const express = require('express');

const app = express();

const packageModel = require('./packageModel');

app.use(express.json());

// Create package

[app.post](http://app.post/)('/packages', async (req, res) => {

  try {

    const packageData = req.body;

    const createdPackage = await packageModel.createPackage(packageData);

    res.status(201).json(createdPackage);

  } catch (error) {

    res.status(500).json({ message: 'Failed to create package' });

  }

});

// Get all packages

app.get('/packages', async (req, res) => {

  try {

    const packages = await packageModel.getPackages();

    res.json(packages);

  } catch (error) {

    res.status(500).json({ message: 'Failed to retrieve packages' });

  }

});

// Get package

app.get('/packages/:packageId', async (req, res) => {

  try {

    const packageId = req.params.packageId;

    const packageData = await packageModel.getPackage(packageId);

    if (!packageData) {

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

    } else {

      res.json(packageData);

    }

  } catch (error) {

    res.status(500).json({ message: 'Failed to retrieve package' });

  }

});

// Update package

app.put('/packages/:packageId', async (req, res) => {

  try {

    const packageId = req.params.packageId;

    const packageData = req.body;

    const updatedPackage = await packageModel.updatePackage(packageId, packageData);

    if (!updatedPackage) {

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

    } else {

      res.json(updatedPackage);

    }

  } catch (error) {

    res.status(500).json({ message: 'Failed to update package' });

  }

});

// Delete package

app.delete('/packages/:packageId', async (req, res) => {

  try {

    const packageId = req.params.packageId;

    await packageModel.deletePackage(packageId);

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

  } catch (error) {

    res.status(500).json({ message: 'Failed to delete package' });

  }

});