Resolución de Repaso CRUD (populado)

Referencia endpoints

Id	Method	Path	Description
1	get	/parks/new	Muestra el formulario para crear un parque
2	post	/parks/new	Guarda en la BBDD un parque
3	get	/coasters/new	Muestra el formulario para crear una montaña rusa
4	post	/coasters/new	Guarda en la BBDD una montaña rusa
5	get	/coasters/	Muestra la lista de montañas rusas
6	get	/coasters/:id	Muestra los detalles de una montaña rusa

Iteraciones

0.- Comienza a trabajar [sobre este repo]

1.- Crea un modelo 'park' con el esquema:

```
name:           Str
description:    Str
active:         Bool
```

```
// models/park.model.js
const parkSchema = new Schema({
 name: String,
 description: String,
 active: { type: Boolean, default: true },
})
```

2.- Crea un modelo 'coaster' con el esquema:

```
name:           Str
description:    Str
inversions:     Num
length:         Num
active:         Bool
park_id:        ObjectId (referenciando al modelo Park)
```

```
// models/coaster.model.js
const coasterModel = new Schema({
 name: String,
 description: String,
 inversions: Number,
 length: Number,
 active: { type: Boolean, default: true },
 park_id: { type: Schema.Types.ObjectId, ref: 'Park' },
})
```

3.- Crea la ruta 1 y muestra la vista de parks/new-park.hbs

```
// routes/park.routes.js

router.get('/new', (req, res) => res.render('parks/new-park'))
```

4.- Crea la ruta 2 para guardar parques en la BBDD mediante el modelo 'park' - no olvides modificar los atributos `action` y `method` del formulario, así como el atributo `name` de cada campo para conectarlo adecuadamente con el controlador.

```
// routes/park.routes.js

router.post('/new', (req, res) => {
 const { name, description } = req.body
 Park.create({ name, description, active: true })
   .then(newPark => console.log(newPark))
   .catch(err => console.log(err))
})
```

```
// views/parks/new-park.hbs

<form action="/parks/new" method="post">
  <div class="form-group">
    <label for="name-input">Nombre</label>
    <input name="name" type="text" class="form-control" id="name-input">
  </div>
  <div class="form-group">
    <label for="description-input">Descripción</label>
    <input name="description" type="text" class="form-control" id="description-input">
  </div>
  <button type="submit" class="btn btn-dark">Crear parque</button>
</form>
```

5.- Testea en la BBDD que el formulario crea parques sin incidencia.

6.- Crea la ruta 3 y muestra la vista de coasters/new-coaster.hbs

```
// routes/coaster.routes.js

router.get('/new', (req, res) => {
 Park.find()
   .then(allTheParks => res.render('coasters/new-coaster', { parks: allTheParks }))
   .catch(err => console.log(err))
})
```

7.- En la vista, las opciones que deben aparecer en la lista los nombres de cada una de los parques de la BBDD (debes enviarlos desde el controlador que genera la vista haciendo uso del modelo 'Park'). Añade a cada `option` el atributo value con el `_id` de ese parque. Ejemplo:

```
      <select name="park" class="form-control" id="park-list">
            <option>Seleccionar</option>
            <option value="5d408c5eff558796d4cf681e">Port Aventura</option>
            <option value="5d408c5eff558796d4cf6894">Isla Mágica</option>
       </select>
```

```
// views/coasters/new-coaster.hbs

[...]
<div class="form-group">
  <label for="park-list">Parque en el que se encuentra:</label>
  <select name="park" class="form-control" id="park-list">
    <option>Seleccionar</option>
    {{#each parks}}
      <option value="{{_id}}">{{name}}</option>
    {{/each}}
  </select>
</div>
[...]
```

8.- Crea la ruta 4 para guardar montañas rusas en la BBDD mediante el modelo ‘coaster’ - no olvides modificar los atributos `action` y `method` del formulario, así como el atributo `name` de cada campo para conectarlo adecuadamente con el controlador. El ID del parque al que pertenece llegará al controlador identificado con el valor `name` de la lista de selección.

```
// routes/coaster.routes.js

router.post('/new', (req, res) => {
 const { name, description, inversions, length, park } = req.body

 Coaster.create({ name, description, inversions, length, park_id: park })
   .then(newCoaster => res.redirect('/coasters'))
   .catch(err => console.log(err))
})
```

9.- Testea en la BBDD que el formulario crea montañas rusas sin incidencia

---

10.- Crea la ruta 5 y muestra la vista de coasters/index-coasters.hbs

```
// routes/coaster.routes.js

router.get('/', (req, res) => {
 Coaster.find()
   .populate('park_id')
   .then(allCoasters => res.render('coasters/coasters-index', { coasters: allCoasters
}))
   .catch(err => console.log(err))
})
```

11.- Adecúa la vista para que muestre todas las montañas rusas, y para que cada botón enlace con la URL /coasters/id_de_la_montaña_rusa. Ejemplo:

http://localhost:3000/coasters/5d408c5eff558796d4cf681e

```
// views/coasters/coasters-index.hbs

[...]
{{#each coasters}}
        <a href="/coasters/{{_id}}" class="btn btn-sm btn-dark">Ver detalles</a>
        <h4>{{name}}</h4>
        <p>{{description}}</p>
        <small>
            <strong>Longitud: </strong> {{length}}m |
            <strong>Inversiones: </strong>   {{inversions}} |
            <strong>Parque: </strong> {{park_id.name}}
         </small>
        <hr>
{{/each}}
[...]
```

12.- Crea la ruta 6 y muestra la vida de coasters/coaster-details.hbs

```
// routes/coaster.routes.js

router.get('/:id', (req, res) => {
 Coaster.findById(req.params.id)
   .populate('park_id')
   .then(theCoaster => res.render('coasters/coaster-details', { coaster: theCoaster
}))
   .catch(err => console.log(err))
})
```

13:- Adecúa la vista para que muestre toda la información de la montaña rusa, incluyendo el nombre y descripción del parque al que pertenece

```
// views/coasters/coaster-details.hbs

 <h1 style="font-weight: 100">{{coaster.name}}</h1>

 <h4>Descripción</h4>
 <p>{{coaster.description}}</p>
 <p><strong>Longitud: </strong> {{coaster.length}}m | <strong>Inversiones: </strong>
{{coaster.inversions}}</p>

 <hr>

 <h5>Sobre el parque</h5>
 <p><strong>Nombre</strong>: {{coaster.park_id.name}}</p>
 <p><strong>Descripción</strong>: {{coaster.park_id.description}}</p>
```
