

1. Què vol dir MEAN Stack?

Es es refereix a grans trets a MongoDB, ExpressJS, AngularJS i NodeJs. Es un framework per el desenvolupament d'aplicacions i pàgines web dinàmiques que estan basades en JavaScript.

Seguiment de la presentació:

1. Iniciar un projecte nou

```
C:\Users\Ana\nodejs-mp6>npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help json` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (nodejs-mp6) api-rest-nodejs-mp6
version: (1.0.0)
description: Projecte Angular m6-uf3
entry point: (index.js) index.js
test command:
git repository:
keywords:
author: AnaValero
license: (ISC) MIT
About to write to C:\Users\Ana\nodejs-mp6\package.json:

{
  "name": "api-rest-nodejs-mp6",
  "version": "1.0.0",
  "description": "Projecte Angular m6-uf3",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "AnaValero",
  "license": "MIT"
}

Is this OK? (yes) y
```

```
nodejs-mp6 > {} package.json > ...
1  [
2    "name": "api-rest-nodejs-mp6",
3    "version": "1.0.0",
4    "description": "Projecte Angular m6-uf3",
5    "main": "index.js",
6    "scripts": {
7      "test": "echo \"Error: no test specified\" && exit 1"
8    },
9    "author": "AnaValero",
10   "license": "MIT"
11  ]
```

2.Instalar Express

```
C:\Users\Ana\nodejs-mp6>npm install express --save
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN api-rest-nodejs-mp6@1.0.0 No repository field.

+ express@4.17.1
added 50 packages from 37 contributors and audited 126 packages in 8.062s
found 0 vulnerabilities
```

```
nodejs-mp6 > {} package.json > ...
1  {
2    "name": "api-rest-nodejs-mp6",
3    "version": "1.0.0",
4    "description": "Projecte Angular m6-uf3",
5    "main": "index.js",
6    "scripts": {
7      "test": "echo \"Error: no test specified\" && exit 1"
8    },
9    "author": "AnaValero",
10   "license": "MIT",
11   "dependencies": {
12     "express": "^4.17.1"
13   }
14 }
```

3.Body parser:

```
C:\Users\Ana\nodejs-mp6>npm install body-parser --save
npm WARN api-rest-nodejs-mp6@1.0.0 No repository field.

+ body-parser@1.19.0
updated 1 package and audited 158 packages in 1.614s
found 0 vulnerabilities
```

```
nodejs-mp6 > {} package.json > ...
1  {
2    "name": "api-rest-nodejs-mp6",
3    "version": "1.0.0",
4    "description": "Projecte Angular m6-uf3",
5    "main": "index.js",
6    "scripts": {
7      "test": "echo \"Error: no test specified\" && exit 1"
8    },
9    "author": "AnaValero",
10   "license": "MIT",
11   "dependencies": {
12     "body-parser": "^1.19.0",
13     "express": "^4.17.1"
14   }
15 }
```

4.Connect-mulltiParty:

```
C:\Users\Ana\nodejs-mp6>npm install connect-multiparty --save
npm WARN api-rest-nodejs-mp6@1.0.0 No repository field.

+ connect-multiparty@2.2.0
added 7 packages from 5 contributors and audited 184 packages in 2.105s
found 0 vulnerabilities
```

```
nodejs-mp6 > {} package.json > ...
1  {
2    "name": "api-rest-nodejs-mp6",
3    "version": "1.0.0",
4    "description": "Projecte Angular m6-uf3",
5    "main": "index.js",
6    "scripts": {
7      "test": "echo \"Error: no test specified\" && exit 1"
8    },
9    "author": "AnaValero",
10   "license": "MIT",
11   "dependencies": {
12     "body-parser": "^1.19.0",
13     "connect-multiparty": "^2.2.0",
14     "express": "^4.17.1"
15   }
}
```

5.Mongoose:

```
C:\Users\Ana\nodejs-mp6>npm install mongoose --save
npm WARN api-rest-nodejs-mp6@1.0.0 No repository field.

+ mongoose@5.8.10
added 19 packages from 13 contributors and audited 210 packages in 4.017s

1 package is looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

```
nodejs-mp6 > {} package.json > ...
1  {
2    "name": "api-rest-nodejs-mp6",
3    "version": "1.0.0",
4    "description": "Projecte Angular m6-uf3",
5    "main": "index.js",
6    "scripts": {
7      "test": "echo \"Error: no test specified\" && exit 1"
8    },
9    "author": "AnaValero",
10   "license": "MIT",
11   "dependencies": {
12     "body-parser": "^1.19.0",
13     "connect-multiparty": "^2.2.0",
14     "express": "^4.17.1",
15     "mongoose": "^5.8.10"
16   }
}
```

6. Nodemon

Per que quedin els canvis reflectits en el servidor i no haver-ho de fer manualment.

```
> nodemon@2.0.2 postinstall C:\Users\Ana\nodejs-mp6\node_modules\nodemon
> node bin/postinstall || exit 0

Love nodemon? You can now support the project via the open collective:
  > https://opencollective.com/nodemon/donate

npm WARN api-rest-nodejs-mp6@1.0.0 No repository field.
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@2.1.2 (node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.1.2: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})

+ nodemon@2.0.2
added 101 packages from 48 contributors and audited 351 packages in 18.56s

2 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

7. Script start

Creem l'arxiu index.js i modifiquem el arxiu package.json per que cridi a index.js

```
nodejs-mp6 > {} package.json > {} scripts > start
1 {
2   "name": "api-rest-nodejs-mp6",
3   "version": "1.0.0",
4   "description": "Projecte Angular m6-uf3",
5   "main": "index.js",
6   "scripts": {
7     "test": "echo \"Error: no test specified\" && exit 1",
8     "start": "nodemon index.js"
9   },
10  "author": "AnaValero",
11  "license": "MIT",
12  "dependencies": {
13    "body-parser": "^1.19.0",
14    "connect-multiparty": "^2.2.0",
15    "express": "^4.17.1",
16    "mongoose": "^5.8.10"
17  },
18  "devDependencies": {
19    "nodemon": "^2.0.2"
20  }
21 }
```

8. Crear la BD a MongoDB

Creem una BD en MongoDB anomenada *portafolio*

```
> use portfolio
switched to db portfolio
```

Creem una col·lecció anomenada *projects*

```
> db.createCollection('projects')
{ "ok" : 1 }
```

```
> show collections
projects
```

Afageix dades

```
> db.projects.insert({name:'Blog',description:'Blog Personal',category:'Informatica',
,langs: ['PHP','MySQL','css'],year:2019})
WriteResult({ "nInserted" : 1 })
```

```
> db.projects.find()
{ "_id" : ObjectId("5e344011403e8a25cb69c36a"), "name" : "Blog", "description" : "Blog Personal", "category" : "Informatica", "langs" : [ "PHP", "MySQL", "css" ], "year" : 2019 }
```

9.Connecta la BD desde NodeJS

```
nodejs-mp6 > JS index.js > ...
1  'use strict'
2
3  var mongoose =require('mongoose');
4
5  mongoose.Promise = global.Promise;
6  mongoose.connect('mongodb://localhost:27017/portfolio')
7    .then(()=> {
8      console.log("Connexio a la BD establerta");
9    })
10   .catch (err=> console.log(err));
```

```
C:\Users\Ana\nodejs-mp6>npm start
```

```
> api-rest-nodejs-mp6@1.0.0 start C:\Users\Ana\nodejs-mp6
> nodemon index.js
```

```
[nodemon] 2.0.2
[nodemon] to restart at any time, enter `rs`
[nodemon] watching dir(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node index.js`
(node:4584) DeprecationWarning: current URL string parser is deprecated, and will be removed in a future version. To use the new parser, pass option { useNewUrlParser: true } to MongoClient.connect.
(node:4584) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated, and will be removed in a future version. To use the new Server Discovery and Monitoring engine, pass option { useUnifiedTopology: true } to the MongoClient constructor.
Connexio a la BD establerta
```

10. Crear servidor nodejs

Fem un arxiu app.js

```
nodejs-mp6 > JS app.js > [⌘] app
1  'use strict'
2
3  var express = require('express');
4  var bodyParser = require('body-parser');
5
6  var app = express();
7
8  app.use(bodyParser.urlencoded({extended:false}));
9  app.use(bodyParser.json());
10
11 module.exports = app;
```

i modifiquem index.js

```
nodejs-mp6 > JS index.js > [⌘] then() callback
1  'use strict'
2
3  var mongoose = require('mongoose');
4  var app = require('./app');
5  var port = 3700;
6
7  mongoose.Promise = global.Promise;
8  mongoose.connect('mongodb://localhost:27017/portfolio')
9    .then(()=> {
10      console.log("Connexio a la BD establerta");
11
12      //creacio del servidor
13      app.listen(port, ()=>{
14        console.log('Servidor funcionant correctament');
15      });
16    })
17    .catch (err=> console.log(err));
```

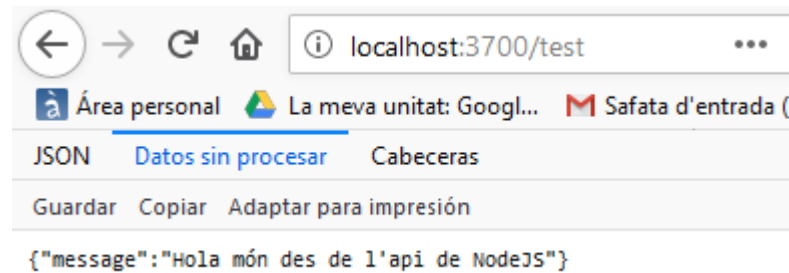
```
Connexio a la BD establerta
Servidor funcionant correctament
```

11. Fer una ruta de test

Modificar l'arxiu app.js

```
app.get('/test',(req,res)=>{  
  res.status(200).send({  
    message: "Hola món des de l'api de NodeJS"  
  });  
});
```

Veiem el resultat a la ruta

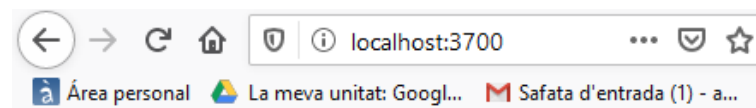


12. Ruta home

Escriurem en l'arxiu app.js

```
app.get('/',(req,res)=>{  
  res.status(200).send("<h1>Projecte M6 Angular NodeJS </h1>");  
});
```

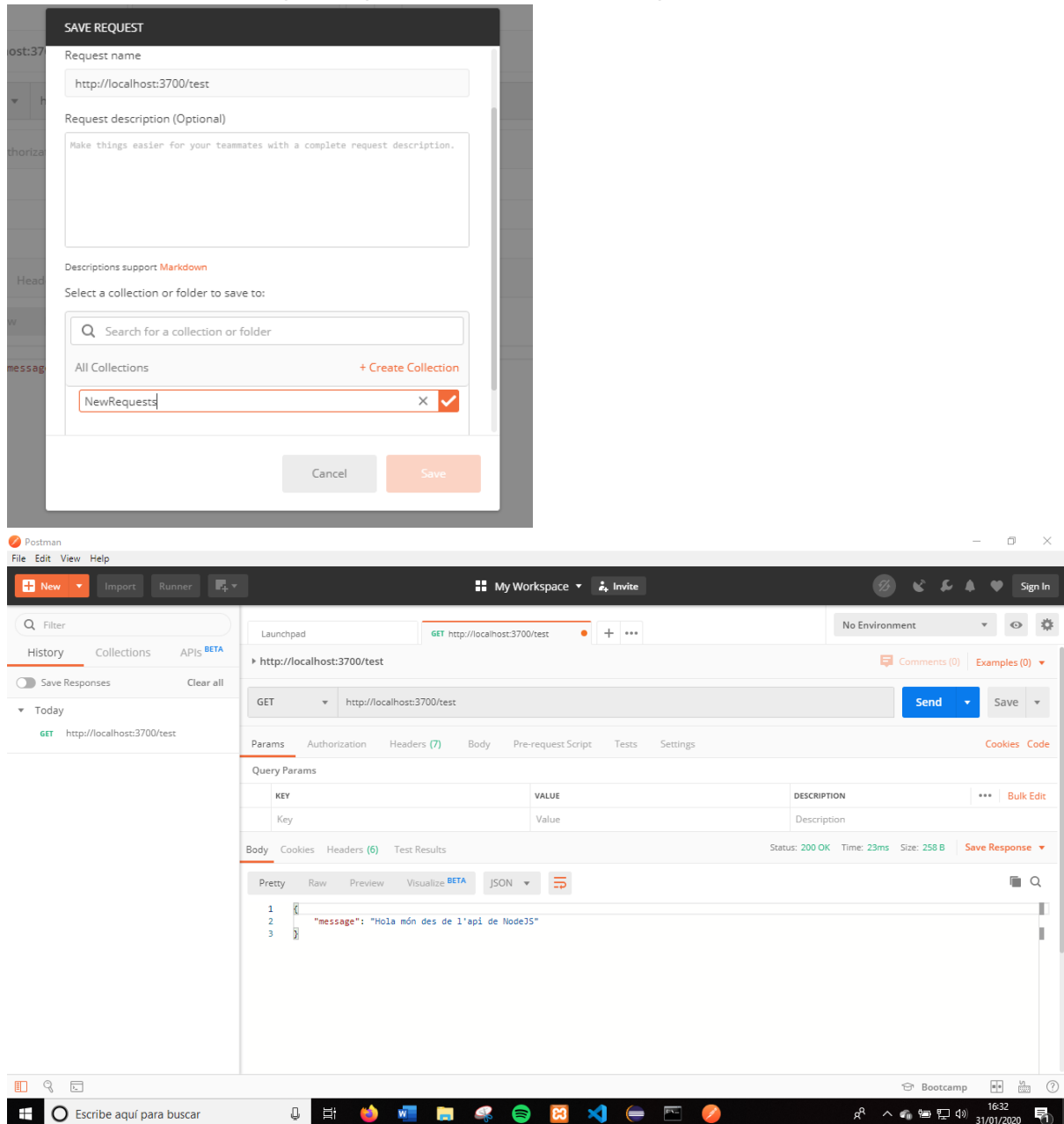
Veiem el resultat



Projecte M6 Angular NodeJS

12.Com utilitzar un client RESTful

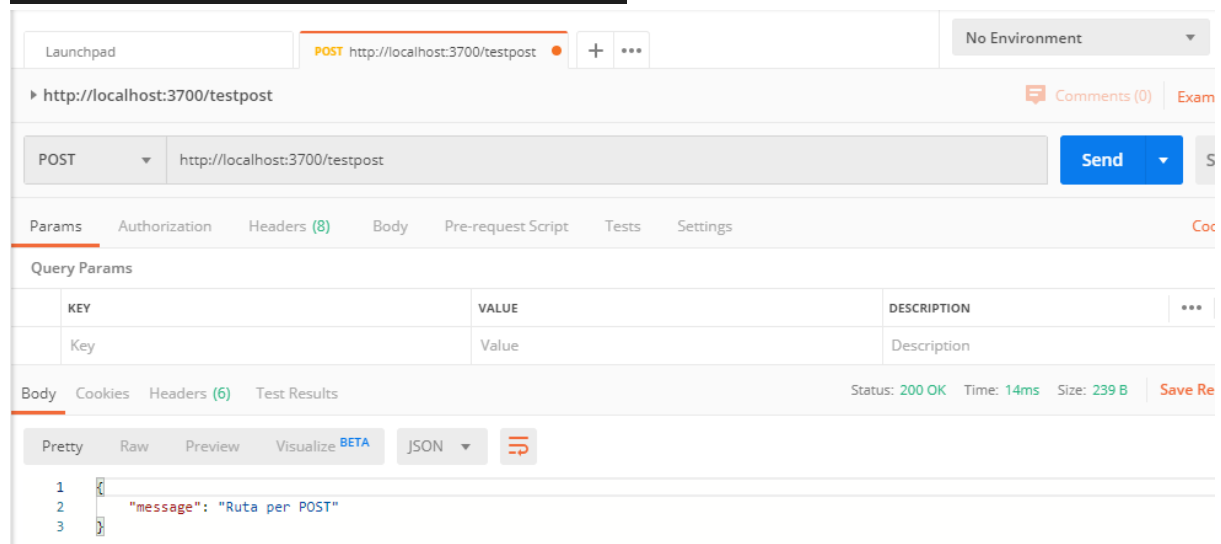
Instal·lar Postman i comprova que funciona creant un request:



13. Proves amb Postman

Creem una nova ruta per post i comprovem que funciona:

```
app.post('/testpost',(req,res)=>{
  res.status(200).send({
    message: "Ruta per POST"
  });
});
```



Launchpad POST http://localhost:3700/testpost No Environment

http://localhost:3700/testpost Comments (0) Exam

POST http://localhost:3700/testpost Send

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

Query Params

KEY	VALUE	DESCRIPTION	...
Key	Value	Description	

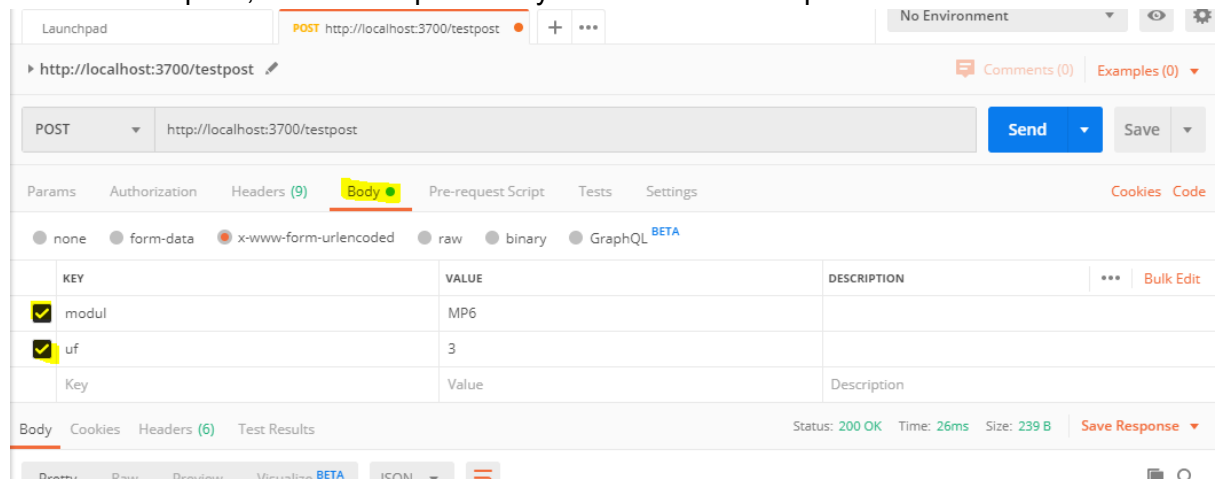
Body Cookies Headers (6) Test Results Status: 200 OK Time: 14ms Size: 239 B Save Re

Pretty Raw Preview Visualize BETA JSON

```
1 {
2   "message": "Ruta per POST"
3 }
```

14. Passem per post paràmetres.

Dins de la request, anirem a l'opció "body" allà escriurem els paràmetres.



Launchpad POST http://localhost:3700/testpost No Environment

http://localhost:3700/testpost Comments (0) Examples (0)

POST http://localhost:3700/testpost Send Save

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies Code

none form-data x-www-form-urlencoded raw binary GraphQL BETA

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/> modul	MP6			
<input checked="" type="checkbox"/> uf	3			
Key	Value	Description		

Body Cookies Headers (6) Test Results Status: 200 OK Time: 26ms Size: 239 B Save Response

Pretty Raw Preview Visualize BETA JSON

Farem que imprimieixi els valor per consola:

```
app.post('/testpost', (req, res) => {
  console.log("Mòdul:", req.param('modul'), "UF:", req.param('uf'));
  console.log("Mòdul:", req.body.modul, "UF:", req.body.uf);
  res.status(200).send({
    message: "Ruta per POST"
  });
});
```

Enviarem la request desde Postman i veurem que ens surten les dades introduïdes en la consola.

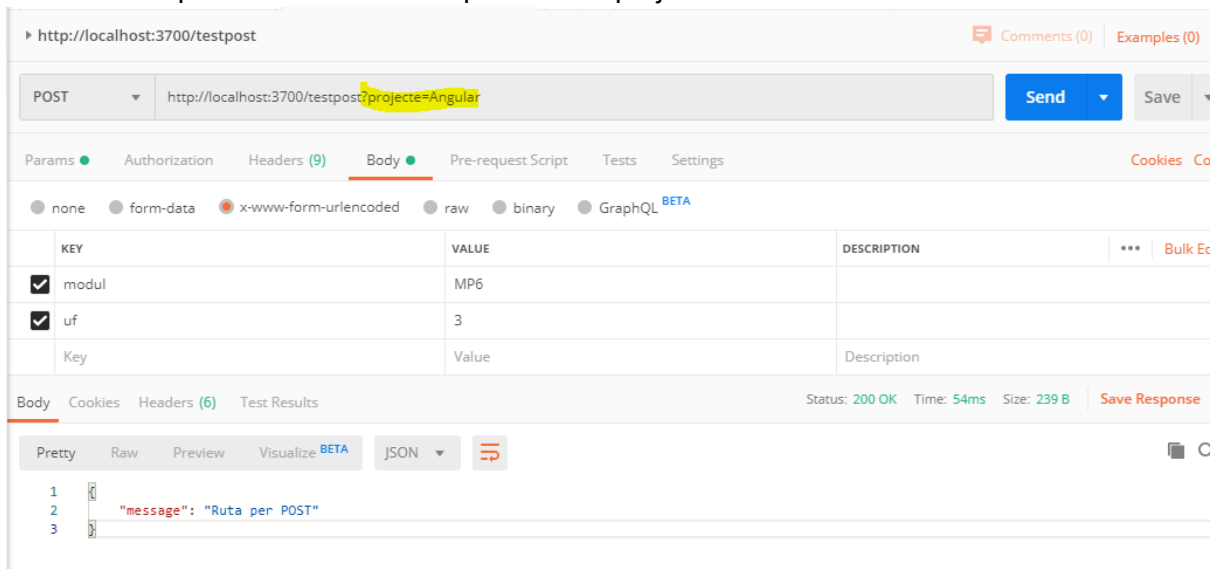
```
express deprecated req.param(name): Use req.params, req.body, or req.query instead app.js:30:30
express deprecated req.param(name): Use req.params, req.body, or req.query instead app.js:30:55
Mòdul: MP6 UF: 3
Mòdul: MP6 UF: 3
```

15. Ara amb query

Recollim les dades del parametre i l'imprimim per consola

```
app.post('/testpost', (req, res) => {
  // console.log("Mòdul:", req.param('modul'), "UF:", req.param('uf'));
  console.log("Mòdul:", req.body.modul, "UF:", req.body.uf);
  console.log("projecte:", req.query.projecte);
  res.status(200).send({
    message: "Ruta per POST"
  });
});
```

Enviem la request amb el valor del parametre "projecte":



POST http://localhost:3700/testpost?projecte=Angular

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies Co

none form-data x-www-form-urlencoded raw binary GraphQL BETA

KEY	VALUE	DESCRIPTION
modul	MP6	
uf	3	
Key	Value	Description

Body Cookies Headers (6) Test Results Status: 200 OK Time: 54ms Size: 239 B Save Response

Pretty Raw Preview Visualize BETA JSON

```
1 {
2   "message": "Ruta per POST"
3 }
```

Veiem el resultat en consola

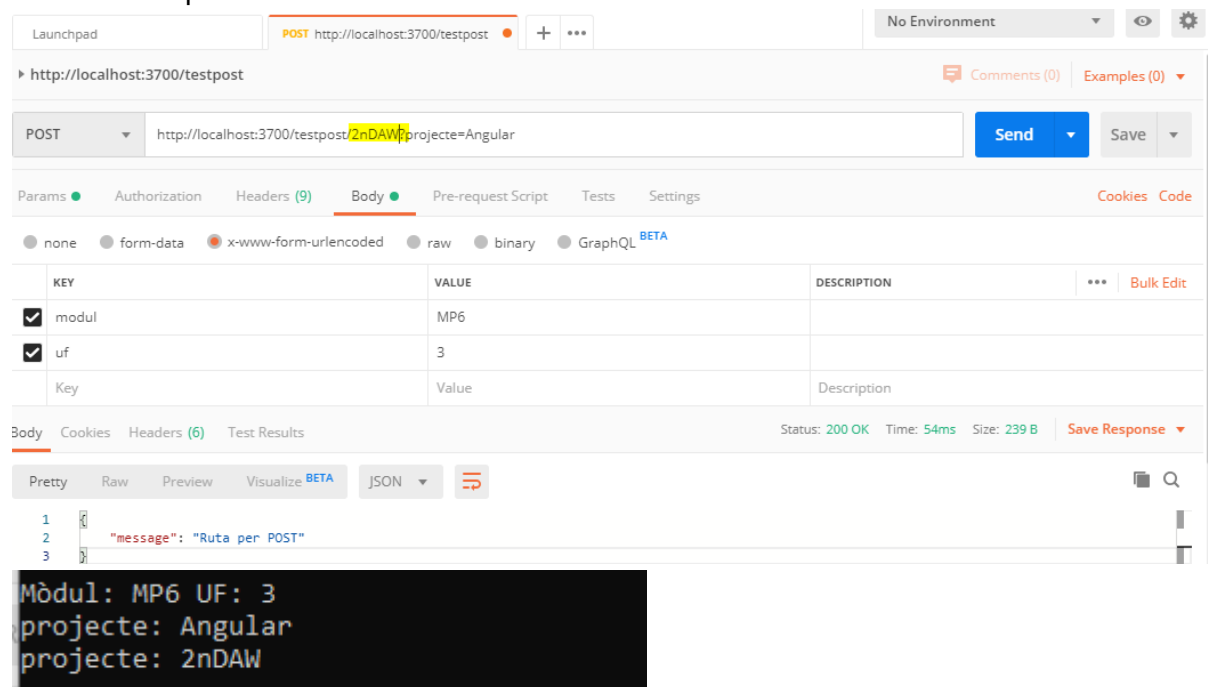
```
Mòdul: MP6 UF: 3
projecte: Angular
```

16. Params

Agafarem els valors que passem per paràmetre:

```
app.post(['/testpost/:curs'], (req, res) => {
  //console.log("Mòdul:", req.param('modul'), "UF:", req.param('uf'));
  console.log("Mòdul:", req.body.modul, "UF:", req.body.uf);
  console.log("projecte:", req.query.projecte);
  console.log("projecte:", req.params.curs);
  res.status(200).send({
    message: "Ruta per POST"
  });
});
```

Enviem la request i veiem el resultat:



Launchpad POST http://localhost:3700/testpost No Environment

http://localhost:3700/testpost Comments (0) Examples (0)

POST http://localhost:3700/testpost?modul=MP6&uf=3&projecte=Angular Send Save

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies Code

none form-data x-www-form-urlencoded raw binary GraphQL BETA

KEY	VALUE	DESCRIPTION
modul	MP6	
uf	3	
Key	Value	Description

Body Cookies Headers (6) Test Results Status: 200 OK Time: 54ms Size: 239 B Save Response

Pretty Raw Preview Visualize BETA JSON

```
1 {
2   "message": "Ruta per POST"
3 }
```

Mòdul: MP6 UF: 3
projecte: Angular
projecte: 2nDAW

17. Crear els models de la BBDD

Un model representa a un document de la BBDD

Creem el fitxer "project.js" dins de la carpeta "model".

```
nodejs-mp6 > model > JS project.js > ...
1  'use strict'
2
3  var mongoose = require('mongoose');
4  var Schema = mongoose.Schema;
5
6  var ProjectSchema = mongoose.Schema({
7    name: String,
8    description: String,
9    category: String,
10   year: Number,
11   langs: [String]
12 });
13
14
15 module.exports = mongoose.model('Project', ProjectSchema);
16 // sempre que no existeixi es generarà una col·lecció anomenada projects
17
```

18. Model vista controlador

Estructura d'un controlador:

Creem el controlador dins de la carpeta "controllers".

```
nodejs-mp6 > controllers > JS project.js > ...
1  'use strict'
2
3  var controller = {
4    home: function() {
5
6    },
7    test: function(){
8
9    }
10 };
11
12 module.exports = controller;
```

19. Desenvolupament del controlador

Primer hem de comentar les rutes del "app.js"

```
nodejs-mp6 > JS app.js > ...
1  'use strict'
2
3  var express = require('express');
4  var bodyParser = require('body-parser');
5
6  var app = express();
7
8  //carregar rutes
9
10 //middlewares
11
12 //codifiquem a json tot el que ens arriba
13 app.use(bodyParser.urlencoded({extended:false}));
14 app.use(bodyParser.json());
15
16 //cors
17
18 //rutes
19
20 /*app.get('/test',(req,res)=>{
21   res.status(200).send({
22     message: "Hola món des de l'api de NodeJS"
23   });
24 });
25 */
26 app.get('/',(req,res)=>{
27   res.status(200).send("<h1>Projecte M6 Angular NodeJS </h1>");
28 });
29 app.post('/testpost/:curs',(req,res)=>{
30   //console.log("Mòdul:",req.param('modul'),'UF:',req.param('uf'));
31   console.log("Mòdul:",req.body.modul,"UF:",req.body.uf);
32   console.log("projecte:",req.query.projecte);
33   console.log("projecte:",req.params.curs);
34   res.status(200).send({
35     message: "Ruta per POST"
36   });
37 });*/
38
39 //exportar
40 module.exports =app;
```

Omplim les dades del controlador

```
nodejs-mp6 > controllers > JS project.js > [e] controller
1  'use strict'
2
3  var controller = {}
4    home: function(req,res) {
5
6      return res.status(200).send({
7        message: 'Pagina'
8      });
9
10   },
11   test: function(req,res){
12
13     return res.status(200).send({
14       message: 'Pagina de test'
15     });
16
17   },
18 };
19
20 module.exports = controller;
```

Creem una carpeta i un arxiu per les rutes:

```
nodejs-mp6 > routes > JS project.js > ...
1  'use strict'
2
3  var express = require('express');
4  var ProjectController = require('../controllers/project');
5
6  var router = express.Router();
7
8  router.get('/home', ProjectController.home);
9  router.post('/test', ProjectController.test);
10
11 module.exports = router;
```

Configurem el arxiu "app.js" per tal de utilitzar les rutes creades en el fitxer anteriorment creat:

```
'use strict'

var express = require('express');
var bodyParser = require('body-parser');

var app = express();

/*-----carregar routes-----*/

var project_routes = require('./routes/project');

/*-----middlewares-----*/

/*-----codifiquem a json tot el que ens arriba-----*/
app.use(bodyParser.urlencoded({extended:false}));
app.use(bodyParser.json());

/*-----cors-----*/

/*-----routes-----*/

app.use('/api',project_routes);
/*app.get('/test',(req,res)=>{
```

Comprovarem que funciona a Postman:

Untitled Request Comments (0)

POST http://localhost:3700/api/test Send Save

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings Cookies Code

☐ none ☐ form-data ☒ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL BETA

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/> modul	MP6			
<input checked="" type="checkbox"/> uf	3			
Key	Value	Description		

Body Cookies Headers (6) Test Results Status: 200 OK Time: 46ms Size: 240 B Save Response

Pretty Raw Preview Visualize BETA JSON Copy

```
1 {
2   "message": "Pagina de test"
3 }
```

20.Mètode saveProject

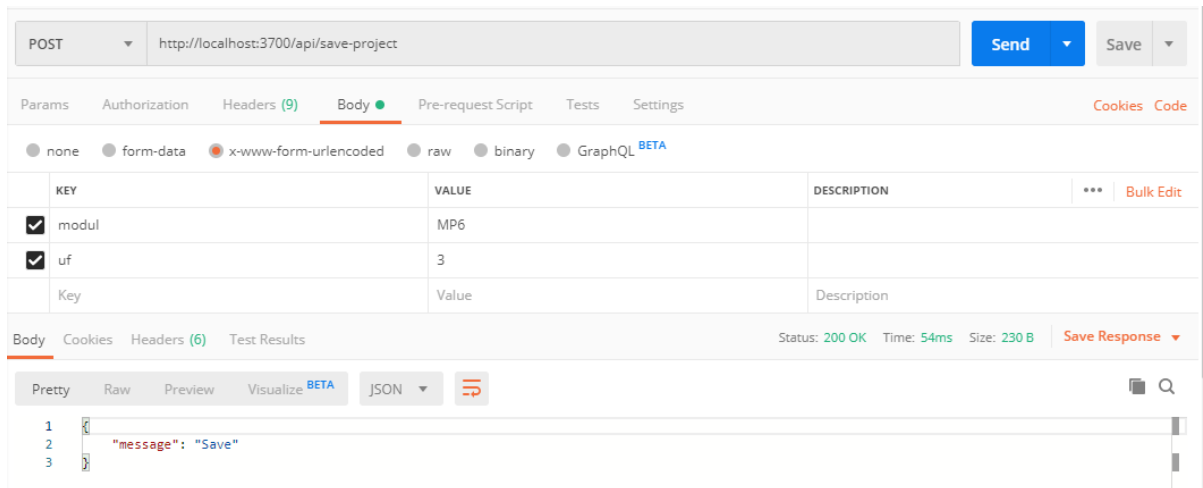
En el controller cal importar el model i crear el mètode saveProject:

```
nodejs-mp6 > controllers > JS project.js > controller > saveProject
1  'use strict'
2
3  var Project = require('../model/project');
4  var controller = {
5    home: function(req,res) {
6
7      return res.status(200).send({
8        message: 'Pagina'
9      });
10
11    },
12    test: function(req,res){
13
14      return res.status(200).send({
15        message: 'Pagina de test'
16      });
17
18    },
19    saveProject: function(req,res){
20
21      var project = new Project();
22
23      return res.status(200).send({
24        message: 'Save'
25      });
26    }
27  };
28
29  module.exports = controller;
```

Afegim la ruta dins del fitxer on tenim les rutes:

```
nodejs-mp6 > routes > JS project.js > ...
1  'use strict'
2
3  var express = require('express');
4  var ProjectController = require('../controllers/project');
5
6  var router = express.Router();
7
8  router.get('/home', ProjectController.home);
9  router.post('/test', ProjectController.test);
10 router.post('/save-project', ProjectController.saveProject);
11
12 module.exports = router;
```

Comprovem a Postman:



POST http://localhost:3700/api/save-project

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies Code

none form-data x-www-form-urlencoded raw binary GraphQL BETA

KEY	VALUE	DESCRIPTION
modul	MP6	
uf	3	
Key	Value	Description

Body Cookies Headers (6) Test Results Status: 200 OK Time: 54ms Size: 230 B Save Response

Pretty Raw Preview Visualize BETA JSON

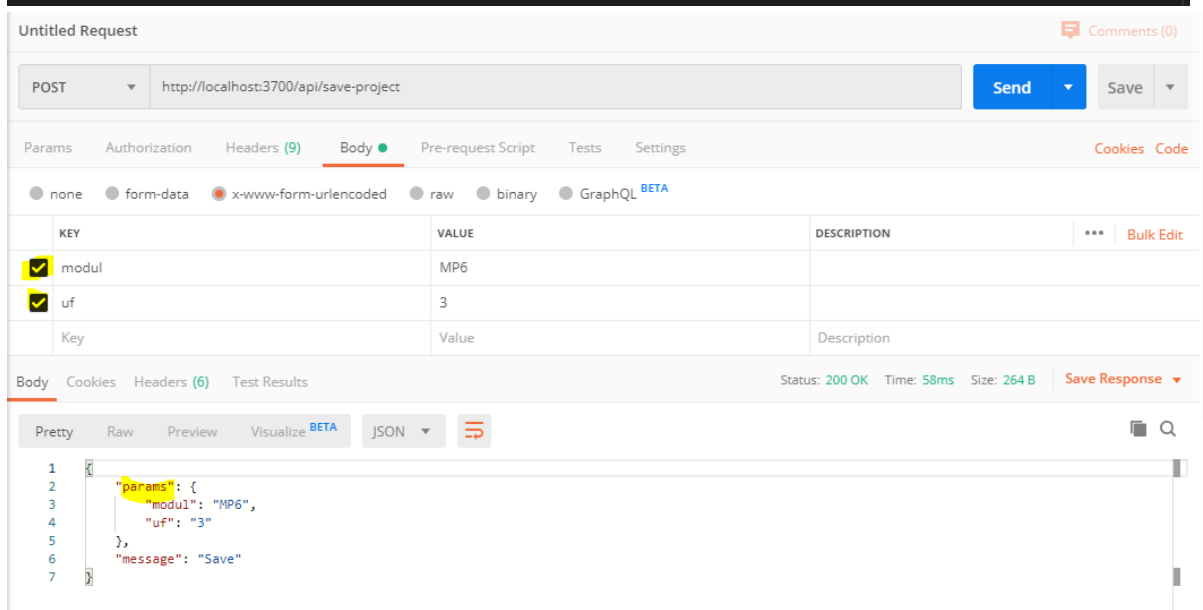
```
1 {
2   "message": "Save"
3 }
```

21. Pasem parametres al saveProject:

En el metode creat en el controlador recollirem les parametres que li pasem per Postman

```
saveProject: function(req,res){
    var project = new Project();
    var params = req.body;

    return res.status(200).send({
        params: params,
        message: 'Save'
    });
};
```



Untitled Request

POST http://localhost:3700/api/save-project

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies Code

none form-data x-www-form-urlencoded raw binary GraphQL BETA

KEY	VALUE	DESCRIPTION
modul	MP6	
uf	3	
Key	Value	Description

Body Cookies Headers (6) Test Results Status: 200 OK Time: 58ms Size: 264 B Save Response

Pretty Raw Preview Visualize BETA JSON

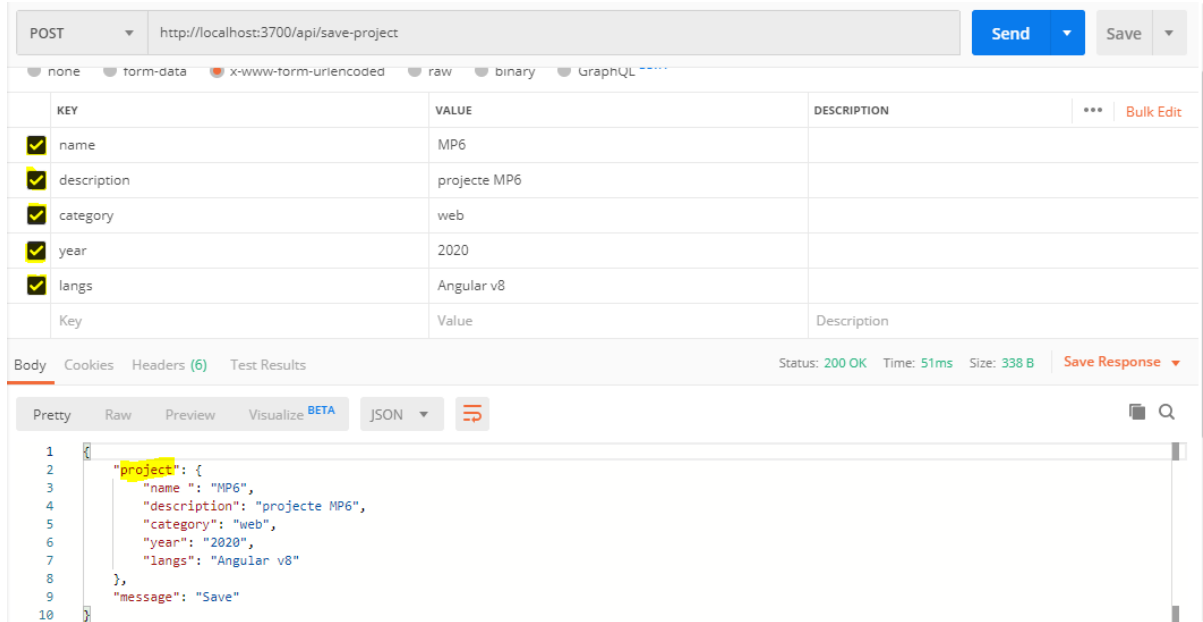
```
1 {
2   "params": {
3     "modul": "MP6",
4     "uf": "3"
5   },
6   "message": "Save"
7 }
```

22.Passem paràmetres segons el model

Posem les mateixes variables que tenim al model, les omplim al body i les recollim amb el nom "project".

```
saveProject: function(req,res){
    var project = new Project();
    var params = req.body;

    return res.status(200).send({
        project: params,
        message: 'Save'
    });
};
```



POST http://localhost:3700/api/save-project

none form-data x-www-form-urlencoded raw binary GraphQL

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> name	MP6	
<input checked="" type="checkbox"/> description	projecte MP6	
<input checked="" type="checkbox"/> category	web	
<input checked="" type="checkbox"/> year	2020	
<input checked="" type="checkbox"/> langs	Angular v8	
Key	Value	Description

Body Cookies Headers (6) Test Results Status: 200 OK Time: 51ms Size: 338 B Save Response

Pretty Raw Preview Visualize BETA JSON

```
1 {
2   "project": {
3     "name": "MP6",
4     "description": "projecte MP6",
5     "category": "web",
6     "year": "2020",
7     "langs": "Angular v8"
8   },
9   "message": "Save"
10 }
```

```

saveProject: function(req,res){

    var project = new Project();

    var params = req.body;
    project.name = params.name;
    project.description = params.description;
    project.category = params.category;
    project.year = params.year;
    project.langs = params.langs;
    project.image = "null";

    project.save((err,projectStored)=>{
        if (err) return res.status(500).send({message: "Error al desar el document"});

        if (!projectStored) return res.status(404).send({message: 'Document no desat'});

        return res.status(200).send({project: projectStored});
    });
};

module.exports = controller;

```

Comprovem Postman i MongoDB:

Pretty Raw Preview Visualize BETA JSON

```

1  {
2    "project": {
3      "langs": [
4        "Angular v8"
5      ],
6      "_id": "5e3998610f62a532bcf4e57c",
7      "description": "projecte MP6",
8      "category": "web",
9      "year": 2020,
10     "__v": 0
11   }
12 }

```

```

{ "_id" : ObjectId("5e3998610f62a532bcf4e57c"), "langs" : [ "Angular v8" ], "description" : "projecte MP6", "category" : "web", "year" : 2020, "__v" : 0 }

```

24. Mètode get Project

Creem el metode "getProject" dins del controllador

```
getProject: function (req,res){
  var projectId = req.params.id;
  console.log(projectId);

  if (projectId == null ) return res.status(500).send({message: "No has especificat cap projecte"}});
  else{

    Project.findById(projectId, (err,project) =>{

      if(err) {
        console.log(err);
        return res.status(500).send({message:"Error al retornar les dades"}});
      }

      if(!project) return res.status(404).send({message: "El projecte no existeix"}});

      return res.status(200).send({
        project
      });
    });
  }
};
```

Esmentem aquest metode dins de les rutes.

```
nodejs-mp6 > routes > JS project.js > ...
1  'use strict'
2
3  var express = require('express');
4  var ProjectController = require('../controllers/project');
5
6  var router = express.Router();
7
8  router.get('/home', ProjectController.home);
9  router.get('/project/:id?', ProjectController.getProject);
10
11 router.post('/test', ProjectController.test);
12 router.post('/save-project',ProjectController.saveProject);
13
14 module.exports = router;
```

Provem el mètode:

GET
http://localhost:3700/api/project/
Send

Params
Authorization
Headers (9)
Body
Pre-request Script
Tests
Settings

none
form-data
x-www-form-urlencoded
raw
binary
GraphQL BETA

This request does not have a body

Body
Cookies
Headers (6)
Test Results
Status: 500 Internal Server Error
Time: 74ms
Size: 276

Pretty
Raw
Preview
Visualize BETA
JSON

1
2
3
{
 "message": "No has especificat cap projecte"
}

GET
http://localhost:3700/api/project/11
Send

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

none
form-data
x-www-form-urlencoded
raw
binary
GraphQL BETA

This request does not have a body

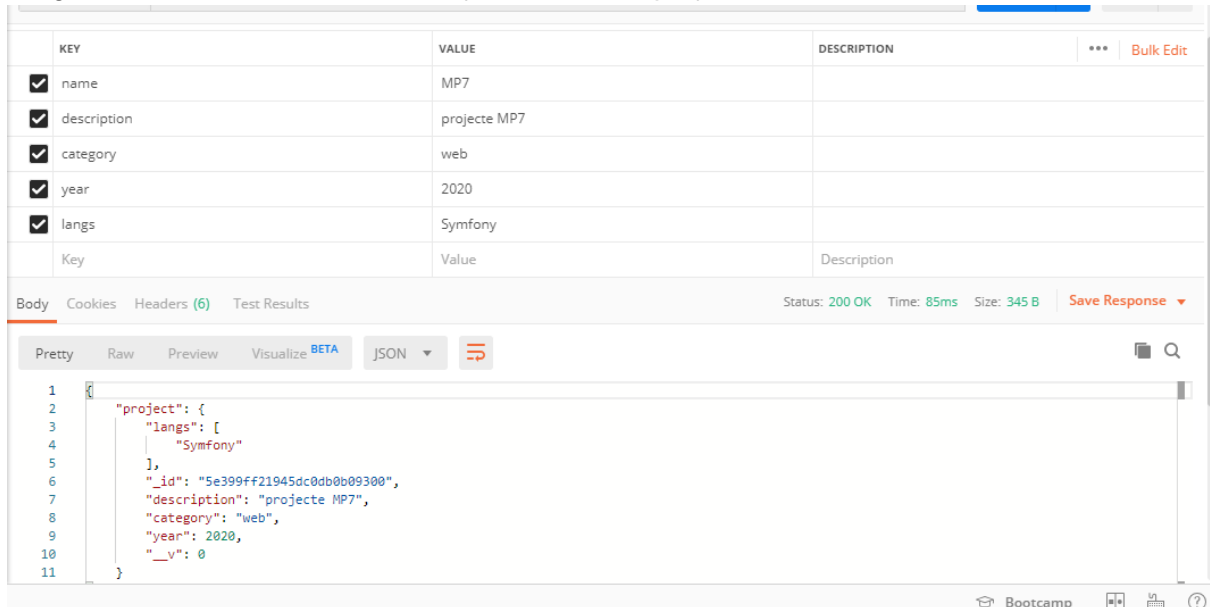
Body
Cookies
Headers (6)
Test Results
Status: 500 Internal Server Error
Time: 33ms
Size: 272

Pretty
Raw
Preview
Visualize BETA
JSON

1
2
3
{
 "message": "Error al retornar les dades"
}

25. Mètodes getProjects

Implementar el mètode getProjects que retorna tots els projectes de la BBDD.
Afegir a través del mètode saveProject un cuants projectes:

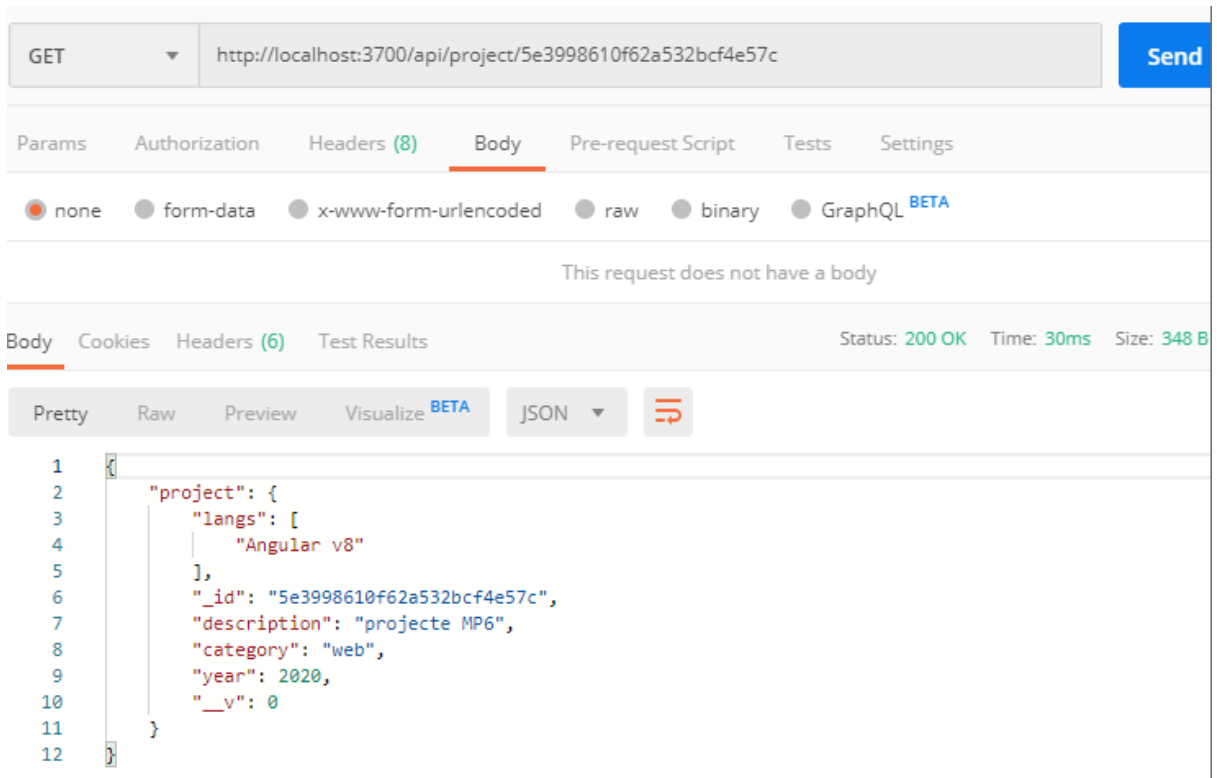


KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> name	MP7	
<input checked="" type="checkbox"/> description	projecte MP7	
<input checked="" type="checkbox"/> category	web	
<input checked="" type="checkbox"/> year	2020	
<input checked="" type="checkbox"/> langs	Symfony	
Key	Value	Description

```

1  {
2    "project": {
3      "langs": [
4        "Symfony"
5      ],
6      "_id": "5e399ff21945dc0db0b09300",
7      "description": "projecte MP7",
8      "category": "web",
9      "year": 2020,
10     "_v": 0
11   }

```



GET http://localhost:3700/api/project/5e3998610f62a532bcf4e57c

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

none form-data x-www-form-urlencoded raw binary GraphQL BETA

This request does not have a body

Body Cookies Headers (6) Test Results Status: 200 OK Time: 30ms Size: 348 B

```

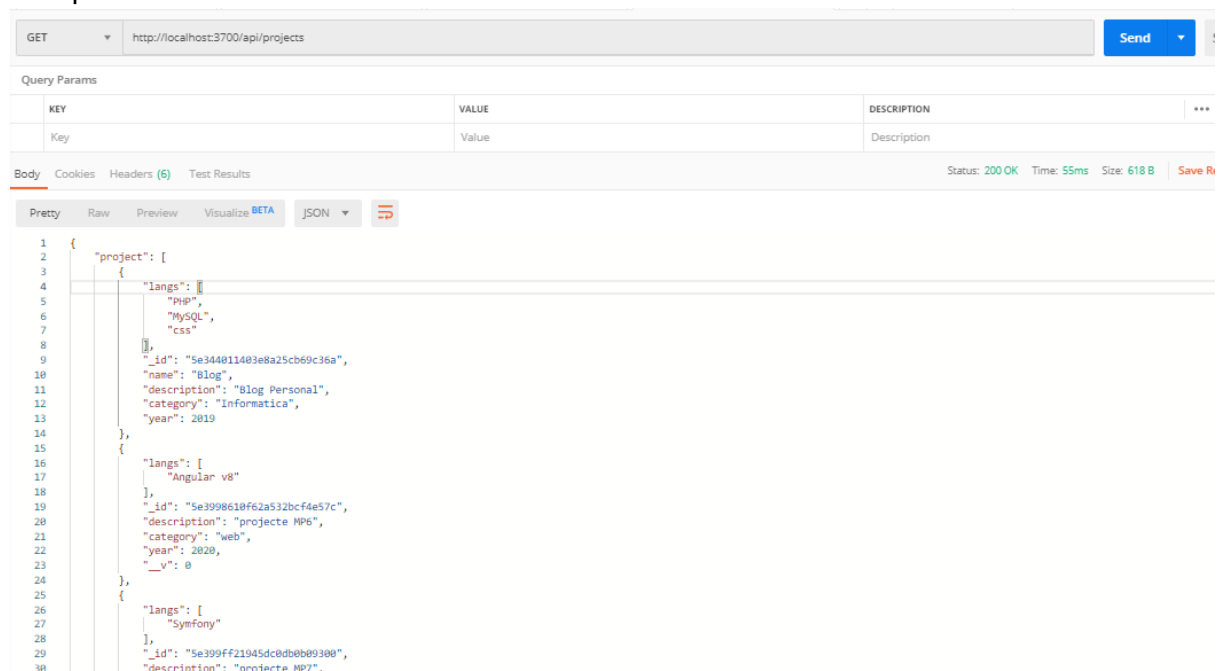
1  {
2    "project": {
3      "langs": [
4        "Angular v8"
5      ],
6      "_id": "5e3998610f62a532bcf4e57c",
7      "description": "projecte MP6",
8      "category": "web",
9      "year": 2020,
10     "_v": 0
11   }
12 }

```

Creem la ruta GET en el fitxer de rutes:

```
nodejs-mp6 > routes > JS project.js > ...
1 'use strict'
2
3 var express = require('express');
4 var ProjectController = require('../controllers/project');
5
6 var router = express.Router();
7
8 router.get('/home', ProjectController.home);
9 router.get('/project/:id?', ProjectController.getProject);
10 router.get(['/projects', ProjectController.getProjects]);
11
12 router.post('/test', ProjectController.test);
13 router.post('/save-project', ProjectController.saveProject);
14
15 module.exports = router;
16
17
```

Comprovem al Postman:



GET http://localhost:3700/api/projects

Send

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (6) Test Results Status: 200 OK Time: 55ms Size: 618 B Save R

Pretty Raw Preview Visualize BETA JSON

```

1 {
2   "project": [
3     {
4       "langs": [
5         "php",
6         "MySQL",
7         "css"
8       ],
9       "_id": "5e344011403e8a25cb69c36a",
10      "name": "Blog",
11      "description": "Blog Personal",
12      "category": "Informatica",
13      "year": 2019
14    },
15    {
16      "langs": [
17        "Angular v8"
18      ],
19      "_id": "5e3998618f62a532bcf4e57c",
20      "description": "projecte MP6",
21      "category": "web",
22      "year": 2020,
23      "__v": 0
24    },
25    {
26      "langs": [
27        "Symfony"
28      ],
29      "_id": "5e399ff21945dc0db0b09300",
30      "description": "projecte MP7",

```


Consulta 1:

```
getProjects: function(req,res){
    Project.find({}).sort('year').exec((err,project) =>{

        if(err) {
            console.log(err);
            return res.status(404).send({message:"Error al retornar les dades"});
        }

        if(!project) return res.status(404).send({message: "El projecte no existeix"});

        return res.status(200).send({
            project
        });
    });
};
```

GET http://localhost:3700/api/projects Send

Pretty Raw Preview Visualize BETA JSON

```
1 {
2   "project": [
3     {
4       "langs": [
5         "PHP",
6         "MySQL",
7         "css"
8       ],
9       "_id": "5e344011403e8a25cb69c36a",
10      "name": "Blog",
11      "description": "Blog Personal1",
12      "category": "Informatica",
13      "year": 2019
14    },
15    {
16      "langs": [
17        "Angular v8"
18      ],
19      "_id": "5e3998610f62a532bcf4e57c",
20      "description": "projecte MP6",
21      "category": "web",
22      "year": 2020,
23      "_v": 0
24    },
25    {
26      "langs": [
27        "Symfony"
28      ],
29    }
30  ],
31}
```

Consulta2:

```
getProjects: function(req,res){
    Project.find({year:2019}).exec((err,project) =>{

        if(err) {
            console.log(err);
            return res.status(404).send({message:"Error al retornar les dades"});
        }

        if(!project) return res.status(404).send({message: "El projecte no existeix"});

        return res.status(200).send({
            project
        });
    });
};
```

GET http://localhost:3700/api/projects Send

Pretty Raw Preview Visualize BETA JSON

```

1 {
2   "project": [
3     {
4       "langs": [
5         "PHP",
6         "MySQL",
7         "css"
8       ],
9       "_id": "5e344011403e8a25cb69c36a",
10      "name": "Blog",
11      "description": "Blog Personal",
12      "category": "Informatica",
13      "year": 2019
14    }
15  ]
16 }

```

26.Metode updateProject

```
//per actualitzar utilitzarem put
router.put('/project/:id', ProjectController.updateProject);
```

```

updateProject: function (req,res){
  var projectId = req.params.id;

  var update = req.body;

  Project.findByIdAndUpdate(projectId, update, (err,projectUpdated)=>{
    if (err) {
      return res.status(500).send({message:"Error actualizan dades"});
    }
    if (!projectUpdated) return res.status(404).send({message:"No existeix el projecte"});
    return res.status(200).send({
      project: projectUpdated
    });
  });
}

```

PUT http://localhost:3700/api/project/5e3998610f62a532bcf4e57c Send Save

none form-data x-www-form-urlencoded raw binary GraphQL

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> name	MP8	
<input checked="" type="checkbox"/> description	Projecte MP8	
<input checked="" type="checkbox"/> category	Interficie	
<input checked="" type="checkbox"/> year	2020	
<input checked="" type="checkbox"/> langs	Node	
Key	Value	Description

Body Cookies Headers (6) Test Results Status: 200 OK Time: 25ms Size: 362 B Save Respon

Pretty Raw Preview Visualize BETA JSON

```

1 {
2   "project": {
3     "langs": [
4       "Node"
5     ],
6     "_id": "5e3998610f62a532bcf4e57c",
7     "description": "Projecte MP8",
8     "category": "Interficie",
9     "year": 2020,
10    "__v": 0,
11    "name": "MP8"
12  }
13 }

```

27. Mètode deleteProject

```
deleteProject: function (req,res){
  //el valor de la id arriba per la URL
  var projectId = req.params.id;
  Project.findByIdAndDelete(projectId, (err, projectRemoved) =>{
    if (err) {
      return res.status(500).send({message: "Error: no s'ha pogut borrar el projecte"});
    }
    if (!projectRemoved) return res.status(404).send({message: "No existeix el projecte a borrar"});
    return res.status(200).send({
      project: projectRemoved
    })
  });
}
```

```
//per eliminar utilitzarem delete
router.delete('/project/:id', ProjectController.deleteProject);
```

DELETE http://localhost:3700/api/project/5e3998610f62a532bcf4e57c

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> name	MP8	
<input checked="" type="checkbox"/> description	Projecte MP8	
<input checked="" type="checkbox"/> category	Interficie	
<input checked="" type="checkbox"/> year	2020	
<input checked="" type="checkbox"/> langs	Node	
Key	Value	Description

Body Cookies Headers (6) Test Results Status: 200 OK Time: 269ms Size: 362 B Save Response

```
1 {
2   "project": {
3     "langs": [
4       "Node"
5     ],
6     "_id": "5e3998610f62a532bcf4e57c",
7     "description": "Projecte MP8",
8     "category": "Interficie",
9     "year": 2020,
10    "__v": 0,
11    "name": "MP8"
12  }
13 }
```

```
> db.projects.find()
{ "_id" : ObjectId("5e344011403e8a25cb69c36a"), "name" : "Blog", "description" : "Blog Personal", "category" : "Informatica", "langs" : [ "PHP", "MySQL", "css" ], "year" : 2019 }
{ "_id" : ObjectId("5e399ff21945dc0db0b09300"), "langs" : [ "Symfony" ], "description" : "projecte MP7", "category" : "web", "year" : 2020, "__v" : 0 }
```

28. Pujar arxius al servidor

```
uploadImage: function(req,res){
    var projectId = req.params.id;
    var fileName = "Image no pujada";

    if (req.files){
        return res.status(200).send({
            files: req.files
        })
    }else{
        return res.status(200).send({
            message: fileName
        })
    }
}
```

```
nodejs-mp6 > routes > JS project.js > ...
1  'use strict'
2
3  var express = require('express');
4  var ProjectController = require('../controllers/project');
5
6  var router = express.Router();
7
8  //cal configurar un middleware (s'executarà abans de que s'executi la ruta)
9  var multipart = require('connect-multiparty');
10 //li hem de passar la carpeta alla on es desaran els arxius
11 var multipartMiddleware = multipart({UploadDir: './uploads'})
12
13 router.get('/home', ProjectController.home);
14 router.get('/project/:id?', ProjectController.getProject);
15 router.get('/projects', ProjectController.getProjects);
16
17 router.post('/test', ProjectController.test);
18 router.post('/save-project',ProjectController.saveProject);
19
20 //per actualitzar utilitzarem put
21 router.put('/project/:id', ProjectController.updateProject);
22
23 //per eliminar utilitzarem delete
24 router.delete('/project/:id', ProjectController.deleteProject);
25
26 //ruta per pujar una imatge, cal indicar el middleware que volem que s'executi
27 router.post('/upload-image/:id',multipartMiddleware,ProjectController.uploadImage);
28 module.exports = router;
29
```

POST http://localhost:3700/api/upload-image/5e3998610f62a532bcf4e57c

none form-data x-www-form-urlencoded raw binary GraphQL

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> image	File Capture001.png X	
Key	Text Value	Description
	File	

Body Cookies Headers (6) Test Result Status: 200 OK Time: 48ms Size: 5

Pretty Raw Preview Visualize BETA JSON

```

1 {
2   "files": {
3     "image": {
4       "fieldName": "image",
5       "originalFilename": "Capture001.png",
6       "path": "uploads\\m1l1QQdDjgoGik0vsikyV90d.png",
7       "headers": {
8         "content-disposition": "form-data; name=\"image\"; filename=\"Capture001.png\"",
9         "content-type": "image/png"
10      },
11       "size": 216781,
12       "name": "Capture001.png",
13       "type": "image/png"
14     }
15   }
16 }

```

< Usuarios > Ana > nodejs-mp6 > uploads

ml1lQQdDjgoGik0vsikyV90d

29. Aconseguir el nom de l'arxiu

```

uploadImage: function(req,res){
  var projectId = req.params.id;
  var fileName = "Image no pujada";

  if (req.files){

    var filePath = req.files.image.path;
    var fileSplit = filePath.split('\\');
    console.log(fileSplit);
    var fileName = fileSplit[1];

    return res.status(200).send({
      files: fileName
    })
  }else{
    return res.status(200).send({
      message: fileName
    })
  }
}

```

POST http://localhost:3700/api/upload-image/5e3998610f62a532bcf4e57c

Params Authorization Headers (9) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL BETA

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> image	Capture001.png X	
Key	Value	Description

Body Cookies Headers (6) Test Results Status: 200 OK Time: 26ms Size: 2

Pretty Raw Preview Visualize BETA JSON

```

1 {
2   "files": "hzA307SL1l8EAk8apjY8QcpK.png"
3 }

```

30.Actualitzar la BBDD:

Afegim el camp al model

```

nodejs-mp6 > model > JS project.js > [e] ProjectSchema > image
1 'use strict'
2
3 var mongoose = require('mongoose');
4 var Schema = mongoose.Schema;
5
6 var ProjectSchema = mongoose.Schema({
7   name: String,
8   description: String,
9   category: String,
10  year: Number,
11  langs: [String],
12  image: String
13 });
14
15
16 module.exports = mongoose.model('Project', ProjectSchema);
17 // sempre que no existeixi es generarà una col·lecció anomenada
18

```

```

uploadImage: function(req,res){
  var projectId = req.params.id;
  var fileName = "imatge no pujada";

  if (req.files){

    var filePath = req.files.image.path;
    var fileSplit = filePath.split('\\');
    //console.log(fileSplit);
    var fileName = fileSplit[1];
    console.log(projectId);

    Project.findByIdAndUpdate(projectId, {image:fileName}, {new:true}, (err, projectUpdated)=>{
      console.log(projectUpdated);
      if (err){
        return res.status(500).send({message:"Error actualitzant la imatge"});
      }
      if (!projectUpdated) return res.status(404).send({message:"No existeix el projecte"});
      return res.status(200).send({
        project: projectUpdated
      });
    });
  }else{
    return res.status(500).send({
      message: fileName
    })
  }
}

//ruta per pujar una imatge, cal indicar el middleware que volem que s'executi
router.post('/upload-image/:id',multipartMiddleware,ProjectController.uploadImage);

```

POST http://localhost:3700/api/upload-image/5e344011403e8a25cb69c36a

Body Cookies Headers (6) Test Results Status: 200 OK Time: 83ms Size: 4

Pretty Raw Preview Visualize BETA JSON

```

1 {
2   "project": {
3     "langs": [
4       "PHP",
5       "MySQL",
6       "css"
7     ],
8     "_id": "5e344011403e8a25cb69c36a",
9     "name": "Blog",
10    "description": "Blog Personal",
11    "category": "Informatica",
12    "year": 2019,
13    "image": "G3l6ahbyh2F7vnoxG4Vpmez4.png"
14  }
15 }

```

31. Millora comprovar extensió:

```
uploadImage: function(req,res){
  var projectId = req.params.id;
  var fileName = "Imatge no pujada";

  if (req.files){

    var filePath = req.files.image.path;
    var fileSplit = filePath.split('\\');
    //console.log(fileSplit);
    var fileName = fileSplit[1];
    var fileSplit2 = fileName.split('.');
    var fileName2 = fileSplit2[1];
    console.log(fileName2);
    //console.log(projectId);
    if (fileName2==="png" || fileName2==="jpg" || fileName2==="jpeg" || fileName2==="")
      return res.status(200).send({message: "Extensió correcta"});
    }
    else{
      fs.unlink(filePath, (err) =>{
        return res.status(200).send({message: "L'extensió no es correcta"});
      })
    }
  }
  Project.findByIdAndUpdate(projectId, {image:fileName}, {new:true}, (err, project)
```

► http://localhost:3700/api/project/5e3998610f62a532bcf4e57c

POST http://localhost:3700/api/upload-image/5e344011403e8a25cb69c36a

Params Authorization Headers (9) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL BETA

	KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/>	image	Capture001.png X	
	Key	Value	Description

Body Cookies Headers (6) Test Results Status: 200 OK Time: 55ms Size: 2

Pretty Raw Preview Visualize BETA JSON

```
1 {
2   "message": "Extensió correcta"
3 }
```


32. Configurer l'accès CORS

```
/*-----cors-----*/  
app.use((req,res,next)=>{  
  res.header('Access-Control-Allow-Origin', '*');  
  res.header('Access-Control-Allow-Headers', 'Authorization, X-API-KEY, Origin, X-Requested-  
  res.header('Access-Control-Allow-Methods', 'GET, POST, OPTIONS, PUT, DELETE');  
  res.header('Allow', 'GET, POST, OPTIONS, PUT, DELETE');  
  next();  
});  
/*-----routes-----*/
```