RESTful API

Build your own restful API :

REST = REpresentational State Transfer
HTTPS => Http secure request.

Client -> (HTTP/FTP request)-> Server
Server -> -> Database

The server has a bunch of API (think like a menu), that it could response to the client

REST is the architecture style of design APIs. (There is also other type, but REST is the gold standard for web.)

HTTPs' 5 verb for RESTful API, think it as database CRUD

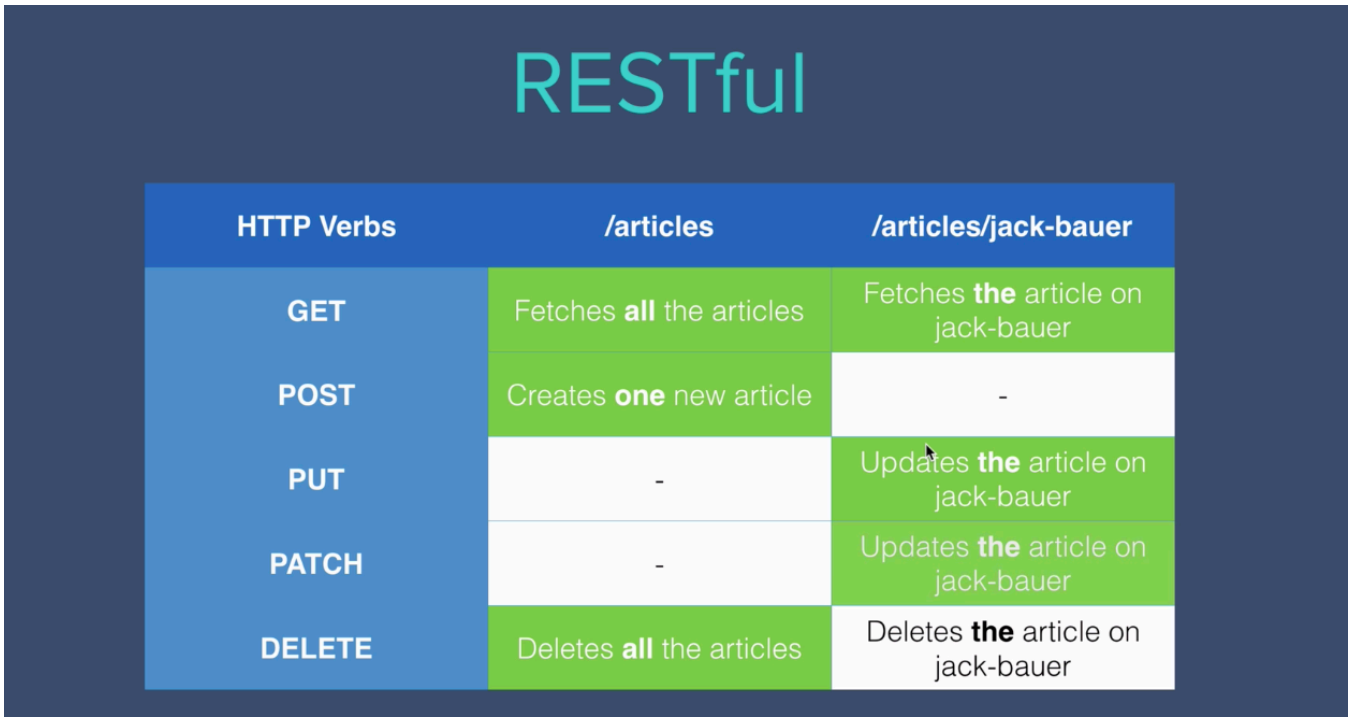
- GET (is refer to "READ" in database : app.get)
- POST (is refer to "CREAT" in database : app.posy)
- PUT
- PATCH
- DELETE (is refer to "DELETE" in database, app.delete)

PUT and PATCH both refer to "UPDATE" the database (app.put/ app.patch)
PUT, updating database with by sending an entire new entry to replace the previous one

PATCH, simply send a piece of data that need to be updated. (Simply update the thing that need to be updated)

Specific pattern and route to make your API

Here we use a mongodb database (a gra[hical user interphase that is used with MOngoDB, use Robo3T) :
https://robomongo.org/



GET ROUTE:

```
app.get('/articles',(req,res)=>{
    if(!err){
    Article.find({},(err,foundArticles)=>{
        res.send(foundArticles);
    });}else{
        console.log(err);
    }

})
```

POST ROUTE:

(Postman will enable us to send data and test our api)

```
//Post RestAPI
app.post("/articles",(req,res)=>{
    console.log(req.body.title);
    console.log(req.body.content);
    //store mongoDB
    const newArticle = new Article({
        title:req.body.title,
        content: req.body.content,
    });
    newArticle.save();
});
```

DELETE ROUTE:

```
//delete RestAPI
app.delete("/articles", (req,res)=> {
    Article.deleteMany(function(err){
        if(!err){
            res.send("successfully deleted all articles ");
        }else{
            res.send(err);
        }
    });
});
```

APP.ROUTE():

When use chained method :
app.route("/").get().post().delete()
Then put your method to each block

ROUTE to specific resource: