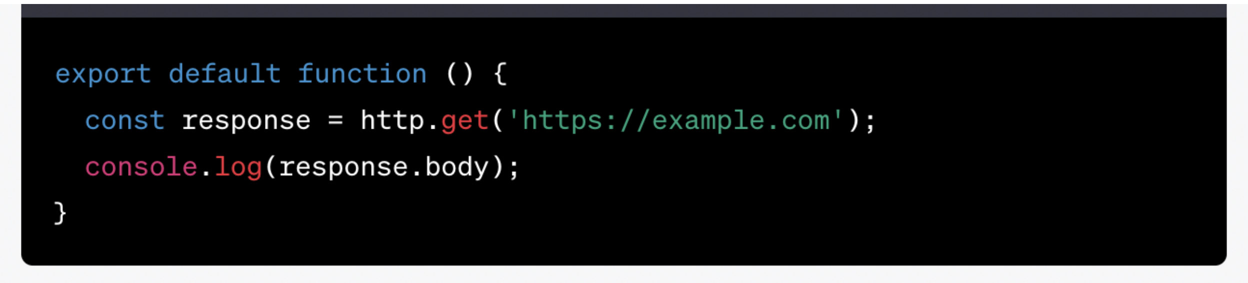
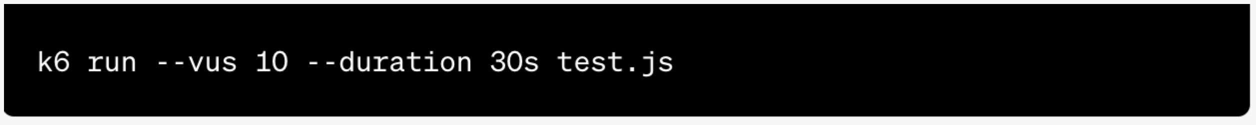
1. Installation:
    - Install k6 using npm: `npm install -g k6`
    - Install k6 using Homebrew (macOS): `brew install k6`
2. Writing a test script:
    - Create a new JavaScript file with a `.js` extension (e.g., `test.js`).
    - Import the k6 module: `import http from 'k6/http';`
    - Write your test script using the k6 API. For example:



```
export default function () {
  const response = http.get('https://example.com');
  console.log(response.body);
}
```

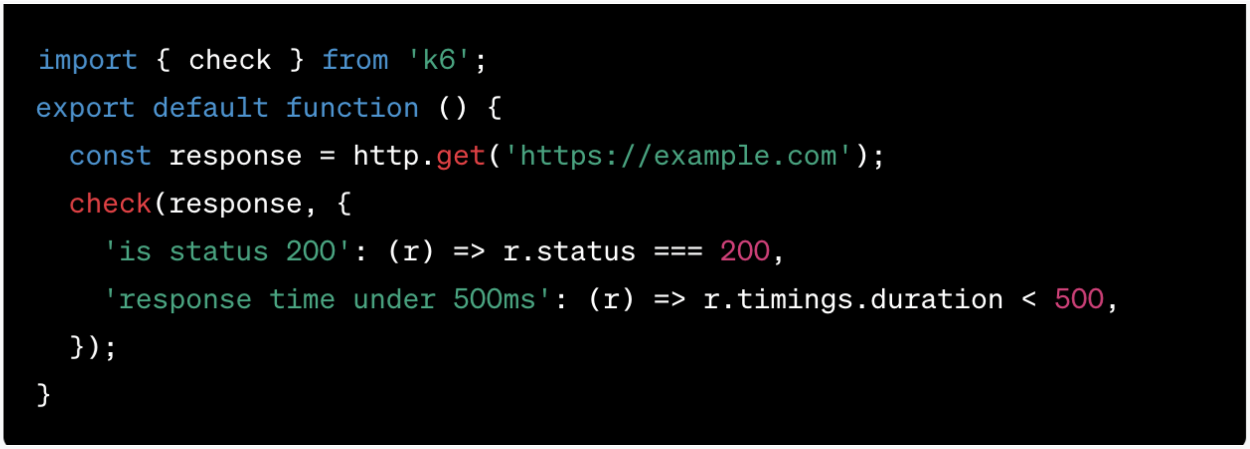
3. Running a test:
    - Run a test script using the command: `k6 run test.js`
    - You can specify various options, such as the number of virtual users and duration:



```
k6 run --vus 10 --duration 30s test.js
```

4. HTTP Requests:
    - Sending a GET request: `http.get(url, [params], [options]);`
    - Sending a POST request: `http.post(url, body, [params], [options]);`
    - Sending other HTTP methods (PUT, DELETE, etc.): `http.<method>(url, [body], [params], [options]);`
5. Virtual Users (VUs):
    - Specify the number of VUs: `--vus <number>`
    - Ramp up VUs over time: `--vus <start>:<stop>`

6. Test duration:
    - Specify the test duration: `--duration <duration>`
    - Use time units like `s` (seconds), `m` (minutes), `h` (hours).
7. Thresholds and Checks:
    - Define thresholds for response times, requests per second, and other metrics using the `check` API.
    - Example:



```
import { check } from 'k6';
export default function () {
  const response = http.get('https://example.com');
  check(response, {
    'is status 200': (r) => r.status === 200,
    'response time under 500ms': (r) => r.timings.duration < 500,
  });
}
```

8. Output and Reporting:
    - Generate an HTML report: `k6 run --out html=report.html test.js`
    - Generate a JSON report: `k6 run --out json=report.json test.js`
    - Integrate with various third-party systems like Grafana, InfluxDB, and Datadog for advanced reporting and monitoring.