

@jridgewell/gen-mapping

Generate source maps

gen-mapping allows you to generate a source map during transpilation or minification. With a source map, you're able to trace the original location in the source file, either in Chrome's DevTools or using a library like [@jridgewell/trace-mapping](#).

You may already be familiar with the [source-map](#) package's SourceMapGenerator. This provides the same addMapping and setSourceContent API.

Installation

```
npm install @jridgewell/gen-mapping
```

Usage

```
import { GenMapping, addMapping,
         setSourceContent, toEncodedMap,
         toDecodedMap } from
         '@jridgewell/gen-mapping';
```

```
const map = new GenMapping({  
    sourceRoot: 'https://example.com/',  
    file: 'output.js',  
    setSourceContent(map, 'input.js',  
        `function foo() {}`),  
    addMapping(map, {  
        // Lines start at line 1, columns at  
        column 0,  
        generated: { line: 1, column: 0 },  
        source: 'input.js',  
        original: { line: 1, column: 0 },  
        generated: { line: 1, column: 0 },  
        addMapping(map, {  
            source: 'input.js',  
            original: { line: 1, column: 9 },  
            generated: { line: 1, column: 9 },  
            name: 'foo',  
            original: { line: 1, column: 9 },  
            generated: { line: 1, column: 9 },  
            addMapping(map, {  
                source: 'input.js',  
                original: { line: 1, column: 9 },  
                generated: { line: 3, column: 9 },  
                file: 'output.js',  
                names: ['foo'],  
                sourceRoot: 'https://example.com/',  
                sources: ['input.js'],  
                sourceContent: [function foo() {}],  
                mapings: [ [0, 0, 0, 0], [9, 0, 0, 9, 0] ]  
            })  
        })  
    })  
});
```

```
assert.deepEqual(toEncodedMap(map), {  
  version: 3,  
  file: 'output.js',  
  names: ['foo'],  
  sourceRoot: 'https://example.com/',  
  sources: ['input.js'],  
  sourcesContent: ['function foo() {}'],  
  mappings: 'AAAA,SAASA',  
})};
```

Smaller Sourcemaps

Not everything needs to be added to a sourcemap, and needless markings can cause significantly larger file sizes. `gen-mapping` exposes `maybeAddSegment`/`maybeAddMapping` APIs that will intelligently determine if this marking adds useful information. If not, the marking will be skipped.

```
import { maybeAddMapping } from  
  '@jridgewell/gen-mapping';  
  
const map = new GenMapping();  
  
// Adding a sourceless marking at the  
// beginning of a line isn't  
// useful.  
maybeAddMapping(map, {  
  generated: { line: 1, column: 0 },  
});
```

source-map-0.6.1	bytes	source-map-0.8.0	bytes	smallest memory usage is gen-mapping:
2885435	2874336	2874336	2874336	addSegment
2885435	2874336	2874336	2874336	adding speed:
2885435	2874336	2874336	2874336	gen-mapping:
456	456	456	456	ops/sec 13% (97 runs sampled)
456	456	456	456	source-map-0.6.1: addMapping x 4,456
456	456	456	456	gen-mapping:
456	456	456	456	ops/sec 15% (100 runs sampled)
456	456	456	456	ops/sec 15%
456	456	456	456	ops/sec 15%
456	456	456	456	Fastest is gen-mapping:
456	456	456	456	addSegment
618	618	618	618	source-map-0.6.1: addMapping x 1,618
618	618	618	618	ops/sec 24% (97 runs sampled)
618	618	618	618	source-map-0.6.1: addMapping x 1,618
618	618	618	618	ops/sec 12% (99 runs sampled)
618	618	618	618	source-map-0.6.1: addMapping x 1,618
618	618	618	618	ops/sec 12%
631	631	631	631	source-map-0.6.1: addMapping x 1,631
631	631	631	631	ops/sec 12% (100 runs sampled)
631	631	631	631	Fastest is gen-mapping:
631	631	631	631	addSegment
695	695	695	695	gen-mapping:
695	695	695	695	ops/sec 0.07% (99 runs
695	695	695	695	379,107,695 ops/sec 0.0:
695	695	695	695	decoded output x 5,421
695	695	695	695	gen-mapping:
695	695	695	695	ops/sec 60.60% (89 runs sampled)
695	695	695	695	source-map-0.6.1: encoded output x 2,113
695	695	695	695	ops/sec 60.10% (100 runs sampled)
695	695	695	695	source-map-0.6.1: encoded output x 2,126
695	695	695	695	ops/sec 60.39% (98 runs sampled)
695	695	695	695	Fastest is gen-mapping:
695	695	695	695	decoded

```
gen-mapping:      addMapping x 13,013  
ops/sec ±0.11% (101 runs sampled)  
source-map-js:   addMapping x 4,564  
ops/sec ±0.21% (98 runs sampled)  
source-map-0.6.1: addMapping x 4,562  
ops/sec ±0.11% (99 runs sampled)  
source-map-0.8.0: addMapping x 4,593  
ops/sec ±0.11% (100 runs sampled)  
Fastest is gen-mapping:      addSegment
```

Generate speed:

```
gen-mapping:      decoded output x  
379,864,020 ops/sec ±0.23% (93 runs  
sampled)  
gen-mapping:      encoded output x  
14,368 ops/sec ±4.07% (82 runs sampled)  
source-map-js:   encoded output x 5,261  
ops/sec ±0.21% (99 runs sampled)  
source-map-0.6.1: encoded output x 5,124  
ops/sec ±0.58% (99 runs sampled)  
source-map-0.8.0: encoded output x 5,434  
ops/sec ±0.33% (96 runs sampled)  
Fastest is gen-mapping:      decoded  
output
```

```
react.js.map  
Memory Usage:  
gen-mapping: addSegment      975096  
bytes  
gen-mapping: addMapping      1102981  
bytes  
source-map-js      2918836  
bytes
```

Memory Usage:

gen-mapping: addSegment	5852872
bytes	
gen-mapping: addMapping	7716042
bytes	
source-map-js	6143250
bytes	
source-map-0.6.1	6124102
bytes	
source-map-0.8.0	6121173
bytes	

Smallest memory usage is gen-mapping:
addSegment

Adding speed:

gen-mapping: addSegment	x 441 ops/ sec ±2.07% (90 runs sampled)
gen-mapping: addMapping	x 350 ops/ sec ±2.40% (86 runs sampled)
source-map-js: addMapping	x 169 ops/ sec ±2.42% (80 runs sampled)
source-map-0.6.1: addMapping	x 167 ops/ sec ±2.56% (80 runs sampled)
source-map-0.8.0: addMapping	x 168 ops/ sec ±2.52% (80 runs sampled)

Fastest is gen-mapping: addSegment

Generate speed:

gen-mapping: decoded output	x 150,824,370 ops/sec ±0.07% (102 runs sampled)
gen-mapping: encoded output	x 663 ops/sec ±0.22% (98 runs sampled)
source-map-js: encoded output	x 197 ops/sec ±0.45% (84 runs sampled)
source-map-0.6.1: encoded output	x 198 ops/sec ±0.33% (85 runs sampled)

