# NEXT.JS

**CRUD** Operations

#### •Next.js supports:

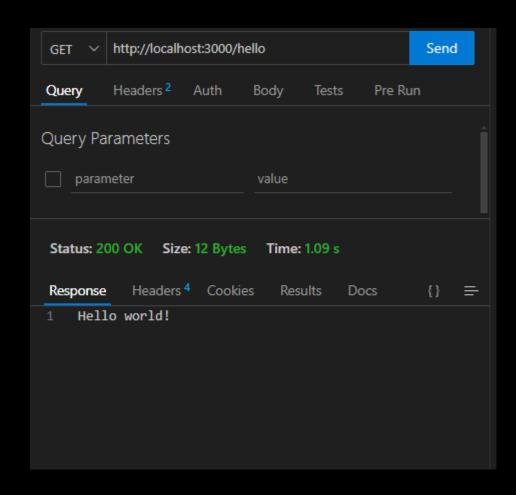
GET POST, PUT, PATCH, DELETE, HEAD, and OPTIONS

•If an unsupported method is called, Next.js Will return a 405 Method Not Allowed response.

# GET()

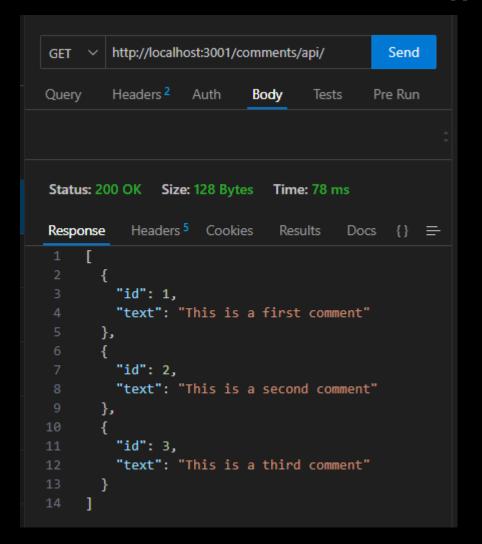
#### Basics

```
export async function GET() {
  return new Response("Hello world!");
}
```



## GET()

```
import { comments } from "../../data";
export async function GET(
  request: Request,
   params,
  }: {
   params: Promise<{ id: string }>;
  const { id } = await params;
  const getIdCommnent = comments.find((comment)
=> comment.id === parseInt(id));
  return Response.json(getIdCommnent);
```



```
import { comments } from "../data";
export async function GET() {
 return Response.json(comments);
export async function POST(request: Request) {
  const comment = await request.json();
  const newComment = {
   id: comments.length + 1,
   text: comment.text,
  comments.push(newComment);
 return new
Response(JSON.stringify(newComment), {
   headers: { "Content-Type":
"application/json" },
   status: 201,
```

#### POST()

```
http://localhost:3001/comments/api/
                                                 Send
Query
         Headers <sup>2</sup>
                    Auth
                             Body 1
                                      Tests
                                              Pre Run
JSON
                      Form Form-encode
                                              GraphQL
JSON Content
                                                Format
         "text" : "New comment"
Status: 201 Created Size: 29 Bytes Time: 26 ms
Response
         Headers 5 Cookies Results Docs {} =
      "id": 4,
      "text": "New comment"
```

#### DYNAMIC ROUTE HANDLERS

```
import { comments } from "../../data";
export async function GET(
 _request: Request,
   params,
 }: {
   params: Promise<{ id: string }>;
  const { id } = await params;
  const getIdCommnent = comments.find((comment)
=> comment.id === parseInt(id));
  return Response.json(getIdCommnent);
```

```
GET 

http://localhost:3001/comments/api/3

Send

Query Headers 

Auth Body 

Tests Pre Run

Status: 200 OK Size: 41 Bytes Time: 46 ms

Response Headers 

Cookies Results Docs 

Time: 46 ms

rid": 3,

"text": "This is a third comment"

This is a third comment"
```

```
import { comments } from "../../data";
export async function PATCH(
  request: Request,
    params,
  }: {
    params: Promise<{ id: string }>;
  const { id } = await params;
  const body = await request.json();
  const { text } = body;
  const index = comments.findIndex((comment) =>
comment.id === parseInt(id));
  comments[index].text = text;
  return Response.json(comments[index]);
```

### PATCH()

```
http://localhost:3001/comments/api/3
                                                Send
PATCH
Query
         Headers <sup>2</sup> Auth
                            Body 1
                                              Pre Run
JSON
                                            GraphQL
               Text
                      Form
                              Form-encode
JSON Content
                                               Format
         "text" : "Updated comment"
Status: 200 OK Size: 33 Bytes Time: 53 ms
Response Headers 5 Cookies Results Docs {} =
      "id": 3,
      "text": "Updated comment"
```

#### import { comments } from "../../data"; export async function DELETE( request: Request, params, **}:** { params: Promise<{ id: string }>; const { id } = await params; const index = comments.findIndex((comment) => comment.id === parseInt(id)); const deletedComment = comments[index]; comments.splice(index, 1); return Response.json(deletedComment);

#### DELETE()

```
http://localhost:3001/comments/api/2
DELETE
                                                Send
                            Body 1
         Headers <sup>2</sup> Auth
                                             Pre Run
Query
JSON
                                            GraphQL
               Text
                     Form Form-encode
JSON Content
                                             Format
Status: 200 OK Size: 42 Bytes Time: 33 ms
          Headers 5 Cookies Results Docs
Response
                                                Copy
      "id": 2,
      "text": "This is a second comment"
```