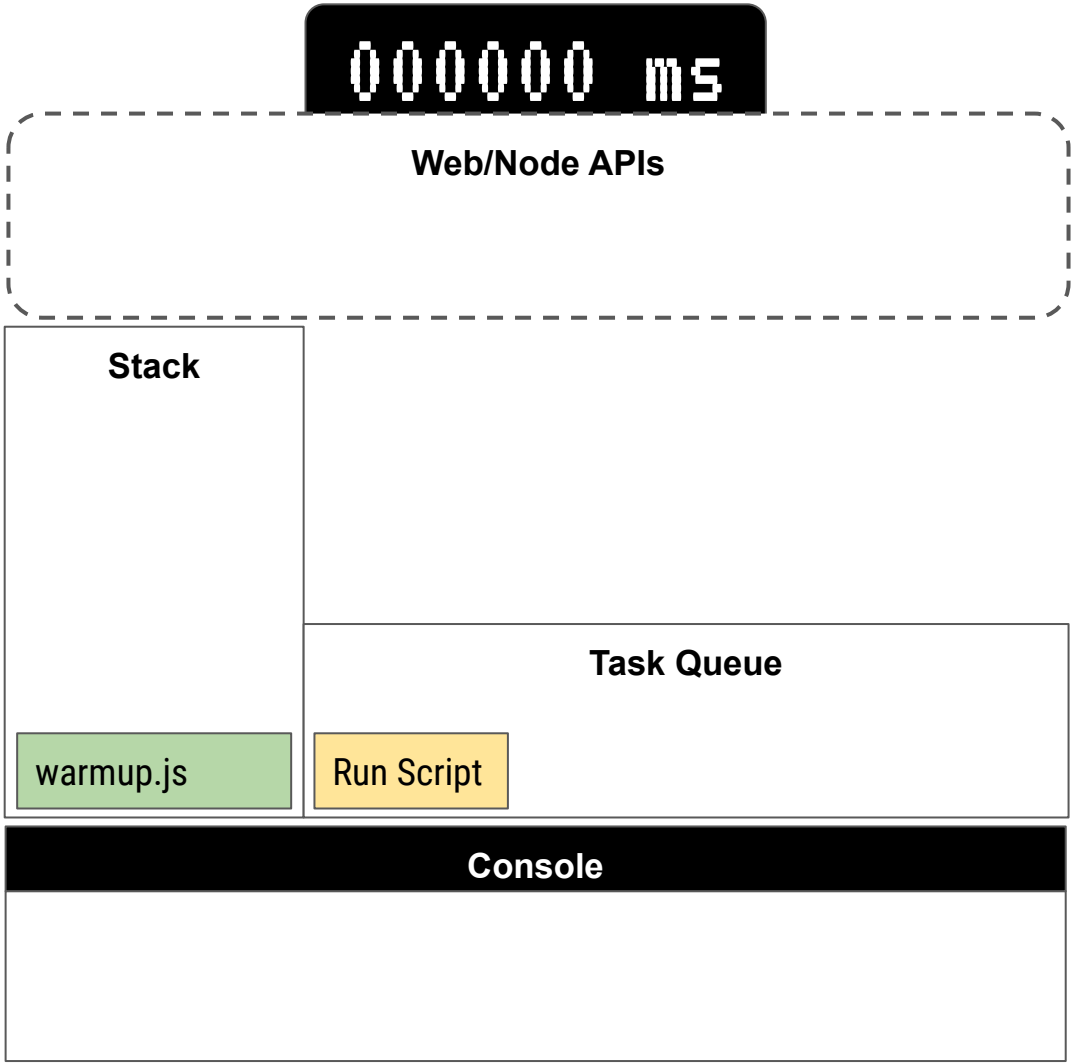
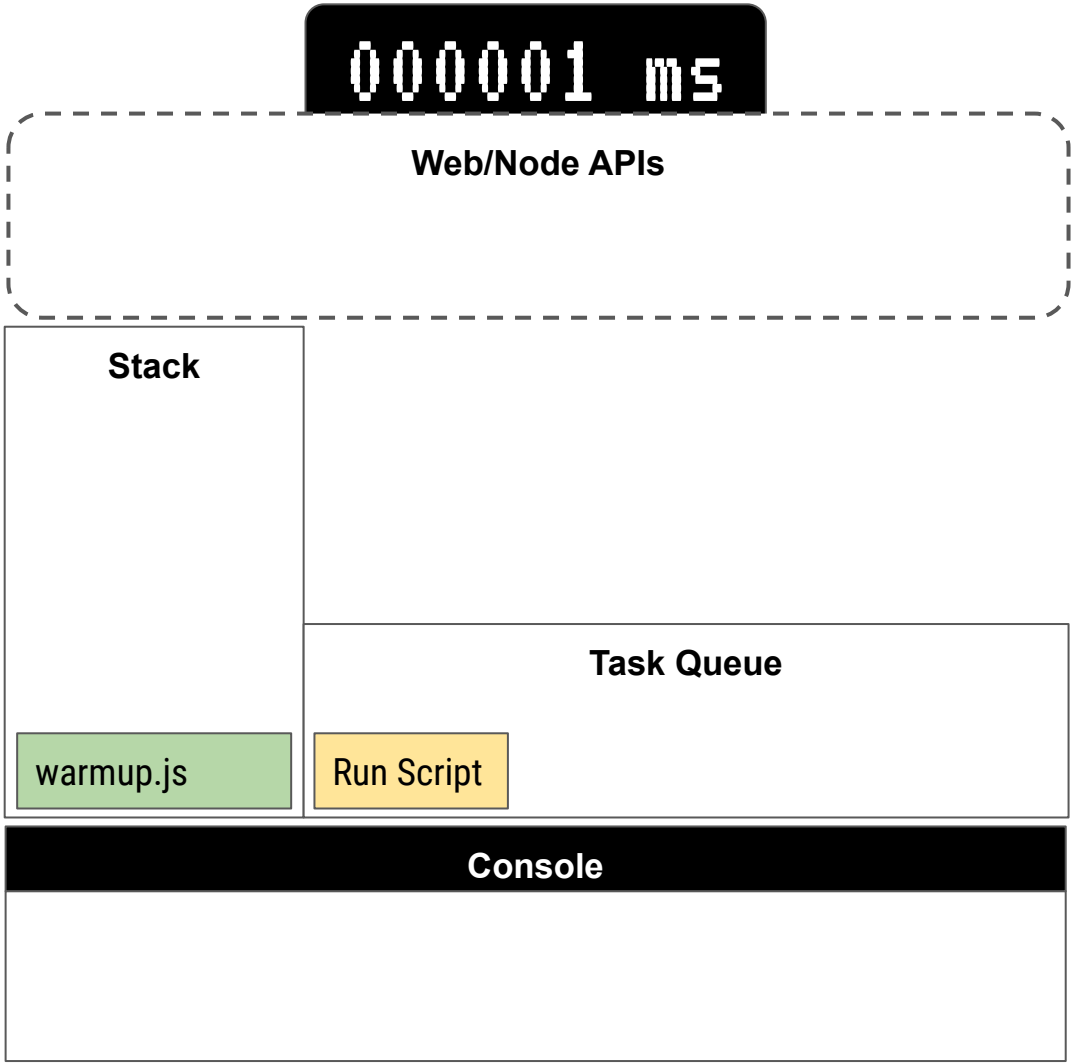
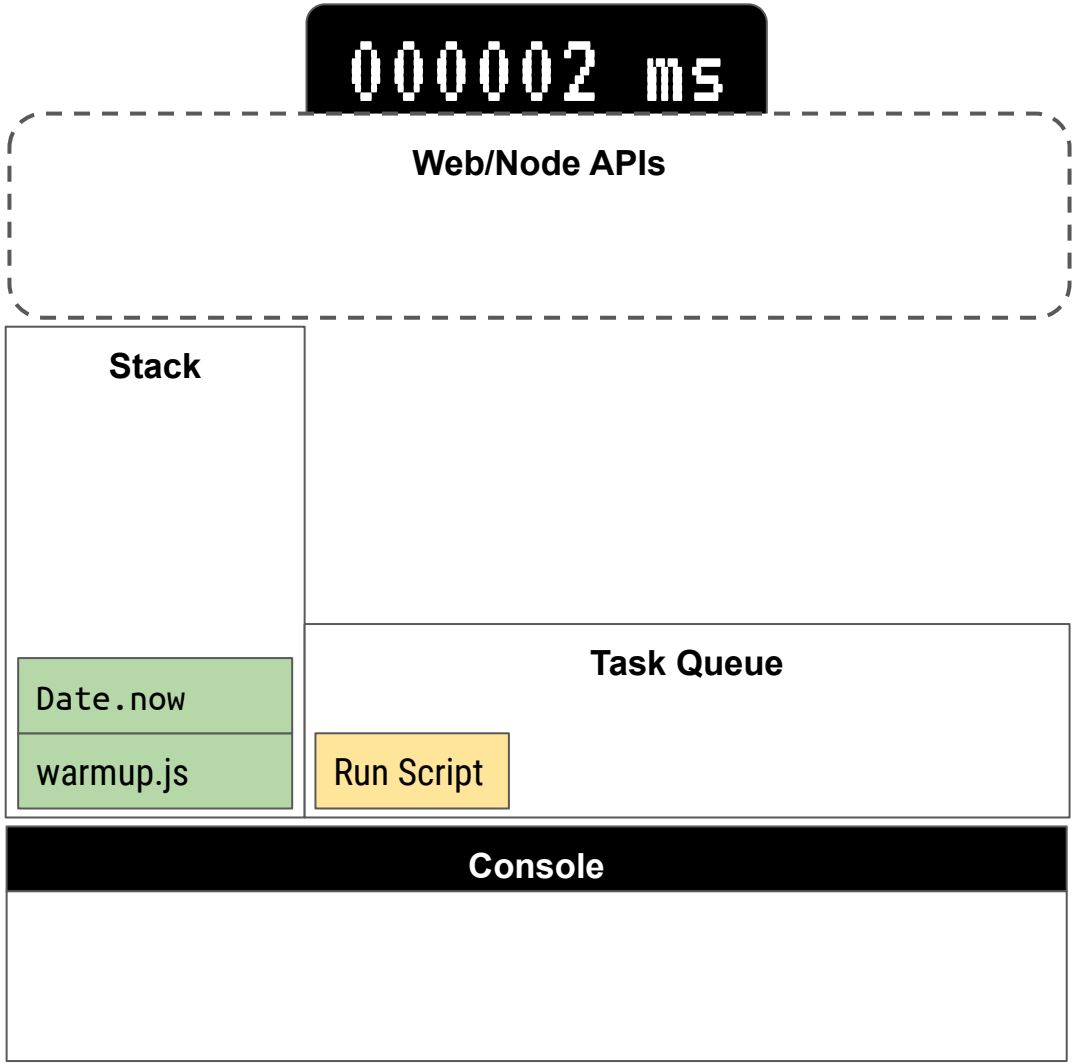
```
01: function waitSync(delayMs) {
02:   const waitStart = Date.now();
03:   while (Date.now() - waitStart < delayMs) {}
04: }
05:
06: const start = Date.now();
07: function elapsedTimeLog(...args) {
08:   const elapsedMs = Date.now() - start;
09:   const sec = Math.floor(elapsedMs / 1000);
10:   console.log(sec, ...args);
11: }
12:
13: function handlerA() {
14:   waitSync(6000);
15:   elapsedTimeLog("A");
16: }
17:
18: function handlerB() {
19:   elapsedTimeLog("B");
20: }
21:
22: setTimeout(handlerA, 1000);
23: setTimeout(handlerB, 2000);
24:
25: elapsedTimeLog("C");
```



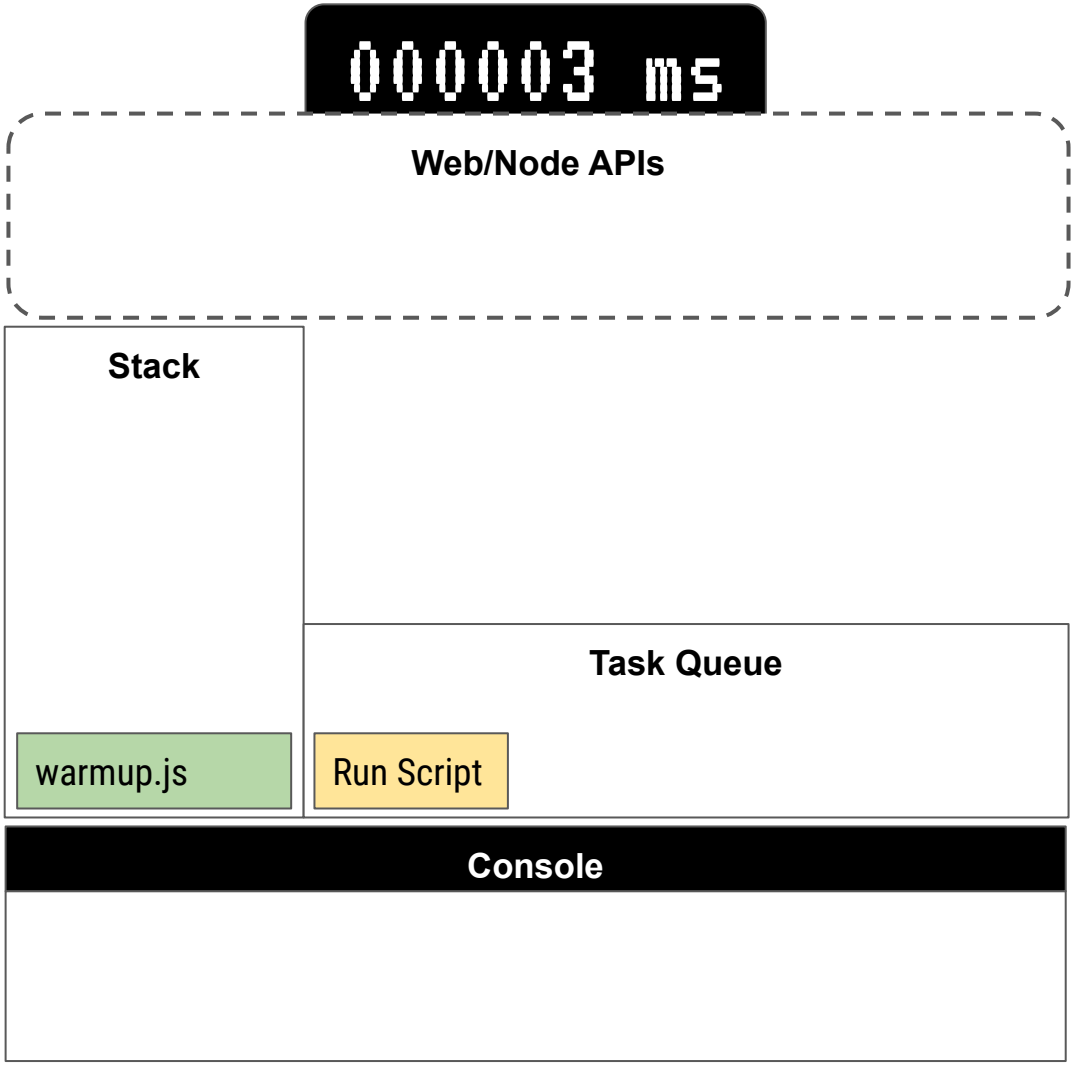
```
01: function waitSync(delayMs) {
02:   const waitStart = Date.now();
03:   while (Date.now() - waitStart < delayMs) {}
04: }
05:
06: const start = Date.now();
07: function elapsedTimeLog(...args) {
08:   const elapsedMs = Date.now() - start;
09:   const sec = Math.floor(elapsedMs / 1000);
10:   console.log(sec, ...args);
11: }
12:
13: function handlerA() {
14:   waitSync(6000);
15:   elapsedTimeLog("A");
16: }
17:
18: function handlerB() {
19:   elapsedTimeLog("B");
20: }
21:
22: setTimeout(handlerA, 1000);
23: setTimeout(handlerB, 2000);
24:
25: elapsedTimeLog("C");
```



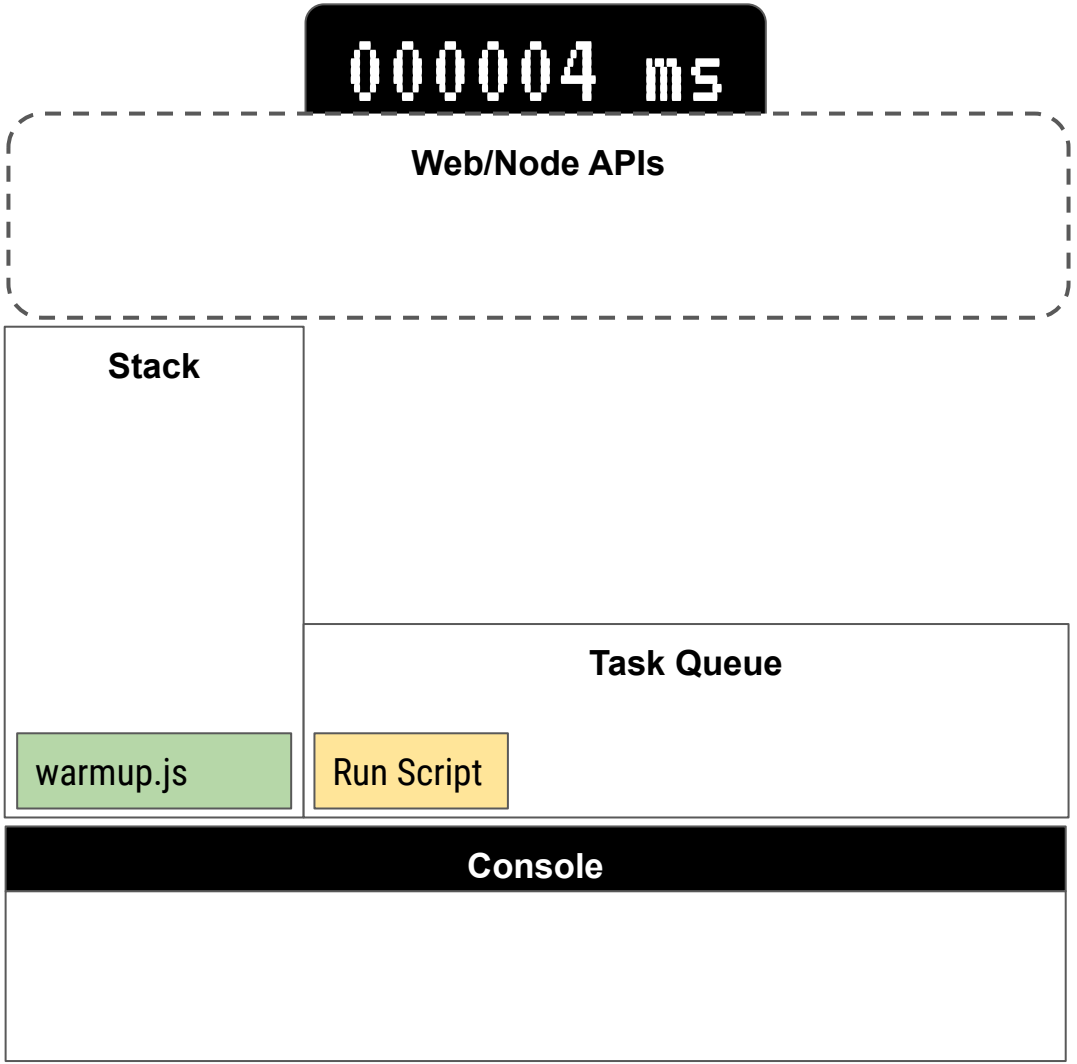
```
01: function waitSync(delayMs) {
02:   const waitStart = Date.now();
03:   while (Date.now() - waitStart < delayMs) {}
04: }
05:
06: const start = Date.now();
07: function elapsedTimeLog(...args) {
08:   const elapsedMs = Date.now() - start;
09:   const sec = Math.floor(elapsedMs / 1000);
10:   console.log(sec, ...args);
11: }
12:
13: function handlerA() {
14:   waitSync(6000);
15:   elapsedTimeLog("A");
16: }
17:
18: function handlerB() {
19:   elapsedTimeLog("B");
20: }
21:
22: setTimeout(handlerA, 1000);
23: setTimeout(handlerB, 2000);
24:
25: elapsedTimeLog("C");
```



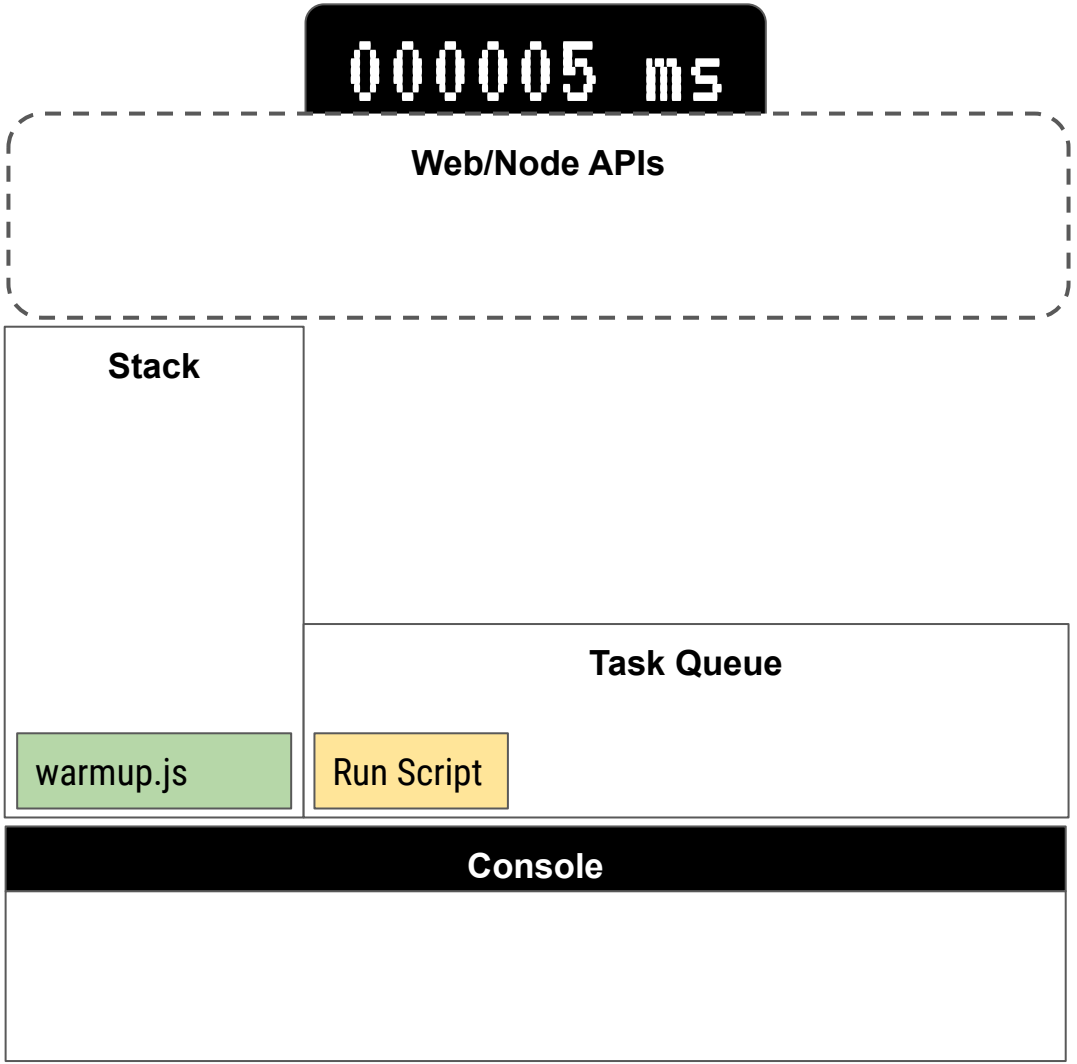
```
01: function waitSync(delayMs) {
02:   const waitStart = Date.now();
03:   while (Date.now() - waitStart < delayMs) {}
04: }
05:
06: const start = Date.now();
07: function elapsedTimeLog(...args) {
08:   const elapsedMs = Date.now() - start;
09:   const sec = Math.floor(elapsedMs / 1000);
10:   console.log(sec, ...args);
11: }
12:
13: function handlerA() {
14:   waitSync(6000);
15:   elapsedTimeLog("A");
16: }
17:
18: function handlerB() {
19:   elapsedTimeLog("B");
20: }
21:
22: setTimeout(handlerA, 1000);
23: setTimeout(handlerB, 2000);
24:
25: elapsedTimeLog("C");
```



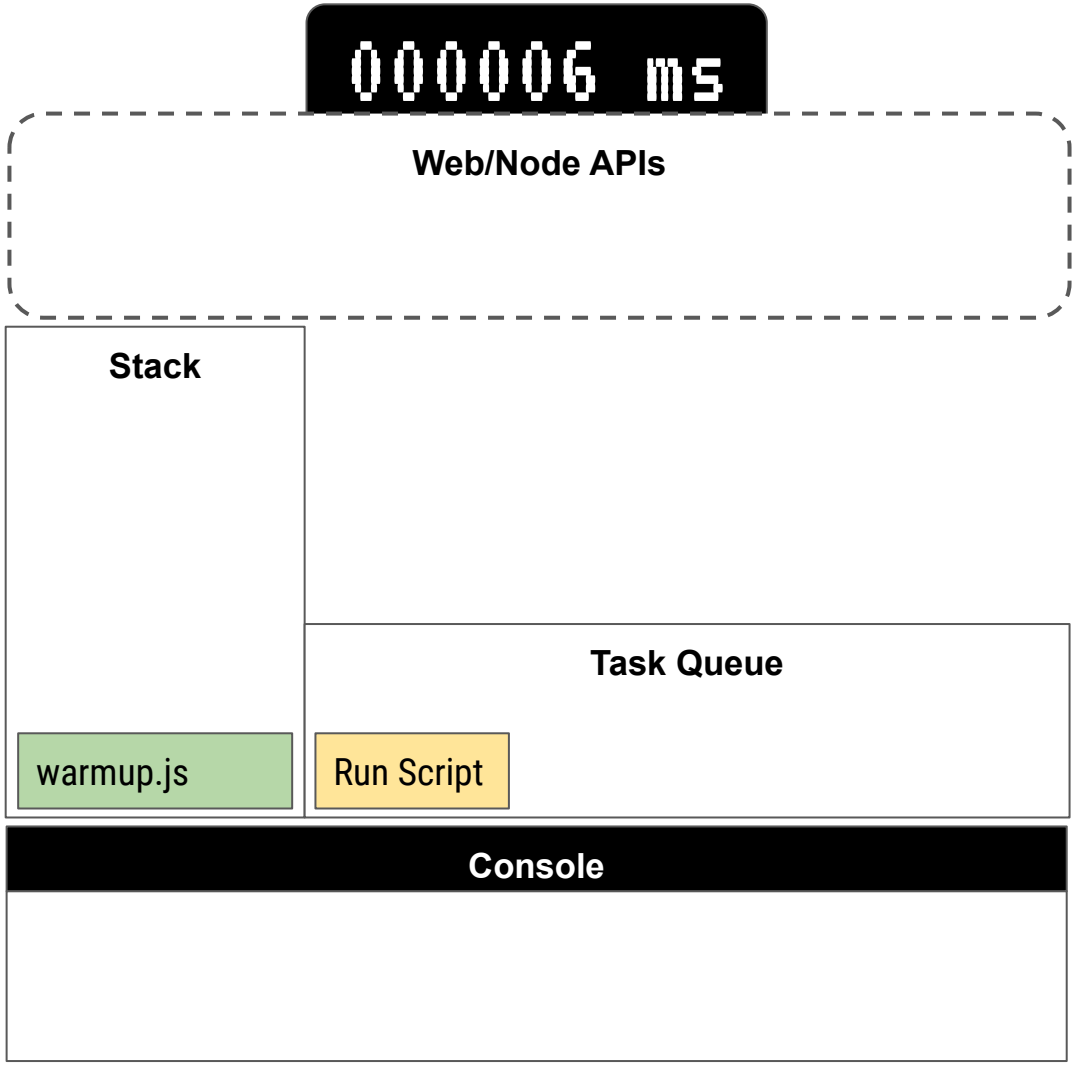
```
01: function waitSync(delayMs) {
02:   const waitStart = Date.now();
03:   while (Date.now() - waitStart < delayMs) {}
04: }
05:
06: const start = Date.now();
07: function elapsedTimeLog(...args) {
08:   const elapsedMs = Date.now() - start;
09:   const sec = Math.floor(elapsedMs / 1000);
10:   console.log(sec, ...args);
11: }
12:
13: function handlerA() {
14:   waitSync(6000);
15:   elapsedTimeLog("A");
16: }
17:
18: function handlerB() {
19:   elapsedTimeLog("B");
20: }
21:
22: setTimeout(handlerA, 1000);
23: setTimeout(handlerB, 2000);
24:
25: elapsedTimeLog("C");
```



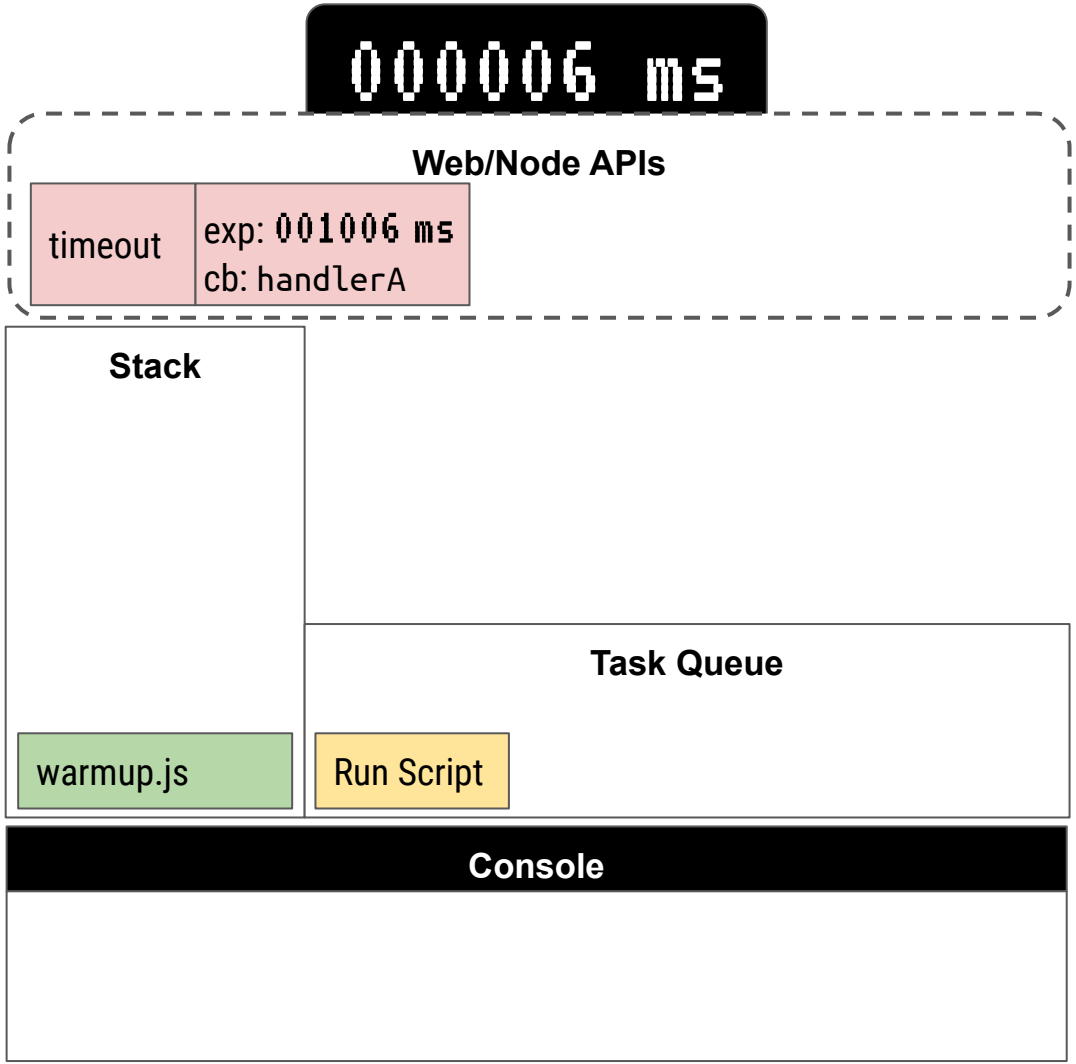
```
01: function waitSync(delayMs) {
02:   const waitStart = Date.now();
03:   while (Date.now() - waitStart < delayMs) {}
04: }
05:
06: const start = Date.now();
07: function elapsedTimeLog(...args) {
08:   const elapsedMs = Date.now() - start;
09:   const sec = Math.floor(elapsedMs / 1000);
10:   console.log(sec, ...args);
11: }
12:
13: function handlerA() {
14:   waitSync(6000);
15:   elapsedTimeLog("A");
16: }
17:
18: function handlerB() {
19:   elapsedTimeLog("B");
20: }
21:
22: setTimeout(handlerA, 1000);
23: setTimeout(handlerB, 2000);
24:
25: elapsedTimeLog("C");
```



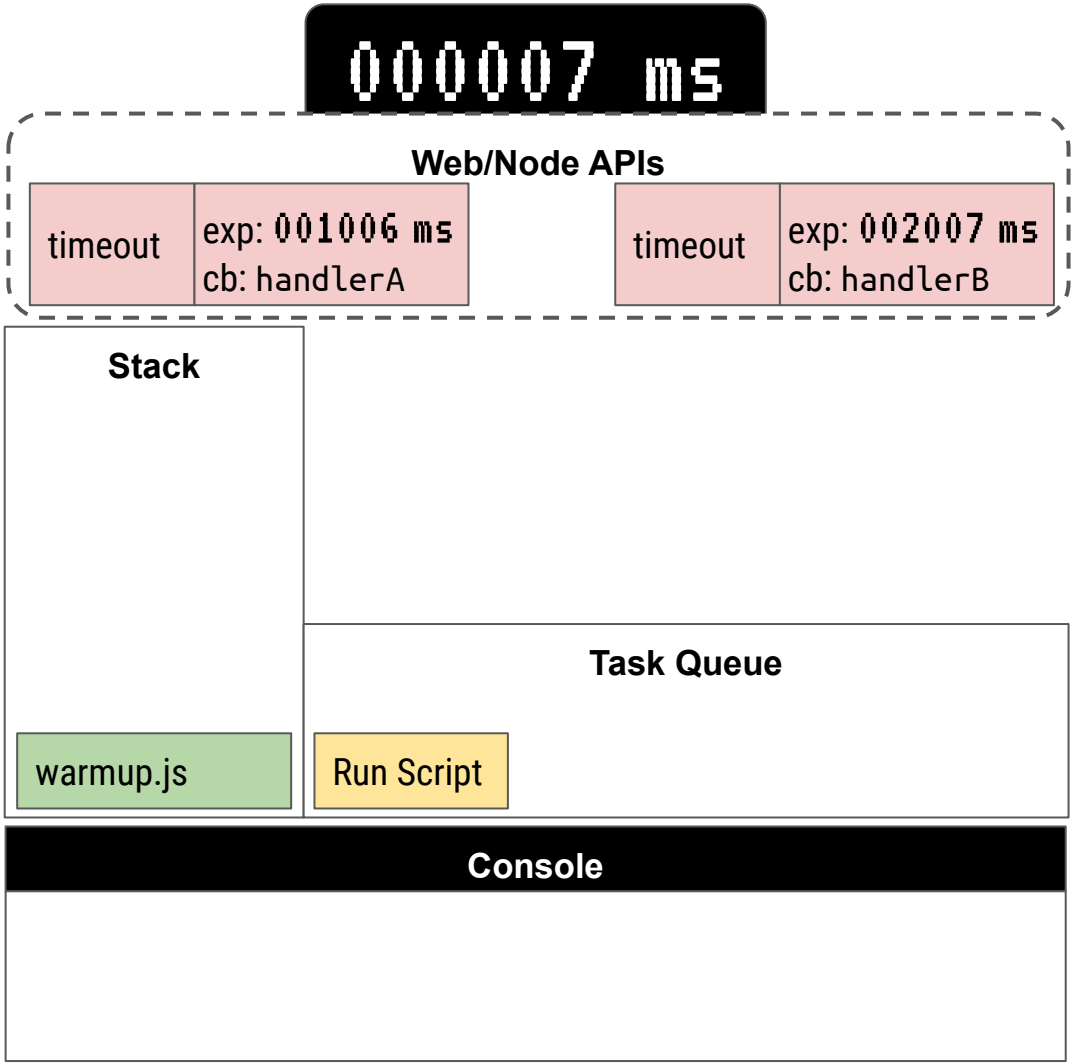
```
01: function waitSync(delayMs) {
02:   const waitStart = Date.now();
03:   while (Date.now() - waitStart < delayMs) {}
04: }
05:
06: const start = Date.now();
07: function elapsedTimeLog(...args) {
08:   const elapsedMs = Date.now() - start;
09:   const sec = Math.floor(elapsedMs / 1000);
10:   console.log(sec, ...args);
11: }
12:
13: function handlerA() {
14:   waitSync(6000);
15:   elapsedTimeLog("A");
16: }
17:
18: function handlerB() {
19:   elapsedTimeLog("B");
20: }
21:
22: setTimeout(handlerA, 1000);
23: setTimeout(handlerB, 2000);
24:
25: elapsedTimeLog("C");
```



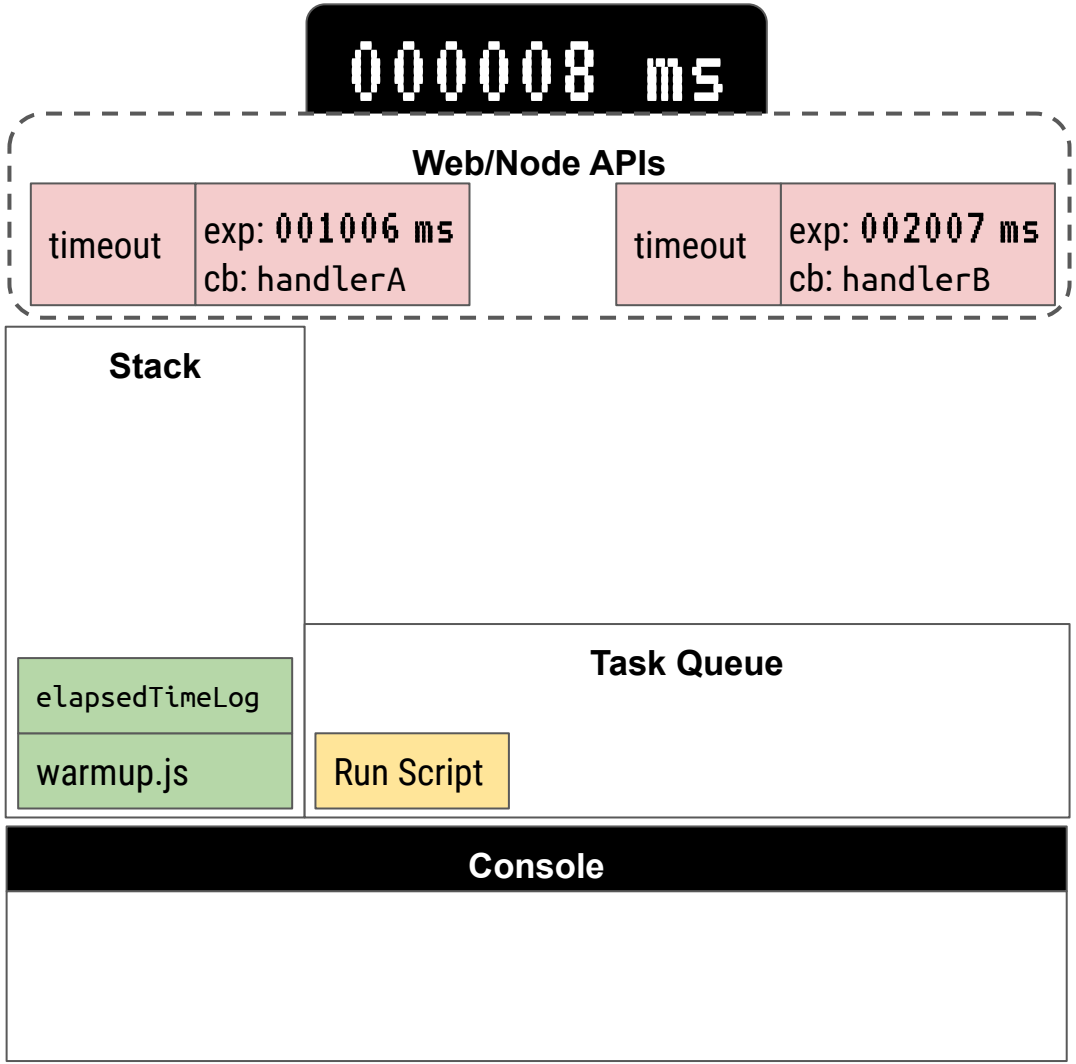
```
01: function waitSync(delayMs) {
02:   const waitStart = Date.now();
03:   while (Date.now() - waitStart < delayMs) {}
04: }
05:
06: const start = Date.now();
07: function elapsedTimeLog(...args) {
08:   const elapsedMs = Date.now() - start;
09:   const sec = Math.floor(elapsedMs / 1000);
10:   console.log(sec, ...args);
11: }
12:
13: function handlerA() {
14:   waitSync(6000);
15:   elapsedTimeLog("A");
16: }
17:
18: function handlerB() {
19:   elapsedTimeLog("B");
20: }
21:
22: setTimeout(handlerA, 1000);
23: setTimeout(handlerB, 2000);
24:
25: elapsedTimeLog("C");
```



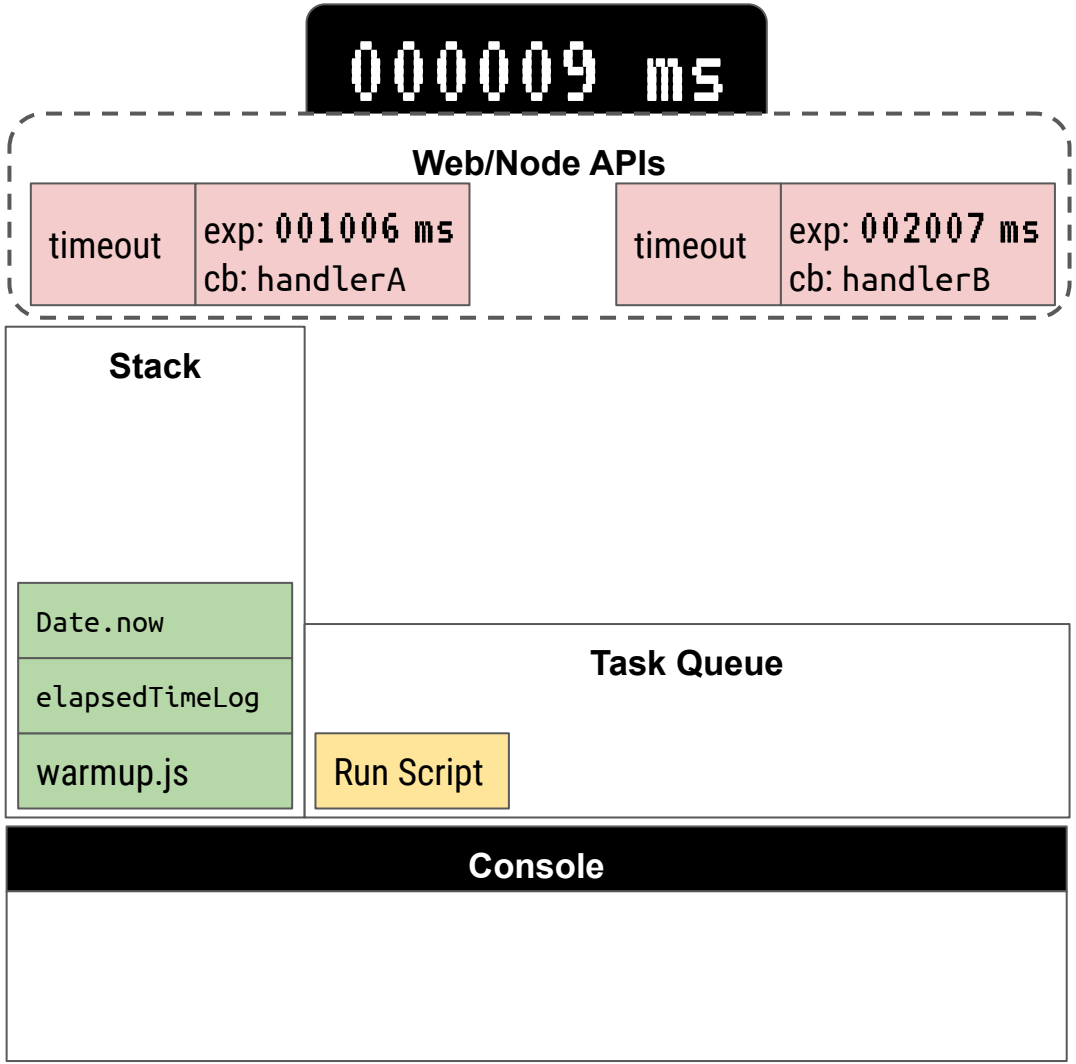
```
01: function waitSync(delayMs) {
02:   const waitStart = Date.now();
03:   while (Date.now() - waitStart < delayMs) {}
04: }
05:
06: const start = Date.now();
07: function elapsedTimeLog(...args) {
08:   const elapsedMs = Date.now() - start;
09:   const sec = Math.floor(elapsedMs / 1000);
10:   console.log(sec, ...args);
11: }
12:
13: function handlerA() {
14:   waitSync(6000);
15:   elapsedTimeLog("A");
16: }
17:
18: function handlerB() {
19:   elapsedTimeLog("B");
20: }
21:
22: setTimeout(handlerA, 1000);
23: setTimeout(handlerB, 2000);
24:
25: elapsedTimeLog("C");
```



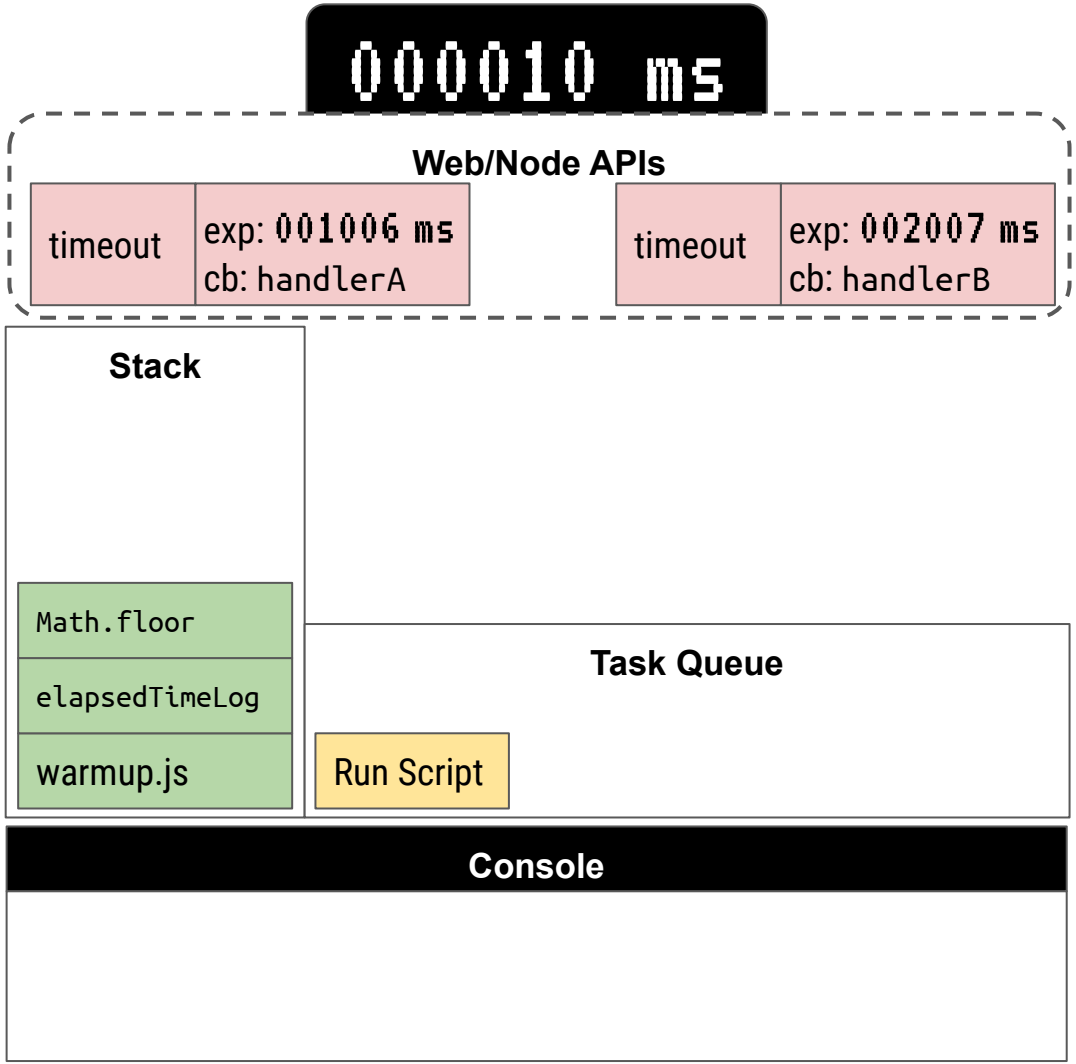
```
01: function waitSync(delayMs) {
02:   const waitStart = Date.now();
03:   while (Date.now() - waitStart < delayMs) {}
04: }
05:
06: const start = Date.now();
07: function elapsedTimeLog(...args) {
08:   const elapsedMs = Date.now() - start;
09:   const sec = Math.floor(elapsedMs / 1000);
10:   console.log(sec, ...args);
11: }
12:
13: function handlerA() {
14:   waitSync(6000);
15:   elapsedTimeLog("A");
16: }
17:
18: function handlerB() {
19:   elapsedTimeLog("B");
20: }
21:
22: setTimeout(handlerA, 1000);
23: setTimeout(handlerB, 2000);
24:
25: elapsedTimeLog("C");
```



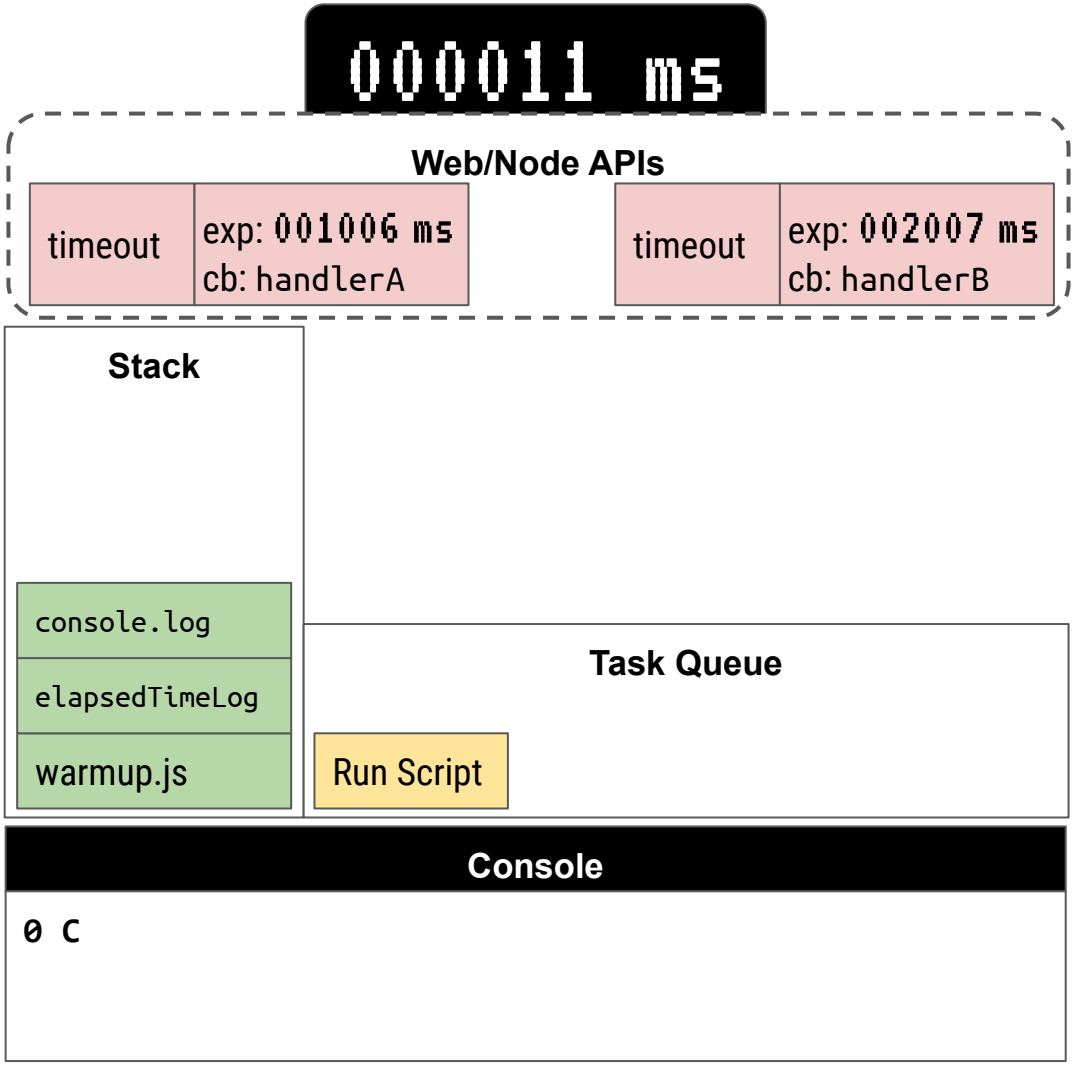
```
01: function waitSync(delayMs) {
02:   const waitStart = Date.now();
03:   while (Date.now() - waitStart < delayMs) {}
04: }
05:
06: const start = Date.now();
07: function elapsedTimeLog(...args) {
08:   const elapsedMs = Date.now() - start;
09:   const sec = Math.floor(elapsedMs / 1000);
10:   console.log(sec, ...args);
11: }
12:
13: function handlerA() {
14:   waitSync(6000);
15:   elapsedTimeLog("A");
16: }
17:
18: function handlerB() {
19:   elapsedTimeLog("B");
20: }
21:
22: setTimeout(handlerA, 1000);
23: setTimeout(handlerB, 2000);
24:
25: elapsedTimeLog("C");
```



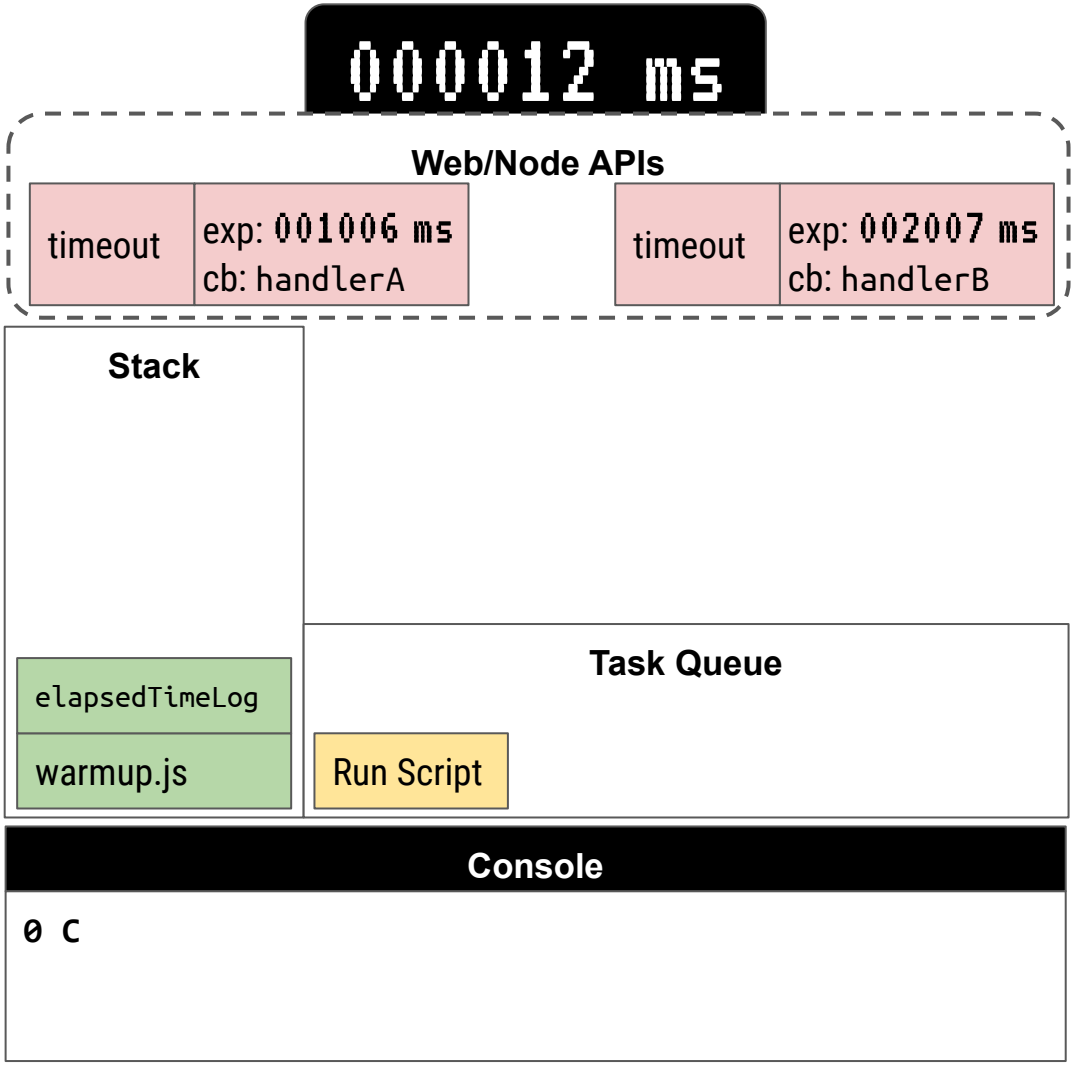
```
01: function waitSync(delayMs) {
02:   const waitStart = Date.now();
03:   while (Date.now() - waitStart < delayMs) {}
04: }
05:
06: const start = Date.now();
07: function elapsedTimeLog(...args) {
08:   const elapsedMs = Date.now() - start;
09:   const sec = Math.floor(elapsedMs / 1000);
10:   console.log(sec, ...args);
11: }
12:
13: function handlerA() {
14:   waitSync(6000);
15:   elapsedTimeLog("A");
16: }
17:
18: function handlerB() {
19:   elapsedTimeLog("B");
20: }
21:
22: setTimeout(handlerA, 1000);
23: setTimeout(handlerB, 2000);
24:
25: elapsedTimeLog("C");
```



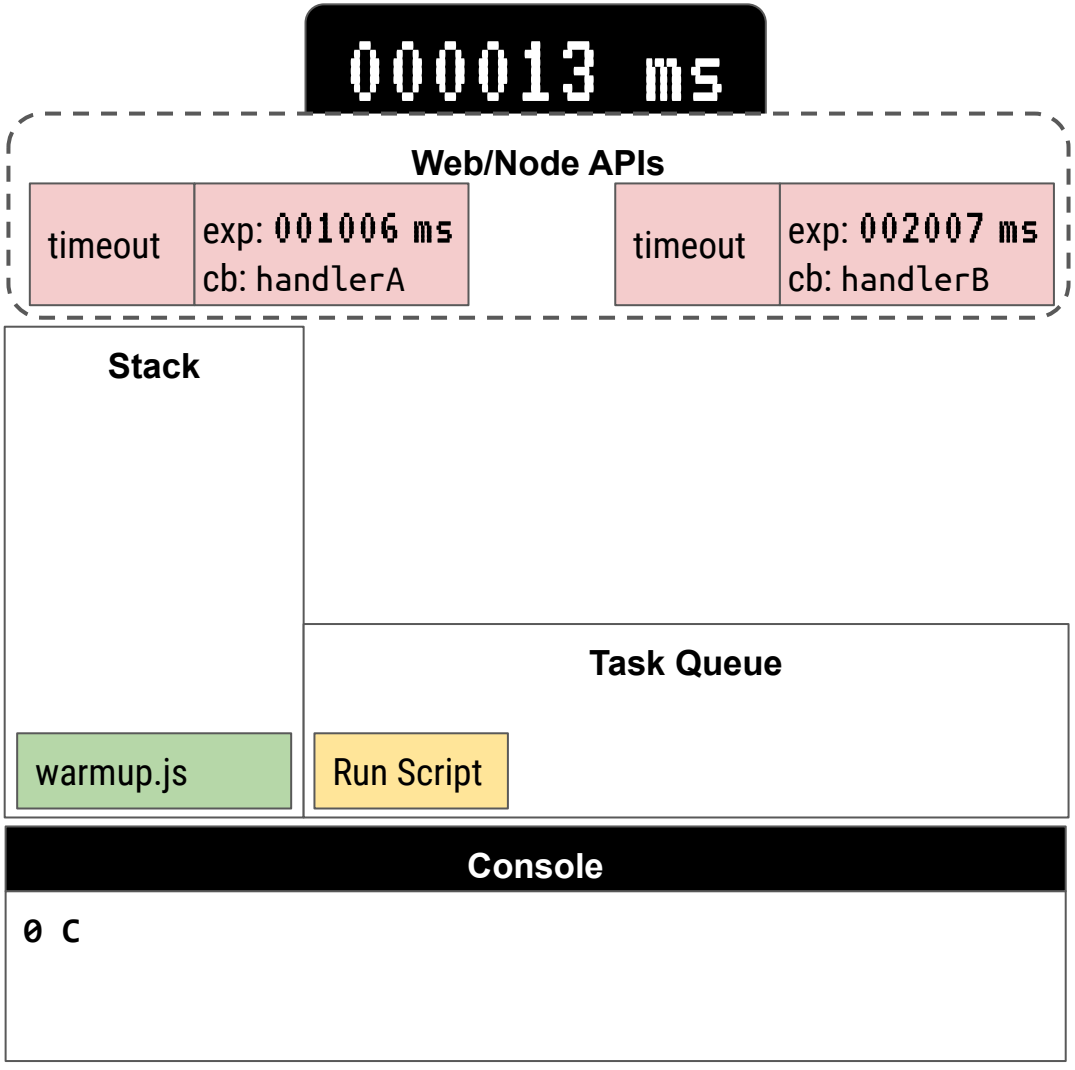
```
01: function waitSync(delayMs) {
02:   const waitStart = Date.now();
03:   while (Date.now() - waitStart < delayMs) {}
04: }
05:
06: const start = Date.now();
07: function elapsedTimeLog(...args) {
08:   const elapsedMs = Date.now() - start;
09:   const sec = Math.floor(elapsedMs / 1000);
10:   console.log(sec, ...args);
11: }
12:
13: function handlerA() {
14:   waitSync(6000);
15:   elapsedTimeLog("A");
16: }
17:
18: function handlerB() {
19:   elapsedTimeLog("B");
20: }
21:
22: setTimeout(handlerA, 1000);
23: setTimeout(handlerB, 2000);
24:
25: elapsedTimeLog("C");
```



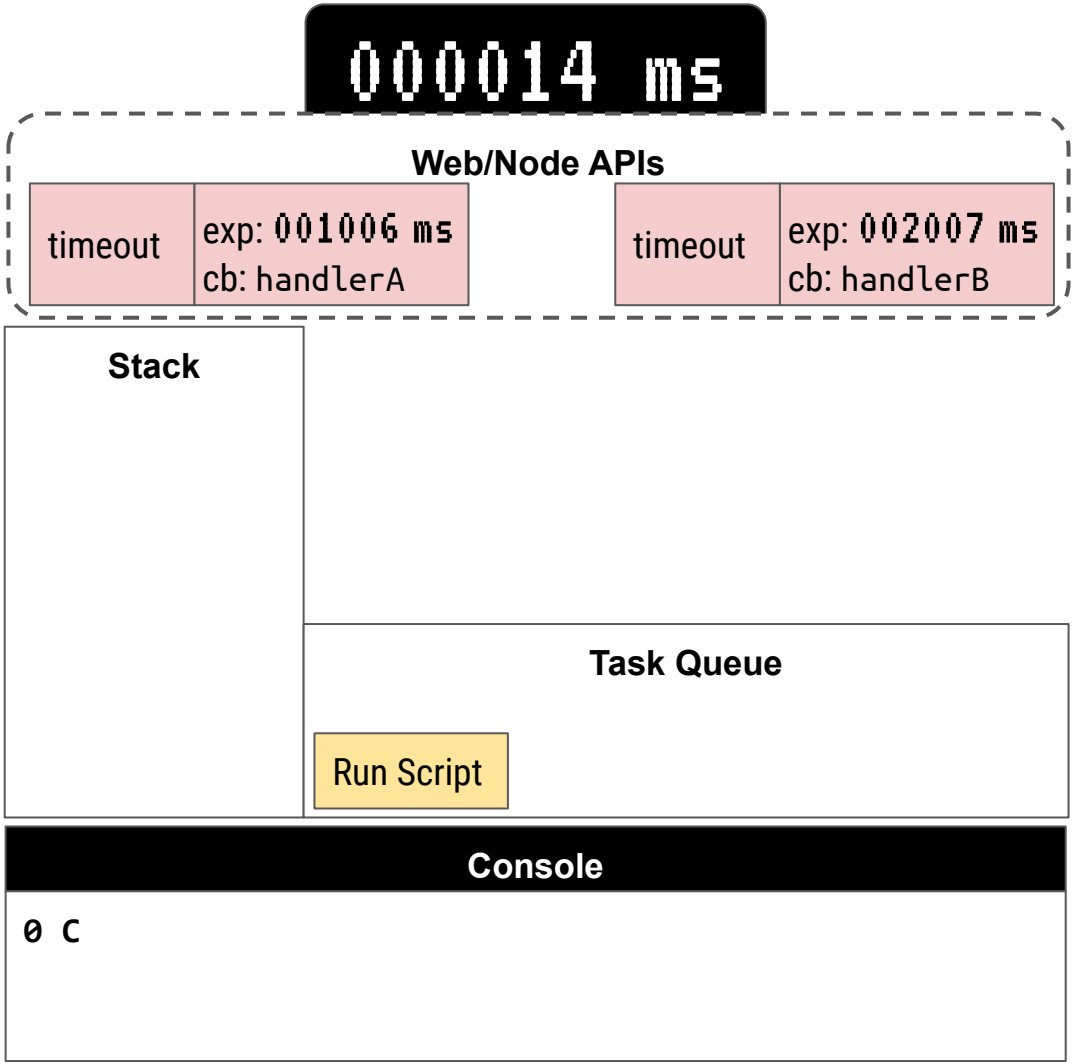
```
01: function waitSync(delayMs) {
02:   const waitStart = Date.now();
03:   while (Date.now() - waitStart < delayMs) {}
04: }
05:
06: const start = Date.now();
07: function elapsedTimeLog(...args) {
08:   const elapsedMs = Date.now() - start;
09:   const sec = Math.floor(elapsedMs / 1000);
10:   console.log(sec, ...args);
11: }
12:
13: function handlerA() {
14:   waitSync(6000);
15:   elapsedTimeLog("A");
16: }
17:
18: function handlerB() {
19:   elapsedTimeLog("B");
20: }
21:
22: setTimeout(handlerA, 1000);
23: setTimeout(handlerB, 2000);
24:
25: elapsedTimeLog("C");
```



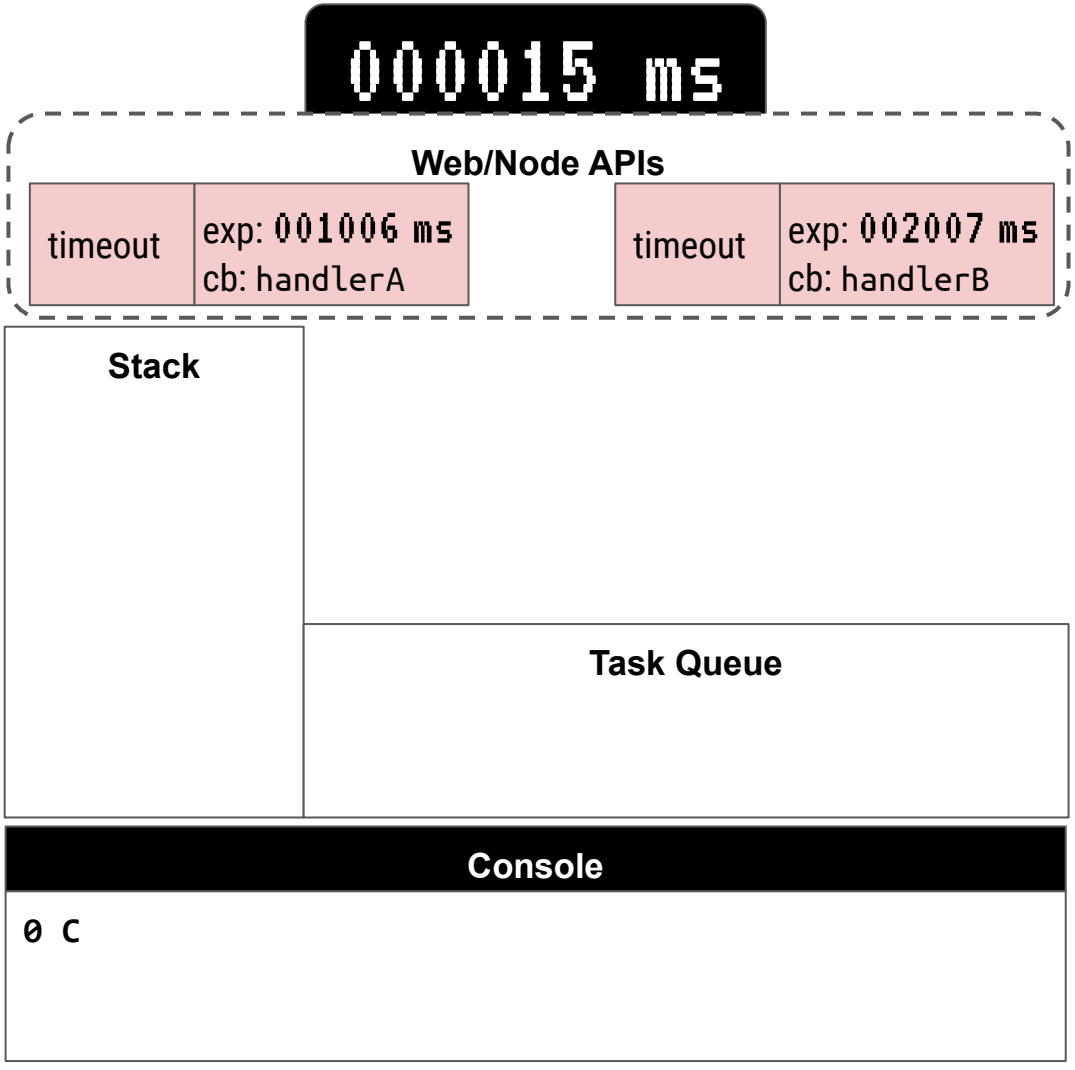
```
01: function waitSync(delayMs) {
02:   const waitStart = Date.now();
03:   while (Date.now() - waitStart < delayMs) {}
04: }
05:
06: const start = Date.now();
07: function elapsedTimeLog(...args) {
08:   const elapsedMs = Date.now() - start;
09:   const sec = Math.floor(elapsedMs / 1000);
10:   console.log(sec, ...args);
11: }
12:
13: function handlerA() {
14:   waitSync(6000);
15:   elapsedTimeLog("A");
16: }
17:
18: function handlerB() {
19:   elapsedTimeLog("B");
20: }
21:
22: setTimeout(handlerA, 1000);
23: setTimeout(handlerB, 2000);
24:
25: elapsedTimeLog("C");
```



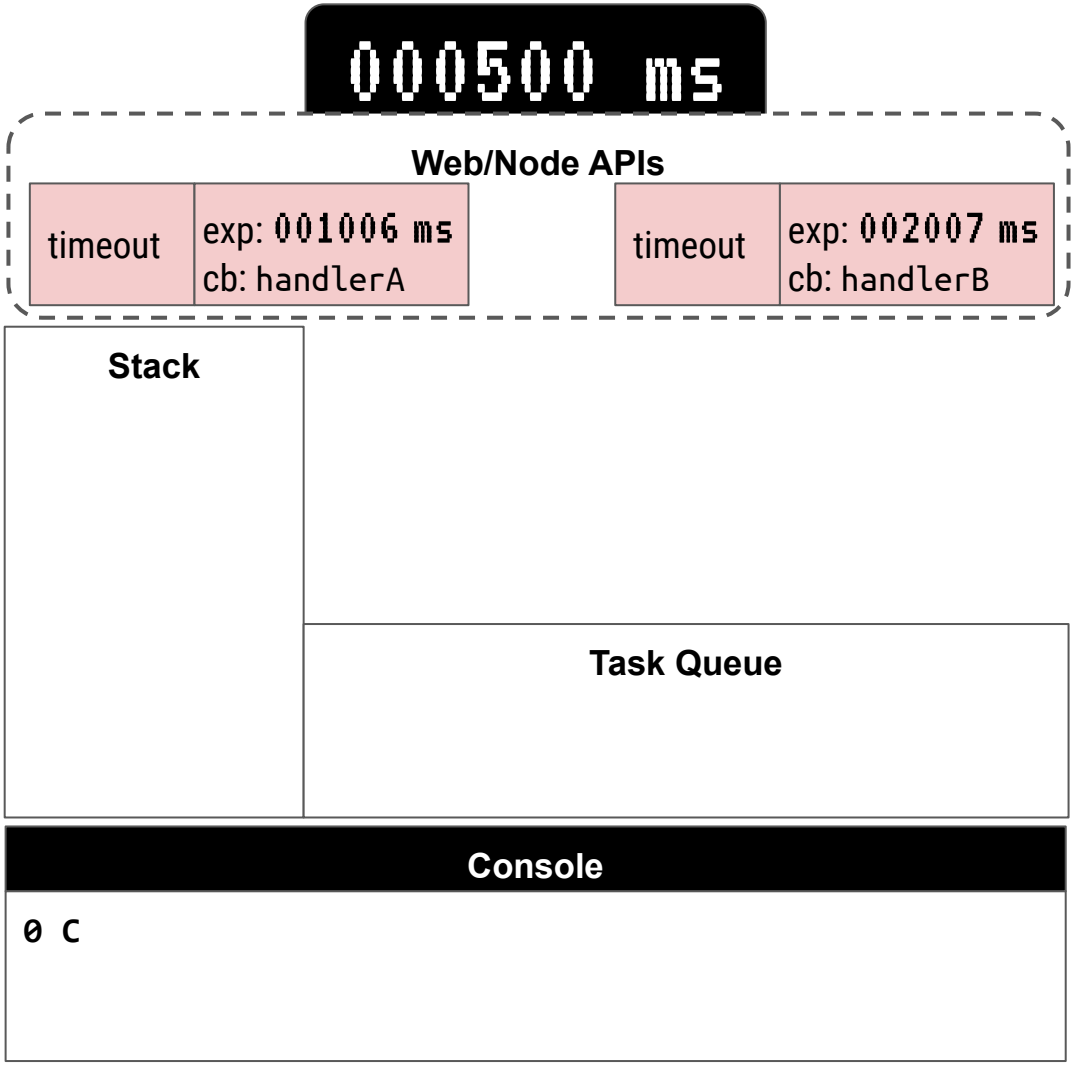
```
01: function waitSync(delayMs) {
02:   const waitStart = Date.now();
03:   while (Date.now() - waitStart < delayMs) {}
04: }
05:
06: const start = Date.now();
07: function elapsedTimeLog(...args) {
08:   const elapsedMs = Date.now() - start;
09:   const sec = Math.floor(elapsedMs / 1000);
10:   console.log(sec, ...args);
11: }
12:
13: function handlerA() {
14:   waitSync(6000);
15:   elapsedTimeLog("A");
16: }
17:
18: function handlerB() {
19:   elapsedTimeLog("B");
20: }
21:
22: setTimeout(handlerA, 1000);
23: setTimeout(handlerB, 2000);
24:
25: elapsedTimeLog("C");
```



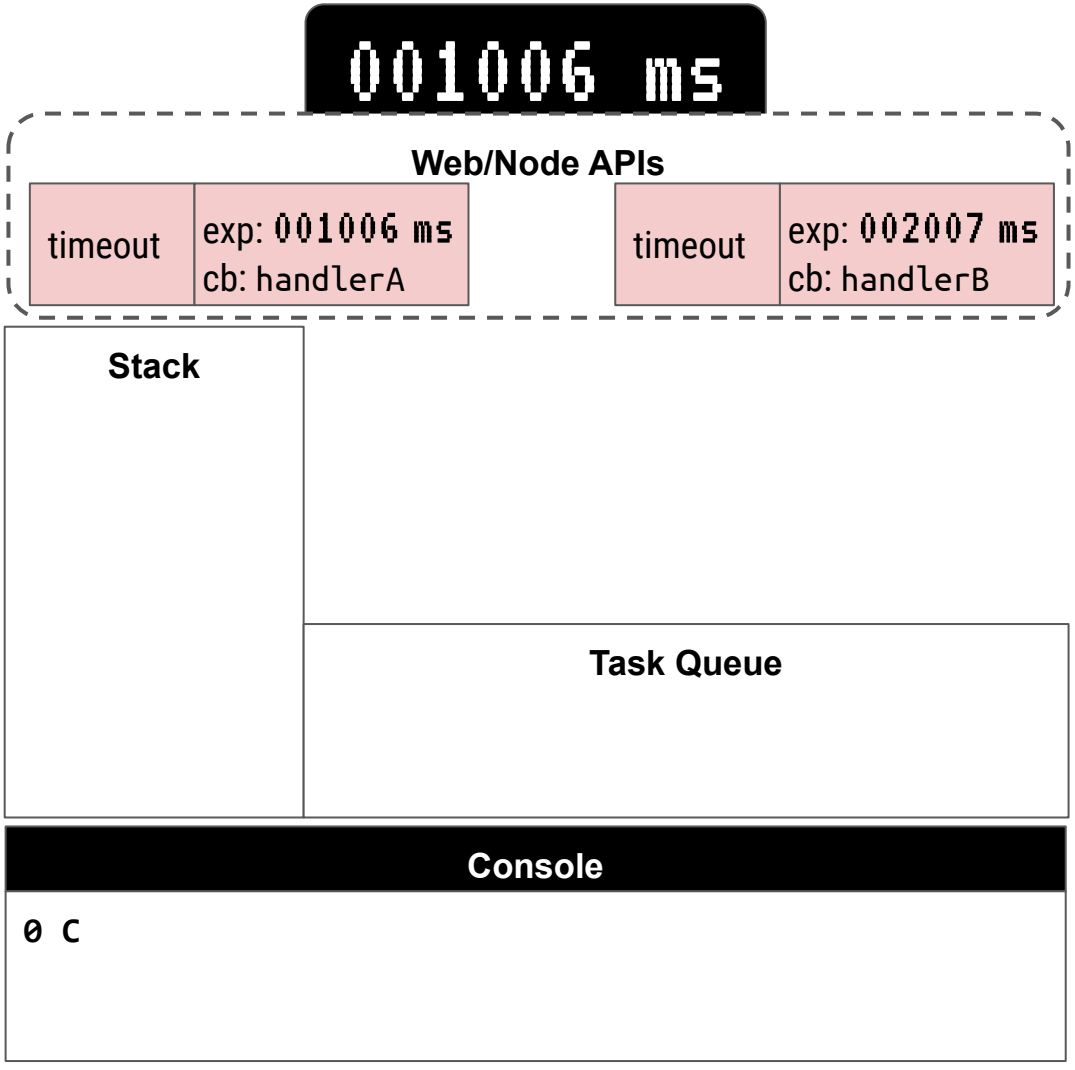
```
01: function waitSync(delayMs) {
02:   const waitStart = Date.now();
03:   while (Date.now() - waitStart < delayMs) {}
04: }
05:
06: const start = Date.now();
07: function elapsedTimeLog(...args) {
08:   const elapsedMs = Date.now() - start;
09:   const sec = Math.floor(elapsedMs / 1000);
10:   console.log(sec, ...args);
11: }
12:
13: function handlerA() {
14:   waitSync(6000);
15:   elapsedTimeLog("A");
16: }
17:
18: function handlerB() {
19:   elapsedTimeLog("B");
20: }
21:
22: setTimeout(handlerA, 1000);
23: setTimeout(handlerB, 2000);
24:
25: elapsedTimeLog("C");
```



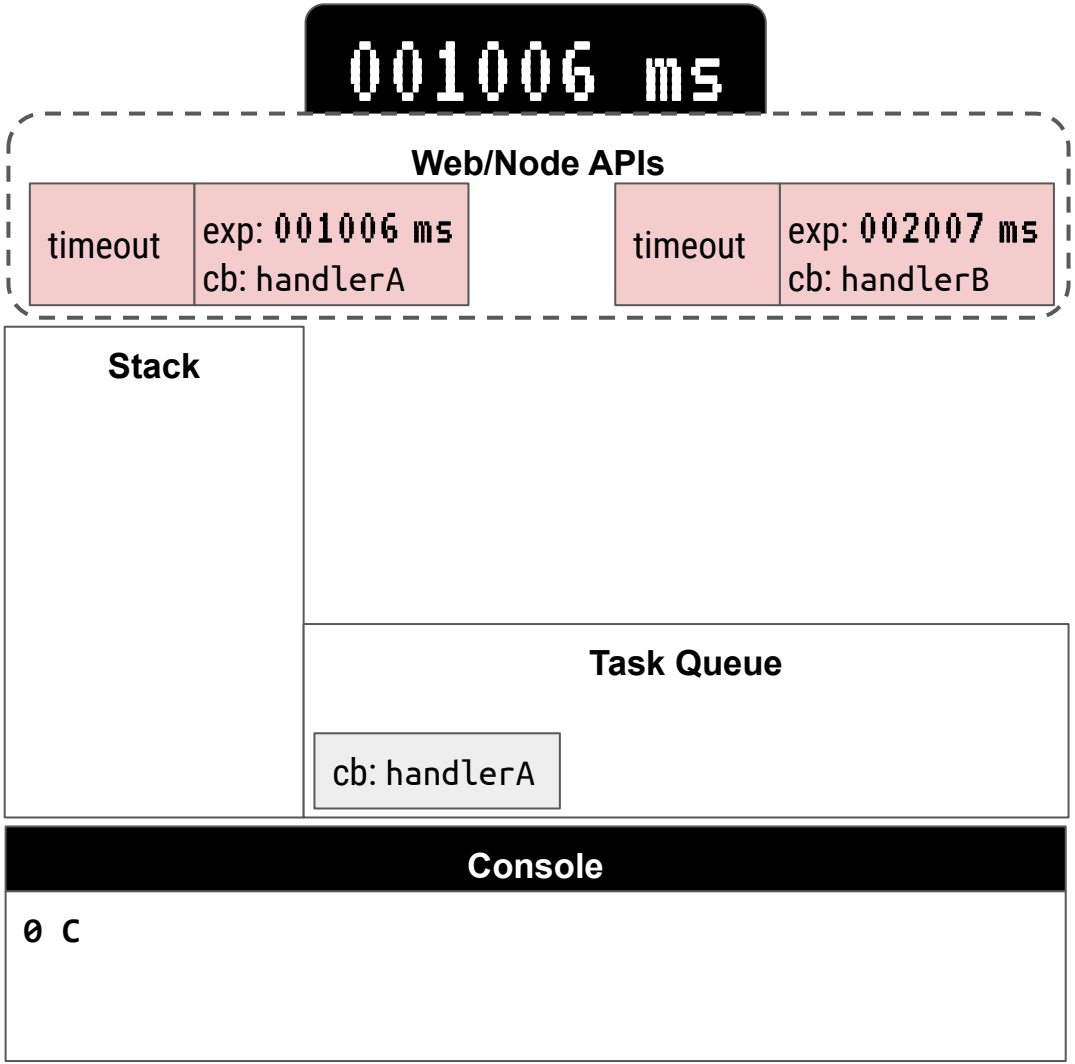
```
01: function waitSync(delayMs) {
02:   const waitStart = Date.now();
03:   while (Date.now() - waitStart < delayMs) {}
04: }
05:
06: const start = Date.now();
07: function elapsedTimeLog(...args) {
08:   const elapsedMs = Date.now() - start;
09:   const sec = Math.floor(elapsedMs / 1000);
10:   console.log(sec, ...args);
11: }
12:
13: function handlerA() {
14:   waitSync(6000);
15:   elapsedTimeLog("A");
16: }
17:
18: function handlerB() {
19:   elapsedTimeLog("B");
20: }
21:
22: setTimeout(handlerA, 1000);
23: setTimeout(handlerB, 2000);
24:
25: elapsedTimeLog("C");
```



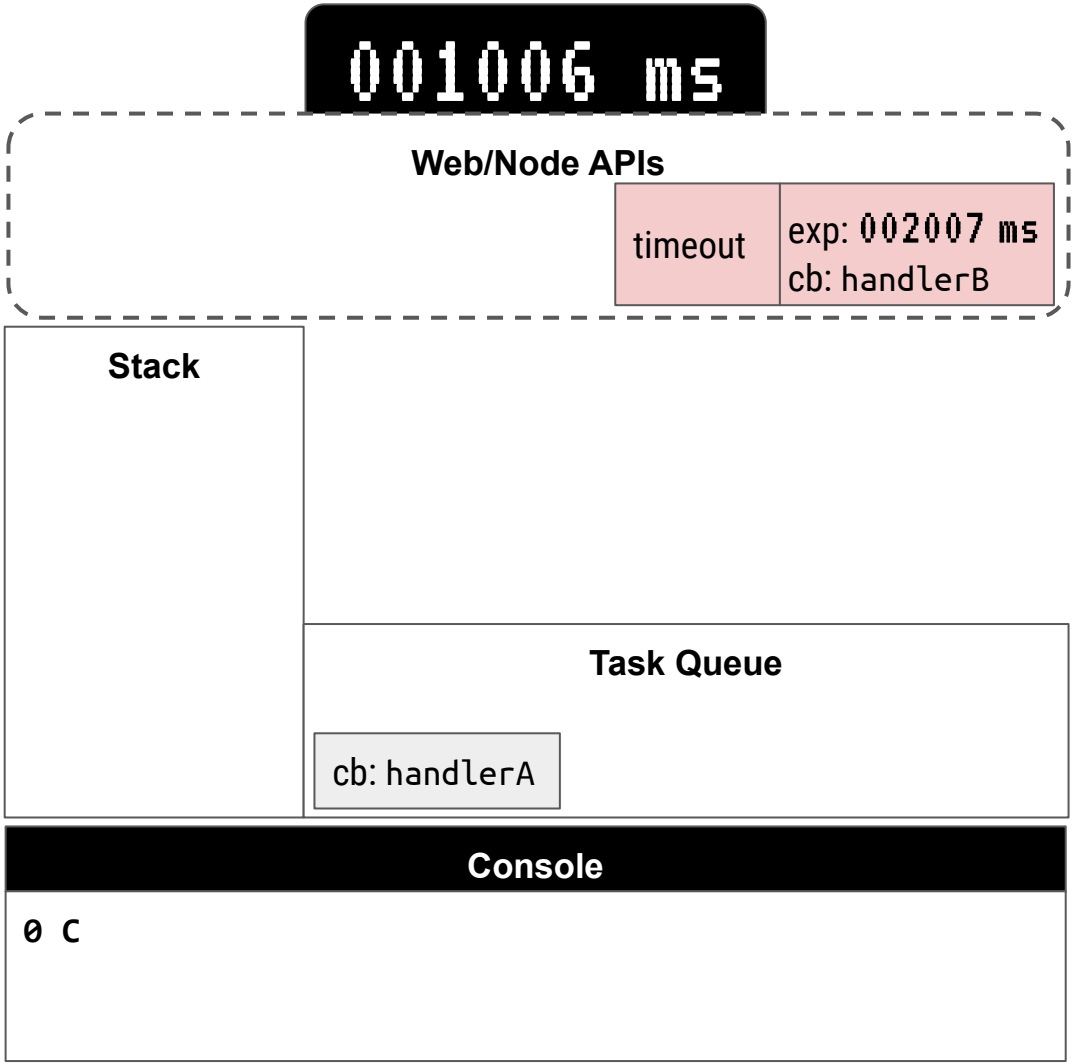
```
01: function waitSync(delayMs) {
02:   const waitStart = Date.now();
03:   while (Date.now() - waitStart < delayMs) {}
04: }
05:
06: const start = Date.now();
07: function elapsedTimeLog(...args) {
08:   const elapsedMs = Date.now() - start;
09:   const sec = Math.floor(elapsedMs / 1000);
10:   console.log(sec, ...args);
11: }
12:
13: function handlerA() {
14:   waitSync(6000);
15:   elapsedTimeLog("A");
16: }
17:
18: function handlerB() {
19:   elapsedTimeLog("B");
20: }
21:
22: setTimeout(handlerA, 1000);
23: setTimeout(handlerB, 2000);
24:
25: elapsedTimeLog("C");
```



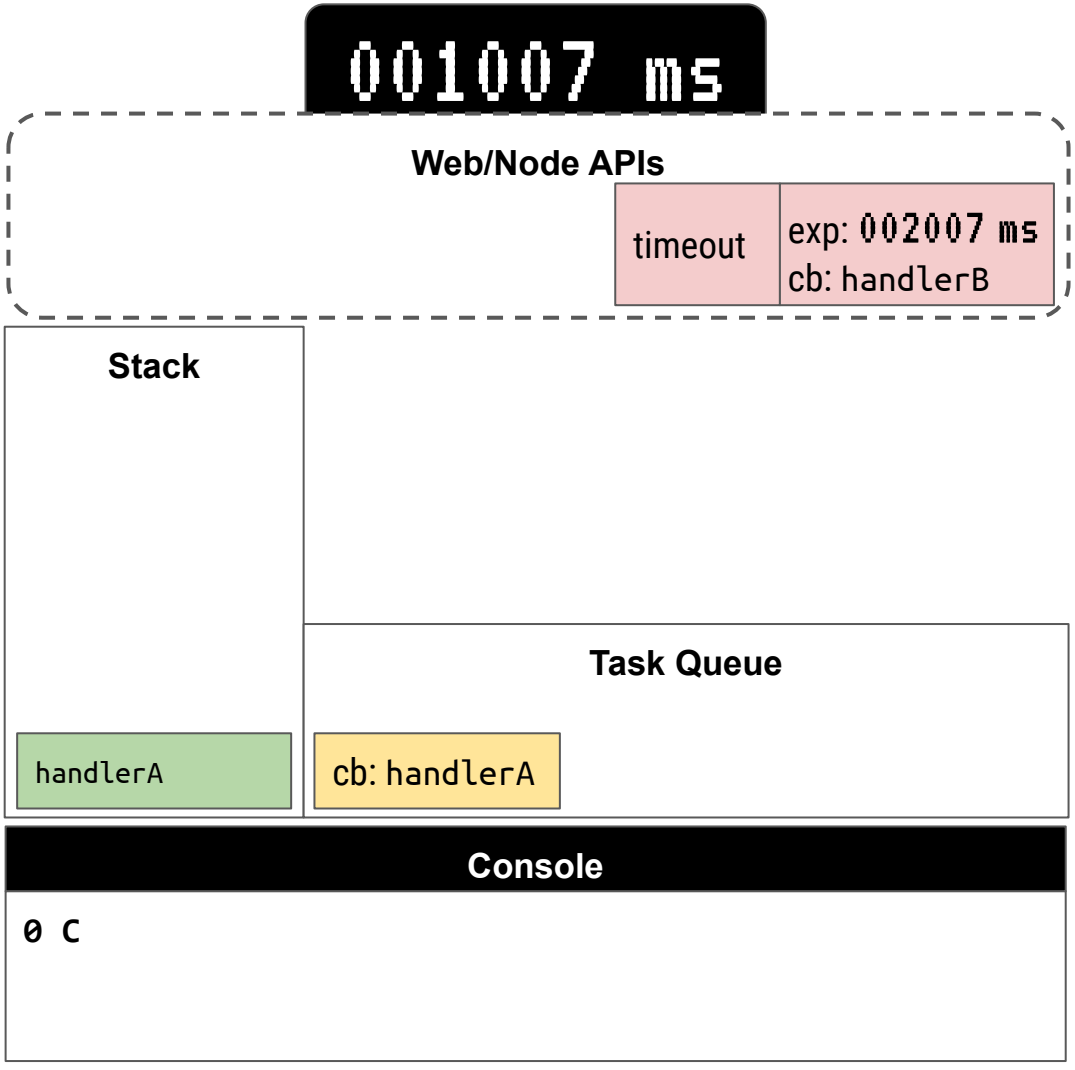
```
01: function waitSync(delayMs) {
02:   const waitStart = Date.now();
03:   while (Date.now() - waitStart < delayMs) {}
04: }
05:
06: const start = Date.now();
07: function elapsedTimeLog(...args) {
08:   const elapsedMs = Date.now() - start;
09:   const sec = Math.floor(elapsedMs / 1000);
10:   console.log(sec, ...args);
11: }
12:
13: function handlerA() {
14:   waitSync(6000);
15:   elapsedTimeLog("A");
16: }
17:
18: function handlerB() {
19:   elapsedTimeLog("B");
20: }
21:
22: setTimeout(handlerA, 1000);
23: setTimeout(handlerB, 2000);
24:
25: elapsedTimeLog("C");
```



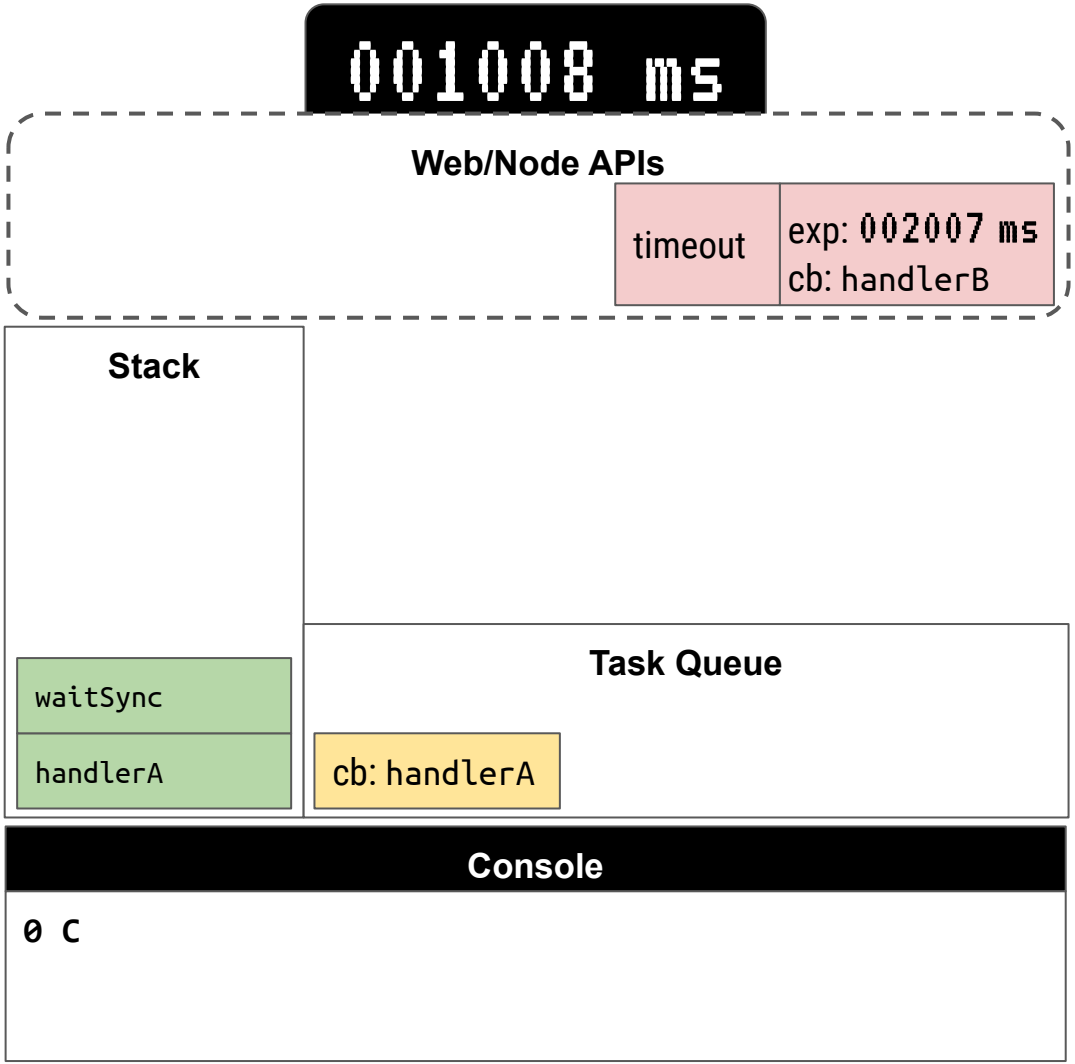
```
01: function waitSync(delayMs) {
02:   const waitStart = Date.now();
03:   while (Date.now() - waitStart < delayMs) {}
04: }
05:
06: const start = Date.now();
07: function elapsedTimeLog(...args) {
08:   const elapsedMs = Date.now() - start;
09:   const sec = Math.floor(elapsedMs / 1000);
10:   console.log(sec, ...args);
11: }
12:
13: function handlerA() {
14:   waitSync(6000);
15:   elapsedTimeLog("A");
16: }
17:
18: function handlerB() {
19:   elapsedTimeLog("B");
20: }
21:
22: setTimeout(handlerA, 1000);
23: setTimeout(handlerB, 2000);
24:
25: elapsedTimeLog("C");
```



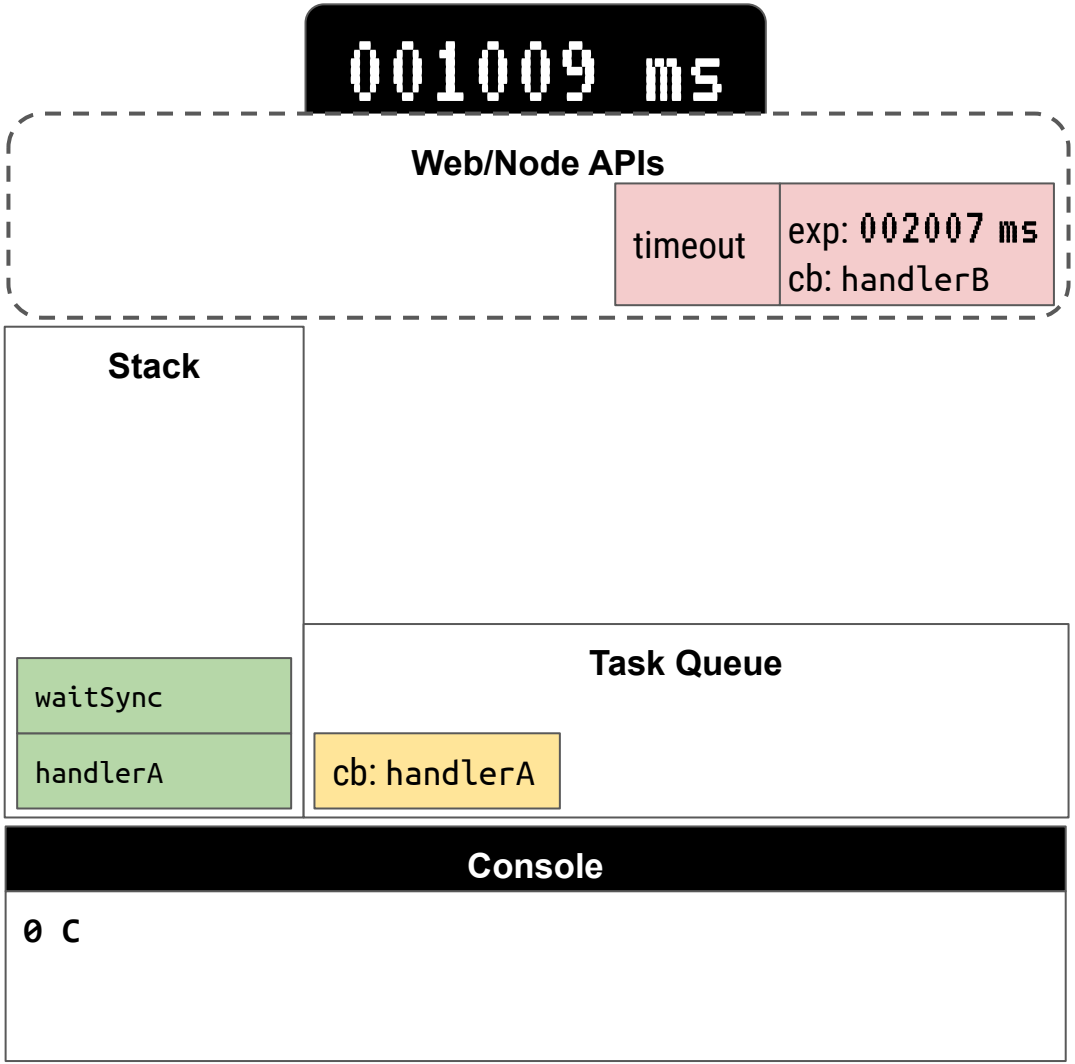
```
01: function waitSync(delayMs) {
02:   const waitStart = Date.now();
03:   while (Date.now() - waitStart < delayMs) {}
04: }
05:
06: const start = Date.now();
07: function elapsedTimeLog(...args) {
08:   const elapsedMs = Date.now() - start;
09:   const sec = Math.floor(elapsedMs / 1000);
10:   console.log(sec, ...args);
11: }
12:
13: function handlerA() {
14:   waitSync(6000);
15:   elapsedTimeLog("A");
16: }
17:
18: function handlerB() {
19:   elapsedTimeLog("B");
20: }
21:
22: setTimeout(handlerA, 1000);
23: setTimeout(handlerB, 2000);
24:
25: elapsedTimeLog("C");
```



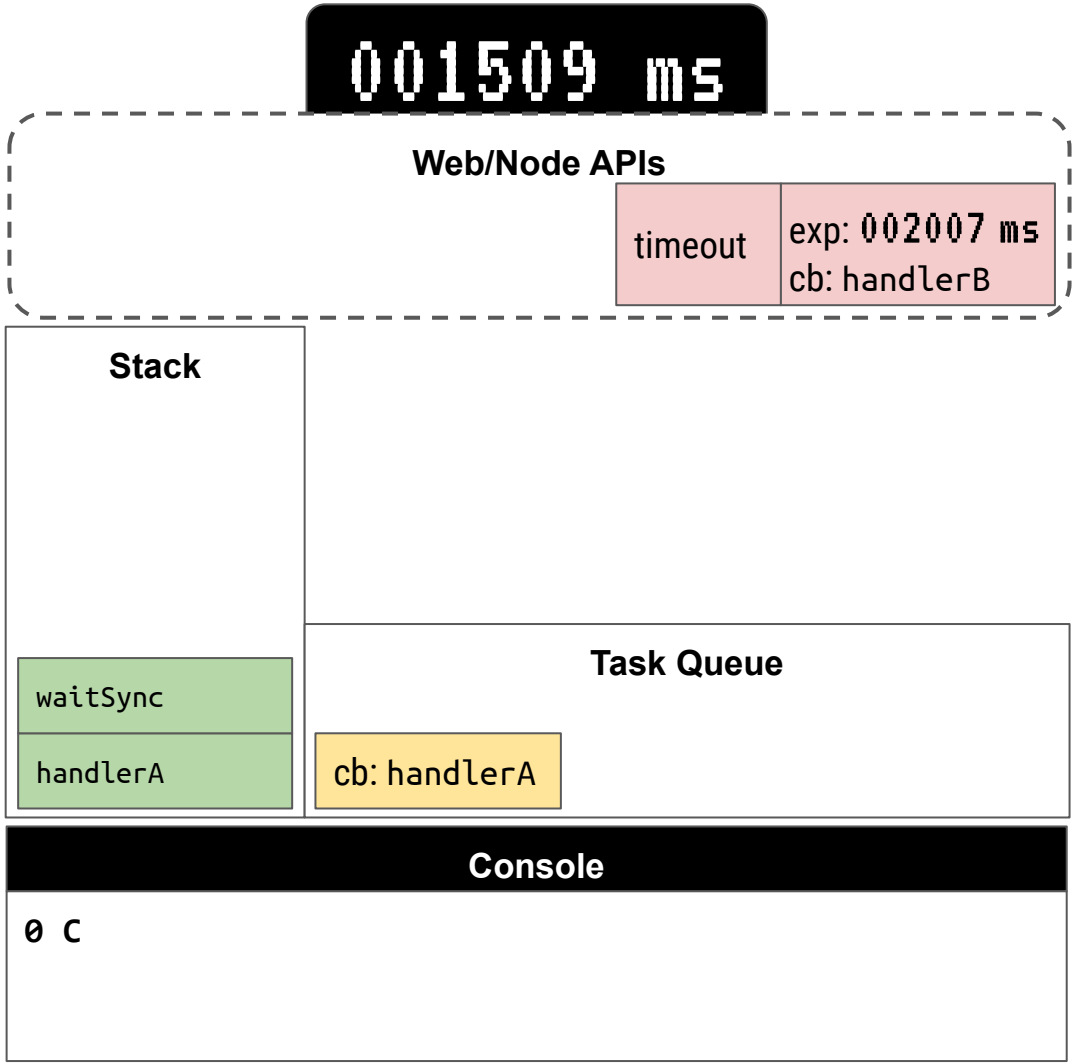
```
01: function waitSync(delayMs) {
02:   const waitStart = Date.now();
03:   while (Date.now() - waitStart < delayMs) {}
04: }
05:
06: const start = Date.now();
07: function elapsedTimeLog(...args) {
08:   const elapsedMs = Date.now() - start;
09:   const sec = Math.floor(elapsedMs / 1000);
10:   console.log(sec, ...args);
11: }
12:
13: function handlerA() {
14:   waitSync(6000);
15:   elapsedTimeLog("A");
16: }
17:
18: function handlerB() {
19:   elapsedTimeLog("B");
20: }
21:
22: setTimeout(handlerA, 1000);
23: setTimeout(handlerB, 2000);
24:
25: elapsedTimeLog("C");
```



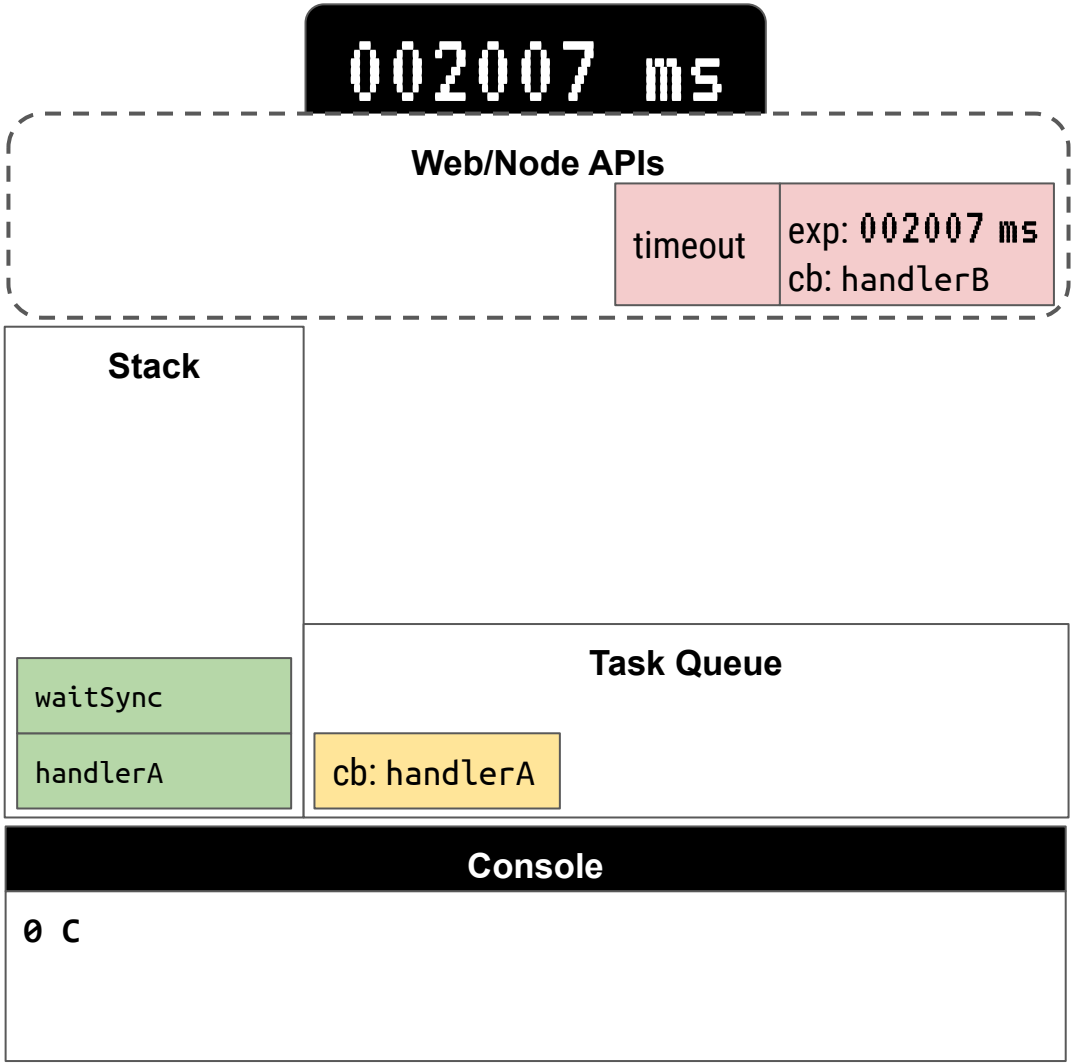
```
01: function waitSync(delayMs) {
02:   const waitStart = Date.now();
03:   while (Date.now() - waitStart < delayMs) {}
04: }
05:
06: const start = Date.now();
07: function elapsedTimeLog(...args) {
08:   const elapsedMs = Date.now() - start;
09:   const sec = Math.floor(elapsedMs / 1000);
10:   console.log(sec, ...args);
11: }
12:
13: function handlerA() {
14:   waitSync(6000);
15:   elapsedTimeLog("A");
16: }
17:
18: function handlerB() {
19:   elapsedTimeLog("B");
20: }
21:
22: setTimeout(handlerA, 1000);
23: setTimeout(handlerB, 2000);
24:
25: elapsedTimeLog("C");
```



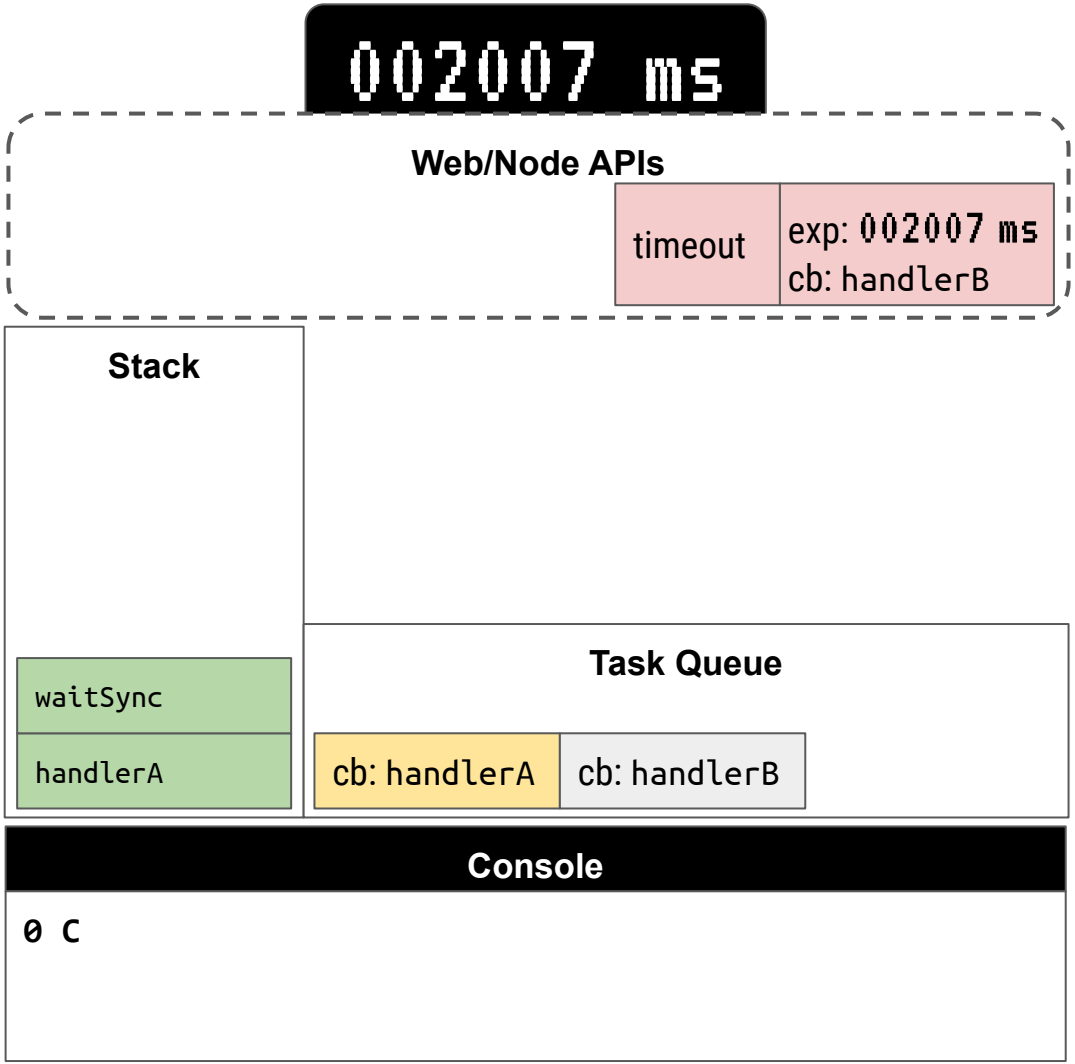
```
01: function waitSync(delayMs) {
02:   const waitStart = Date.now();
03:   while (Date.now() - waitStart < delayMs) {}
04: }
05:
06: const start = Date.now();
07: function elapsedTimeLog(...args) {
08:   const elapsedMs = Date.now() - start;
09:   const sec = Math.floor(elapsedMs / 1000);
10:   console.log(sec, ...args);
11: }
12:
13: function handlerA() {
14:   waitSync(6000);
15:   elapsedTimeLog("A");
16: }
17:
18: function handlerB() {
19:   elapsedTimeLog("B");
20: }
21:
22: setTimeout(handlerA, 1000);
23: setTimeout(handlerB, 2000);
24:
25: elapsedTimeLog("C");
```



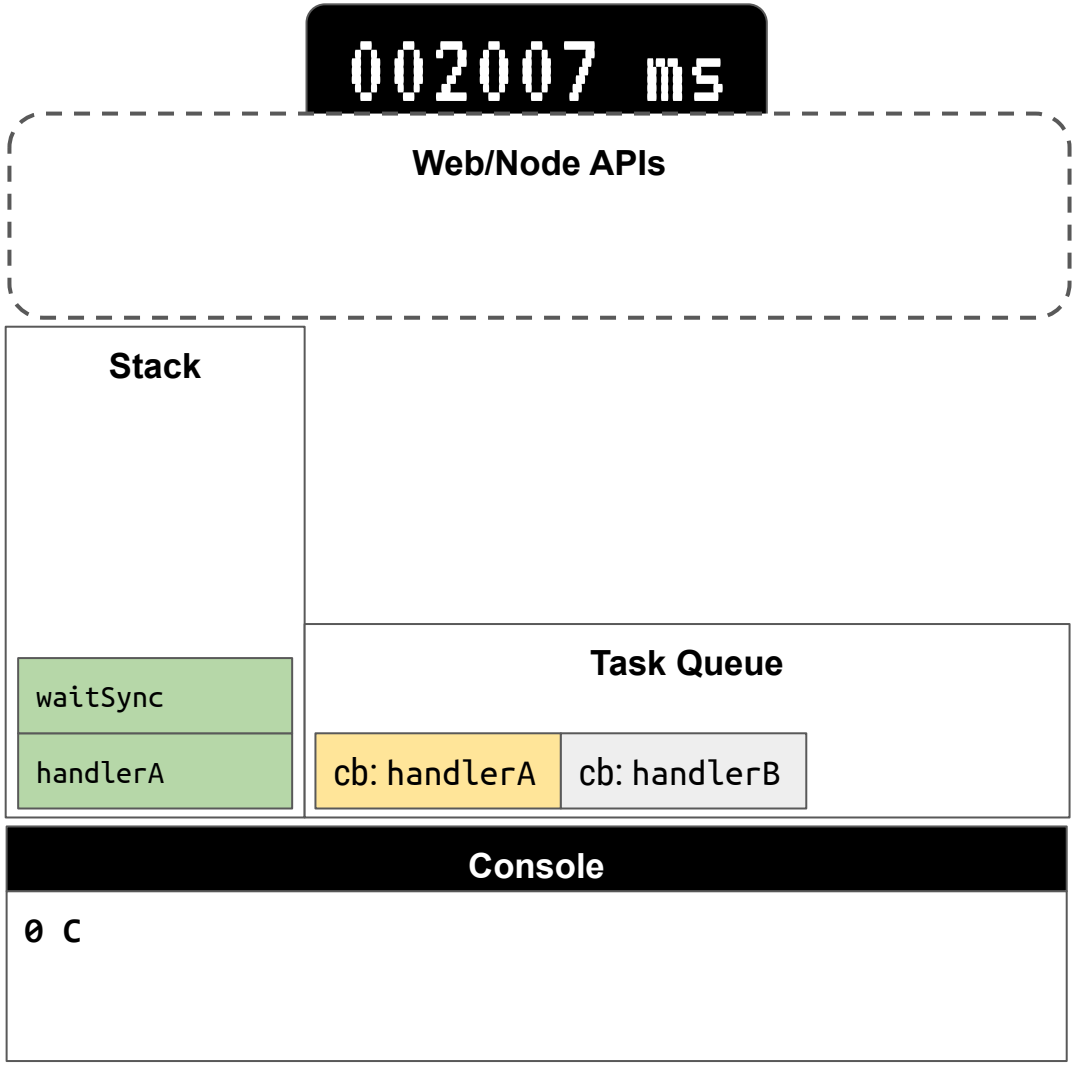
```
01: function waitSync(delayMs) {
02:   const waitStart = Date.now();
03:   while (Date.now() - waitStart < delayMs) {}
04: }
05:
06: const start = Date.now();
07: function elapsedTimeLog(...args) {
08:   const elapsedMs = Date.now() - start;
09:   const sec = Math.floor(elapsedMs / 1000);
10:   console.log(sec, ...args);
11: }
12:
13: function handlerA() {
14:   waitSync(6000);
15:   elapsedTimeLog("A");
16: }
17:
18: function handlerB() {
19:   elapsedTimeLog("B");
20: }
21:
22: setTimeout(handlerA, 1000);
23: setTimeout(handlerB, 2000);
24:
25: elapsedTimeLog("C");
```



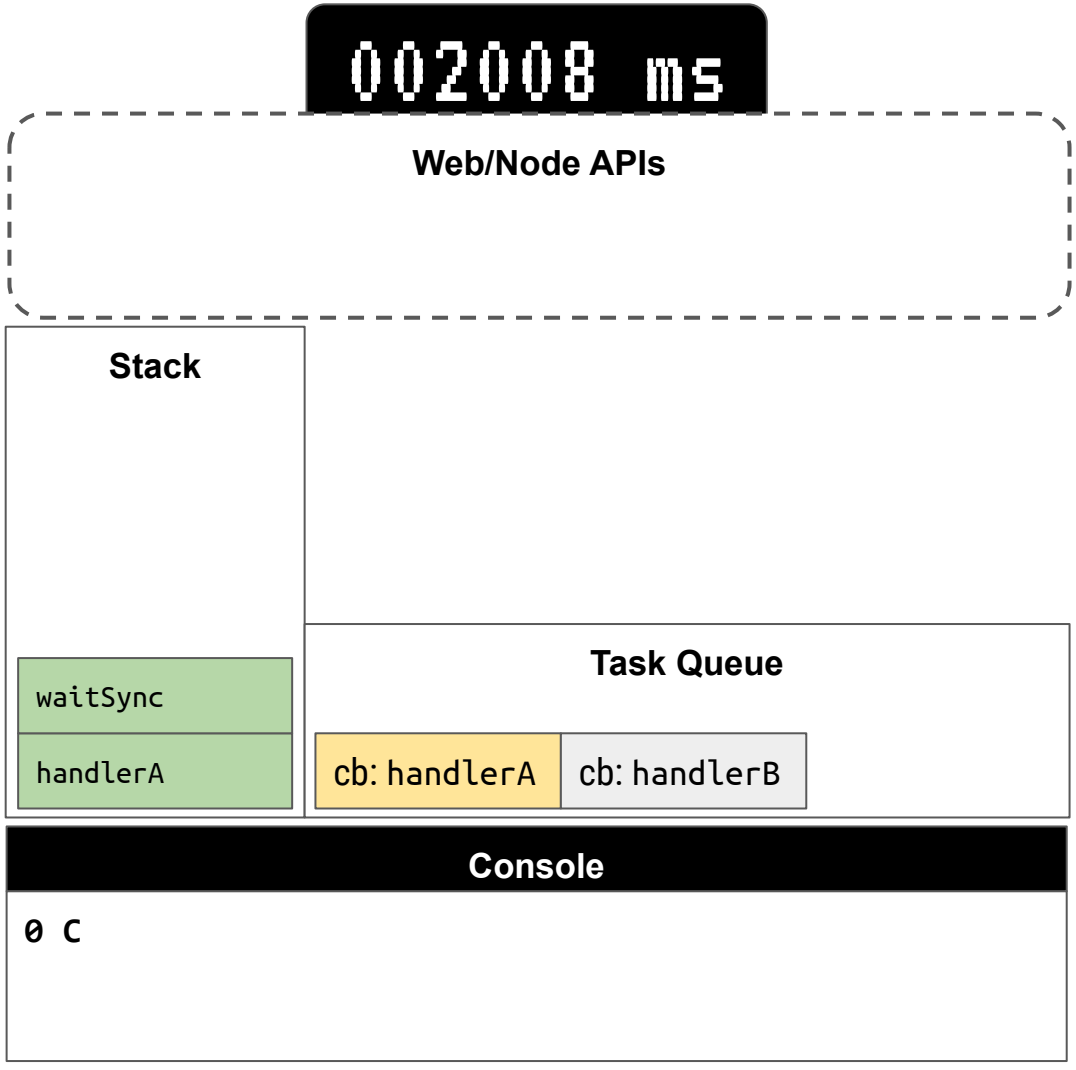
```
01: function waitSync(delayMs) {
02:   const waitStart = Date.now();
03:   while (Date.now() - waitStart < delayMs) {}
04: }
05:
06: const start = Date.now();
07: function elapsedTimeLog(...args) {
08:   const elapsedMs = Date.now() - start;
09:   const sec = Math.floor(elapsedMs / 1000);
10:   console.log(sec, ...args);
11: }
12:
13: function handlerA() {
14:   waitSync(6000);
15:   elapsedTimeLog("A");
16: }
17:
18: function handlerB() {
19:   elapsedTimeLog("B");
20: }
21:
22: setTimeout(handlerA, 1000);
23: setTimeout(handlerB, 2000);
24:
25: elapsedTimeLog("C");
```



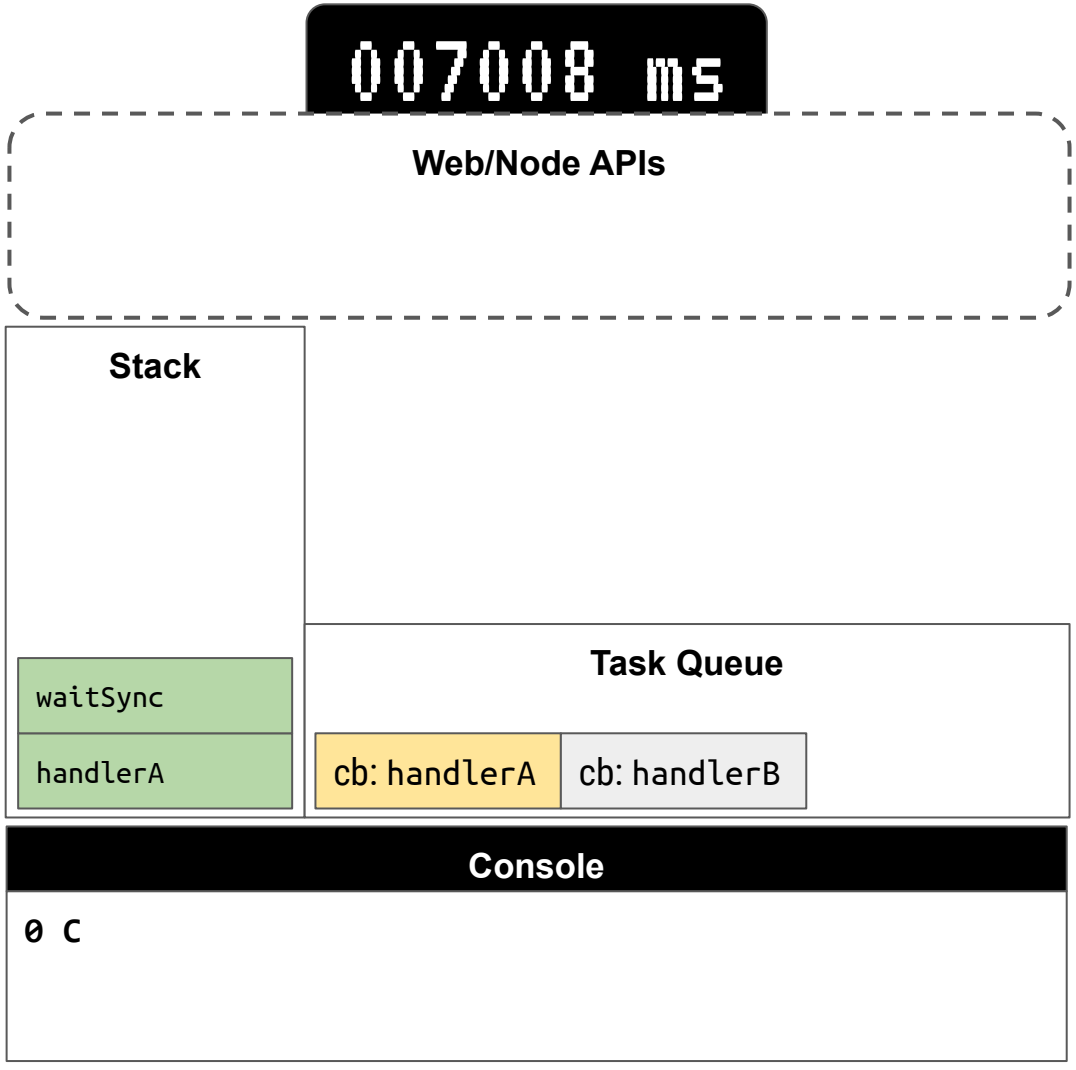
```
01: function waitSync(delayMs) {
02:   const waitStart = Date.now();
03:   while (Date.now() - waitStart < delayMs) {}
04: }
05:
06: const start = Date.now();
07: function elapsedTimeLog(...args) {
08:   const elapsedMs = Date.now() - start;
09:   const sec = Math.floor(elapsedMs / 1000);
10:   console.log(sec, ...args);
11: }
12:
13: function handlerA() {
14:   waitSync(6000);
15:   elapsedTimeLog("A");
16: }
17:
18: function handlerB() {
19:   elapsedTimeLog("B");
20: }
21:
22: setTimeout(handlerA, 1000);
23: setTimeout(handlerB, 2000);
24:
25: elapsedTimeLog("C");
```



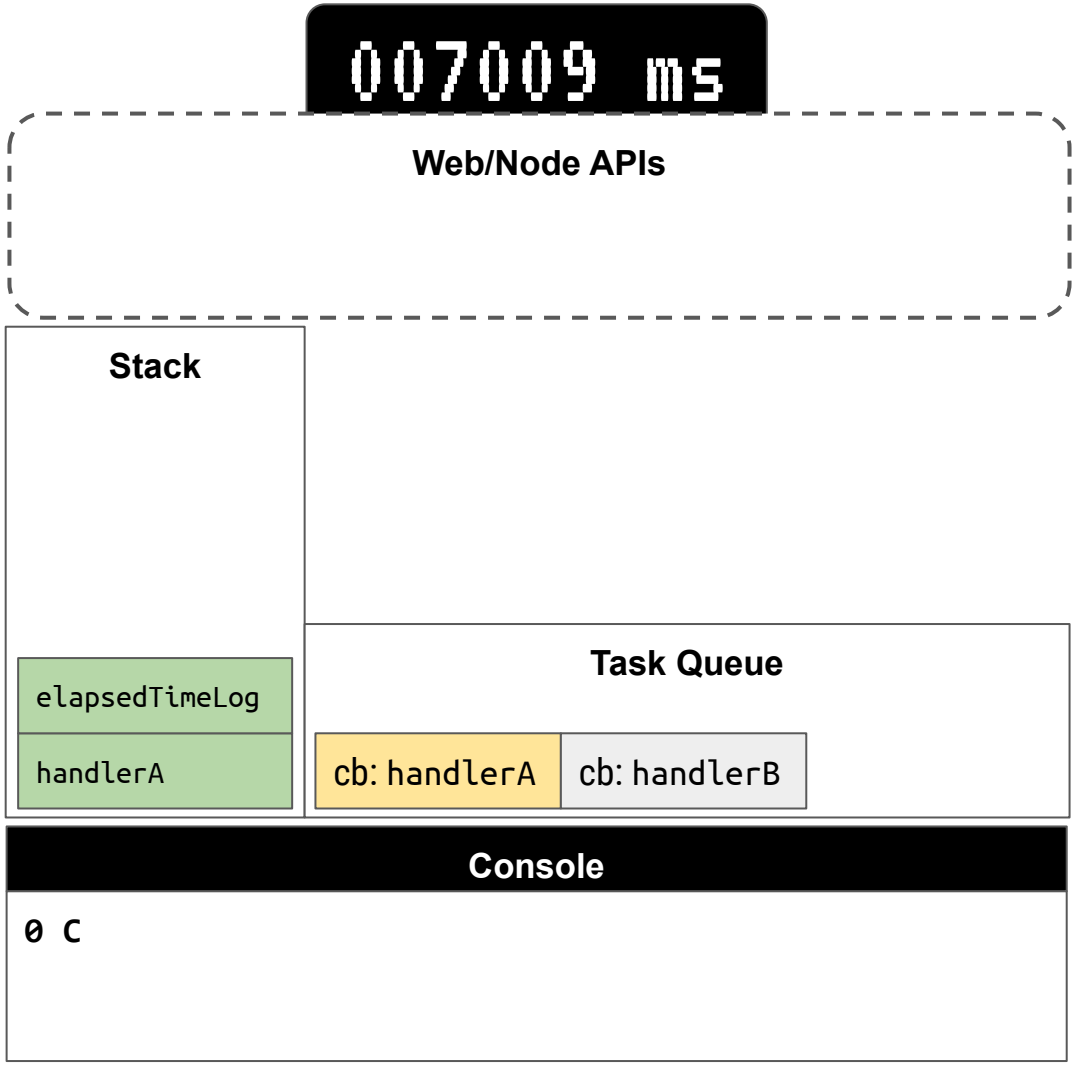
```
01: function waitSync(delayMs) {
02:   const waitStart = Date.now();
03:   while (Date.now() - waitStart < delayMs) {}
04: }
05:
06: const start = Date.now();
07: function elapsedTimeLog(...args) {
08:   const elapsedMs = Date.now() - start;
09:   const sec = Math.floor(elapsedMs / 1000);
10:   console.log(sec, ...args);
11: }
12:
13: function handlerA() {
14:   waitSync(6000);
15:   elapsedTimeLog("A");
16: }
17:
18: function handlerB() {
19:   elapsedTimeLog("B");
20: }
21:
22: setTimeout(handlerA, 1000);
23: setTimeout(handlerB, 2000);
24:
25: elapsedTimeLog("C");
```



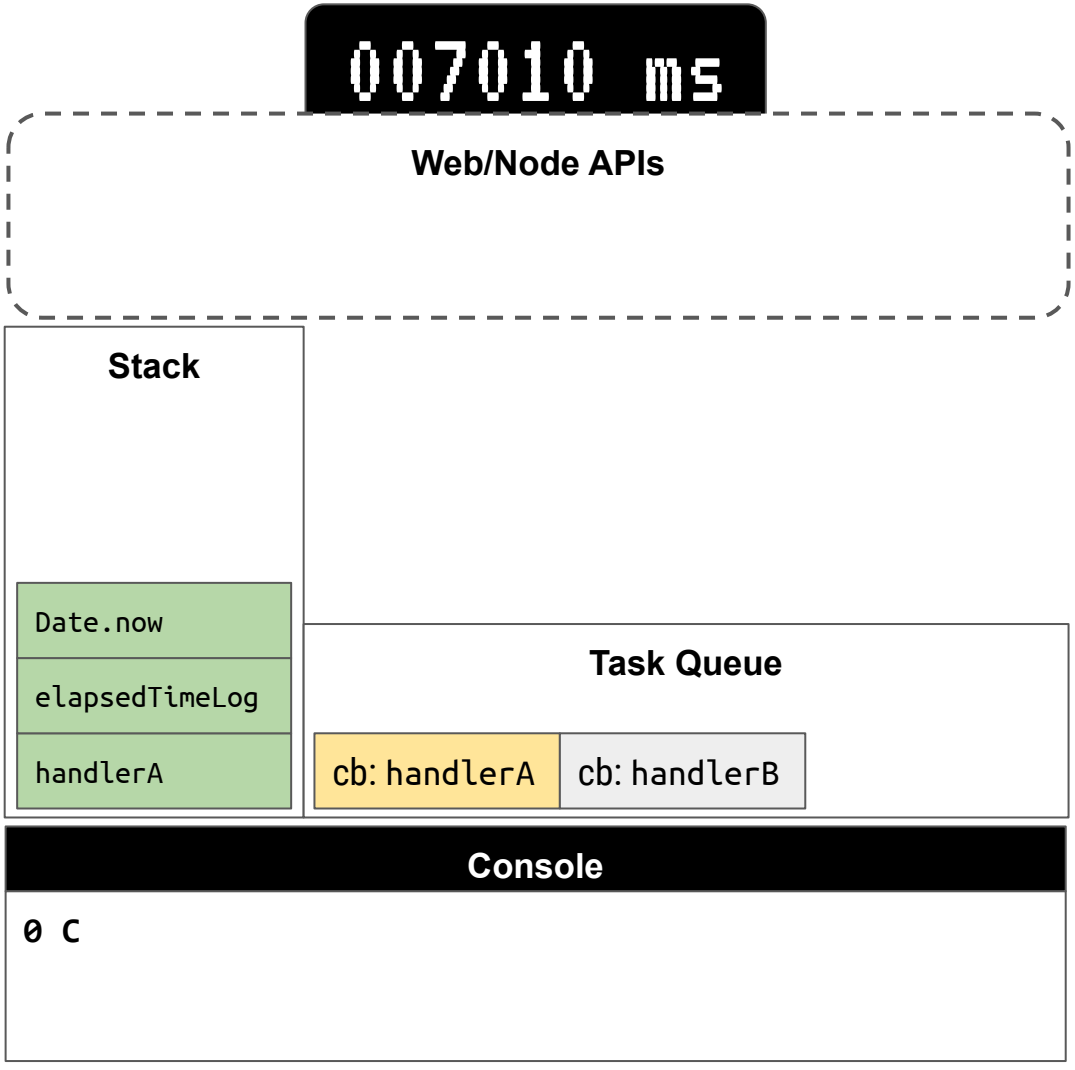
```
01: function waitSync(delayMs) {
02:   const waitStart = Date.now();
03:   while (Date.now() - waitStart < delayMs) {}
04: }
05:
06: const start = Date.now();
07: function elapsedTimeLog(...args) {
08:   const elapsedMs = Date.now() - start;
09:   const sec = Math.floor(elapsedMs / 1000);
10:   console.log(sec, ...args);
11: }
12:
13: function handlerA() {
14:   waitSync(6000);
15:   elapsedTimeLog("A");
16: }
17:
18: function handlerB() {
19:   elapsedTimeLog("B");
20: }
21:
22: setTimeout(handlerA, 1000);
23: setTimeout(handlerB, 2000);
24:
25: elapsedTimeLog("C");
```



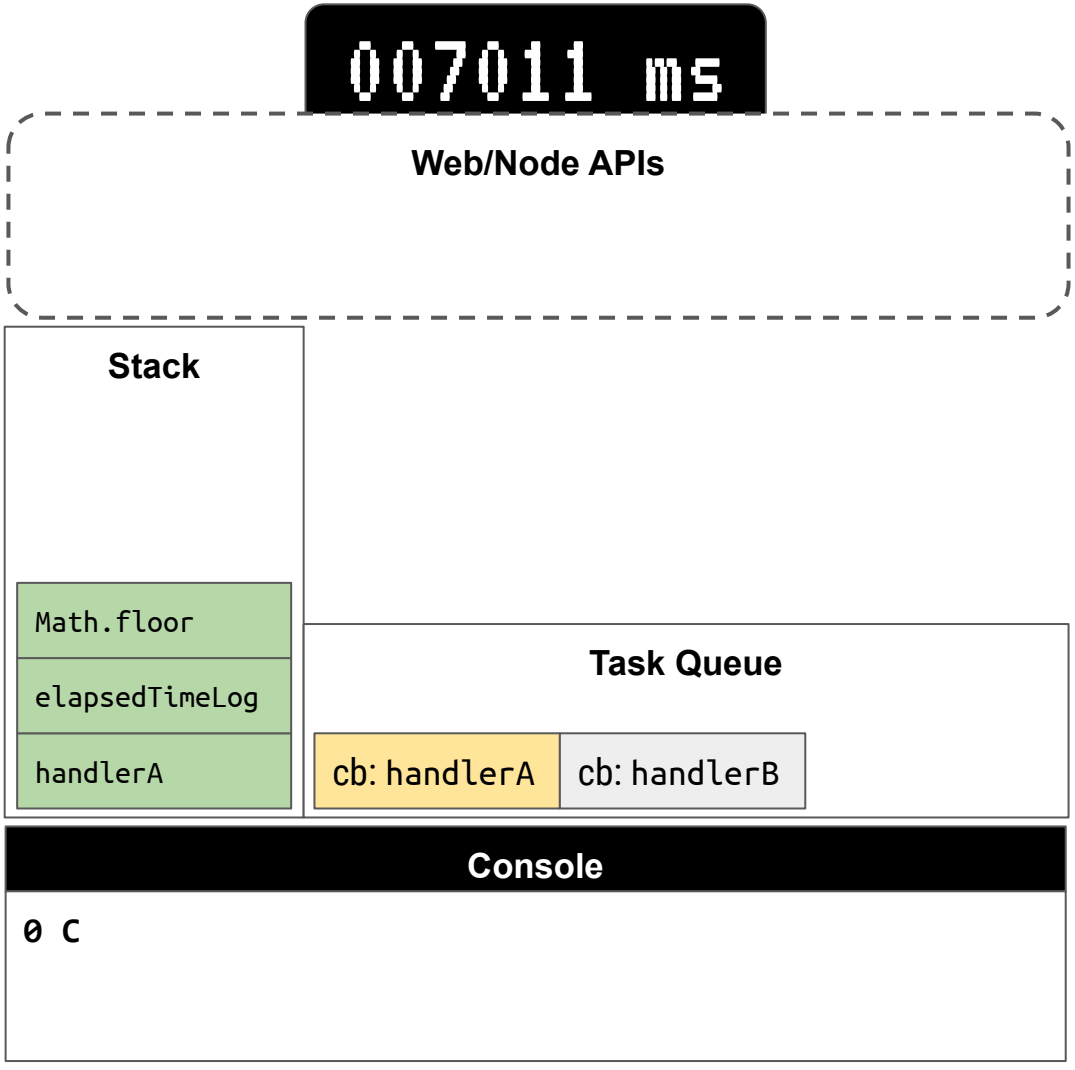
```
01: function waitSync(delayMs) {
02:   const waitStart = Date.now();
03:   while (Date.now() - waitStart < delayMs) {}
04: }
05:
06: const start = Date.now();
07: function elapsedTimeLog(...args) {
08:   const elapsedMs = Date.now() - start;
09:   const sec = Math.floor(elapsedMs / 1000);
10:   console.log(sec, ...args);
11: }
12:
13: function handlerA() {
14:   waitSync(6000);
15:   elapsedTimeLog("A");
16: }
17:
18: function handlerB() {
19:   elapsedTimeLog("B");
20: }
21:
22: setTimeout(handlerA, 1000);
23: setTimeout(handlerB, 2000);
24:
25: elapsedTimeLog("C");
```



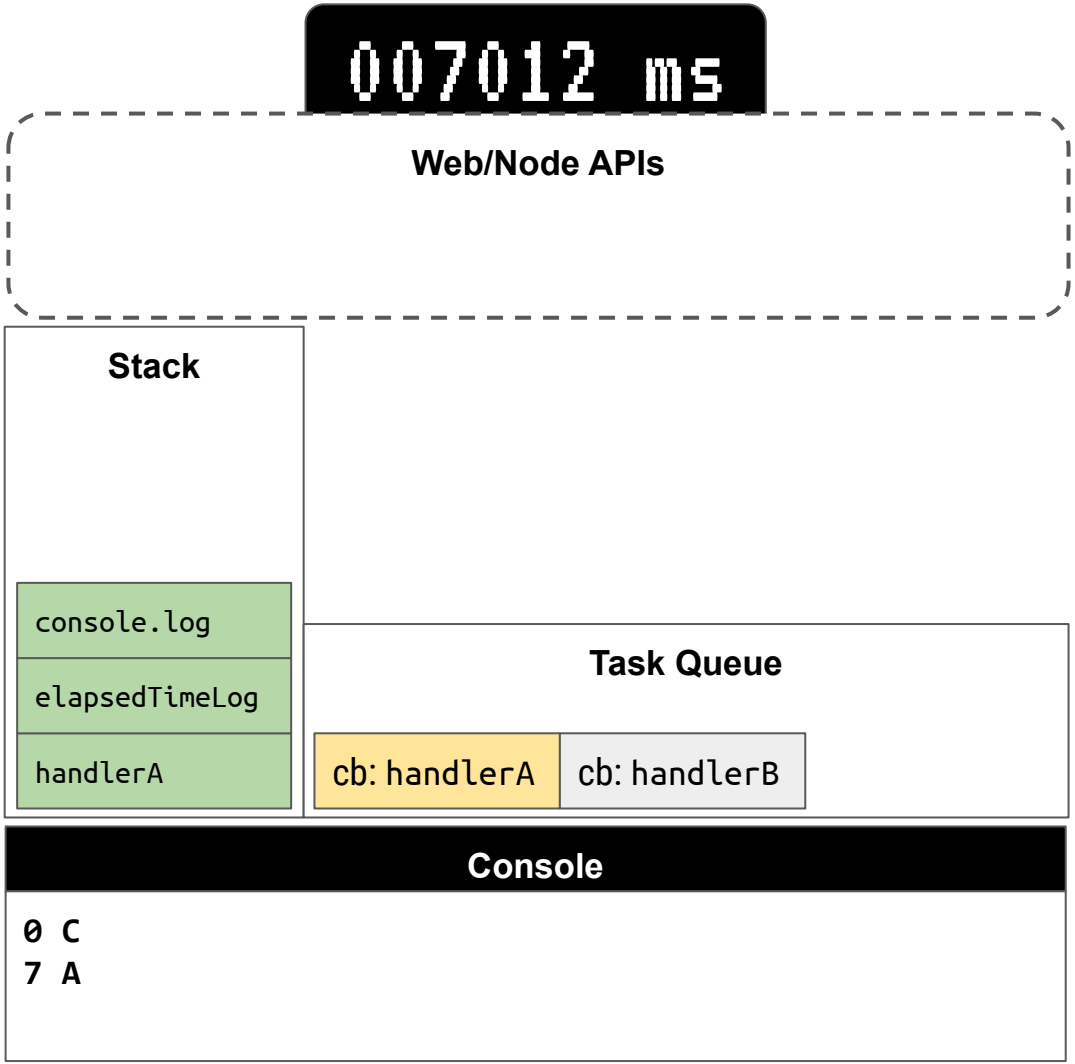
```
01: function waitSync(delayMs) {
02:   const waitStart = Date.now();
03:   while (Date.now() - waitStart < delayMs) {}
04: }
05:
06: const start = Date.now();
07: function elapsedTimeLog(...args) {
08:   const elapsedMs = Date.now() - start;
09:   const sec = Math.floor(elapsedMs / 1000);
10:   console.log(sec, ...args);
11: }
12:
13: function handlerA() {
14:   waitSync(6000);
15:   elapsedTimeLog("A");
16: }
17:
18: function handlerB() {
19:   elapsedTimeLog("B");
20: }
21:
22: setTimeout(handlerA, 1000);
23: setTimeout(handlerB, 2000);
24:
25: elapsedTimeLog("C");
```



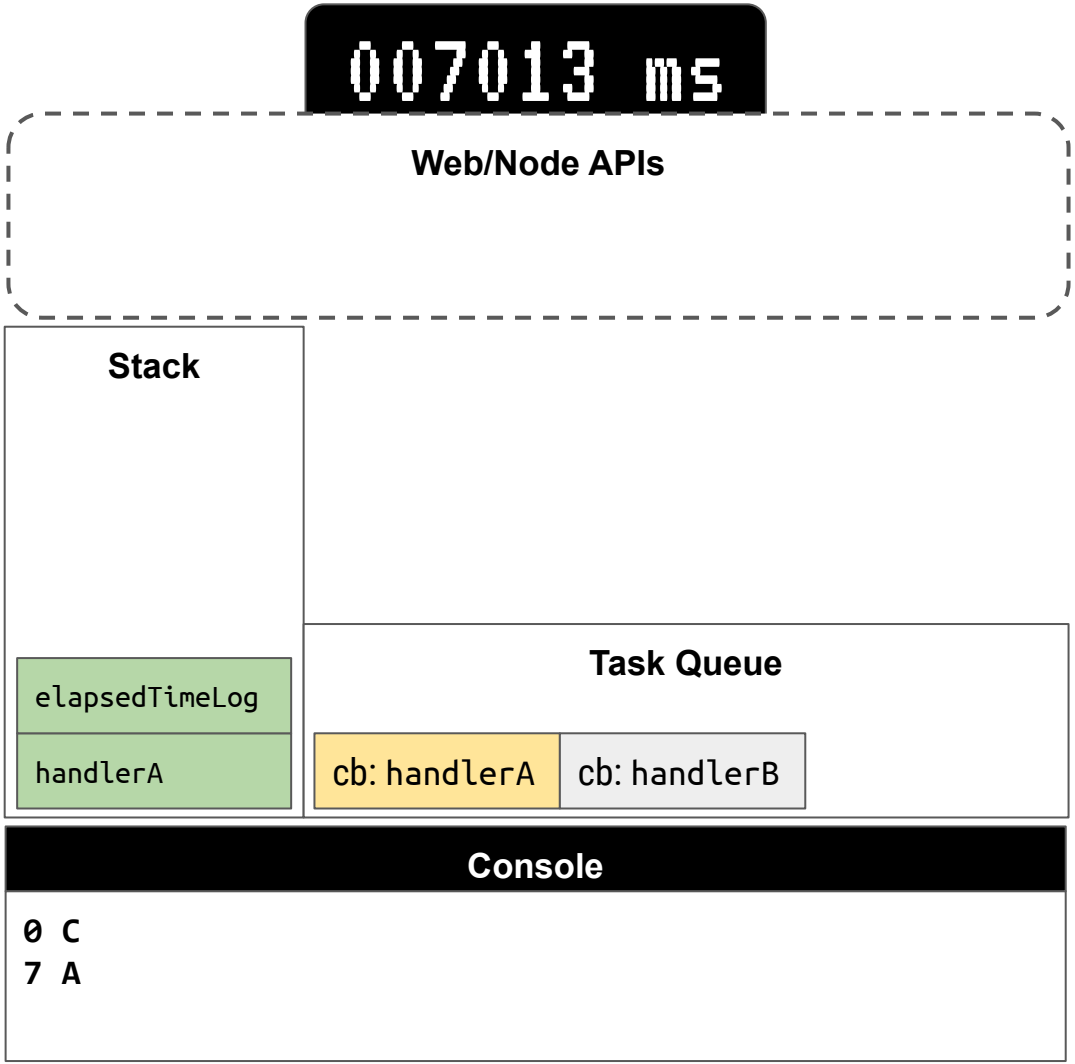
```
01: function waitSync(delayMs) {
02:   const waitStart = Date.now();
03:   while (Date.now() - waitStart < delayMs) {}
04: }
05:
06: const start = Date.now();
07: function elapsedTimeLog(...args) {
08:   const elapsedMs = Date.now() - start;
09:   const sec = Math.floor(elapsedMs / 1000);
10:   console.log(sec, ...args);
11: }
12:
13: function handlerA() {
14:   waitSync(6000);
15:   elapsedTimeLog("A");
16: }
17:
18: function handlerB() {
19:   elapsedTimeLog("B");
20: }
21:
22: setTimeout(handlerA, 1000);
23: setTimeout(handlerB, 2000);
24:
25: elapsedTimeLog("C");
```



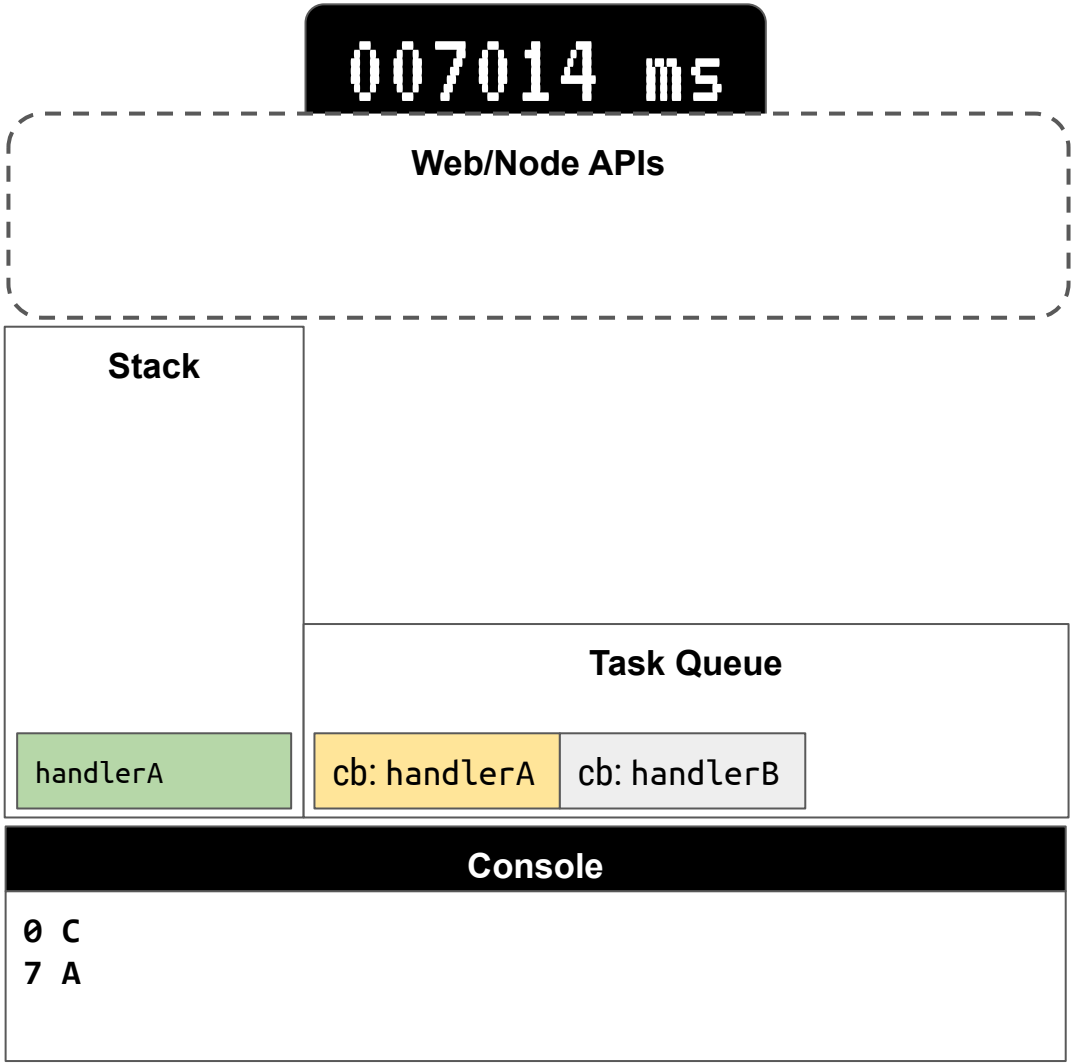
```
01: function waitSync(delayMs) {
02:   const waitStart = Date.now();
03:   while (Date.now() - waitStart < delayMs) {}
04: }
05:
06: const start = Date.now();
07: function elapsedTimeLog(...args) {
08:   const elapsedMs = Date.now() - start;
09:   const sec = Math.floor(elapsedMs / 1000);
10:   console.log(sec, ...args);
11: }
12:
13: function handlerA() {
14:   waitSync(6000);
15:   elapsedTimeLog("A");
16: }
17:
18: function handlerB() {
19:   elapsedTimeLog("B");
20: }
21:
22: setTimeout(handlerA, 1000);
23: setTimeout(handlerB, 2000);
24:
25: elapsedTimeLog("C");
```



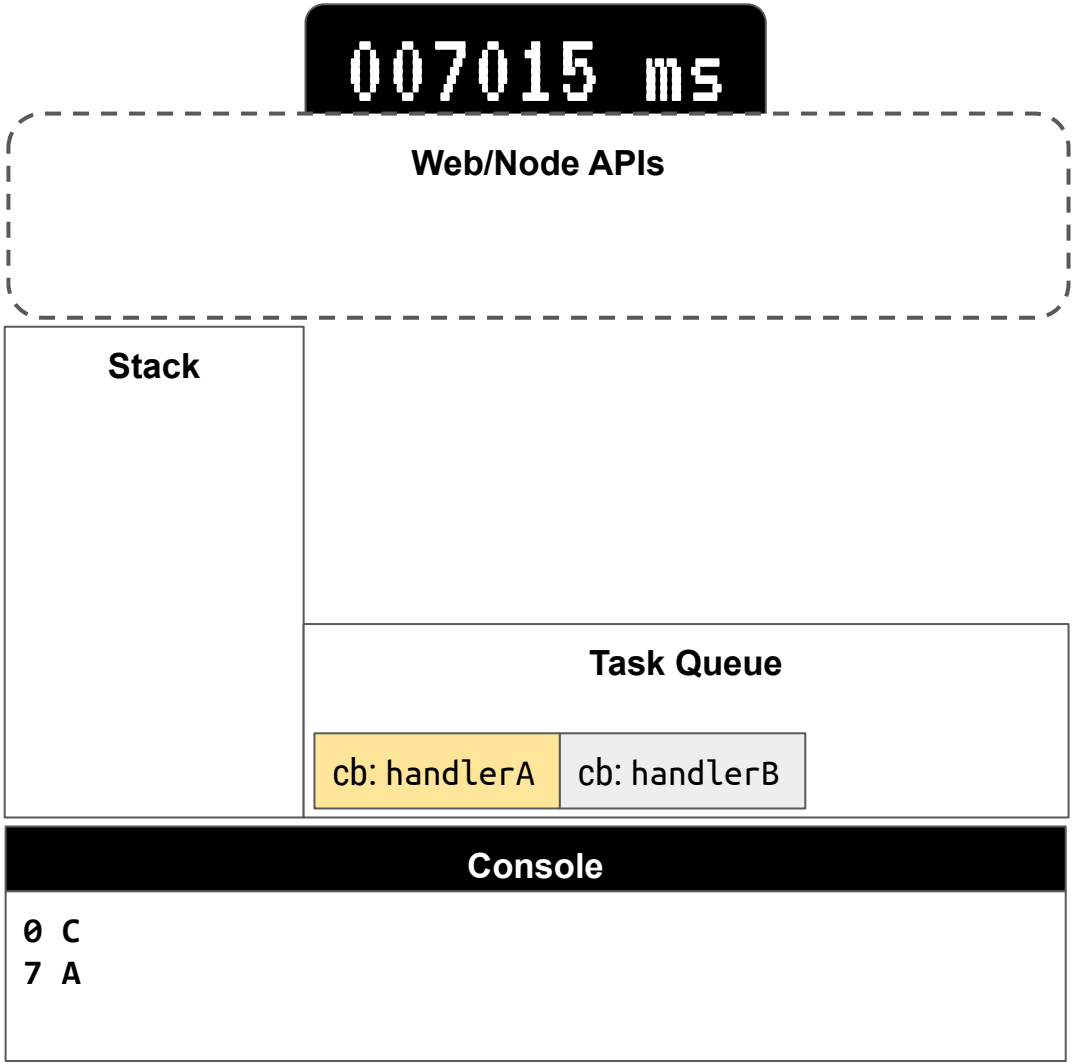
```
01: function waitSync(delayMs) {
02:   const waitStart = Date.now();
03:   while (Date.now() - waitStart < delayMs) {}
04: }
05:
06: const start = Date.now();
07: function elapsedTimeLog(...args) {
08:   const elapsedMs = Date.now() - start;
09:   const sec = Math.floor(elapsedMs / 1000);
10:   console.log(sec, ...args);
11: }
12:
13: function handlerA() {
14:   waitSync(6000);
15:   elapsedTimeLog("A");
16: }
17:
18: function handlerB() {
19:   elapsedTimeLog("B");
20: }
21:
22: setTimeout(handlerA, 1000);
23: setTimeout(handlerB, 2000);
24:
25: elapsedTimeLog("C");
```



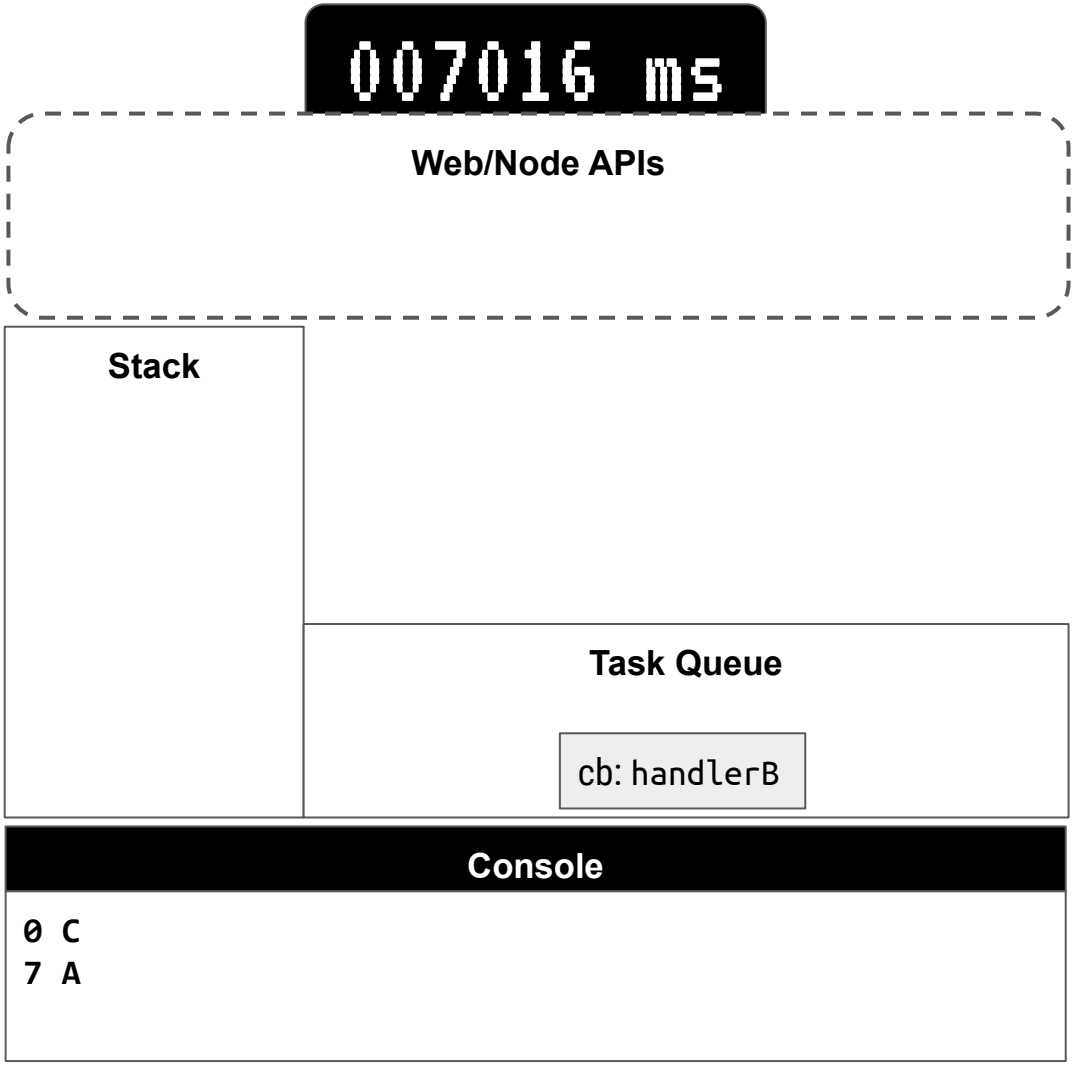
```
01: function waitSync(delayMs) {
02:   const waitStart = Date.now();
03:   while (Date.now() - waitStart < delayMs) {}
04: }
05:
06: const start = Date.now();
07: function elapsedTimeLog(...args) {
08:   const elapsedMs = Date.now() - start;
09:   const sec = Math.floor(elapsedMs / 1000);
10:   console.log(sec, ...args);
11: }
12:
13: function handlerA() {
14:   waitSync(6000);
15:   elapsedTimeLog("A");
16: }
17:
18: function handlerB() {
19:   elapsedTimeLog("B");
20: }
21:
22: setTimeout(handlerA, 1000);
23: setTimeout(handlerB, 2000);
24:
25: elapsedTimeLog("C");
```



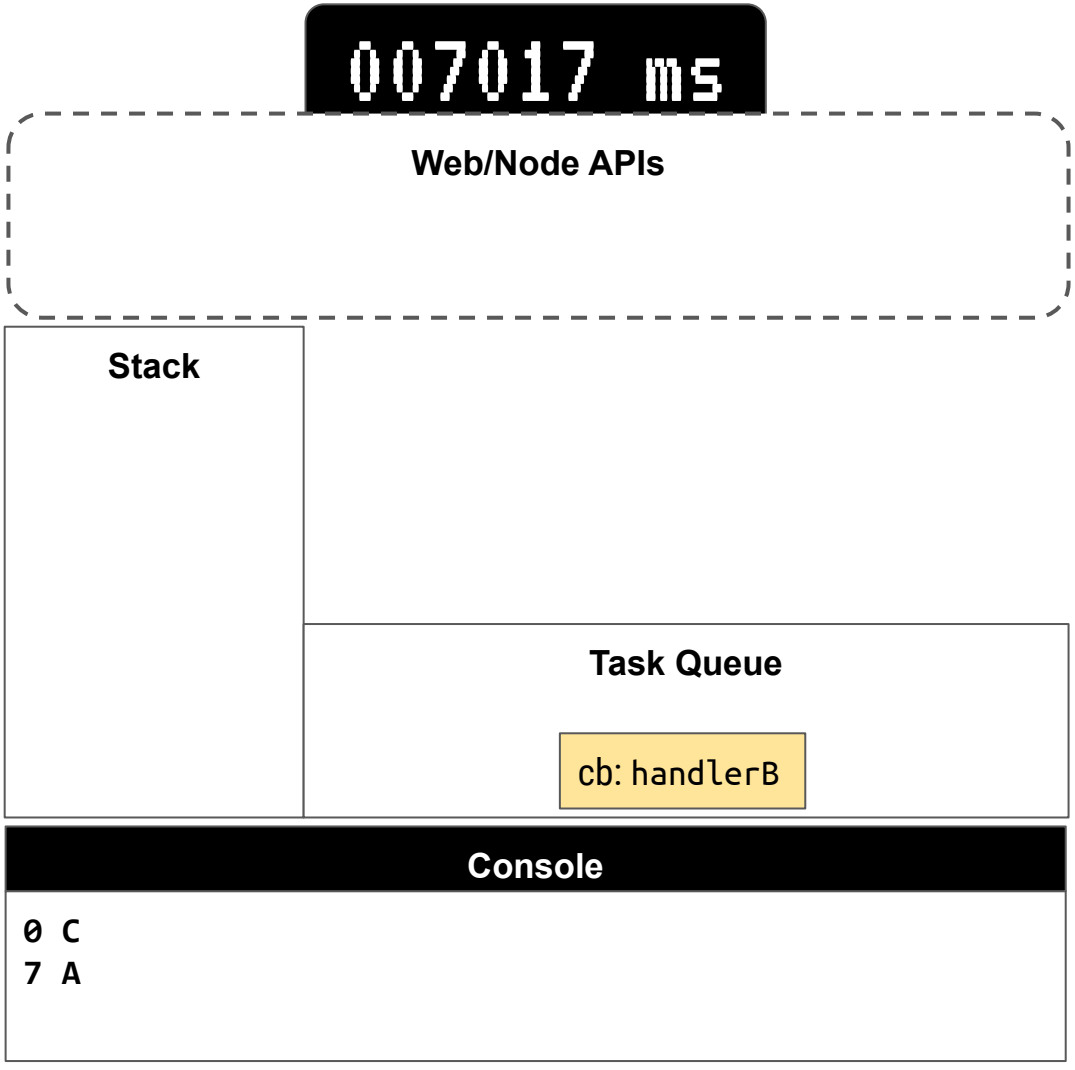
```
01: function waitSync(delayMs) {
02:   const waitStart = Date.now();
03:   while (Date.now() - waitStart < delayMs) {}
04: }
05:
06: const start = Date.now();
07: function elapsedTimeLog(...args) {
08:   const elapsedMs = Date.now() - start;
09:   const sec = Math.floor(elapsedMs / 1000);
10:   console.log(sec, ...args);
11: }
12:
13: function handlerA() {
14:   waitSync(6000);
15:   elapsedTimeLog("A");
16: }
17:
18: function handlerB() {
19:   elapsedTimeLog("B");
20: }
21:
22: setTimeout(handlerA, 1000);
23: setTimeout(handlerB, 2000);
24:
25: elapsedTimeLog("C");
```



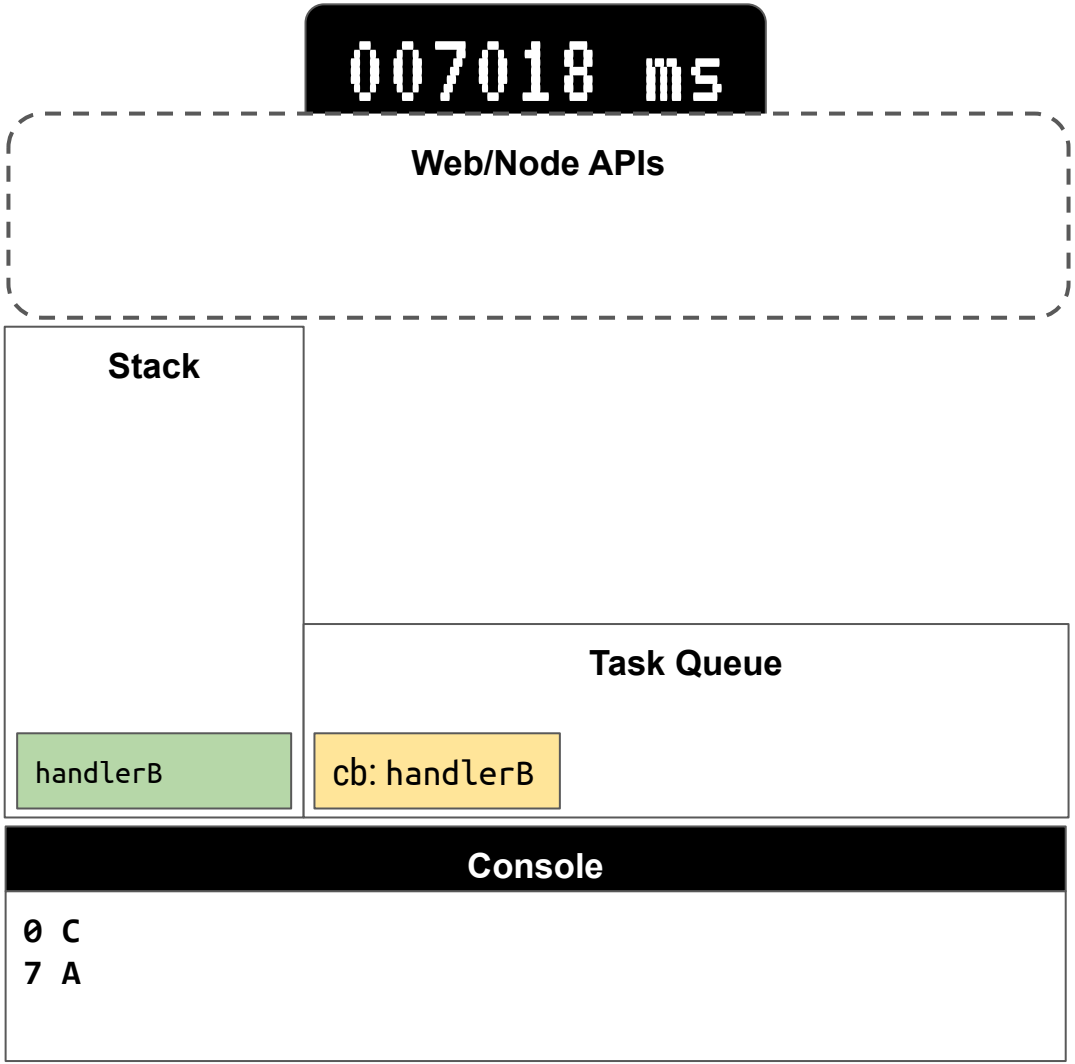
```
01: function waitSync(delayMs) {
02:   const waitStart = Date.now();
03:   while (Date.now() - waitStart < delayMs) {}
04: }
05:
06: const start = Date.now();
07: function elapsedTimeLog(...args) {
08:   const elapsedMs = Date.now() - start;
09:   const sec = Math.floor(elapsedMs / 1000);
10:   console.log(sec, ...args);
11: }
12:
13: function handlerA() {
14:   waitSync(6000);
15:   elapsedTimeLog("A");
16: }
17:
18: function handlerB() {
19:   elapsedTimeLog("B");
20: }
21:
22: setTimeout(handlerA, 1000);
23: setTimeout(handlerB, 2000);
24:
25: elapsedTimeLog("C");
```



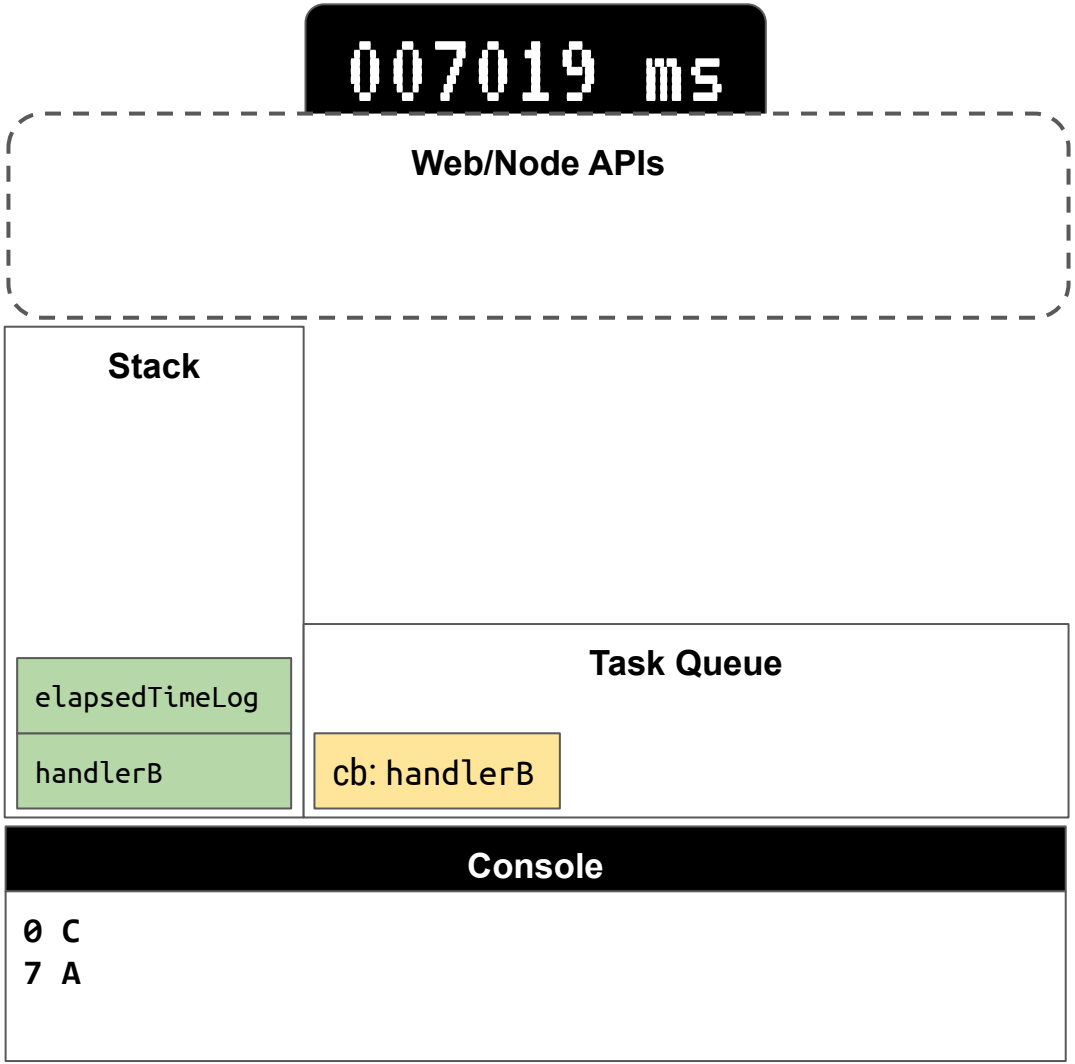
```
01: function waitSync(delayMs) {
02:   const waitStart = Date.now();
03:   while (Date.now() - waitStart < delayMs) {}
04: }
05:
06: const start = Date.now();
07: function elapsedTimeLog(...args) {
08:   const elapsedMs = Date.now() - start;
09:   const sec = Math.floor(elapsedMs / 1000);
10:   console.log(sec, ...args);
11: }
12:
13: function handlerA() {
14:   waitSync(6000);
15:   elapsedTimeLog("A");
16: }
17:
18: function handlerB() {
19:   elapsedTimeLog("B");
20: }
21:
22: setTimeout(handlerA, 1000);
23: setTimeout(handlerB, 2000);
24:
25: elapsedTimeLog("C");
```



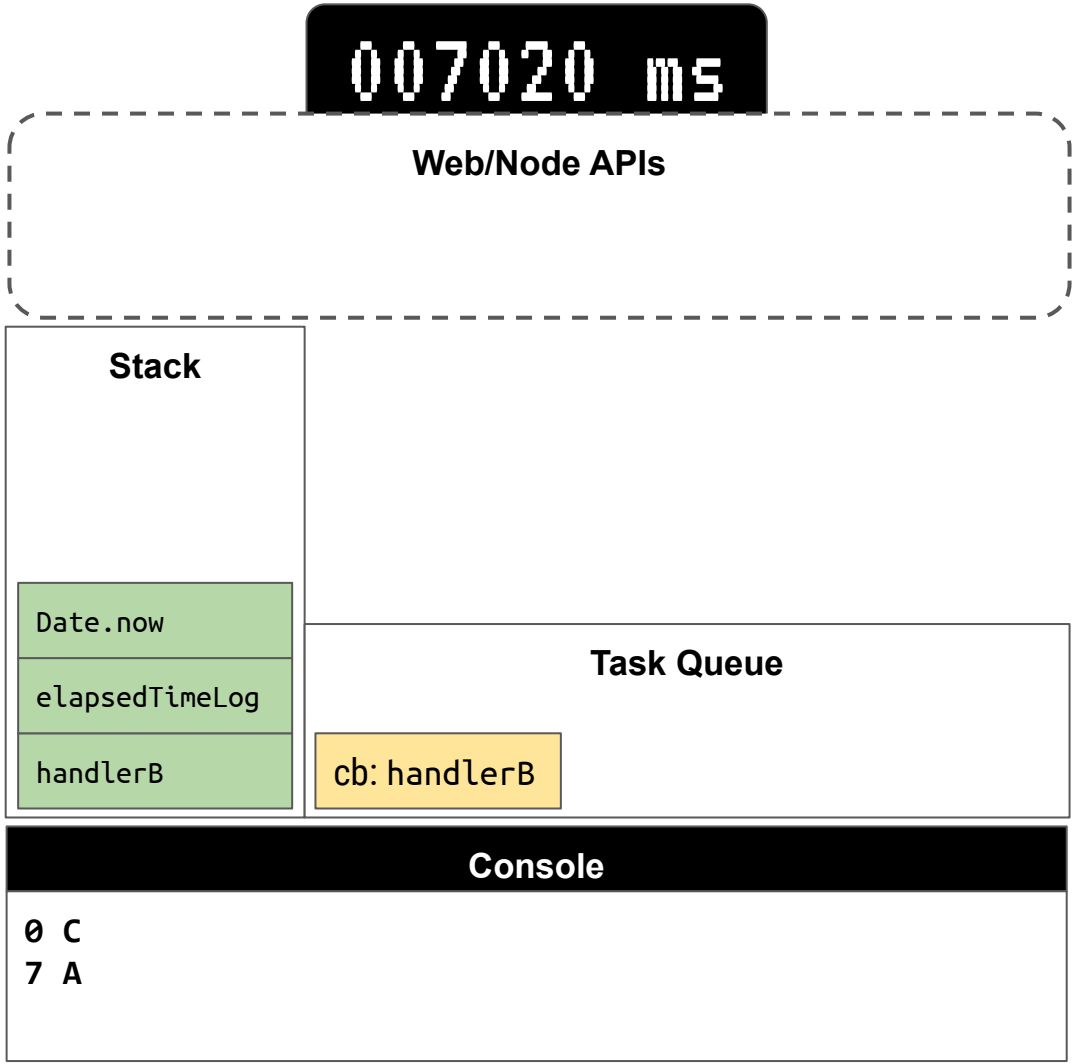
```
01: function waitSync(delayMs) {
02:   const waitStart = Date.now();
03:   while (Date.now() - waitStart < delayMs) {}
04: }
05:
06: const start = Date.now();
07: function elapsedTimeLog(...args) {
08:   const elapsedMs = Date.now() - start;
09:   const sec = Math.floor(elapsedMs / 1000);
10:   console.log(sec, ...args);
11: }
12:
13: function handlerA() {
14:   waitSync(6000);
15:   elapsedTimeLog("A");
16: }
17:
18: function handlerB() {
19:   elapsedTimeLog("B");
20: }
21:
22: setTimeout(handlerA, 1000);
23: setTimeout(handlerB, 2000);
24:
25: elapsedTimeLog("C");
```



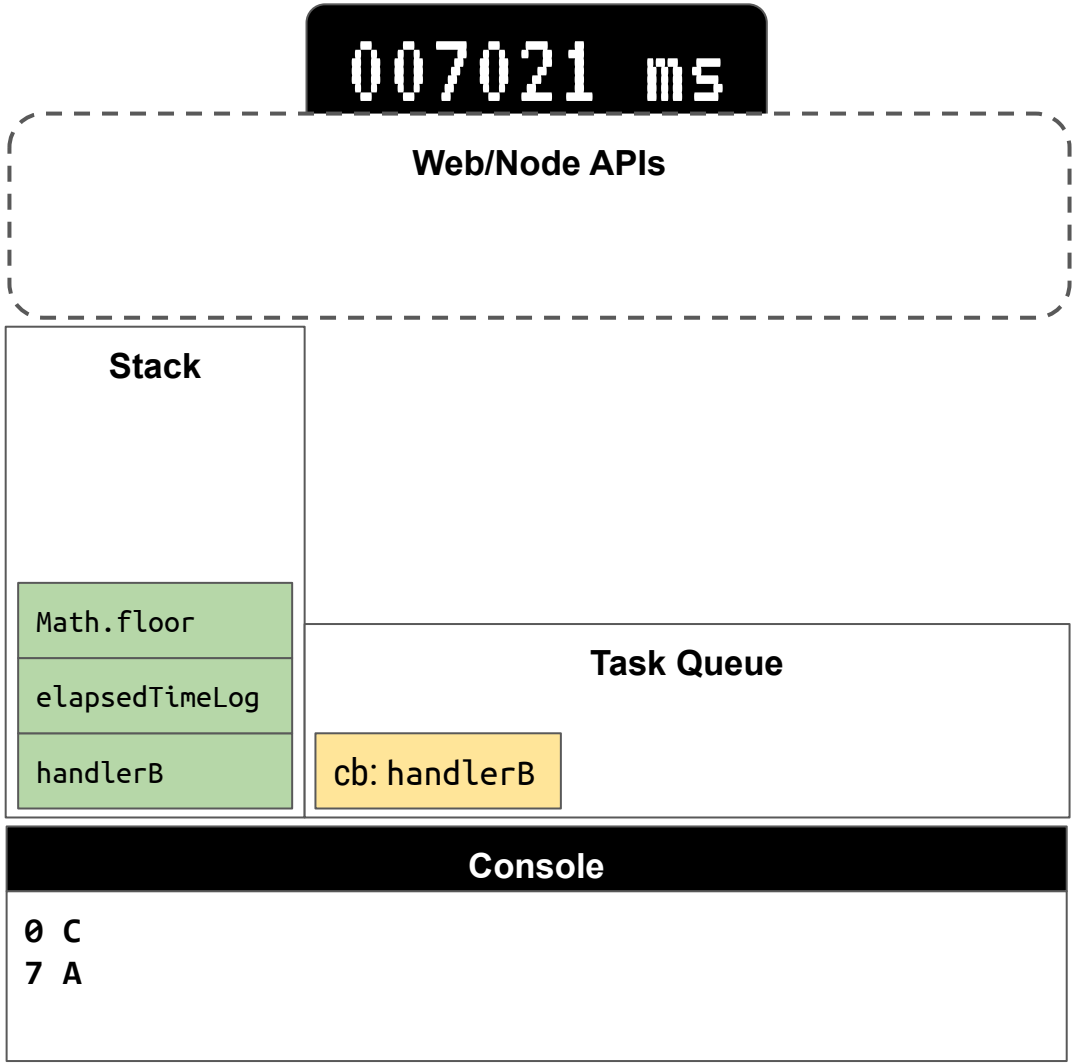
```
01: function waitSync(delayMs) {
02:   const waitStart = Date.now();
03:   while (Date.now() - waitStart < delayMs) {}
04: }
05:
06: const start = Date.now();
07: function elapsedTimeLog(...args) {
08:   const elapsedMs = Date.now() - start;
09:   const sec = Math.floor(elapsedMs / 1000);
10:   console.log(sec, ...args);
11: }
12:
13: function handlerA() {
14:   waitSync(6000);
15:   elapsedTimeLog("A");
16: }
17:
18: function handlerB() {
19:   elapsedTimeLog("B");
20: }
21:
22: setTimeout(handlerA, 1000);
23: setTimeout(handlerB, 2000);
24:
25: elapsedTimeLog("C");
```



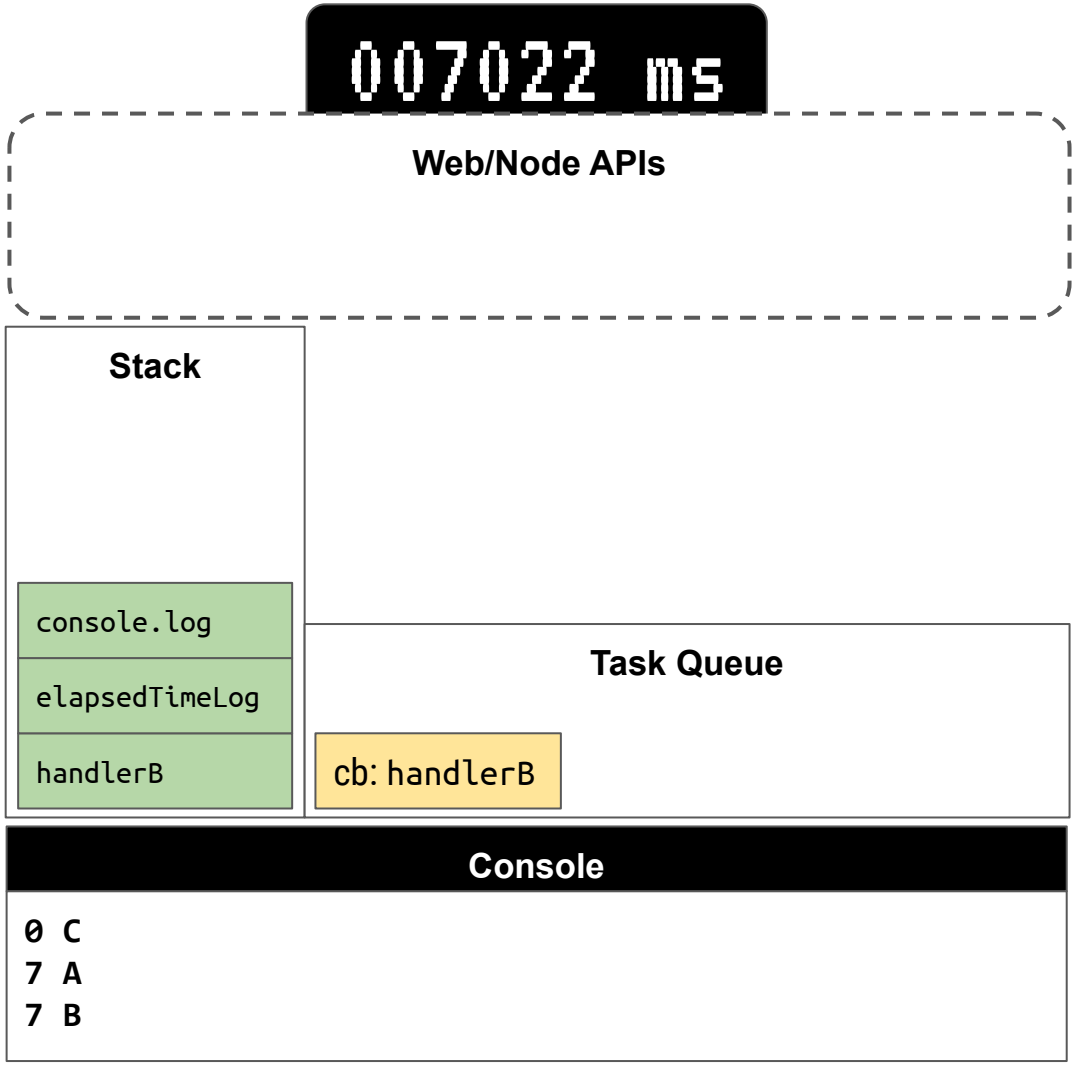
```
01: function waitSync(delayMs) {
02:   const waitStart = Date.now();
03:   while (Date.now() - waitStart < delayMs) {}
04: }
05:
06: const start = Date.now();
07: function elapsedTimeLog(...args) {
08:   const elapsedMs = Date.now() - start;
09:   const sec = Math.floor(elapsedMs / 1000);
10:   console.log(sec, ...args);
11: }
12:
13: function handlerA() {
14:   waitSync(6000);
15:   elapsedTimeLog("A");
16: }
17:
18: function handlerB() {
19:   elapsedTimeLog("B");
20: }
21:
22: setTimeout(handlerA, 1000);
23: setTimeout(handlerB, 2000);
24:
25: elapsedTimeLog("C");
```



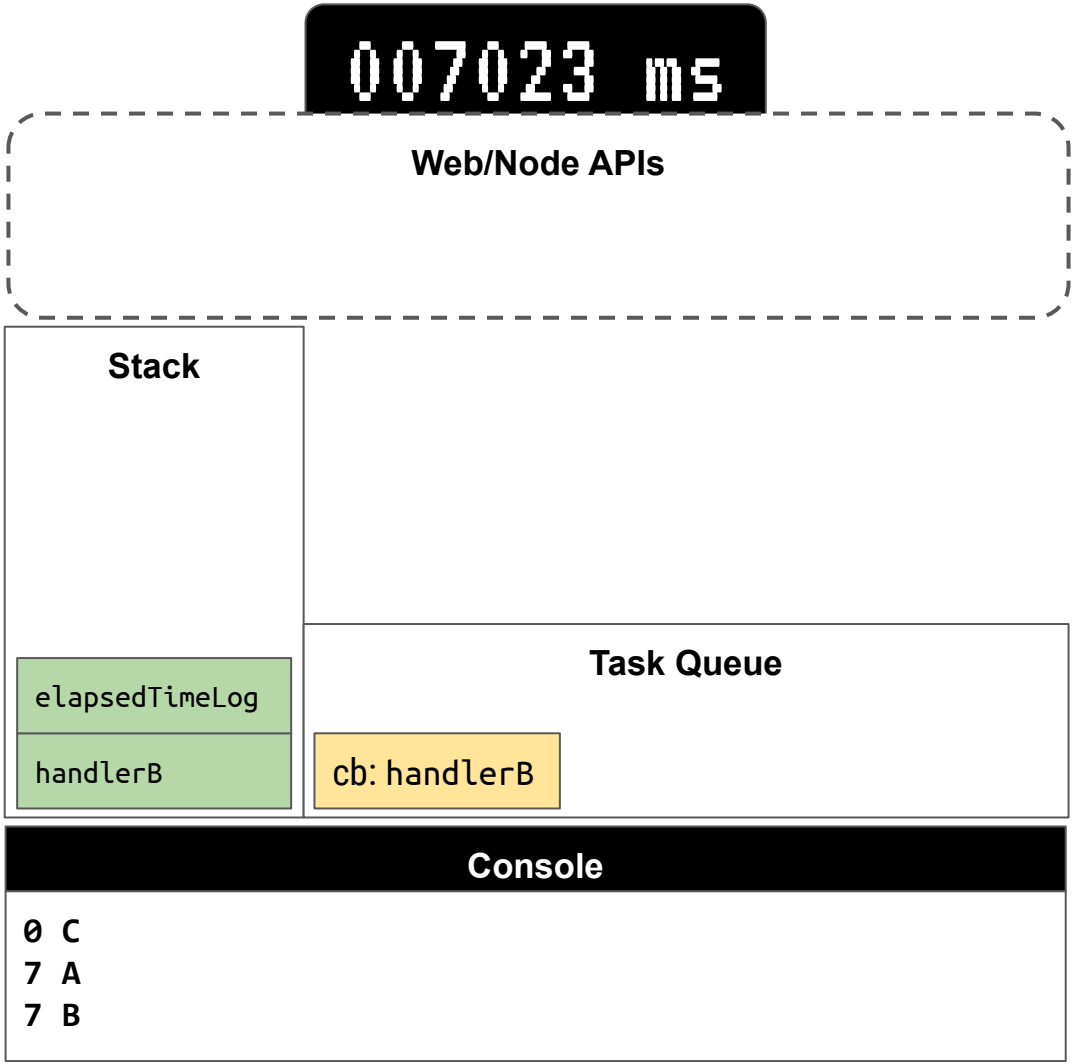
```
01: function waitSync(delayMs) {
02:   const waitStart = Date.now();
03:   while (Date.now() - waitStart < delayMs) {}
04: }
05:
06: const start = Date.now();
07: function elapsedTimeLog(...args) {
08:   const elapsedMs = Date.now() - start;
09:   const sec = Math.floor(elapsedMs / 1000);
10:   console.log(sec, ...args);
11: }
12:
13: function handlerA() {
14:   waitSync(6000);
15:   elapsedTimeLog("A");
16: }
17:
18: function handlerB() {
19:   elapsedTimeLog("B");
20: }
21:
22: setTimeout(handlerA, 1000);
23: setTimeout(handlerB, 2000);
24:
25: elapsedTimeLog("C");
```



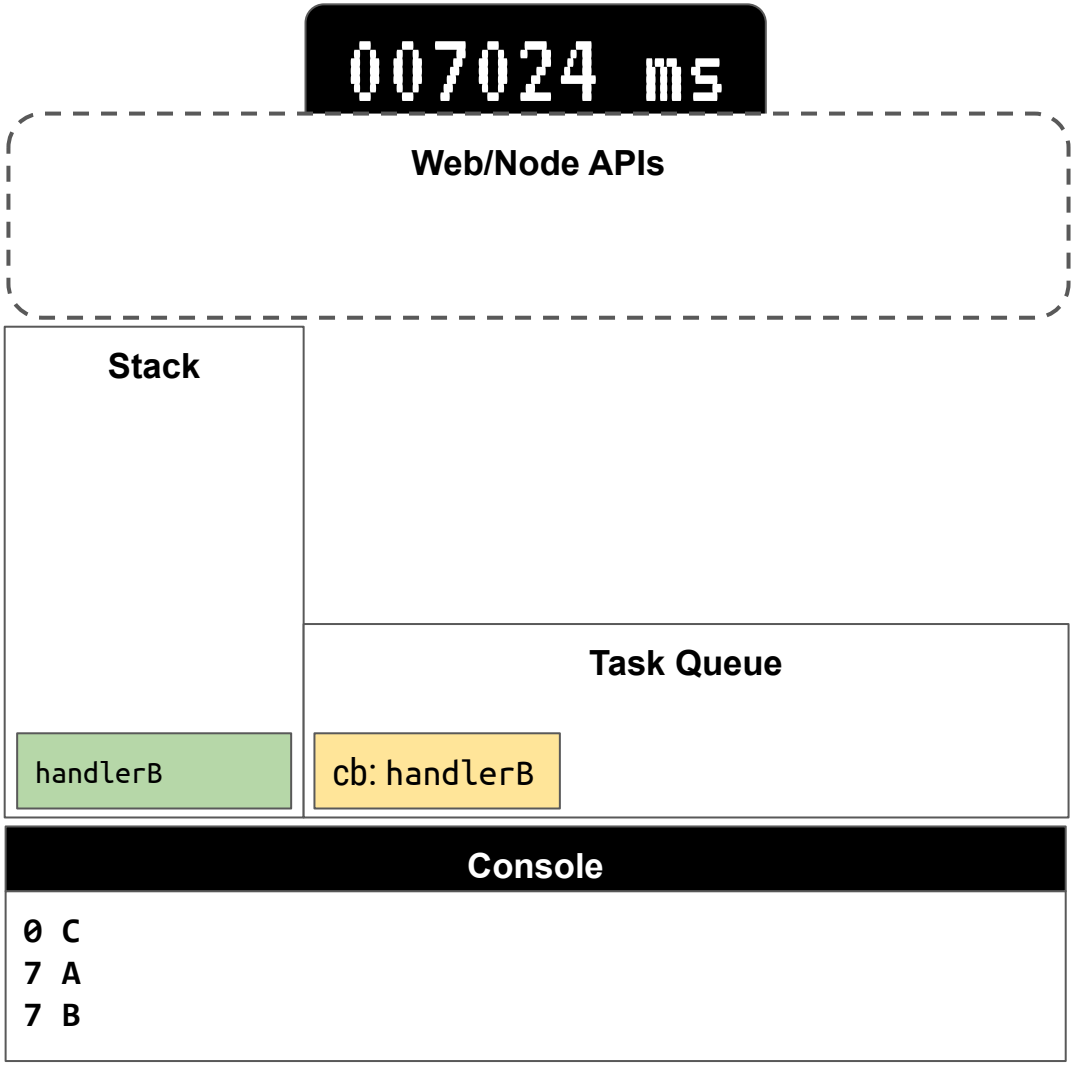
```
01: function waitSync(delayMs) {
02:   const waitStart = Date.now();
03:   while (Date.now() - waitStart < delayMs) {}
04: }
05:
06: const start = Date.now();
07: function elapsedTimeLog(...args) {
08:   const elapsedMs = Date.now() - start;
09:   const sec = Math.floor(elapsedMs / 1000);
10:   console.log(sec, ...args);
11: }
12:
13: function handlerA() {
14:   waitSync(6000);
15:   elapsedTimeLog("A");
16: }
17:
18: function handlerB() {
19:   elapsedTimeLog("B");
20: }
21:
22: setTimeout(handlerA, 1000);
23: setTimeout(handlerB, 2000);
24:
25: elapsedTimeLog("C");
```



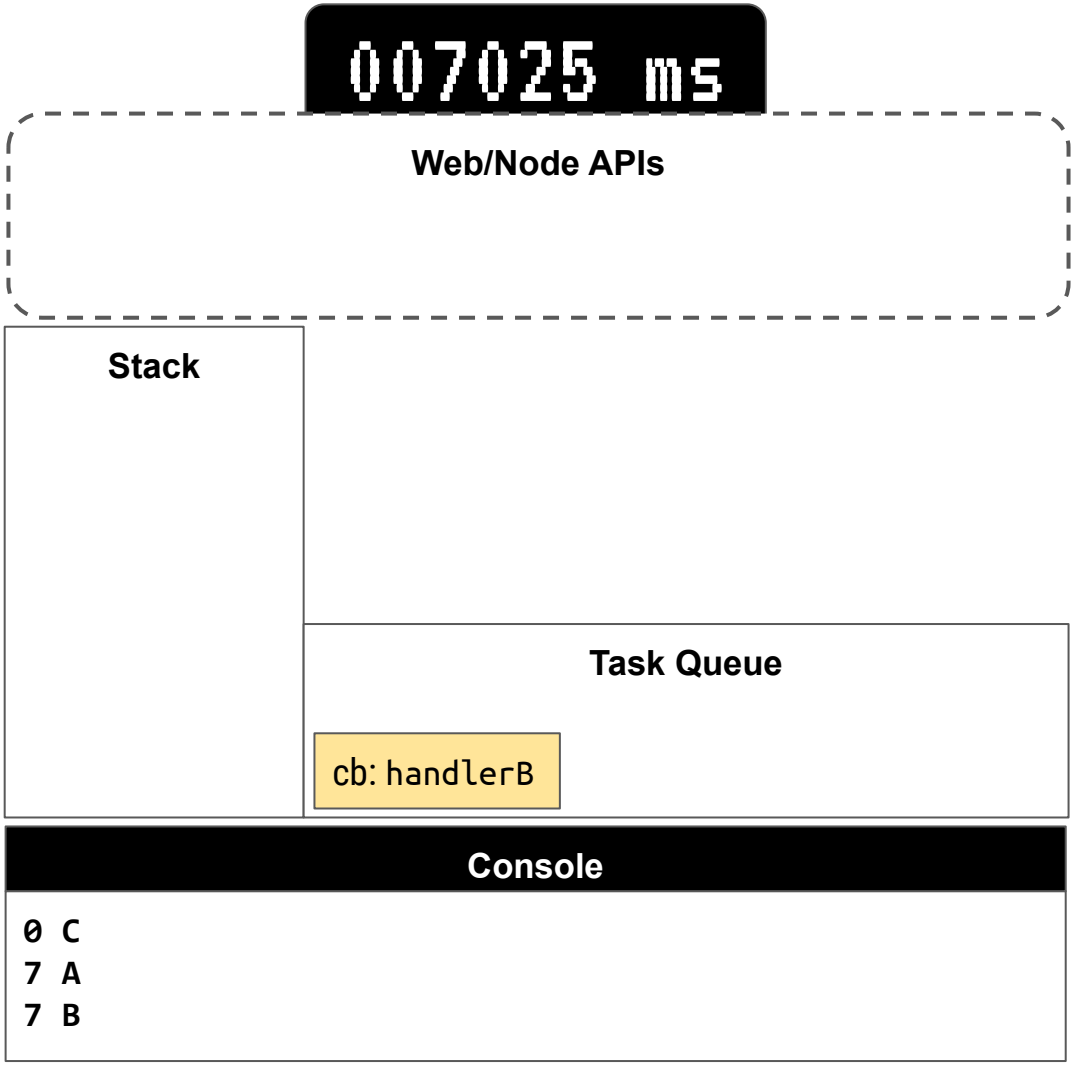
```
01: function waitSync(delayMs) {
02:   const waitStart = Date.now();
03:   while (Date.now() - waitStart < delayMs) {}
04: }
05:
06: const start = Date.now();
07: function elapsedTimeLog(...args) {
08:   const elapsedMs = Date.now() - start;
09:   const sec = Math.floor(elapsedMs / 1000);
10:   console.log(sec, ...args);
11: }
12:
13: function handlerA() {
14:   waitSync(6000);
15:   elapsedTimeLog("A");
16: }
17:
18: function handlerB() {
19:   elapsedTimeLog("B");
20: }
21:
22: setTimeout(handlerA, 1000);
23: setTimeout(handlerB, 2000);
24:
25: elapsedTimeLog("C");
```



```
01: function waitSync(delayMs) {
02:   const waitStart = Date.now();
03:   while (Date.now() - waitStart < delayMs) {}
04: }
05:
06: const start = Date.now();
07: function elapsedTimeLog(...args) {
08:   const elapsedMs = Date.now() - start;
09:   const sec = Math.floor(elapsedMs / 1000);
10:   console.log(sec, ...args);
11: }
12:
13: function handlerA() {
14:   waitSync(6000);
15:   elapsedTimeLog("A");
16: }
17:
18: function handlerB() {
19:   elapsedTimeLog("B");
20: }
21:
22: setTimeout(handlerA, 1000);
23: setTimeout(handlerB, 2000);
24:
25: elapsedTimeLog("C");
```



```
01: function waitSync(delayMs) {
02:   const waitStart = Date.now();
03:   while (Date.now() - waitStart < delayMs) {}
04: }
05:
06: const start = Date.now();
07: function elapsedTimeLog(...args) {
08:   const elapsedMs = Date.now() - start;
09:   const sec = Math.floor(elapsedMs / 1000);
10:   console.log(sec, ...args);
11: }
12:
13: function handlerA() {
14:   waitSync(6000);
15:   elapsedTimeLog("A");
16: }
17:
18: function handlerB() {
19:   elapsedTimeLog("B");
20: }
21:
22: setTimeout(handlerA, 1000);
23: setTimeout(handlerB, 2000);
24:
25: elapsedTimeLog("C");
```



```
01: function waitSync(delayMs) {
02:   const waitStart = Date.now();
03:   while (Date.now() - waitStart < delayMs) {}
04: }
05:
06: const start = Date.now();
07: function elapsedTimeLog(...args) {
08:   const elapsedMs = Date.now() - start;
09:   const sec = Math.floor(elapsedMs / 1000);
10:   console.log(sec, ...args);
11: }
12:
13: function handlerA() {
14:   waitSync(6000);
15:   elapsedTimeLog("A");
16: }
17:
18: function handlerB() {
19:   elapsedTimeLog("B");
20: }
21:
22: setTimeout(handlerA, 1000);
23: setTimeout(handlerB, 2000);
24:
25: elapsedTimeLog("C");
```