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

```
import { history, add } from "./chat-module";

onconnect = function (e) {
    for (const port of e.ports) {
        port.onmessage = event => {
            const msg = event.data;
            switch (msg.type) {
                case "message":
                    add(msg.content, msg.from);
                // fallthrough
                case "history":
                    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

```
/******/ (() => { // webpackBootstrap
/******/ 	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__.* */
/*! ModuleConcatenation bailout: Module is not an ECMAScript module */
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: undefined
    }
);

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: undefined
    /* 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

```
/******/ (() => { // webpackBootstrap
/******/        "use strict";
var __webpack_exports__ = {};
/*!************************************!*\
  !*** ./chat-worker.js + 1 modules ***!
  \************************************/
/*! namespace exports */
/*! runtime requirements:  */

;// CONCATENATED MODULE: ./chat-module.js
const chat_module_history = [];

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

;// CONCATENATED MODULE: ./chat-worker.js


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

/******/ })()
;
```

```
(()=>{"use strict";const s=[];onconnect=function(t){for(const o of
t.ports)o.onmessage=t=>{const e=t.data;switch(e.type)
{case"message":n=e.content,c=e.from,s.length>10&&s.shift(),s.push(`${c}:
${n}`);case"history":o.postMessage({type:"history",history:s})}var n,c}}})();
```

dist/workers/fibonacci.js

```
/******/ (() => { // webpackBootstrap
/******/        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__.* */
/*! ModuleConcatenation bailout: Module is not an ECMAScript module */
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={},r={};function o(t){var a=r[t];if(void 0!==a)return a.exports;var
n=r[t]={exports:{}};return e[t](n,n.exports,o),n.exports}o.m=e,o.d=(e,r)=>{for(var t
in r)o.o(r,t)&&!o.o(e,t)&&Object.defineProperty(e,t,{enumerable:!0,get:r[t]})},o.f=
{},o.e=e=>Promise.all(Object.keys(o.f).reduce(((r,t)=>(o.f[t](e,r),r)),
[])),o.u=e=>e+".js",o.o=(e,r)=>Object.prototype.hasOwnProperty.call(e,r),o.r=e=>
{"undefined"!=typeof
Symbol&&Symbol.toStringTag&&Object.defineProperty(e,Symbol.toStringTag,
{value:"Module"}),Object.defineProperty(e,"__esModule",{value:!0})},o.p="/dist/",
(()=>{var e={631:1};o.f.i=(r,t)=>{e[r]||importScripts(o.p+o.u(r))};var
r=self.webpackChunk=self.webpackChunk||[],t=r.push.bind(r);r.push=r=>
{var[a,n,p]=r;for(var s in n)o.o(n,s)&&
(o.m[s]=n[s]);for(p&&p(o);a.length;)e[a.pop()]=1;t(r)}})(),onmessage=async e=>
{const{fibonacci:r}=await
o.e(129).then(o.bind(o,129)),t=JSON.parse(e.data);postMessage(`fib(${t}) =
${r(t)}`)}})();
```

dist/129.js

```
"use strict";
(self["webpackChunk"] = self["webpackChunk"] || []).push([[129],{

/***/ 2:
/*!**********************!*\
  !*** ./fibonacci.js ***!
  \**********************/
/*! namespace exports */
/*! export fibonacci [provided] [maybe used in main, 9a81d90cfd0dfd13d748 (runtime-
defined)] [usage prevents renaming] */
/*! other exports [not provided] [maybe used in main, 9a81d90cfd0dfd13d748 (runtime-
defined)] */
/*! 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 12.3 KiB [emitted] (name: main)
asset workers/fibonacci.js 5.43 KiB [emitted] (name: fibonacci)
asset 129.js 931 bytes [emitted]
asset chat.js 911 bytes [emitted] (name: chat)
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) 5.72 KiB (runtime) [entry] [rendered]
  > ./example.js main
  runtime modules 5.72 KiB 8 modules
  ./example.js 2.25 KiB [built] [code generated]
    [no exports used]
    entry ./example.js main
chunk (runtime: 1fad8bf8de78b0a77bfd) chat.js (chat) 527 bytes [entry] [rendered]
  > ./example.js 25:19-31:1
  ./chat-worker.js + 1 modules 527 bytes [built] [code generated]
    [no exports]
    [no exports used]
    new Worker() ./chat-worker.js ./example.js 25:19-31:1
chunk (runtime: 9a81d90cfd0dfd13d748) workers/fibonacci.js (fibonacci) 176 bytes (javascript) 2.14 KiB (runtime) [entry] [rendered]
  > ./example.js 80:18-84:2
  runtime modules 2.14 KiB 7 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
```

Production mode

```
asset main.js 3.47 KiB [emitted] [minimized] (name: main)
asset workers/fibonacci.js 945 bytes [emitted] [minimized] (name: fibonacci)
asset chat.js 270 bytes [emitted] [minimized] (name: chat)
asset 129.js 166 bytes [emitted] [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) 5.72 KiB (runtime) [entry]
[rendered]
  > ./example.js main
  runtime modules 5.72 KiB 8 modules
  ./example.js 2.25 KiB [built] [code generated]
    [no exports used]
    entry ./example.js main
chunk (runtime: 1fad8bf8de78b0a77bfd) chat.js (chat) 527 bytes [entry] [rendered]
  > ./example.js 25:19-31:1
  ./chat-worker.js + 1 modules 527 bytes [built] [code generated]
    [no exports]
    [no exports used]
    new Worker() ./chat-worker.js ./example.js 25:19-31:1
chunk (runtime: 9a81d90cfd0dfd13d748) workers/fibonacci.js (fibonacci) 176 bytes
(javascript) 2.14 KiB (runtime) [entry] [rendered]
  > ./example.js 80:18-84:2
  runtime modules 2.14 KiB 7 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
```