Save the Animals

Node.js

Get out of the Browser

Your challenge is to code the save-the-animals Node app

Setup git, make sure to add a *.gitignore* file to your project

The app reads a CSV file (comma separated values)

rare-animals.csv:

```
id,name,count
101, Malayan Tiger, 787
102, Mountain Gorilla, 212
103, Fin Whale, 28
```

Process the CSV line by line:

- Fetch an image for the animal (see below)
- Save the img locally to imgs folder (see below)

When this is done - generate a PDF: save-the-animals.pdf with a section for each animal

Let's put that into code:

```
utilService.loadCSV('./data/rare-animals.csv')

.then(animals => {
  // TODO: for every animal call the imgService.suggestImg,
  // get back a url and store it in the animal object
  // return a promise that resolved when ALL img urls are set.
})
.then(animalsWithImgUrls => {
  // TODO: For each animal, for each of his imgUrl, download the file
  // then return a promise that resolved when ALL imgs were downloaded.
})
.then(animalsWithImgs => {
    // TODO: Use the pdfService to build the animals PDF
})
```

Services

utilService

Use the provided service, it has the following method:

```
export const utilService = {
  readJsonFile,
  download,
  httpGet
}
```

Add the following method:

`loadCSV`(filePath)

It uses the *csv-parser npm module* and returns a promise for an array of objects

pdfService

This service should use the *pdfkit* npm module and exports the following function:
`buildAnimalsPDF`*(animals, filename='SaveTheAnimals.pdf')*

imgService (img.service.js)

This service has the following method: `suggestImg`(term) that (asynchronously) returns a single image url

Bonus: After PDF is built change to an array of 3 image urls that match the term (refactor to: `suggestImgs`).

Directions

- Use `utilService.httpGet` to get from: `https://www.istockphoto.com/search/2/image?phrase=${term}`
  Note: this URL has an auto-redirect, so in `httpGet` we use a module that follows redirects
- use *cheerio* (another node module) to turn the retrieved string to DOM
- Use a selector: `[class*="yGh0CfFS4AMLWjEE9W7v"]` to get the relevant data

Here is some code to help you with that service:

```
import { load } from 'cheerio'

function suggestImgs(term) {
   const url = `https://www.istockphoto.com/search/2/image?phrase=${term}`
   return utilService.httpGet(url).then((res) => {
      const $ = load(res)
      const topImg = Array.from($('[class*="yGh0CfFS4AMLWjEE9W7v"]'))[0]
      const imgUrl = topImg.attribs.src
      return imgUrl
   })
}
```

Bonuses

- In the PDF show several photos of the animal
- Add some content to your PDF from Wikipedia about this animal