**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
* Nodemialer 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).



Imagen 1. *Ruta y utilización de la ruta*

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

Texto

Descripción generada automáticamente

Imagen 2. *Directorios de rutas a los servicios*

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

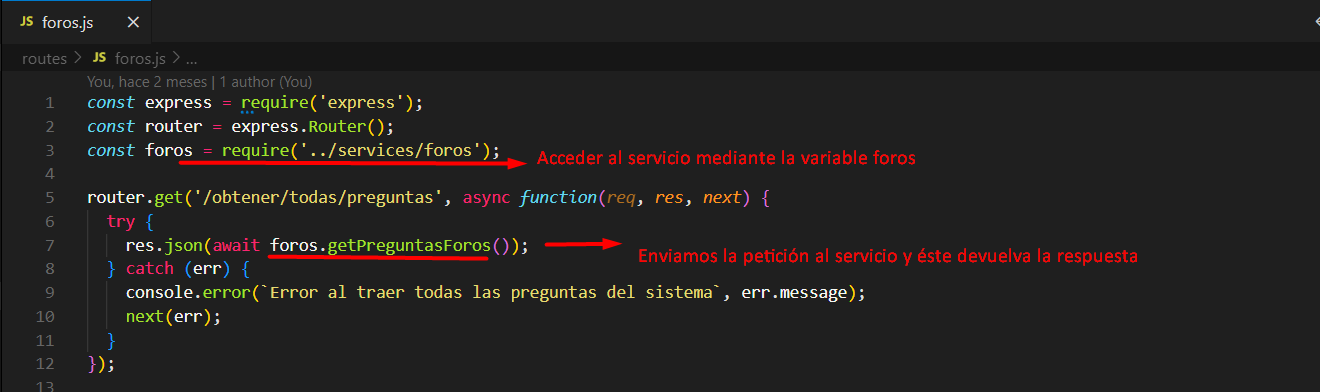


Imagen 7. *Rutas y petición al servicio*

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

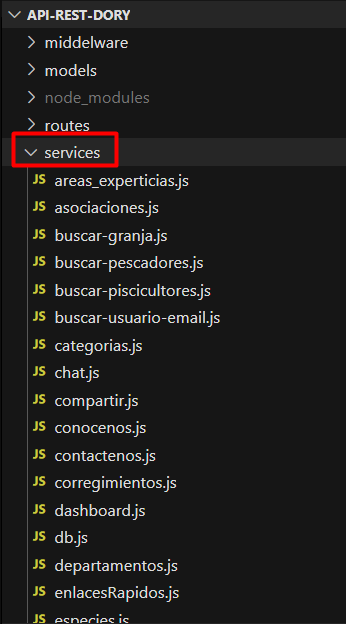


Imagen 8. *Directorio de servicios*

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

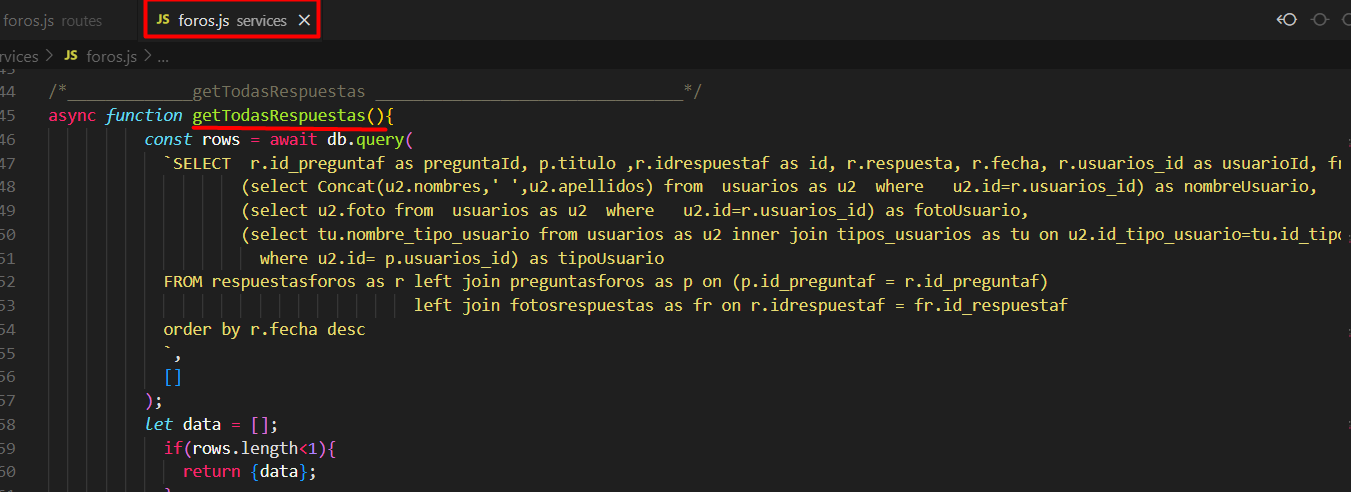


Imagen 9. *Implementación de servicio*

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

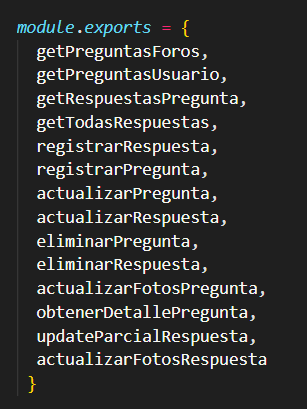


Imagen 10. *Exportación del servicio creado*