

7 Easy GitLab CI Jobs for ES6-Compatible JavaScript

Minification	Obfuscation	Testing	Coverage	Documentation	Releases	Announcements
<p>Purpose: Reduce the size of production .JS module files to decrease load time</p>	<p>Purpose: Mitigate the possibility of reverse-engineering .JS modules to protect your code</p>	<p>Purpose: Assert success of unit-level tests for an individual .JS module</p>	<p>Purpose: Assert completion of testing metrics over the entirety of .JS module source code.</p>	<p>Purpose: Automatically generate developer documentation directly from .JS modules source code comments</p>	<p>Purpose: Utilize the API to generate "Release" artifacts, including builds and notes, for specific tagged commits to .JS module development</p>	<p>Purpose: Let stakeholders know about new releases automatically so they can use the latest and greatest .JS modules</p>
<p>In Source:</p> <pre> class Person { constructor() { this.name = "Tythos"; this.age = 1337; } saySomething(msg="hey") { console.log(`\${this.name}, age \${this.age}, says "\${msg}"`); } } export default Object.assign(Person, { __metadata__: "..." }); </pre>	<p>In Source:</p> <pre> class Person { constructor() { this.name = "Tythos"; this.age = 1337; } saySomething(msg="hey") { console.log(`\${this.name}, age \${this.age}, says "\${msg}"`); } } export default Object.assign(Person, { __metadata__: "..." }); </pre>	<p>In Source:</p> <pre> class Person { constructor() { this.name = "Tythos"; this.age = 1337; } saySomething(msg="hey") { console.log(`\${this.name}, age \${this.age}, says "\${msg}"`); } } export default Object.assign(Person, { __metadata__: "..." }); </pre>	<p>In Source:</p> <pre> class Person { constructor() { this.name = "Tythos"; this.age = 1337; } saySomething(msg="hey") { console.log(`\${this.name}, age \${this.age}, says "\${msg}"`); } } export default Object.assign(Person, { __metadata__: "..." }); </pre>	<p>In Source:</p> <pre> class Person { constructor() { this.name = "Tythos"; this.age = 1337; } saySomething(msg="hey") { console.log(`\${this.name}, age \${this.age}, says "\${msg}"`); } } export default Object.assign(Person, { __metadata__: "..." }); </pre>	<p>In Source: [Body]</p> <pre> class Person { constructor() { this.name = "Tythos"; this.age = 1337; } saySomething(msg="hey") { console.log(`\${this.name}, age \${this.age}, says "\${msg}"`); } } export default Object.assign(Person, { __metadata__: "..." }); </pre>	
<p>In CI:</p> <pre> minification: script: - npm install -g terser - terser --compress --output Person.min.mjs Person.mjs artifacts: when: always paths: - Person.min.mjs expire_in: never </pre>	<p>In CI:</p> <pre> obfuscation: script: - npm install -g terser - terser --mangle --output Person.min.mjs Person.mjs artifacts: when: always paths: - Person.min.mjs expire_in: never </pre>	<p>In CI:</p> <pre> testing: script: - npm install -g jasmine - npm install jasmine-reporters - jasmine --config=.jasmine-conf.json artifacts: when: always expire_in: never paths: - junitresults.xml reports: - junit: - junitresults.xml </pre>	<p>In CI:</p> <pre> coverage: script: - npm install -g nyc - nyc --reporter=cobertura --include=jasmine-test.js,jasmine-config,jasmine.conf.json </pre>	<p>In CI:</p> <pre> documentation: script: - npm install -g esdoc - esdoc --main=src --output=docs artifacts: when: always paths: - docs expire_in: never </pre>	<p>In CI:</p> <pre> releases: script: - npm install -g @gitlab-ci-examples/releases - releases --tag=v0.1.0 artifacts: when: always paths: - dist expire_in: never </pre>	<p>In CI:</p> <pre> announcements: script: - npm install -g @gitlab-ci-examples/announcements - announcements --tag=v0.1.0 artifacts: when: always paths: - dist expire_in: never </pre>
<p>Result:</p> <pre> class Person(constructor){this.name="Tythos",this... </pre>	<p>Result:</p> <pre> class Person(constructor){this.name="Tythos",this... </pre>	<p>Result:</p> <pre> class Person(constructor){this.name="Tythos",this... </pre>	<p>Result:</p> <pre> class Person(constructor){this.name="Tythos",this... </pre>	<p>Result:</p> <pre> class Person(constructor){this.name="Tythos",this... </pre>	<p>Result:</p>	<p>Result:</p>