const Player = require('../models/playerModel');

// Example: Update a player

const updatePlayer = async (req, res) => {

const { playerId } = req.params;

const { name, age, team, position } = req.body;

try {

const updatedPlayer = await Player.findByIdAndUpdate(

playerId,

{ name, age, team, position },

{ new: true }

);

if (!updatedPlayer) {

return res.status(404).json({ message: 'Player not found' });

}

res.json(updatedPlayer);

} catch (error) {

res.status(500).json({ message: error.message });

}

};

// Example: Delete a player

const deletePlayer = async (req, res) => {

const { playerId } = req.params;

try {

const deletedPlayer = await Player.findByIdAndDelete(playerId);

if (!deletedPlayer) {

return res.status(404).json({ message: 'Player not found' });

}

res.json({ message: 'Player deleted successfully' });

} catch (error) {

res.status(500).json({ message: error.message });

}

};

module.exports = {

getAllPlayers,

addPlayer,

updatePlayer,

deletePlayer,

};

In this example, I've added the updatePlayer and deletePlayer functions. The updatePlayer function updates a player based on the provided player ID, and the deletePlayer function deletes a player based on the player ID. Adjust these functions according to your specific requirements and the structure of your data.

// routes/playerRoutes.js

// ... Existing imports and code ...

// Routes for handling CRUD operations

router.put('/players/:playerId', playerController.updatePlayer);

router.delete('/players/:playerId', playerController.deletePlayer);

// ... Define other routes for queries

Now, you can test your API with tools like Postman or CURL by making requests to the corresponding endpoints (e.g., PUT /api/players/:playerId for updating, DELETE /api/players/:playerId for deleting).