```
const db = require('../config/db.config.js');
const File = db.file; //  引入表模型
const Sequelize = require('sequelize');
const Op = Sequelize.Op;
const path = require('path');
const fs = require('fs');
const childProcess = require('child_process');
//var toPdf = require("office-to-pdf")
const libre = require('libreoffice-convert');



/**
 * v3-文件转码:
 * 将office文档、txt、java、python、cpp等文本文件转成pdf，存入临时文件夹，返回给前端显示
 * 调用了Libreoffice
 *
 */
exports.transcode = (req, res) => {
  const sourceFileName = req.body.hash_name;
  const arr = sourceFileName.split('.');
  const sourceFileExt = '.'+arr[arr.length-1]
  const outputExtend = '.pdf'
  const enterPath = path.join(__dirname, `../resource/${sourceFileName}`);
  const outputPath = path.join(__dirname, `../convertPDF/$
{sourceFileName.replace(sourceFileExt,outputExtend)}`);
  // Read file
  const enter = fs.readFileSync(enterPath);
  // Convert it to pdf format with undefined filter (see Libreoffice doc about
filter)
  libre.convert(enter, outputExtend, undefined, (err, done) => {
    if (err) {
      let msg = {
        flag:0,
        msg:`Error converting file: ${err}`
      }
      res.status(500).json(msg);
      console.log(`Error converting file: ${err}`);
    }else{
      fs.writeFileSync(outputPath, done);
      let msg = {
        flag:1,
        msg:'converted!'
      }
      res.status(200).json(msg)
    }
  });

}
  //return await convertPromise();
  //let hash_name = req.body.hash_name;

  //convertPath = `${__dirname}/../convertPDF/${hash_name}`;
  //child_process.execFile('soffice',['--convert-to','pdf','note.txt'],(error,
stdout, stderr) => {
  // if(error){
  //     throw error;
  // }
  // if(stdout&&!stderr){
  //     let msg = {
  //       flag:1
  //     }
  //     res.status(200).json(msg)
  //     console.log("converted!!!")
```

```
  // }
  //console.log(`stdout: ${stdout}`);
  //console.log(`stderr: ${stderr}`);
  //})
  //console.log("converted!!!")

//  添加文件
exports.create = (req, res) => {
  //填充数据库中file表的字段
  let params = {
    file_name: req.files[0].originalname,
    hash_name: req.files[0].filename,
    upload_time: new Date(),
    type: path.parse(req.files[0].originalname).ext,
    size: req.files[0].size,
    download: 0,
    uid: req.body.uid
  };
  File.create(params)
    .then(file => {
      if (file) {
        let msg = {
          flag: 1,
          msg: 'Uploaded successfully!'
        };
        res.status(200).json(msg);
      } else {
        let msg = {
          flag: 0,
          msg: 'Failed!try again!'
        };
        res.status(500).json(msg);
      }
    })
    .catch(err => {
      res.status(500).json('Error->' + err);
    });
};
/**
 * 保存分享的文件信息到file表
 */
exports.save = (req, res) => {
  req.body.fileList.forEach((e, index) => {
    let params = {
      file_name: e.file_name,
      hash_name: e.hash_name,
      upload_time: new Date(),
      type: e.type,
      size: e.size,
      download: 0,
      uid: req.params.uid
    };
    File.create(params).then(file => {
      if (index == req.body.fileList.length - 1) {
        if (file) {
          let msg = {
            flag: 1,
            msg: `Save successfully!`
          };
          res.status(200).json(msg);
        } else {
          let msg = {
            flag: 0,
            msg: `Fail in save!`
```

```
          };
          res.status(500).json(msg);
        }
      }
    })

  });
}

//  删除文件
exports.delete = (req, res) => {
  const id = req.params.fileId;
  //  从数据库删除记录
  File.destroy({
    where: { id: id }
  })
    .then(_ => {
      //  从文件夹删除
      let fileName = req.params.fileName;
      let path = `${__dirname}/../resource/${fileName}`;
      fs.unlink(path, err => {
        if (err) {
          let msg = {
            flag: 0,
            msg: 'Failed!'
          };
          res.status(200).json(msg);
        } else {
          let msg = {
            flag: 1,
            msg: 'Done!'
          };
          res.status(200).json(msg);
        }
      });
    })
    .catch(err => {
      res.status(500).json('Error=>', err);
    });
};

//  下载文件
exports.download = (req, res) => {
  let fileId = req.params.fileId;
  //File.findByPrimary(fileId)
  File.findOne({
    where: {
      // [Op.and]: [
      //   {
      id: fileId
    }
    // ]
    // }
  })
    .then(file => {
      // file
      //   //.increment('download')
      //   .then(() => {
      file.increment('download').then(() => {
        let fileName = req.params.fileName;
        let path = `${__dirname}/../resource/${fileName}`;
        console.log(path);
        //   res.set({
        //     "Content-type":"application/octet-stream",
```

```
        //     "Content-Disposition":"attachment;filename="+file.file_name
        // });
        // console.log(file.file_name);
        res.download(path, fileName);
      });
      // let fileName = req.params.fileName;
      // let path = `${__dirname}/../resource/${fileName}`;
      // console.log(path);
      // res.download(path, fileName);
      // res.send({
      //   "status":"OK"
      // })
    })
    .catch(err => {
      res.status(500).json('Error=>', err);
      // });
    });
};

//  获取文件列表信息
exports.findAll = (req, res) => {
  File.findAll({
    where: { uid: req.body.uid }
  })
    .then(file => {
      res.status(200).json(file);
    })
    .catch(err => {
      res.status(500).json('Error=>', err);
    });
};
```