Intégration continue pour le web

Correction TP n° 5 Les tests

Ajout des tests unitaires

src/services/cardService.test.js

```js
import * as cardService from "./cardService"

describe("csvToJson", () => {
  test("should transform a csv to a javascript object", async () => {
    const testCsv = `
wood;stone;victory point
1;2;3
4;5;6`

    expect(cardService.csvToJson(testCsv)).toEqual([
      { wood: 1, stone: 2, victoryPoint: 3 },
      { wood: 4, stone: 5, victoryPoint: 6 }
    ])
  })
})
```

Ajout des tests d'intégration

src/routes/cardRouter.test.js

```javascript
import request from "supertest"
import app from "../app"
import readFile from "../utils/readFile"
jest.mock("../utils/readFile")

describe("/cards/workers", () => {
  test("should response the GET method", async () => {
    const testCsv = `\
wood;stone;victory point
1;2;3
4;5;6`
    readFile.mockResolvedValue(testCsv)
    const response = await request(app).get("/cards/workers")
    expect(response.statusCode).toBe(200)
    expect(response.body).toStrictEqual([
      { wood: 1, stone: 2, victoryPoint: 3 },
      { wood: 4, stone: 5, victoryPoint: 6 }
    ])
  })

  test("should return 500 if reading failed", async () => {
    readFile.mockRejectedValue(Error("Error test"))
    const response = await request(app).get("/cards/workers")
    expect(response.statusCode).toBe(500)
    expect(response.text).toEqual("Can't read workers cards.")
  })
})

describe("/cards/buildings", () => {
  test("should response the GET method", async () => {
    const testCsv = `\
wood;stone;victory point
1;2;3
4;5;6`
    readFile.mockResolvedValue(testCsv)
    const response = await request(app).get("/cards/buildings")
    expect(response.statusCode).toBe(200)
    expect(response.body).toStrictEqual([
      { wood: 1, stone: 2, victoryPoint: 3 },
      { wood: 4, stone: 5, victoryPoint: 6 }
    ])
  })

  test("should return 500 if reading failed", async () => {
    readFile.mockRejectedValue(Error("Error test"))
    const response = await request(app).get("/cards/buildings")
    expect(response.statusCode).toBe(500)
    expect(response.text).toEqual("Can't read buildings cards.")
  })
})
```