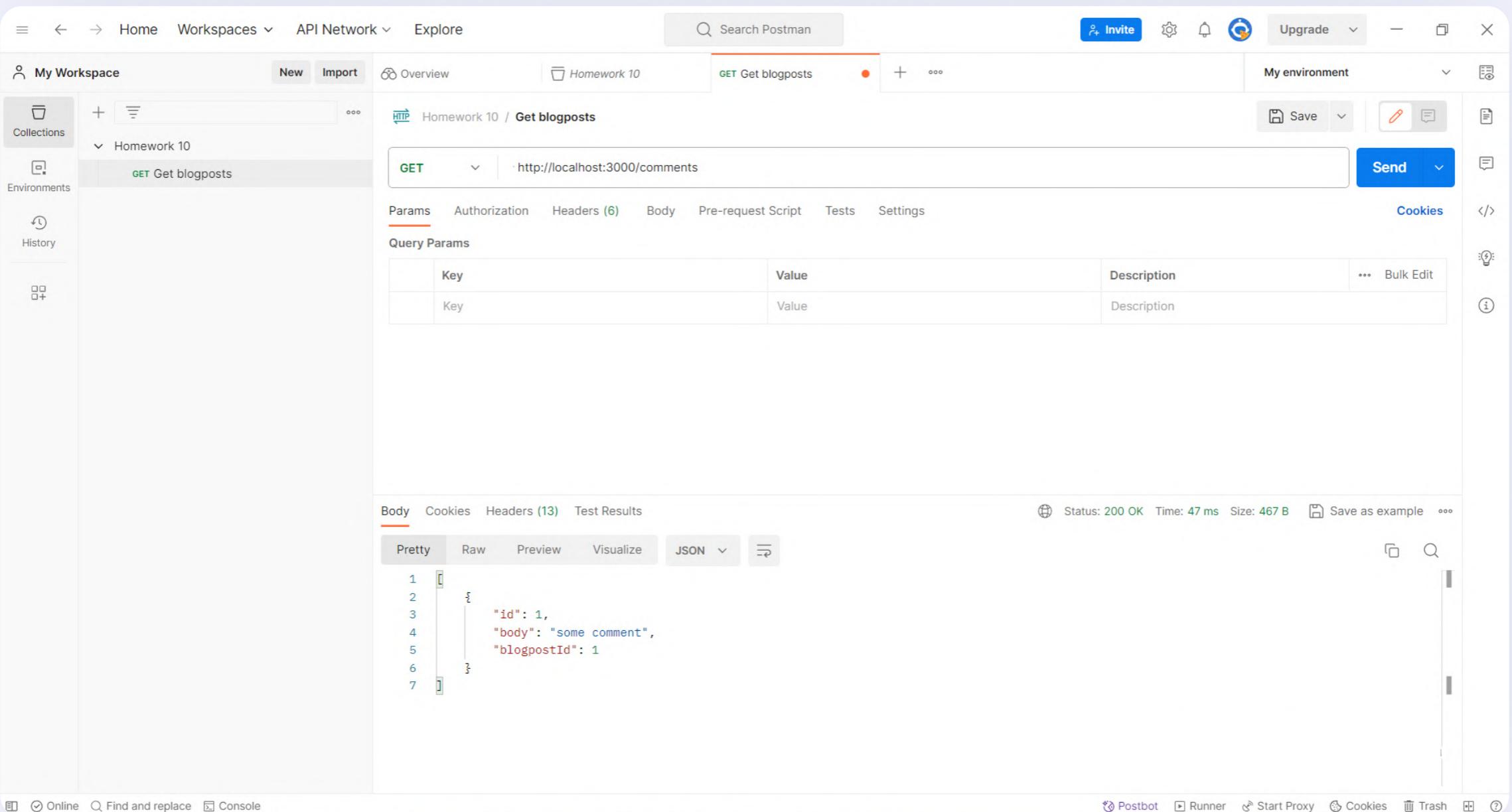
# GET one blogpost

The screenshot shows the Postman application interface. The top navigation bar includes 'Home', 'Workspaces', 'API Network', and 'Explore' buttons, along with a search bar and user account options. The left sidebar features 'My Workspace' with 'Collections' (Homework 10), 'Environments', and 'History' sections. The main workspace displays a collection named 'Homework 10' with a single item: 'GET Get blogposts'. The request details show a 'GET' method and the URL 'http://localhost:3000/blogposts/1'. The 'Params' tab is selected, showing a table for 'Query Params' with one entry: 'Key'. The 'Headers' tab shows 6 items. The 'Body' tab is selected, showing the response in 'Pretty' format:

```
1  "id": 1,
2  "title": "The FIRST blog post",
3  "body": "This is my first blog post",
4  "profileId": 1
```

The response status is 200 OK, with a time of 19 ms and a size of 501 B. The bottom navigation bar includes 'Online', 'Find and replace', 'Console', 'Postbot', 'Runner', 'Start Proxy', 'Cookies', 'Trash', and a help icon.

# GET all comments



The screenshot shows the Postman application interface. The left sidebar displays 'My Workspace' with a 'Collections' section containing 'Homework 10' and a 'GET Get blogposts' request. The main workspace shows a 'GET' request to 'http://localhost:3000/comments'. The 'Body' tab in the results section shows a single comment in JSON format:

```
1
2  {
3    "id": 1,
4    "body": "some comment",
5    "blogpostId": 1
6  }
7
```

The status bar at the bottom indicates the request was successful with a status of 200 OK, a time of 47 ms, and a size of 467 B.

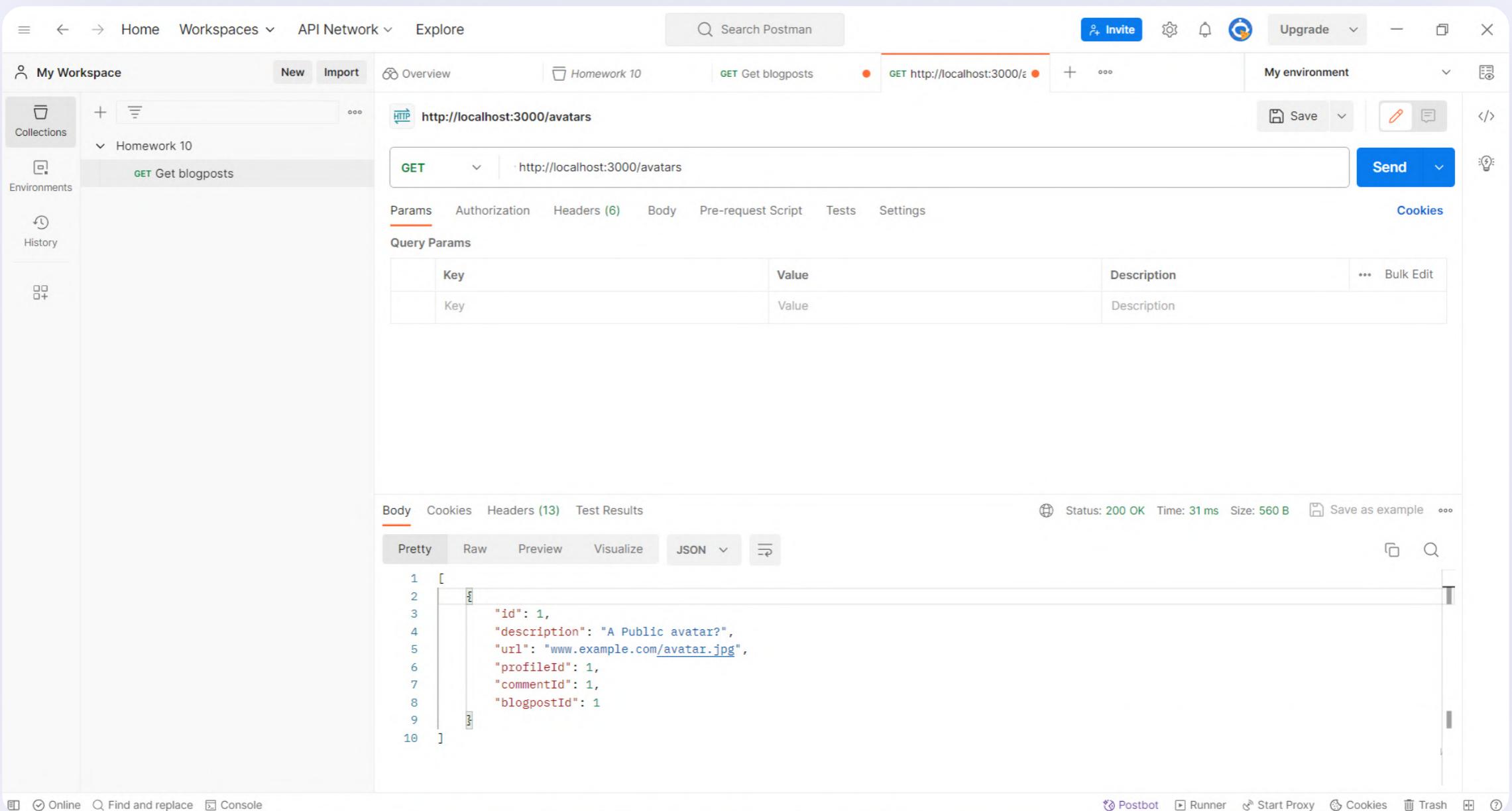
# GET one comment

The screenshot shows the Postman application interface. The left sidebar displays 'My Workspace' with a 'Collections' section containing 'Homework 10' and a 'GET Get blogposts' request. The main workspace shows a 'GET' request to 'http://localhost:3000/comments/1'. The 'Body' tab in the results section displays a JSON response with the following structure:

```
1  "id": 1,
2  "body": "some comment",
3  "blogpostId": 1
```

The status bar at the bottom indicates the request was successful (200 OK) with a time of 24 ms and a size of 453 B.

# GET avatars



The screenshot shows the Postman application interface. The top navigation bar includes Home, Workspaces, API Network, and Explore. The search bar is empty. The top right features an Invite button, settings, notifications, and an upgrade link. The left sidebar shows 'My Workspace' with 'Collections' (empty), 'Environments' (empty), and 'History' (empty). The main workspace shows a collection named 'Homework 10' with a 'GET Get blogposts' request. The request details show a GET method to 'http://localhost:3000/avatars'. The 'Params' tab is selected, showing a table for 'Query Params' with one row: 'Key' and 'Value'. The 'Headers' tab shows 6 items. The 'Body' tab is collapsed. The 'Tests' and 'Settings' tabs are also visible. Below the request details, the 'Body' tab is selected, showing the response in 'Pretty' format. The response is a JSON array with one element:

```
1 [  
2   {  
3     "id": 1,  
4     "description": "A Public avatar?",  
5     "url": "www.example.com/avatar.jpg",  
6     "profileId": 1,  
7     "commentId": 1,  
8     "blogpostId": 1  
9   }  
10 ]
```

The status bar at the bottom shows 'Status: 200 OK' with a timestamp of 'Time: 31 ms' and a size of 'Size: 560 B'. Other buttons in the bottom right include 'Save as example', 'Postbot', 'Runner', 'Start Proxy', 'Cookies', 'Trash', and a help icon.

# POST requests

# POST profile

The screenshot shows the Postman application interface. The left sidebar displays a 'My Workspace' section with a 'Collections' tree. The 'POST Profile' item under the 'POST' collection is currently selected. The main workspace shows a POST request to `http://localhost:3000/blogposts/3`. The 'Body' tab is selected, showing the following JSON payload:

```
1
2   "name": "Tanya Pylypenko"
3
```

The response status is 201 Created, with a response body showing:

```
1
2   "name": "Tanya Pylypenko",
3   "id": 2
4
```

At the bottom, there are navigation links for 'Online', 'Find and replace', 'Console', 'Postbot', 'Runner', 'Start Proxy', 'Cookies', 'Trash', and a help icon.

# POST blogpost

The screenshot shows the Postman application interface. The left sidebar is titled 'My Workspace' and contains a 'Collections' section with 'Homework 10' expanded, showing 'GET' and 'POST' requests. The 'POST' section is currently selected, with 'Blogpost' highlighted. The main workspace shows a 'POST' request for 'Blogpost' with the URL 'http://localhost:3000/blogposts'. The 'Body' tab is selected, showing the following JSON payload:

```
1
2   "title": "The THIRD blog post",
3   "body": "This is my third blog post",
4   "profileId": 1
5
```

The 'Headers' tab shows 8 headers. The 'Body' tab shows the JSON payload. The 'Tests' and 'Settings' tabs are also visible. At the bottom, the status is 'Status: 201 Created' with a 'Save as example' button. The bottom navigation bar includes 'Online', 'Find and replace', 'Console', 'Postbot', 'Runner', 'Start Proxy', 'Cookies', 'Trash', and a help icon.

# POST comment

The screenshot shows the Postman application interface. The left sidebar is titled 'My Workspace' and contains a 'Collections' section with 'Homework 10' expanded, showing 'GET' and 'POST' requests. The 'POST' section is currently selected, with 'Comment' highlighted. The main workspace shows a 'POST' request to 'http://localhost:3000/comments'. The 'Body' tab is selected, showing the following JSON payload:

```
1  "body": "It is the interesting major",
2  "blogpostId": 3
3
4
```

The 'Headers' tab shows 8 headers. The 'Body' tab shows the JSON content. The 'Tests' and 'Settings' tabs are also visible. At the bottom, the status bar shows 'Status: 201 Created' and other details. The bottom navigation bar includes 'Postbot', 'Runner', 'Start Proxy', 'Cookies', 'Trash', and a help icon.

# POST avatar

The screenshot shows the Postman application interface. The left sidebar displays 'My Workspace' with a collection named 'Homework 10' containing various API endpoints: GET, POST, and PUT methods for Blogpost, Comment, Comment/1, Profile, and Avatar. The main workspace shows a POST request to 'http://localhost:3000/avatars'. The 'Body' tab is selected, showing a JSON payload with fields: description, url, profileId, blogpostId, commentId, and id. The 'Pretty' tab in the results section shows the same JSON data with line numbers. The status bar at the bottom indicates the request was successful with a 201 Created status, 61 ms time, and 755 B size.

```
1
2   ...
3   ...
4   ...
5   ...
6   ...
7   ...
8 }
```

```
1
2   ...
3   ...
4   ...
5   ...
6   ...
7   ...
8 }
```

# PUT requests

# PUT profile

The screenshot shows the Postman application interface. The top navigation bar includes 'Home', 'Workspaces', 'API Network', and 'Explore' buttons, along with a search bar and an 'Invite' button. The left sidebar shows 'My Workspace' with a 'Collections' section containing 'Homework 10', which has 'GET', 'POST', and 'PUT' sections. The 'PUT' section is currently selected and expanded, showing 'Profile' under it. The main workspace shows a 'PUT' request to 'http://localhost:3000/profiles/2'. The 'Body' tab is selected, showing the following JSON payload:

```
1
2   "name": "Tetiana Pylypenko",
3   "id": 2
4
```

The 'Headers' tab shows 8 headers. The 'Tests' and 'Settings' tabs are also visible. Below the request, the 'Body' tab is selected, showing the same JSON payload. The status bar at the bottom indicates 'Status: 200 OK', 'Time: 26 ms', and 'Size: 439 B'. The bottom navigation bar includes 'Online', 'Find and replace', 'Console', 'Postbot', 'Runner', 'Start Proxy', 'Cookies', 'Trash', and a help icon.

# PUT blogpost

The screenshot shows the Postman application interface. The left sidebar displays 'My Workspace' with collections like 'Homework 10' and environments. The main area shows a 'PUT' request to `http://localhost:3000/blogposts/3`. The 'Body' tab is selected, showing a JSON payload:

```
1
2   "title": "I study QA",
3   "body": "This is interesting major",
4   "profileId": 2,
5   "id": 3
```

The 'Body' tab also includes tabs for 'Pretty', 'Raw', 'Preview', and 'Visualize'. Below the request, the response is shown with a status of 200 OK, time 31 ms, and size 490 B. The response body is identical to the request body. The bottom navigation bar includes 'Postbot', 'Runner', 'Start Proxy', 'Cookies', 'Trash', and a help icon.

# PUT comment

The screenshot shows the Postman application interface. The left sidebar displays 'My Workspace' with a collection named 'Homework 10' containing various API endpoints: GET Blogposts, GET Profiles, GET Comments, GET Avatars, POST Blogpost, POST Comment, POST Profile, and PUT Blogpost. The 'PUT Comments' endpoint is currently selected. The main workspace shows a 'PUT' request to the URL `http://localhost:3000/comments/2`. The 'Body' tab is selected, showing the following JSON payload:

```
1
2   "body": "Oh it is realy interesting job",
3   "blogpostId": 3,
4   "id": 2
5
```

The 'Body' tab also includes tabs for 'Pretty', 'Raw', 'Preview', 'Visualize', and 'JSON'. Below the request, the response status is shown as 'Status: 200 OK' with a response size of '471 B'. The bottom navigation bar includes 'Online', 'Find and replace', 'Console', 'Postbot', 'Runner', 'Start Proxy', 'Cookies', 'Trash', and a help icon.

# PUT avatar

The screenshot shows the Postman application interface. The top navigation bar includes 'Home', 'Workspaces', 'API Network', 'Explore', a search bar 'Search Postman', and various global buttons like 'Invite', 'Upgrade', and environment dropdowns.

The left sidebar is titled 'My Workspace' and contains sections for 'Collections', 'Environments', and 'History'. The 'Collections' section shows 'Homework 10' expanded, containing 'GET' and 'POST' requests. The 'GET' section includes 'Blogposts', 'Blogposts/1', 'Comments', 'Comment/1', 'Profiles', 'Profile/1', 'Avatars', and 'Avatar/1'. The 'POST' section includes 'Blogpost', 'Comment', 'Comment/1', 'Profile', and 'Avatar'. The 'PUT' section is currently selected and expanded, showing 'Blogpost', 'Comments', 'Profile', and 'Avatar'. The 'Avatar' item is highlighted with a gray background.

The main workspace shows a 'PUT' request for 'http://localhost:3000/avatars/1'. The 'Body' tab is selected, showing the following JSON payload:

```
1
2   "description": "My first avatar",
3   "url": "https://zooclub.ru/attach/4236.jpg",
4   "profileId": 1,
5   "commentId": 1,
6   "blogpostId": 1
7 }
```

Below the request, the 'Body' tab is selected in the results panel, showing the same JSON response:

```
1
2   "description": "My first avatar",
3   "url": "https://zooclub.ru/attach/4236.jpg",
4   "profileId": 1,
5   "commentId": 1,
6   "blogpostId": 1
7
8   "id": 1
9 }
```

The status bar at the bottom indicates 'Status: 200 OK', 'Time: 34 ms', and 'Size: 547 B'. Other buttons in the bottom right include 'Save as example', 'Pretty', 'Raw', 'Preview', 'Visualize', and 'JSON' dropdown.

# 500 Error

My Workspace

Homework 10 / POST / New Request

POST http://localhost:3000/comments

Body (8) Params Authorization Headers (8) Body (1) Pre-request Script Tests Settings

```

1
2     "id": 1,
3     "body": "Hello world",
4     "blogpostId": 2
5

```

Body Cookies Headers (13) Test Results

Pretty Raw Preview Visualize HTML

```

1 Error: Insert failed, duplicate id
2 at Function.insert
3 (C:\Users\Таня\Downloads\API_Course\API_Course\api-testing-foundations\node_modules\lodash-id\src\index.js:49:18)
4 at C:\Users\Таня\Downloads\API_Course\API_Course\api-testing-foundations\node_modules\lodash\lodash.js:4430:28
5 at arrayReduce
6 (C:\Users\Таня\Downloads\API_Course\API_Course\api-testing-foundations\node_modules\lodash\lodash.js:697:21)
7 at baseWrapperValue
8 (C:\Users\Таня\Downloads\API_Course\API_Course\api-testing-foundations\node_modules\lodash\lodash.js:4429:14)
9 at LodashWrapper.wrapperValue
10 (C:\Users\Таня\Downloads\API_Course\API_Course\api-testing-foundations\node_modules\lodash\lodash.js:9114:14)
11 at create
12 (C:\Users\Таня\Downloads\API_Course\API_Course\api-testing-foundations\node_modules\json-server\lib\server\router\plural.js:239:48)

```

Online Find and replace Console

Postbot Runner Start Proxy Cookies Trash

My Workspace

Overview Homework 10

GET Get blogposts

GET http://localhost:3000/comments?\_expand=blogpost

Params (1) Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

Key	Value	Description	Bulk Edit
_expand	blogpost		
Key	Value	Description	

Body Cookies Headers (14) Test Results

Pretty Raw Preview Visualize HTML

```

1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="utf-8">
6   <title>Error</title>
7 </head>
8
9 <body>
10  <pre>TypeError: Cannot read properties of undefined (reading &#39;indexOf&#39;)<br> &nbsp;at C:\Users\Таня\Downloads\API_Course\API_Course\api-testing-foundations\node_modules\lodash-id\src\index.js:49:18<br> &nbsp;at C:\Users\Таня\Downloads\API_Course\API_Course\api-testing-foundations\node_modules\lodash\lodash.js:4430:28<br> &nbsp;at arrayReduce<br> &nbsp;at baseWrapperValue<br> &nbsp;at LodashWrapper.wrapperValue<br> &nbsp;at create<br> &nbsp;at (C:\Users\Таня\Downloads\API_Course\API_Course\api-testing-foundations\node_modules\lodash\lodash.js:9114:14)<br> &nbsp;at (C:\Users\Таня\Downloads\API_Course\API_Course\api-testing-foundations\node_modules\json-server\lib\server\router\plural.js:239:48)<br>
11 </body>
12

```

Online Find and replace Console

Postbot Runner Start Proxy Cookies Trash

Thank you for your attention!