

Práctica 2. BBDD con SpringBoot+MySQL y Node+MongoDB

Enunciado

Parte 1: SpringBoot+MySQL

Partiendo de la práctica anterior:

- Añadir persistencia a la aplicación con Spring Data y MySQL
- Quitar la parte web: eliminar templates, dependencia moustache y eliminar el controlador web.
- Los autores pasan a ser una entidad propia: dejan de ser un string de la clase Comentario a ser una clase propia con las siguientes propiedades: Nombre y Edad
- Cambios en la API REST:
 - Nuevas operaciones REST:
 - Se debe añadir un endpoint para crear Autores.
 - Se deberá añadir un endpoint de consulta de todos los comentarios de un Autor concreto. En este caso, los Comentarios deberán traer el id del Post.
 - Cambios en las operaciones REST ya existentes:
 - Al consultar un Post, se devolverán todos sus comentarios. Los comentarios en este caso deberán traer el nombre e id del Autor.
 - Al crear un Comentario, se deberá proveer el id del Autor asociado. No se podrá crear un Comentario sin este id.

Parte 2: Node+MongoDB

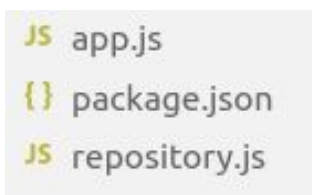
Se desea implementar una **aplicación REST con persistencia (sin web)** para manejar un foro.

- El foro está formado por 2 entidades:
 - **Entradas**. Cada entrada contiene los siguientes campos: Nombre del autor, Nickname del autor, Titulo, Texto, Lista de comentarios.
 - **Comentarios**. Cada comentario contiene los siguientes campos: Nickname del autor del comentario, Contenido, Fecha del comentario.

- La aplicación ofrecerá **7 endpoints REST**, a saber:
 - Recuperación de todas las entradas
 - Recuperación de una entrada
 - Adición de una nueva entrada
 - Borrado de una entrada existente
 - Modificación de una entrada existente (se modificará la entrada al completo)
 - Adición de un nuevo comentario a una entrada
 - Borrado de un comentario existente

NOTA: se proporcionará el archivo de **POSTMAN** con todas las operaciones ya creadas. De esta manera los alumnos tendrán ya predefinidas las URLs y tendrán preparadas para ejecutar todas las operaciones REST. En definitiva, las APIs REST de todos los alumnos expondrán exactamente los mismos métodos definidos en el archivo de **POSTMAN**.

- La aplicación estará **dividida en, al menos, 2 módulos**: un módulo “app.js” (o similar) que ejerza de controlador REST y un módulo “repository.js” o similar que sea el que único que acceda a la BBDD. Cualquier otra lógica de negocio que se quiera añadir (por ejemplo, si se desea, comprobación de los campos enviados por el cliente) deberá estar en su propio módulo.



```
JS app.js
{} package.json
JS repository.js
```

Apecto mínimo de la aplicación Node

- Aspectos técnicos a tener en cuenta:
 - La versión de **Node** debe ser la última LTS (12.13.1)
 - Como BBDD se usará **MongoDB**. La versión de MongoDB debe ser la última LTS (4.2.x)
 - Se utilizará **express** para implementar el controlador REST y el driver native de MongoDB para NodeJS (paquete NPM **mongodb**). De esta manera, el package.json tendrá este aspecto:

```
{  
  "name": "practica",  
  "version": "1.0.0",  
  "dependencies": {  
    "express": "4.17.1",  
    "mongodb": "3.3.5"  
  }  
}
```

Aspecto del package.json (“name” y “version” pueden variar). La versión de las dependencias debe ser la última disponible en NPM

- La API REST deberá cumplir con el nivel de madurez 2 y el formato de las URLs deberá identificar recursos, no acciones. *Esto viene ya proporcionado por la obligatoriedad de utilizar el archivo de POSTMAN para implementar la API REST.*

Formato de entrega

La práctica se entregará teniendo en cuenta los siguientes aspectos:

- La práctica se entregará como un fichero .zip. Dentro del fichero .zip deberá haber **2 carpetas**: una llamada “**Java**” y otra llamada “**Node**”. Dentro de la carpeta Java se encontrará el proyecto Maven (el *pom.xml* estará en la raíz de la carpeta Java). Dentro de la carpeta Node se encontrará el proyecto NPM (el *package.json* estará en la raíz de la carpeta Node). El nombre del fichero .zip será el correo URJC del alumno (sin @alumnos.urjc.es).
- En la carpeta Java, actualizar el fichero API.md y el JSON de POSTMAN de la práctica anterior para incluir las 2 nuevas operaciones REST y los cambios en las operaciones existentes.
- Los proyectos se pueden crear con cualquier editor o IDE (para el proyecto Node se recomienda VSCode).
- La práctica se entregará por el aula virtual con la fecha indicada.

Las prácticas se podrán realizar de forma individual o por parejas. En caso de que la práctica se haga por parejas:

- Sólo será entregada por uno de los alumnos
- El nombre del fichero .zip contendrá el correo de ambos alumnos separado por guión. Por ejemplo p.perezf2019-z.gonzalez2019.zip