Intégration continue pour le web

Correction TP n°4 Le developpement web

src/routes/cardRouter.js

```javascript
import express from "express"
import HttpError from "../middlewares/HttpError"
import * as cardService from "../services/cardService"
const router = express.Router()

// Accessible via http://localhost:3000/cards/workers
router.get("/workers", async function(req, res) {
  try {
    const workers = await cardService.importWorkers()
    res.json(workers)
  } catch (e) {
    // Retourne une erreur 500 si la lecture à échouée
    throw new HttpError(500, "Can't read workers cards.")
  }
})

// Accessible via http://localhost:3000/cards/buildings
router.get("/buildings", async function(req, res) {
  try {
    const buildings = await cardService.importBuildings()
    res.json(buildings)
  } catch (e) {
    // Retourne une erreur 500 si la lecture à échouée
    throw new HttpError(500, "Can't read buildings cards.")
  }
})

export default router
```

src/utils/readFile.js

```
import fs from "fs"

export default function readFile(path) {
  return new Promise((resolve, reject) => {
    fs.readFile(path, "utf8", (err, data) => {
      if (err) {
        reject(err)
        return
      }
      resolve(data)
    })
  })
}
```

src/services/cardService.js

```
import readFile from "../utils/readFile"
import camelCase from "lodash/camelCase"

export function csvToJson(file) {
  const [headerLine, ...lines] = file.split("\n")
  const headers = headerLine.split(";")
  return lines.map(line => {
    const cells = line.split(";")
    const tmpObject = {}
    for (let i = 0; i < cells.length; i++) {
      tmpObject[camelCase(headers[i])] = Number.parseInt(cells[i])
    }
    return tmpObject
  })
}

export async function importBuildings() {
  const buildingsFile = await readFile(
    `${__dirname}/../ressources/buildings.csv`
  )
  return csvToJson(buildingsFile)
}

export async function importWorkers() {
  const buildingsFile = await readFile(`${__dirname}/../ressources/workers.csv`)
  return csvToJson(buildingsFile)
}
```