

Como crear un servicio en el Api Rest Dory

Objetivo: Registrar un servicio en el Api Rest Dory.

Procedimiento:

XXX = nombre del enrutador

Index.js

- Crear y utilizar ruta.
`Const XXXRouter = requiere ('./routes/XXX');`
`App.use ('/api/XXX', XXXRouter);`

./routes/XXX.js

- Crear la ruta al servicio:
`const express = require('express');`
`const router = express.Router();`
`const XXX = require('../services/XXX');`
- Crear endpoint del servicio: (Tomando como ejemplo el verbo get)
`router.get('/', async function(req, res, next) {`
 `try {`
 `res.json(await XXX.getServicio());`
 `} catch (err) {`
 `console.error(`Error al traer los datos del servicio`, err.message);`
 `next(err);`
 `}`
`});`
- Exportar ruta:
`module.exports = router;`

./services/XXX.js

- Requisitos necesarios en el servicio:
`const db = require('../db');`
`const helper = require('../helper');`
`const config = require('../config');`
`var createError = require('http-errors');`
- Implementación del servicio.
`async function getServicio(){`
 `secuencia de código del servicio.... return { }`
`}`

Recursos necesarios:

Es necesario instalar las siguientes aplicaciones o programas:

- Visual studio code versión 1.18.1
- MySQL+ MySQL Workbench versión 8.0.32 (MYSQL Community Server–GPL)
- Node.js versión 16.14.1 + npm versión 8.5.0
- Express versión 4.17.1
- Postman versión 10.17.3
- GIT versión 4.17.1

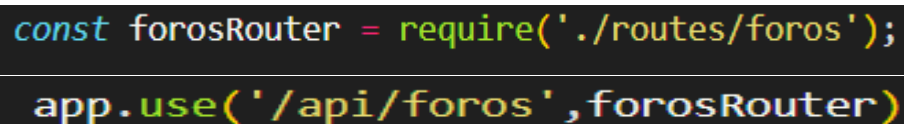
Piezas de software necesarias:

- MySQL2 versión 2.2.5
- Nodemon
- Body-parser versión 1.19.2
- Cors versión 2.8.5
- JWT-simple versión 0.5.6
- Moment versión 2.29.1
- Nodemailer versión 6.7.2
- Http-errors versión 2.0.0
- Dotenv versión 16.0.3

Ver Anexos (tecnologías y piezas de software)

Pasos:

1. Retomando la receta "Abrir Api Rest Dory", en el archivo principal "index.js" se crea la ruta al servicio y se utiliza (ver imagen 1).



```
const forosRouter = require('./routes/foros');  
app.use('/api/foros', forosRouter)
```

Imagen 1. Ruta y utilización de la ruta

2. En la carpeta "routes" se encuentran las rutas de los archivos ".js" en donde se accede a los servicios (ver imagen 2).

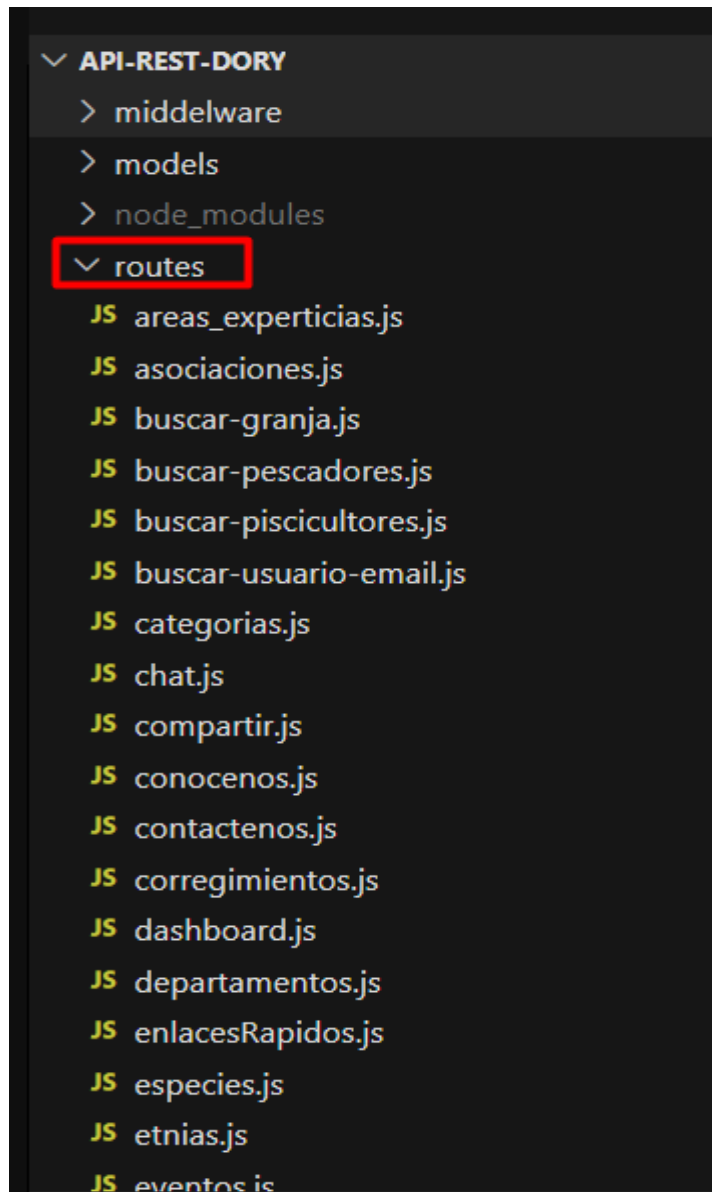


Imagen 2. Directorios de rutas a los servicios

3. Abrir el archivo de javascript de la ruta correspondiente y llamar el servicio (ver imagen 7).

```
JS foros.js x
routes > JS foros.js > ...
You, hace 2 meses | 1 author (You)
1 const express = require('express');
2 const router = express.Router();
3 const foros = require('../services/foros'); // Acceder al servicio mediante la variable foros
4
5 router.get('/obtener/todas/preguntas', async function(req, res, next) {
6   try {
7     res.json(await foros.getPreguntasForos()); // Enviamos la petición al servicio y éste devuelve la respuesta
8   } catch (err) {
9     console.error('Error al traer todas las preguntas del sistema', err.message);
10    next(err);
11  }
12 });
13
```

Imagen 7. Rutas y petición al servicio

4. En el directorio “services” se encuentran los servicios en archivos “.js” en donde se implementan los servicios (ver imagen 8).

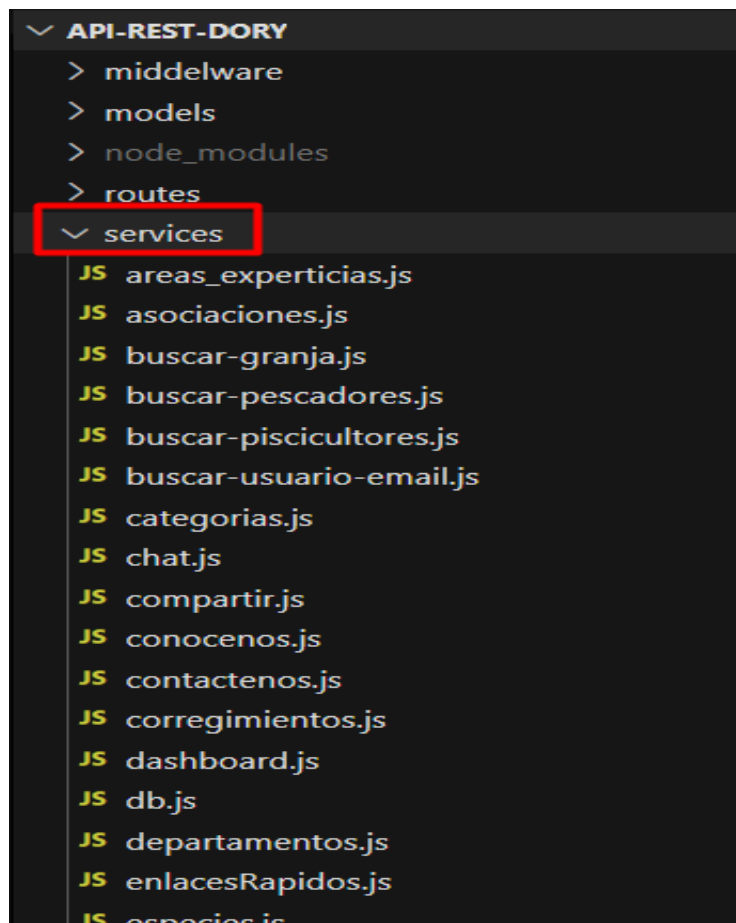
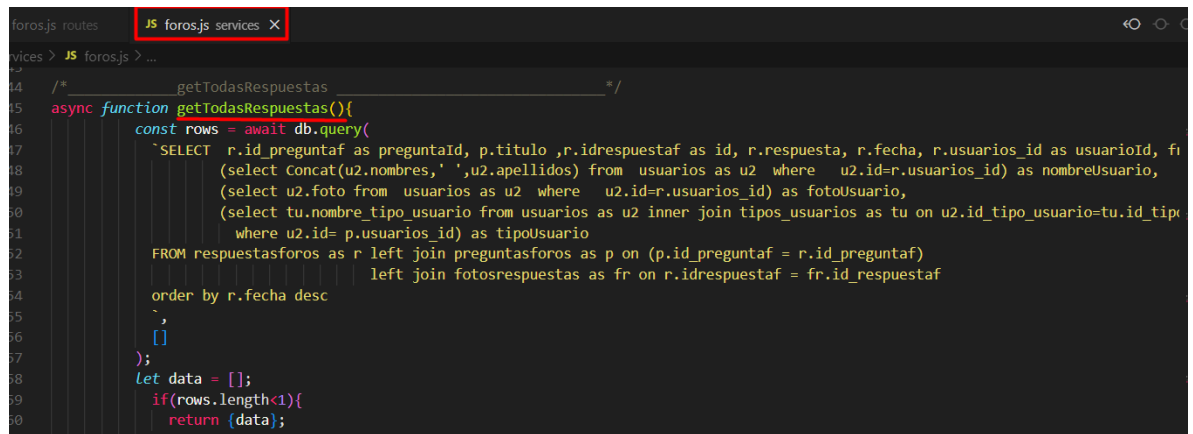


Imagen 8. Directorio de servicios

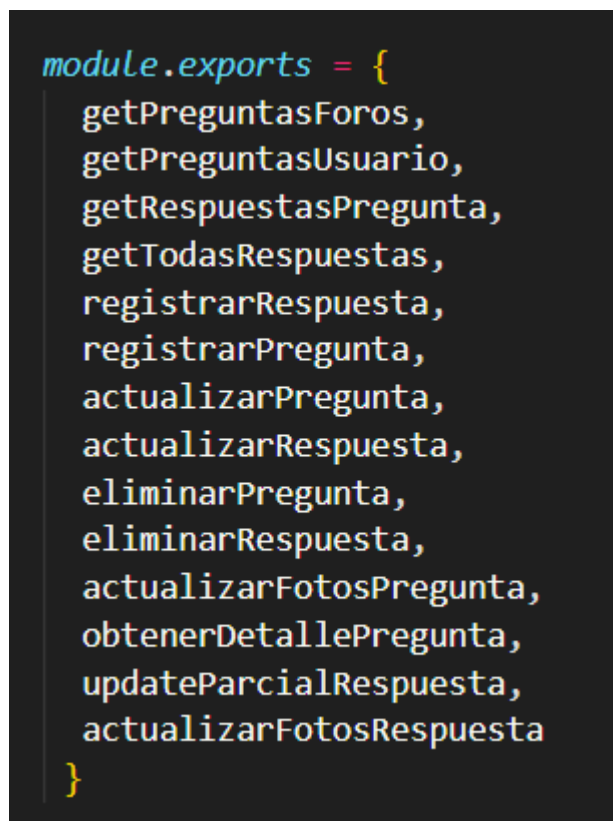
5. Abrir el archivo de javascript del servicio correspondiente e implementar el servicio en un archivo “.js” (ver imagen 9).



```
foros.js routes JS foros.js services X
vices > JS foros.js > ...
24 /* getTodasRespuestas */
25 async function getTodasRespuestas(){
26   const rows = await db.query(
27     `SELECT r.id_preguntaf as preguntaId, p.titulo ,r.idrespuestaf as id, r.respuesta, r.fecha, r.usuarios_id as usuarioId, fr
28     (select Concat(u2.nombres,' ',u2.apellidos) from usuarios as u2 where u2.id=r.usuarios_id) as nombreUsuario,
29     (select u2.foto from usuarios as u2 where u2.id=r.usuarios_id) as fotoUsuario,
30     (select tu.nombre_tipo_usuario from usuarios as u2 inner join tipos_usuarios as tu on u2.id_tipo_usuario=tu.id_tipo
31     where u2.id= p.usuarios_id) as tipoUsuario
32     FROM respuestasforos as r left join preguntasforos as p on (p.id_preguntaf = r.id_preguntaf)
33     left join fotosrespuestas as fr on r.idrespuestaf = fr.id_respuestaf
34     order by r.fecha desc
35   `,
36   []
37 );
38 let data = [];
39 if(rows.length<1){
40   return {data};
41 }
```

Imagen 9. Implementación de servicio

Nota: Se debe exportar el servicio dentro del archivo javascript donde se implementó (ver imagen 10)



```
module.exports = {
  getPreguntasForos,
  getPreguntasUsuario,
  getRespuestasPregunta,
  getTodasRespuestas,
  registrarRespuesta,
  registrarPregunta,
  actualizarPregunta,
  actualizarRespuesta,
  eliminarPregunta,
  eliminarRespuesta,
  actualizarFotosPregunta,
  obtenerDetallePregunta,
  updateParcialRespuesta,
  actualizarFotosRespuesta
}
```

Imagen 10. Exportación del servicio creado