Running Prism.js from a web worker

I've spent a lot of time trying to figure out if Prism.js will run inside a worker to highlight syntax on the web page.

The answer is sort of.

Given the way that Prism uses web workers, it still would not work in AMP pages. You still need to call this snippet of Javascript in the page that needs syntax highlighting.

```
Prism.highlightAll(true, function() {
  console.log('Syntax highlight completed');
});
```

Furthermore, the async functionality of Prism.js is off by default. The Prism.js FAQ explains why:

> [...]]Furthermore, using Web Workers is actually slower than synchronously highlighting, due to the overhead of creating and terminating the Worker. It just appears faster in these cases because it doesn't block the main thread.[...] Lastly, Web Workers cannot interact with the DOM and most other APIs (e.g. the console), so they are notoriously hard to debug.

So we can't run Prism inside a worker, can we?

No, we cannot. As far as I can tell, Prism will build a worker to execute the highlight portion of the work and then return that work to the main thread.

The relevant portion of the code is in prism-core.js, lines 250 to 265.

```
if (async && _self.Worker) {
  const worker = new Worker(_.filename);
```

```
  worker.onmessage = function(evt) {
    insertHighlightedCode(evt.data);
  };

  worker.postMessage(JSON.stringify({
    language: env.language,
    code: env.code,
    immediateClose: true
  }));
}
else {
  insertHighlightedCode(_.highlight(env.code,
    env.grammar, env.language));
}
```

If I'm understanding the code correctly if we've called Prism with `highlightAll` method with true as the first parameter, Prism will create a worker and delegates the highlighting to the worker that returns the highlighted code when the task completes.

I don't know enough about workers to understand if we can put the `Prism.highlightAll` code in a worker and let it handle it in a sub-worker and whether that would be too much of a performance hit to be worth the effort.