# example.js

```
document.body.innerHTML = `
    <pre id="history"></pre>
    <form>
    <input id="message" type="text">
    <button id="send">Send Message</button>
    </form>
    <p>Computing fibonacci without worker:</p>
    <input id="fib1" type="number">
    <pre id="output1"></pre>
    <p>Computing fibonacci with worker:</p>
    <input id="fib2" type="number">
    <pre id="output2"></pre>
`;

const history = document.getElementById("history");
const message = document.getElementById("message");
const send = document.getElementById("send");
const fib1 = document.getElementById("fib1");
const output1 = document.getElementById("output1");
const fib2 = document.getElementById("fib2");
const output2 = document.getElementById("output2");

/// CHAT with shared worker ///

const chatWorker = new SharedWorker(
    new URL("./chat-worker.js", import.meta.url),
    {
        name: "chat",
        type: "module"
    }
);

let historyTimeout;
const scheduleUpdateHistory = () => {
    clearTimeout(historyTimeout);
    historyTimeout = setTimeout(() => {
        chatWorker.port.postMessage({ type: "history" });
    }, 1000);
};
scheduleUpdateHistory();

const from = `User ${Math.floor(Math.random() * 10000)}`;

send.addEventListener("click", e => {
    chatWorker.port.postMessage({
        type: "message",
        content: message.value,
        from
    });
    message.value = "";
    message.focus();
    e.preventDefault();
});

chatWorker.port.onmessage = event => {
    const msg = event.data;
    switch (msg.type) {
        case "history":
            history.innerText = msg.history.join("\n");
            scheduleUpdateHistory();
            break;
    }
};

/// FIBONACCI without worker ///

fib1.addEventListener("change", async () => {
    try {
        const value = parseInt(fib1.value, 10);
        const { fibonacci } = await import("./fibonacci");
        const result = fibonacci(value);
        output1.innerText = `fib(${value}) = ${result}`;
    } catch (e) {
        output1.innerText = e.message;
    }
});

/// FIBONACCI with worker ///

const fibWorker = new Worker(new URL("./fib-worker.js", import.meta.url), {
    name: "fibonacci",
    type: "module"
    /* webpackEntryOptions: { filename: "workers/[name].js" } */
});

fib2.addEventListener("change", () => {
    try {
        const value = parseInt(fib2.value, 10);
        fibWorker.postMessage(`${value}`);
    } catch (e) {
        output2.innerText = e.message;
    }
});

fibWorker.onmessage = event => {
    output2.innerText = event.data;
};
```

# fib-worker.js

```
onmessage = async event => {
    const { fibonacci } = await import("./fibonacci");
    const value = JSON.parse(event.data);
    postMessage(`fib(${value}) = ${fibonacci(value)}`);
};
```

# fibonacci.js

```
export function fibonacci(n) {
    return n < 1 ? 0 : n <= 2 ? 1 : fibonacci(n - 1) + fibonacci(n - 2);
}
```

# chat-worker.js

```
onconnect = function (e) {
    for (const port of e.ports) {
        port.onmessage = async event => {
            const msg = event.data;
            switch (msg.type) {
                case "message":
                    const { add } = await import("./chat-module");
                    add(msg.content, msg.from);
                // fallthrough
                case "history":
                    const { history } = await import("./chat-module");
                    port.postMessage({
                        type: "history",
                        history
                    });
                    break;
            }
        };
    }
};
```

# chat-module.js

```
export const history = [];

export const add = (content, from) => {
    if (history.length > 10) history.shift();
    history.push(`${from}: ${content}`);
};
```

# dist/main.js

```
/******/ var __webpack_modules__ = ({});
```

▶ `/* webpack runtime code */`

```
var __webpack_exports__ = {};
/*!********************!*\
  !*** ./example.js ***!
  \********************/
/*! unknown exports (runtime-defined) */
/*! runtime requirements: __webpack_require__.p, __webpack_require__.b,
__webpack_require__.u, __webpack_require__.e, __webpack_require__,
__webpack_require__.* */
document.body.innerHTML = `
    <pre id="history"></pre>
    <form>
    <input id="message" type="text">
    <button id="send">Send Message</button>
    </form>
    <p>Computing fibonacci without worker:</p>
    <input id="fib1" type="number">
    <pre id="output1"></pre>
    <p>Computing fibonacci with worker:</p>
    <input id="fib2" type="number">
    <pre id="output2"></pre>
`;

const history = document.getElementById("history");
const message = document.getElementById("message");
const send = document.getElementById("send");
const fib1 = document.getElementById("fib1");
const output1 = document.getElementById("output1");
const fib2 = document.getElementById("fib2");
const output2 = document.getElementById("output2");

/// CHAT with shared worker ///

const chatWorker = new SharedWorker(
    new URL(/* worker import */ __webpack_require__.p + __webpack_require__.u(348),
__webpack_require__.b),
    {
        name: "chat",
        type: "module"
    }
```

```
);

let historyTimeout;
const scheduleUpdateHistory = () => {
    clearTimeout(historyTimeout);
    historyTimeout = setTimeout(() => {
        chatWorker.port.postMessage({ type: "history" });
    }, 1000);
};
scheduleUpdateHistory();

const from = `User ${Math.floor(Math.random() * 10000)}`;

send.addEventListener("click", e => {
    chatWorker.port.postMessage({
        type: "message",
        content: message.value,
        from
    });
    message.value = "";
    message.focus();
    e.preventDefault();
});

chatWorker.port.onmessage = event => {
    const msg = event.data;
    switch (msg.type) {
        case "history":
            history.innerText = msg.history.join("\n");
            scheduleUpdateHistory();
            break;
    }
};

/// FIBONACCI without worker ///

fib1.addEventListener("change", async () => {
    try {
        const value = parseInt(fib1.value, 10);
        const { fibonacci } = await __webpack_require__.e(/*! import() */
129).then(__webpack_require__.bind(__webpack_require__, /*! ./fibonacci */ 2));
        const result = fibonacci(value);
        output1.innerText = `fib(${value}) = ${result}`;
    } catch (e) {
        output1.innerText = e.message;
    }
});

/// FIBONACCI with worker ///

const fibWorker = new Worker(new URL(/* worker import */ __webpack_require__.p +
__webpack_require__.u(631), __webpack_require__.b), {
    name: "fibonacci",
    type: "module"
    /* webpackEntryOptions: { filename: "workers/[name].js" } */
});

fib2.addEventListener("change", () => {
    try {
        const value = parseInt(fib2.value, 10);
        fibWorker.postMessage(`${value}`);
    } catch (e) {
        output2.innerText = e.message;
    }
});

fibWorker.onmessage = event => {
    output2.innerText = event.data;
};
```

## dist/chat.js

```
/******/ var __webpack_modules__ = ({});
```

▶ /* webpack runtime code */

```
var __webpack_exports__ = {};
/*!************************!*\
  !*** ./chat-worker.js ***!
  \************************/
/*! unknown exports (runtime-defined) */
/*! runtime requirements: __webpack_require__.e, __webpack_require__,
__webpack_require__.* */
onconnect = function (e) {
    for (const port of e.ports) {
        port.onmessage = async event => {
            const msg = event.data;
            switch (msg.type) {
                case "message":
                    const { add } = await __webpack_require__.e(/*! import() */
192).then(__webpack_require__.bind(__webpack_require__, /*! ./chat-module */ 4));
                    add(msg.content, msg.from);
                // fallthrough
                case "history":
                    const { history } = await __webpack_require__.e(/*! import() */
192).then(__webpack_require__.bind(__webpack_require__, /*! ./chat-module */ 4));
                    port.postMessage({
                        type: "history",
                        history
                    });
                    break;
            }
        };
    }
};
```

```
var e,o,t={},r={};function n(e){var o=r[e];if(void 0!==o)return o.exports;var
s=r[e]={exports:{}};return t[e](s,s.exports,n),s.exports}n.m=t,n.d=(e,o)=>{for(var t
in o)n.o(o,t)&&!n.o(e,t)&&Object.defineProperty(e,t,{enumerable:!0,get:o[t]})},n.f=
{},n.e=e=>Promise.all(Object.keys(n.f).reduce(((o,t)=>(n.f[t](e,o),o)),
[])),n.u=e=>e+".js",n.o=(e,o)=>Object.prototype.hasOwnProperty.call(e,o),n.r=e=>
{"undefined"!=typeof
Symbol&&Symbol.toStringTag&&Object.defineProperty(e,Symbol.toStringTag,
{value:"Module"}),Object.defineProperty(e,"__esModule",{value:!0})},e={348:0},o=o=>
{var t,r,{ids:s,modules:i,runtime:a}=o,c=0;for(t in i)n.o(i,t)&&
(n.m[t]=i[t]);for(a&&a(n);c<s.length;c++)r=s[c],n.o(e,r)&&e[r]&&e[r][0]
(),e[s[c]]=0},n.f.j=(t,r)=>{var s=n.o(e,t)?e[t]:void
0;if(0!==s)if(s)r.push(s[1]);else{var i=import("./"+n.u(t)).then(o,(o=>{throw
0!==e[t]&&(e[t]=void 0),o}));i=Promise.race([i,new Promise((o=>s=e[t]=
[o]))]),r.push(s[1]=i)}},onconnect=function(e){for(const o of
e.ports)o.onmessage=async e=>{const t=e.data;switch(t.type)
{case"message":const{add:e}=await
n.e(192).then(n.bind(n,192));e(t.content,t.from);case"history":const{history:r}=await
 n.e(192).then(n.bind(n,192));o.postMessage({type:"history",history:r})}}};
```

# dist/workers/fibonacci.js

```
/******/ var __webpack_modules__ = ({});
```

▶ /* webpack runtime code */

```
var __webpack_exports__ = {};
/*!***********************!*\
  !*** ./fib-worker.js ***!
  \***********************/
/*! unknown exports (runtime-defined) */
/*! runtime requirements: __webpack_require__.e, __webpack_require__,
__webpack_require__.* */
onmessage = async event => {
    const { fibonacci } = await __webpack_require__.e(/*! import() */
129).then(__webpack_require__.bind(__webpack_require__, /*! ./fibonacci */ 2));
    const value = JSON.parse(event.data);
    postMessage(`fib(${value}) = ${fibonacci(value)}`);
};
```

```
var e,o,r={},t={};function i(e){var o=t[e];if(void 0!==o)return o.exports;var
a=t[e]={exports:{}};return r[e](a,a.exports,i),a.exports}i.m=r,i.d=(e,o)=>{for(var r
in o)i.o(o,r)&&!i.o(e,r)&&Object.defineProperty(e,r,{enumerable:!0,get:o[r]})},i.f=
{},i.e=e=>Promise.all(Object.keys(i.f).reduce(((o,r)=>(i.f[r](e,o),o)),
[])),i.u=e=>e+".js",i.o=(e,o)=>Object.prototype.hasOwnProperty.call(e,o),i.r=e=>
{"undefined"!=typeof
Symbol&&Symbol.toStringTag&&Object.defineProperty(e,Symbol.toStringTag,
{value:"Module"}),Object.defineProperty(e,"__esModule",{value:!0})},e={631:0},o=o=>
{var r,t,{ids:a,modules:n,runtime:s}=o,f=0;for(r in n)i.o(n,r)&&
(i.m[r]=n[r]);for(s&&s(i);f<a.length;f++)t=a[f],i.o(e,t)&&e[t]&&e[t][0]
(),e[a[f]]=0},i.f.j=(r,t)=>{var a=i.o(e,r)?e[r]:void
0;if(0!==a)if(a)t.push(a[1]);else{var n=import("../"+i.u(r)).then(o,(o=>{throw
0!==e[r]&&(e[r]=void 0),o}));n=Promise.race([n,new Promise((o=>a=e[r]=
[o]))]),t.push(a[1]=n)}},onmessage=async e=>{const{fibonacci:o}=await
i.e(129).then(i.bind(i,129)),r=JSON.parse(e.data);postMessage(`fib(${r}) =
${o(r)}`)};
```

# dist/129.js

```
export const id = 129;
export const ids = [129];
export const modules = {

/***/ 2:
/*!**********************!*\
  !*** ./fibonacci.js ***!
  \**********************/
/*! namespace exports */
/*! export fibonacci [provided] [no usage info] [missing usage info prevents
renaming] */
/*! other exports [not provided] [no usage info] */
/*! runtime requirements: __webpack_require__.r, __webpack_exports__,
__webpack_require__.d, __webpack_require__.* */
/***/ ((__unused_webpack_module, __webpack_exports__, __webpack_require__) => {

__webpack_require__.r(__webpack_exports__);
/* harmony export */ __webpack_require__.d(__webpack_exports__, {
/* harmony export */   "fibonacci": () => (/* binding */ fibonacci)
/* harmony export */ });
function fibonacci(n) {
    return n < 1 ? 0 : n <= 2 ? 1 : fibonacci(n - 1) + fibonacci(n - 2);
}


/***/ })

};
```

# Info

## Unoptimized

```
asset main.js 8.56 KiB [emitted] [javascript module] (name: main)
asset chat.js 6.34 KiB [emitted] [javascript module] (name: chat)
asset workers/fibonacci.js 5.99 KiB [emitted] [javascript module] (name: fibonacci)
asset 192.js 1.01 KiB [emitted] [javascript module]
asset 129.js 847 bytes [emitted] [javascript module]
chunk (runtime: 9a81d90cfd0dfd13d748, main) 129.js 103 bytes [rendered]
  > ./fibonacci ./example.js 70:30-51
  > ./fibonacci ./fib-worker.js 2:29-50
  ./fibonacci.js 103 bytes [built] [code generated]
    [exports: fibonacci]
    [used exports unknown]
    import() ./fibonacci ./example.js 70:30-51
    import() ./fibonacci ./fib-worker.js 2:29-50
chunk (runtime: main) main.js (main) 2.25 KiB (javascript) 3.07 KiB (runtime) [entry]
[rendered]
  > ./example.js main
  runtime modules 3.07 KiB 7 modules
  ./example.js 2.25 KiB [built] [code generated]
    [used exports unknown]
    entry ./example.js main
chunk (runtime: 1fad8bf8de78b0a77bfd) 192.js 152 bytes [rendered]
  > ./chat-module ./chat-worker.js 11:31-54
  > ./chat-module ./chat-worker.js 7:27-50
  ./chat-module.js 152 bytes [built] [code generated]
    [exports: add, history]
    [used exports unknown]
    import() ./chat-module ./chat-worker.js 7:27-50
    import() ./chat-module ./chat-worker.js 11:31-54
chunk (runtime: 1fad8bf8de78b0a77bfd) chat.js (chat) 442 bytes (javascript) 2.86 KiB
(runtime) [entry] [rendered]
  > ./example.js 25:19-31:1
  runtime modules 2.86 KiB 6 modules
  ./chat-worker.js 442 bytes [built] [code generated]
    [used exports unknown]
    new Worker() ./chat-worker.js ./example.js 25:19-31:1
chunk (runtime: 9a81d90cfd0dfd13d748) workers/fibonacci.js (fibonacci) 176 bytes
(javascript) 2.87 KiB (runtime) [entry] [rendered]
  > ./example.js 80:18-84:2
  runtime modules 2.87 KiB 6 modules
  ./fib-worker.js 176 bytes [built] [code generated]
    [used exports unknown]
    new Worker() ./fib-worker.js ./example.js 80:18-84:2
webpack 5.51.1 compiled successfully
```

## Production mode

```
asset main.js 2.5 KiB [emitted] [javascript module] [minimized] (name: main)
asset chat.js 1.19 KiB [emitted] [javascript module] [minimized] (name: chat)
asset workers/fibonacci.js 1.04 KiB [emitted] [javascript module] [minimized] (name:
fibonacci)
asset 192.js 187 bytes [emitted] [javascript module] [minimized]
asset 129.js 161 bytes [emitted] [javascript module] [minimized]
chunk (runtime: 9a81d90cfd0dfd13d748, main) 129.js 103 bytes [rendered]
  > ./fibonacci ./example.js 70:30-51
  > ./fibonacci ./fib-worker.js 2:29-50
  ./fibonacci.js 103 bytes [built] [code generated]
    [exports: fibonacci]
    import() ./fibonacci ./example.js 70:30-51
    import() ./fibonacci ./fib-worker.js 2:29-50
chunk (runtime: main) main.js (main) 2.25 KiB (javascript) 3.07 KiB (runtime) [entry]
[rendered]
  > ./example.js main
  runtime modules 3.07 KiB 7 modules
  ./example.js 2.25 KiB [built] [code generated]
    [no exports used]
    entry ./example.js main
chunk (runtime: 1fad8bf8de78b0a77bfd) 192.js 152 bytes [rendered]
  > ./chat-module ./chat-worker.js 11:31-54
  > ./chat-module ./chat-worker.js 7:27-50
  ./chat-module.js 152 bytes [built] [code generated]
    [exports: add, history]
    import() ./chat-module ./chat-worker.js 7:27-50
    import() ./chat-module ./chat-worker.js 11:31-54
chunk (runtime: 1fad8bf8de78b0a77bfd) chat.js (chat) 442 bytes (javascript) 2.86 KiB
(runtime) [entry] [rendered]
  > ./example.js 25:19-31:1
  runtime modules 2.86 KiB 6 modules
  ./chat-worker.js 442 bytes [built] [code generated]
    [no exports used]
    new Worker() ./chat-worker.js ./example.js 25:19-31:1
chunk (runtime: 9a81d90cfd0dfd13d748) workers/fibonacci.js (fibonacci) 176 bytes
(javascript) 2.87 KiB (runtime) [entry] [rendered]
  > ./example.js 80:18-84:2
  runtime modules 2.87 KiB 6 modules
  ./fib-worker.js 176 bytes [built] [code generated]
    [no exports used]
    new Worker() ./fib-worker.js ./example.js 80:18-84:2
webpack 5.51.1 compiled successfully
```