

Práctica 1

Desarrollo de una API REST para la gestión de una clínica de fisioterapia con Express y Mongoose

Parte 2. Definición de servicios REST

En esta parte de la práctica se van a definir los servicios REST que accederán a la base de datos, llamada **physiocare**, para realizar las diferentes operaciones sobre pacientes, fisioterapeutas (fisios) y expedientes médicos.

Instala *express* en la aplicación (si aún no lo has hecho) y crea una carpeta *routes* donde irás ubicando cada uno de los enrutadores que se explican a continuación.

1. Enrutadores

En TODOS los servicios, se debe enviar:

- Un *código de estado* apropiado: 200 si todo ha ido bien, 400 si hay un fallo en la petición y 500 si hay un fallo en el servidor
- Un *objeto JSON* con estos atributos:
 - **error**: contendrá el mensaje con el error que se haya producido
 - **result**: contendrá el resultado que se envía como respuesta. Dicho resultado se detalla a continuación para cada servicio.

Cada servicio tendrá una calificación independiente de 0,25 puntos, que serán 0 puntos automáticamente si:

- La ruta del endpoint no coincide EXACTAMENTE con la que se proporciona
- Los nombres de los campos de retorno no coinciden EXACTAMENTE con los indicados.

Enrutador para Pacientes

Este enrutador responderá a URIs que comiencen por **/patients**. Se pide implementar los siguientes servicios:

- **Obtener un listado de todos los pacientes:**
 - **Endpoint**: GET /patients
 - **Descripción**: devuelve el listado de todos los pacientes registrados en la clínica.
 - **Si todo ha ido bien**: devolverá un código 200 y una lista con la información de todos los pacientes.
 - **Si no hay pacientes registrados**: devolverá un código 404 y un mensaje indicando que no hay pacientes en el sistema

- **En caso de error en el servidor:** se devolverá un código 500 y un mensaje indicando que ocurrió un error interno del servidor
- **Obtener detalles de un paciente específico:**
 - **Endpoint:** GET /patients/:id
 - **Descripción:** devuelve el detalle de un paciente a partir de su ID.
 - **Si todo ha ido bien:** devolverá un código 200 y la información detallada del paciente.
 - **Si el paciente no existe:** devolverá un código 404 y un mensaje indicando que el paciente no se ha encontrado.
 - **Si ocurre un error en el servidor:** devolverá un código 500 y un mensaje indicando error interno del servidor.
- **Buscar pacientes por nombre o apellidos:**
 - **Endpoint:** GET /patients/find
 - **Descripción:** permitirá buscar pacientes que coincidan parcial o completamente con el apellido proporcionado en los parámetros de consulta (query parameters). Ejemplo: GET /patients/find?surname=Sanz. Si no se proporciona ningún apellido, se mostrará todo el listado de pacientes.
 - **Si todo ha ido bien:** devolverá un código 200 y una lista con los pacientes que coincidan con la búsqueda.
 - **Si no se encuentra ningún paciente:** devolverá un código 404 y un mensaje indicando que no se encontraron pacientes con esos criterios.
 - **Si ocurre un error en el servidor:** devolverá un código 500 y un mensaje indicando error interno del servidor.

Para resolver este servicio REST, puedes aprovechar el operador \$regex de MongoDB (consulta la documentación)

- **Insertar un paciente:**
 - **Endpoint:** POST /patients
 - **Descripción:** se añadirá el paciente que se reciba en la petición a la colección de pacientes. En dicha petición se enviarán todos los campos básicos del paciente (excepto el ID).
 - **Si todo ha ido bien:** devolverá un código 201 y los detalles del paciente insertado.
 - **Si ocurre un error al insertar el paciente o cualquier otro error:** devolverá un código 400 y el mensaje de error correspondiente.

- **Actualizar los datos de un paciente:**
 - **Endpoint:** PUT /patients/:id
 - **Descripción:** permitirá modificar los datos de un paciente existente (todo menos el ID). De nuevo, se enviarán en la petición los campos básicos del paciente.
 - **Si todo ha ido bien:** devolverá un código 200 y los detalles del paciente actualizado.
 - **Si ocurre un error al actualizar los datos (no se encuentra el paciente):** devolverá un código 400 y el mensaje "Error actualizando los datos del paciente".
 - **Si ocurre un error en el servidor:** devolverá un código 500 y un mensaje indicando error interno del servidor.
- **Eliminar un paciente:**
 - **Endpoint:** DELETE /patients/:id
 - **Descripción:** permitirá eliminar un paciente a partir de su ID. Como resultado, se devolverá el paciente eliminado.
 - **Si todo ha ido bien:** devolverá un código 200 y los detalles del paciente eliminado.
 - **Si el paciente no existe:** devolverá un código 404 y un mensaje de error indicando que el paciente a eliminar no existe.
 - **Si ocurre algún error :** devolverá un código 500 y un mensaje de error.

Enrutador para Fisios

Este enrutador responderá a URIs que comiencen por **/physios**. Se pide implementar los siguientes servicios:

- **Obtener un listado de todos los fisioterapeutas:**
 - **Endpoint:** GET /physios
 - **Descripción:** devolverá el listado de todos los fisioterapeutas registrados en la clínica.
 - **Si todo ha ido bien:** devolverá un código 200 y una lista con la información de todos los fisios.
 - **Si no hay fisios registrados:** devolverá un código 404 y un mensaje indicando que no hay fisios en el sistema
 - **En caso de error en el servidor:** se devolverá un código 500 y un mensaje indicando que ocurrió un error interno del servidor
- **Buscar fisioterapeutas por especialidad:**

- **Endpoint:** GET /physios/find
- **Descripción:** permitirá buscar fisioterapeutas que coincidan con la especialidad proporcionada en los parámetros de consulta (query parameters). Ej: GET /physios/find?specialty=Sports.
 - **Si todo ha ido bien:** devolverá un código 200 y una lista con los fisios que coincidan con la búsqueda.
 - **Si no se encuentra ningún fisio:** devolverá un código 404 y un mensaje indicando que no se encontraron fisios con esos criterios.
 - **Si ocurre un error en el servidor:** devolverá un código 500 y un mensaje indicando error interno del servidor.
- **Obtener detalles de un fisioterapeuta específico:**
 - **Endpoint:** GET /physios/:id
 - **Descripción:** devolverá el detalle de un fisio a partir de su ID.
 - **Si todo ha ido bien:** devolverá un código 200 y la información detallada del fisio.
 - **Si el fisio no existe:** devolverá un código 404 y un mensaje indicando que el fisio no se ha encontrado.
 - **Si ocurre un error en el servidor:** devolverá un código 500 y un mensaje indicando error interno del servidor.
- **Insertar un fisioterapeuta:**
 - **Endpoint:** POST /physios
 - **Descripción:** se añadirá el fisioterapeuta que se reciba en la petición a la colección de fisios. En dicha petición se enviarán todos los campos básicos del fisio (excepto el ID).
 - **Si todo ha ido bien:** devolverá un código 201 y los detalles del fisio insertado.
 - **Si ocurre un error al insertar el paciente o cualquier otro error:** devolverá un código 400 y el mensaje de error correspondiente.
- **Actualizar los datos de un fisioterapeuta:**
 - **Endpoint:** PUT /physios/:id
 - **Descripción:** permitirá modificar los datos del fisioterapeuta (todo menos el ID). Se devolverá el detalle del fisio modificado.
 - **Si todo ha ido bien:** devolverá un código 200 y los detalles del fisio actualizado.
 - **Si ocurre un error al actualizar los datos** (no se encuentra el fisio): devolverá un código 400 y el mensaje "Error actualizando los datos del fisio".

- **Si ocurre un error en el servidor:** devolverá un código 500 y un mensaje indicando error interno del servidor.
- **Eliminar un fisioterapeuta:**
 - **Endpoint:** DELETE /physios/:id
 - **Descripción:** permitirá eliminar un fisio a partir de su ID. Como resultado, se devolverá el fisio eliminado.
 - **Si todo ha ido bien:** devolverá un código 200 y los detalles del fisio eliminado.
 - **Si el fisio no existe:** devolverá un código 404 y un mensaje de error indicando que el fisioterapeuta a eliminar no existe.
 - **Si ocurre algún error:** devolverá un código 500 y un mensaje de error.

Enrutador para Expedientes Médicos

Este enrutador responderá a URLs que comiencen por **/records**. Se pide implementar los siguientes servicios:

- **Obtener un listado de todos los expedientes médicos:**
 - **Endpoint:** GET /records
 - **Descripción:** devolverá como resultado todo el listado de expedientes médicos, incluyendo las consultas de cada paciente.
 - **Si todo ha ido bien:** devolverá un código 200 y una lista con los expedientes médicos que existan.
 - **Si no se encuentra ningún expediente:** devolverá un código 404 y un mensaje indicando que no se encontraron expedientes en el sistema.
 - **Si ocurre un error en el servidor:** devolverá un código 500 y un mensaje indicando error interno del servidor.
- **Buscar expedientes por nombre de paciente:**
 - **Endpoint:** GET /records/find
 - **Descripción:** permitirá buscar expedientes de pacientes que coincidan con el apellido proporcionado en los parámetros de consulta. Ej: GET /records/find?surname=Calle. Se devolverá una lista de expedientes que coincidan con el nombre o apellidos.
 - **Si todo ha ido bien:** devolverá un código 200 y una lista con el expediente del paciente que coincida con la búsqueda.
 - **Si no se encuentra ningún expediente:** devolverá un código 404 y un mensaje indicando que no se encontraron expedientes.

- **Si ocurre un error en el servidor:** devolverá un código 500 y un mensaje indicando error interno del servidor.
- **Obtener detalles de un expediente específico:**
 - **Endpoint:** GET /records/:id
 - **Descripción:** devolverá el expediente de un paciente a partir de su ID, incluyendo todas las consultas.
 - **Si todo ha ido bien:** devolverá un código 200 y la información detallada del expediente, incluyendo las consultas.
 - **Si el expediente no existe:** devolverá un código 404 y un mensaje indicando que el expediente no se ha encontrado.
 - **Si ocurre un error en el servidor:** devolverá un código 500 y un mensaje indicando error interno del servidor.
- **Insertar un expediente médico:**
 - **Endpoint:** POST /records
 - **Descripción:** se encarga únicamente de crear el expediente médico de un paciente, sin las citas. Se devolverá como resultado el expediente insertado.
 - **Si todo ha ido bien:** devolverá un código 201 y los detalles del expediente creado.
 - **Si ocurre un error al insertar el expediente o cualquier otro error:** devolverá un código 400 y el mensaje de error correspondiente.
- **Añadir consultas a un expediente:**
 - **Endpoint:** POST /records/:id/appointments
 - **Descripción:** se encarga de añadir una nueva cita a un expediente médico ya existente. Se devolverá el expediente con las consultas.
 - **Si todo ha ido bien:** devolverá un código 201 y la información detallada del expediente, incluyendo las consultas.
 - **Si el expediente no existe:** devolverá un código 404 y un mensaje indicando que el expediente no se ha encontrado.
 - **Si ocurre cualquier error:** devolverá un código 500 y un mensaje indicando error interno del servidor.
- **Eliminar un expediente médico:**
 - **Endpoint:** DELETE /records/:id
 - **Descripción:** permitirá eliminar un expediente completo a partir del ID del expediente. Como resultado, se devolverá el expediente eliminado.

- **Si todo ha ido bien:** devolverá un código 200 y los detalles del expediente eliminado.
- **Si el expediente no existe:** devolverá un código 404 y un mensaje de error indicando que el expediente no existe.
- **Si ocurre algún otro error:** devolverá un código 500 y un mensaje de error.

2. Servidor Principal (0,75 puntos)

Se deberá implementar en el archivo **index.js** la funcionalidad general del servidor principal:

- Carga de librerías/módulos necesarios (al menos Express y Mongoose).
- Conexión con la base de datos (la base de datos se llama physiocare).
- Definición del middleware necesario (carga de enrutadores).
- Puesta en marcha del servidor en el puerto 8080.

3. Prueba de los servicios (0,75 puntos)

Se debe definir una colección de pruebas (en ThunderClient, Postman o una herramienta similar) para verificar el correcto funcionamiento de cada uno de estos servicios.