

# READ FOR STEP 5: BASIC CRUD OPERATIONS

- The basic operations in every database comprises of the following four operations:

**CREATE** (http verb "POST")

**READ** (http verb "get")

**UPDATE** (http verbs "put"/"patch")

**DELETE** (http verb "delete")

- In MongoDB these four basic operations are achieved using the unique style of writing.
- *We've already tried "find()" to respond to "get" request.*
- *We're also know the function "save()" – we could use it for "post" request, but there is also "create()"*

# STEP 5 – FIRST OF ALL, LET’S TRY “POST”, BUT HOW?

- A) For this we’ll need some helper tool, called “Postman” – install it
- B) Meanwhile in server.js create one more route, for
- `App.post('/api/sports/create'`

And use function like

```
(req, res)=>{  
  try{  
    const newSport = await Sport.create(req.body)  
    res.status(201).json(newSport)  
  }catch(err){  
    console.error("There is an error:",err)  
    res.status(500).json({err: err.message})  
  }  
}
```

- C) Open “Postman”, create “Olympics” collection



## STEP 5 - CONTINUED

- D) Create new request (or use template for several requests) – start from “GET” and see that it succeeds
- E) Now create POST request – pay attention, that you should fill the body, for example, if you use “raw”, fill just JSON:
  - `{"name": "badminton",`
  - `"isOlympic": "true",`
  - `"isSummerOlympic": "true",`
  - `"img": "badminton_pictogram.svg" }`
- F) You see an error, right? Show yourself the error message
- G) It means, that your input is not parsed as JSON, and you should add to the MIDDLEWARE of your server
- `app.use(express.json());`



## STEP 5 - CONTINUED

- H) See that now it works OK
- I) Now create POST request again, but use the “urlencoded”, it is what comes from form.
- J) Again does not work?
- K) We need to parse the “form” now, so we use one more MIDDLEWARE:

```
app.use(express.urlencoded({extended:true}));
```