'use strict'

var validator = require('validator');

var fs = require('fs');

var path = require('path');

var Article = require('../models/article');

var controller = {

    datosCurso: (req, res) => {

        var hola = req.body.hola;

        return res.status(200).send({

            curso: 'Master en Frameworks JS',

            autor: 'Godoy Guillermo GG',

            url: 'gg230592.com',

            hola

        });

    },

    test: (req, res) => {

        return res.status(200).send({

            message: 'Soy la accion test de mi controlador de articulos'

        });

    },

    save: (req, res) => {

        // Recoger parametros por post

        var params = req.body;

        //validar datos (Validator)

        try{

            var validate\_title = !validator.isEmpty(params.title);

            var validate\_content = !validator.isEmpty(params.content);

        }catch(err){

            return res.status(200).send({

                status: 'error',

                message: 'faltan datos por enviar !!!'

            });

        }

        if(validate\_title && validate\_content){

        // Crear el objeto a guardar

        var article = new Article();

        //Asignar valores

        article.title = params.title;

        article.content = params.content;

        if(params.image){

            article.image = params.image;

        }else{

            article.image = null;

        }

        //Guardar el articulo

        article.save((err, articleStored) => {

            if(err || !articleStored){

                return res.status(404).send({

                    status: 'error',

                    message: 'El articulo no se ha guardado !!'

                });

            }

            //Devolver una respuesta

            return res.status(200).send({

                status: 'success',

                article: articleStored

            });

        });

        }else{

            return res.status(200).send({

                status: 'error',

                message: 'Los datos no son válidos !!!'

            });

        }

    },

    getArticles: (req, res) =>{

        var query = Article.find({});

        var last = req.params.last;

        if(last || last != undefined){

            query.limit(5);

        }

        //Find

        query.sort('-\_id').exec((err, articles) => {

            if(err){

                return res.status(500).send({

                    status: 'error',

                    message: 'Error al devolver los articulos !!!'

                });

            }

            if(!articles){

                return res.status(404).send({

                    status: 'error',

                    message: 'No hay articulos para mostrar !!!'

                });

            }

            return res.status(200).send({

                status: 'success',

                articles

            });

        });

    },

     getArticle: (req, res) =>{

        // Recoger el id de la url

        var articleId = req.params.id;

        // Comprobar que existe

        if(!articleId || articleId == null){

            return res.status(404).send({

                status: 'error',

                message: 'No existe el articulo !!!'

            });

        }

        // Buscar el articulo

        Article.findById(articleId, (err, article) =>{

            if(err || !article){

                return res.status(404).send({

                    status: 'error',

                    message: 'No existe el articulo !!!'

                });

            }

            // Devolverlo en json

            return res.status(200).send({

                status: 'success',

                article

            });

        });

    },

    update: (req, res) =>{

        // Recoger el id del articulo por la url

        var articleId = req.params.id;

        // Recoger los datos que llegan por put

        var params = req.body;

        // Validar datos

        try{

            var validate\_title = !validator.isEmpty(params.title);

            var validate\_content = !validator.isEmpty(params.content);

        }catch(err){

            return res.status(200).send({

                status: 'error',

                message: 'Faltan datos por enviar !!!'

            });

        }

        if(validate\_title && validate\_content){

            // Find and update

            Article.findOneAndUpdate({\_id: articleId}, params, {new:true}, (err, articleUpdated) => {

                if(err){

                    return res.status(500).send({

                        status: 'error',

                        message: 'Error al actualizar !!!'

                    });

                }

                if(!articleUpdated){

                    return res.status(404).send({

                        status: 'error',

                        message: 'No existe el articulo !!!'

                    });

                }

                return res.status(200).send({

                    status: 'success',

                    article: articleUpdated

                });

            });

        }else{

            // Devolver respuesta

            return res.status(200).send({

                status: 'error',

                message: 'la validacion no es correcta !!!'

            });

        }

    },

    delete: (req, res) => {

        // Recoger el id de la url

        var articleId = req.params.id;

        // Find and delete

        Article.findOneAndDelete({\_id: articleId}, (err, articleRemoved) =>{

            if(err){

                return res.status(500).send({

                    status: 'error',

                    message: 'Error al borrar!!!'

                });

            }

            if(!articleRemoved){

                return res.status(404).send({

                    status: 'error',

                    message: 'No se ha borrado/ no existe!!!'

                });

            }

            return res.status(200).send({

                status:'success',

                article: articleRemoved

            });

        });

    },

    upload: (req, res)=> {

        // Configurar el modulo connect multiparty router/article.js (T hecho)

        // Recoger el fichero de la peticion

        var file\_name = 'Imagen no subida...';

        if(!req.files){

            return res.status(404).send({

                status: 'error',

                message: file\_name

            });

        }

        // Conseguir nombre y la extrension del archivo

        var file\_path = req.files.file0.path;

        var file\_split = file\_path.split('\\');

        /\* "ADVERTENCIA" en linux o MAC

          var file\_split = file\_path.split('/');

        \*/

        // Nombre del archivo

        var file\_name = file\_split[2];

        // Extension del fichero

        var extension\_split = file\_name.split('\.');

        var file\_ext = extension\_split[1];

        // Comprobar la extension, solo imgs, si no es valida borrar el fichero

        if(file\_ext != 'png' && file\_ext != 'jpg' && file\_ext != 'jpeg' && file\_ext != 'gif'){

            //borrar el archivo subido

            fs.unlink(file\_path, (err)=>{

                return res.status(200).send({

                    status: 'error',

                    message: 'La extension de la img no es valida'

                });

            });

        }else{

            // Si todo es valido, sacando la id de la url

            var articleId = req.params.id;

            if(articleId){

                // Buscar el articulo, asignarle el nombre de la imagen y actualizarlo

                Article.findOneAndUpdate({\_id: articleId}, {image: file\_name}, {new:true}, (err, articleUpdated) => {

                    if(err || !articleUpdated){

                        return res.status(200).send({

                            status: 'error',

                            message: 'Error al guardar la imagen de articulo !!!'

                        });

                    }

                    return res.status(200).send({

                        status: 'success',

                        article: articleUpdated

                    });

                });

             }else{

                return res.status(200).send({

                    status: 'success',

                    image: file\_name

                });

             }

        }

    }, // end upload file

    getImage: (req, res) => {

        var file = req.params.image;

        var path\_file = './upload/articles/'+file;

        fs.exists(path\_file, (exists)=>{

            if(exists){

                return res.sendfile(path.resolve(path\_file));

            }else{

                return res.status(404).send({

                    status: 'error',

                    message: 'La imagen no existe !!!'

                });

            }

        });

    },

    search: (req, res) => {

        // Sacar el string a buscar

        var searchString = req.params.search;

        //Find or

        article.find({ "$or": [

            {"title": { "$regex": searchString, "$options": "i"}},

            {"content": { "$regex": searchString, "$options": "i"}}

        ]})

        .sort([['date', 'descending']])

        .exec((err, articles) => {

            if(err){

                return res.status(500).send({

                    status: 'error',

                    message: 'Error en la peticion !!!'

                });

            }

            if(!articles || articles.length <= 0){

                return res.status(404).send({

                    status: 'error',

                    message: 'No hay articulos que coincidan con tu busqueda!!!'

                });

            }

            return res.status(200).send({

                status: 'success',

                articles

            });

        });

    }

}; //end controller

module.exports = controller;