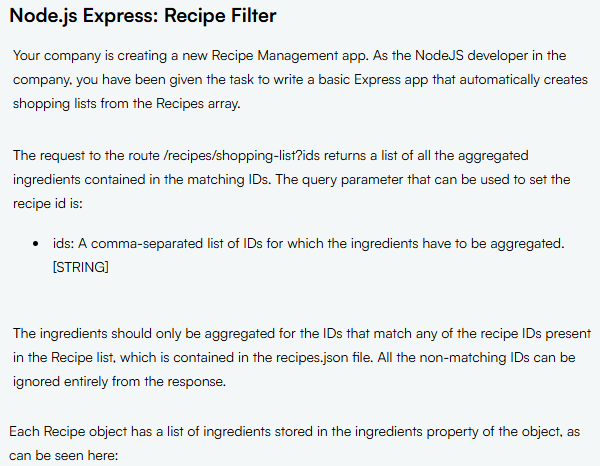
# Node.js Express: Recipe Filter



## Challenge/routes/recipes.js

var recipes = require('../recipes.json');

var router = require('express').Router();

router.get('/shopping-list', (req, res) => {

  const { ids } = req.query;

  // If ids are missing, return 400

  if (!ids) {

    return res.status(400).send('Missing ids parameter');

  }

  // Parse the ids

  const requestedIds = ids.split(',');

  // Filter the recipes based on ids

  const matchingRecipes = recipes.filter(recipe => requestedIds.includes(String(recipe.id)));

  // If no recipes match, return 404

  if (matchingRecipes.length === 0) {

    return res.status(404).send('NOT\_FOUND');

  }

  // Aggregate ingredients

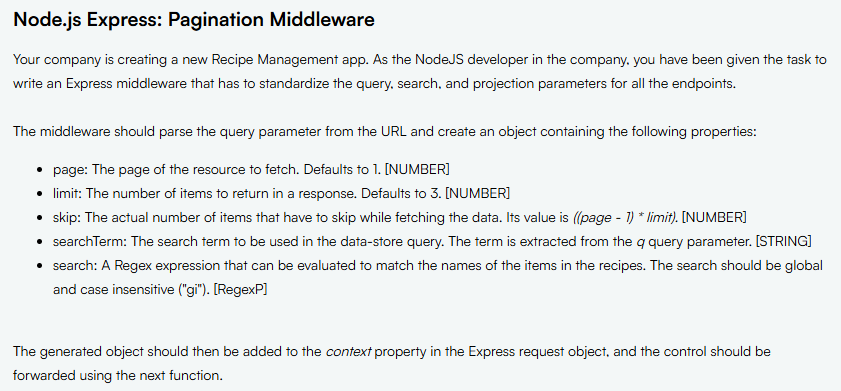
  const ingredientsList = matchingRecipes.flatMap(recipe => recipe.ingredients);

  res.json(ingredientsList);

});

module.exports = router;

# Node.js Express: Pagination Middleware



## Challenge/middleware.js

module.exports = (req, res, next) => {

  const { page = 1, limit = 3, q = '' } = req.query;

  req.context = {

    page: +page,

    limit: +limit,

    skip: (page - 1) \* limit,

    searchTerm: q,

    search: new RegExp(q, "gi")

  };

  next();

};

## Challenge/routes/recipes.js

var recipes = require('../recipes.json');

var router = require('express').Router();

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

    const {context} = req;

    const page = context && context.page || req.query.page

    const limit = context && context.limit || req.query.limit

    const skip = context && context.skip || req.query.skip || ((page-1) \* limit)

    const search = context && context.searchTerm || req.query.q;

    let results = [];

     if (search) {

        results = recipes.filter(({name}) => name.match(search))

    } else {

        results = recipes

    }

    results = await results.slice(skip, (skip + limit))

    res.json({

        page: Number(page),

        limit: Number(limit),

        skip: Number(skip),

        search: search,

        data: results || []

    });

});

module.exports = router;

## challenge/app.js

var express = require('express');

var path = require('path');

var cookieParser = require('cookie-parser');

var logger = require('morgan');

var middleware = require('./middleware');

var indexRouter = require('./routes/index');

var recipesRouter = require('./routes/recipes');

var app = express();

// view engine setup

app.set('view engine', 'jade');

app.use(logger('dev'));

app.use(express.json());

app.use(express.urlencoded({ extended: false }));

app.use(cookieParser());

app.use(express.static(path.join(\_\_dirname, 'public')));

app.use('/', indexRouter);

app.use('/recipes', recipesRouter);

module.exports = app;