class APIFeatures{

//construction is a function that automatically call as soon as we create a new object out of this class

constructor(query, queryString){

this.query = query;

this.queryString = queryString;

}

filter(){

const queryObj = {...this.queryString}; //... restructuring object = this 3 dots take the fields out of the object; implement something similar

const excludedFields = ['page','sort','limit','fields'];

excludedFields.forEach(el => delete queryObj[el]);

let queryStr = JSON.stringify(queryObj);

queryStr = queryStr.replace(/\b(gte|gt|lte|lt)\b/g, match => `$${match}`);

this.query.find(JSON.parse(queryStr))

return this;

// let query = TourModel.find(JSON.parse(queryStr));

}

sort(){

//2 Sorting

if(this.queryString.sort){

const sortBy = this.queryString.sort.split(',').join(' ');

this.query = this.query.sort(sortBy);

}else{

this.query = this.query.sort('-createdAt');

}

return this;

}

limitFields(){

//3 Field Limiting

if(this.queryString.fields){

const fields = this.queryString.fields.split(',').join(' ');

this.query = this.query.select(fields);

}else{

this.query = this.query.select('-\_\_v'); //selecting all except \_\_v field add(- sign)

}

return this;

}

paginate(){

const page = this.queryString.page \* 1 || 1; // this is to convert string to number; || is defining the default value of one

const limit = this.queryString.limit \* 1 || 100;

const skip = (page - 1) \* limit;

this.query = this.query.skip(skip).limit(limit);

return this;

}

} module.exports = APIFeatures;