



Pruebas de rendimiento web con Lighthouse

Lighthouse para comprobación tamaño, velocidad, SEO y otras métricas.

"Cualquier optimización que no sea sobre el cuello de botella es una ilusión de mejora."

-- 🖋 Federico Toledo

- Medir antes de optimizar. [Lighthouse](#).
- Puedes usar *lighthouse* como un complemento de *Chrome*.
- Integración con *Puppeteer* **preparada y automatizada** para uso sencillo y barato y frecuente.
- Dependencia a [su librería desde npm](#)

Arrange

```
async function arrangeBrowser() {
  const chrome = await launchChrome();
  const browser = await connectToChrome(chrome);
  return { chrome, browser };
}
async function launchChrome() {
  const chrome = await chromeLauncher.launch(config);
  config.port = chrome.port;
  return chrome;
}
async function connectToChrome(chrome) {
  const resp = await util.promisify(request)(`http://localhost:${chrome.port}/json/version`);
  const { websocketDebuggerUrl } = JSON.parse(resp.body);
  const browser = await puppeteer.connect({ browserWSEndpoint: websocketDebuggerUrl });
  return browser;
}
```

Act

- Detectar y un cuello de botella y corregirlo

```
async function actGetReport(url) {
  lh_desktop_config.settings.skipAudits = null;
  lh_desktop_config.settings.onlyAudits = ['speed-index'];
  const report = await lighthouse(url, config, lh_desktop_config).then(results => {
    return results;
  });
  const audits = getSimpleArray(report.lhr.audits);
  return audits;
}

function getSimpleArray(property) {
  return Object.keys(property).map(x => ({
    id: x,
    title: property[x].title,
    score: property[x].score
  }));
}
```

Assert

Medir y comparar con un umbral.

```
module.exports = async function itShouldBeFast() {
  const inputPageUrl = 'https://www.bitademy.com';
  const { chrome, browser } = await arrangeBrowser();
  console.info(`GIVEN chrome attached : ${chrome.port}`);
  console.info(`  WHEN ${inputPageUrl} is scanned with lighthouse`);
  const actualAudits = await actGetReport(inputPageUrl);
  const actual = getSpeedIndex(actualAudits).score;
  const minimunExpected = 0.89;
  console.info(`    THEN it Should be faster than: ${minimunExpected}`);
  const failMessage = `    SpeedIndex ${actual} lower than ${minimunExpected}`;
  await afterAll({ chrome, browser });
  return assertTrue(actual > minimunExpected, failMessage);
};
```

After

Al acabar tus pruebas deberías liberar los recursos, que en este caso es simplemente desconectar y cerrar la instancia de *chrome*

```
async function afterAll({ chrome, browser }) {  
  browser.disconnect();  
  await chrome.kill();  
}
```

En [el laboratorio](#) tienes más ejemplos de lo que es capaz *Lighthouse*. Y si aún quieres más puede mirar este otro repositorio aún más completo

[AtomicBuilders/muon](#)

Empezar en el mundo de las pruebas de las aplicaciones web

Lo más fácil de probar.

Comprobar que las páginas respondan rápido **es rentable.**

Después garantizar que son correctas.



 Alberto Basalo

Online Mayo 2020