

```

const pally = require('pally')
const fs = require('fs')
const path = require('path')
const appRoot = require('app-root-path')
const packageJson = require(appRoot + '/package.json')
const colors = require('colors/safe')
let data;

// Checks if a config file exists, and if it does, add the path to it to config variable.
try {
  if (fs.existsSync(appRoot.path + packageJson.giffgaff.allyconfig)) {
    data = require(appRoot.path + packageJson.giffgaff.allyconfig).urls
  } else {
    data = packageJson.giffgaff.accessibility
  }
} catch(err) {
  console.log("No config file found, please check the README file for help")
}

async function runPally(url) {
  try {
    const result = await pally(url, {
      hideElements: '#lightningjs-frame-usabilla_live, .cbot-chunky-buttons,
      iframe[src*="doubleclick"], #AWIN_CDT'
    });

    // print out message to user with celebratory message
    console.log(colors.yellow("Ally summary for " + url))
    if (result.issues.length == 0) {
      console.log(colors.green(`
=====
CONGRATULATIONS, YOUR SITE MEETS THE WCAG AA STANDARD!
The Design System team thanks you. As does our members!
=====
`))
    } else {
      console.log(colors.yellow("There are " + result.issues.length + " issues with this page:"))
    }

    // Output the raw result object
    for (i = 0; i < result.issues.length; i++) {
      console.log(colors.red("A11Y ERROR! " + result.issues[i].message))
      console.log(colors.brightCyan("The offending code: " + result.issues[i].context))
      console.log("=====")
    }

  } catch (error) {

    // Output an error if it occurred
    console.error(colors.red(error.message))

  }
}

// RUN THROUGH TEAM'S PROJECTS
if (data) {
  for (i = 0; i < data.length; i++) {
    runPally(data[i])
  }
}

```

```
## Usage

### Install this project.

To install giffgaff accessibility, run ``npm install @giffgaff/accessibility-testing --save`` in your
repo

### Specify your config

If you only have a few files to test, copy the below, and put your links in place of "www.my-url.com",
"www.my-other-url.com".
Please note that the accessibility tester will run through an array, so even if you have one URL you
want to test, please put it into an array

```json
"giffgaff": {
  "accessibility": [
    "www.my-url.com",
    "www.my-other-url.com"
  ]
}
```

If you have several files to test, create a config file:
i.e. config/ally.json

And copy the below into it. Then specify the URLs that you need testing inside your config file

```json
{
  "urls": [
    "www.my-url.com",
    "www.my-other-url.com"
  ]
}
```

Add the below into your package.json, updating the path inside allyconfig so that it routes to your new
config file.

```json
"giffgaff": {
  "allyconfig": [
    "/config/ally.json"
  ]
}
```

### Add scripts to your package.json

```
"scripts": {
  "test-accessibility": "node node_modules/@giffgaff/accessibility-testing/scripts/pally.js"
}
```
```