example.js

```
This example illustrates how to specify the chunk name in require.ensure()
and import() to separated modules into separate chunks manually.
import("./templates/foo" /* webpackChunkName: "chunk-foo" */ ).then(function(foo) {
    console.log('foo:', foo);
})
require.ensure([], function(require) {
    var foo = require("./templates/foo");
    console.log('foo:', foo);
}, "chunk-foo1");
var createContextVar = "r";
import("./templates/ba" + createContextVar /* webpackChunkName: "chunk-bar-baz" */ ).then(f
    console.log('bar:', bar);
})
templates/
  • foo.js
  • baz.js
  • bar.js
All templates are of this pattern:
var foo = "foo";
export default foo;
```

dist/output.js

```
"./bar": [
       3,
       791
    "./bar.js": [
       3,
       791
   ],
    "./baz": [
       4,
       548
    "./baz.js": [
       4,
       548
   ]
};
function webpackAsyncContext(req) {
   if(!__webpack_require__.o(map, req)) {
       return Promise.resolve().then(() => {
           var e = new Error("Cannot find module '" + req + "'");
           e.code = 'MODULE_NOT_FOUND';
           throw e;
       });
   }
   var ids = map[req], id = ids[0];
   return __webpack_require__.e(ids[1]).then(() => {
       return __webpack_require__(id);
   });
}
webpackAsyncContext.keys = () => (Object.keys(map));
webpackAsyncContext.id = 1;
module.exports = webpackAsyncContext;
/***/ })
/*****/
           ]);
/* webpack runtime code */
/*****/
           // The module cache
/*****/
           var __webpack_module_cache__ = {};
/*****/
/*****/
           // The require function
/*****/
           function __webpack_require__(moduleId) {
/*****/
               // Check if module is in cache
```

```
var cachedModule = __webpack_module_cache__[moduleId];
/*****/
/*****/
               if (cachedModule !== undefined) {
/*****/
                   return cachedModule.exports;
/*****/
               }
/*****/
               // Create a new module (and put it into the cache)
/*****/
               var module = __webpack_module_cache__[moduleId] = {
/*****/
                   // no module.id needed
/*****/
                   // no module.loaded needed
/*****/
                   exports: {}
/*****/
               };
/*****/
/*****/
               // Execute the module function
/*****/
                __webpack_modules__[moduleId] (module, module.exports, __webpack_require__);
/*****/
/*****/
               // Return the exports of the module
/*****/
               return module.exports;
/*****/
           }
/*****/
/*****/
           // expose the modules object (__webpack_modules__)
/*****/
            __webpack_require__.m = __webpack_modules__;
/*****/
/*****
           ***********************
/*****/
            /* webpack/runtime/define property getters */
/*****/
            (() => {
/*****/
               // define getter functions for harmony exports
/*****/
                __webpack_require__.d = (exports, definition) => {
/*****/
                   for(var key in definition) {
/*****/
                       if(__webpack_require__.o(definition, key) && !__webpack_require__.o
/*****/
                           Object.defineProperty(exports, key, { enumerable: true, get: de
/*****/
                       }
/*****/
                   }
/*****/
               };
/*****/
           })();
/*****/
/*****/
            /* webpack/runtime/ensure chunk */
/*****/
            (() => {
/*****/
               __webpack_require__.f = {};
/*****/
               // This file contains only the entry chunk.
/*****/
               // The chunk loading function for additional chunks
/*****/
               __webpack_require__.e = (chunkId) => {
                   return Promise.all(Object.keys(__webpack_require__.f).reduce((promises,
/*****/
/*****/
                       __webpack_require__.f[key](chunkId, promises);
/*****/
                       return promises;
/*****/
                   }, []));
/*****/
               };
/*****/
           })();
```

```
/*****/
/*****/
            /* webpack/runtime/get javascript chunk filename */
/*****/
            (() => {
/*****/
                // This function allow to reference async chunks
/*****/
                __webpack_require__.u = (chunkId) => {
/*****/
                    // return url for filenames based on template
/*****/
                    return "" + chunkId + ".output.js";
/*****/
                };
/*****/
            })();
/*****/
/*****/
            /* webpack/runtime/hasOwnProperty shorthand */
/*****/
            (() => {
/*****/
                __webpack_require__.o = (obj, prop) => (Object.prototype.hasOwnProperty.cal
/*****/
            })();
/*****/
/*****/
            /* webpack/runtime/load script */
/*****/
            (() => {
/*****/
                var inProgress = {};
/*****/
                // data-webpack is not used as build has no uniqueName
/*****/
                // loadScript function to load a script via script tag
                __webpack_require__.l = (url, done, key, chunkId) => {
/*****/
/*****/
                    if(inProgress[url]) { inProgress[url].push(done); return; }
/*****/
                    var script, needAttach;
                    if(key !== undefined) {
/*****/
/*****/
                        var scripts = document.getElementsByTagName("script");
/*****/
                        for(var i = 0; i < scripts.length; i++) {</pre>
/*****/
                            var s = scripts[i];
/*****/
                            if(s.getAttribute("src") == url) { script = s; break; }
/*****/
                        }
/*****/
                    }
/*****/
                    if(!script) {
/*****/
                        needAttach = true;
/*****/
                        script = document.createElement('script');
/*****/
/*****/
                        script.charset = 'utf-8';
/*****/
                        script.timeout = 120;
/*****/
                        if (_webpack_require__.nc) {
/*****/
                            script.setAttribute("nonce", __webpack_require__.nc);
/*****/
                        }
/*****/
/*****/
                        script.src = url;
/*****/
/*****/
                    inProgress[url] = [done];
/*****/
                    var onScriptComplete = (prev, event) => {
/*****/
                        // avoid mem leaks in IE.
/*****/
                        script.onerror = script.onload = null;
```

```
/*****/
                        clearTimeout(timeout);
/*****/
                        var doneFns = inProgress[url];
/*****/
                        delete inProgress[url];
/*****/
                        script.parentNode && script.parentNode.removeChild(script);
/*****/
                        doneFns && doneFns.forEach((fn) => (fn(event)));
/*****/
                        if(prev) return prev(event);
/*****/
                    }
/*****/
/*****/
                    var timeout = setTimeout(onScriptComplete.bind(null, undefined, { type:
/*****/
                    script.onerror = onScriptComplete.bind(null, script.onerror);
/*****/
                    script.onload = onScriptComplete.bind(null, script.onload);
/*****/
                    needAttach && document.head.appendChild(script);
/*****/
                };
/*****/
            })();
/*****/
/*****/
            /* webpack/runtime/make namespace object */
/*****/
            (() => {
/*****/
                // define __esModule on exports
/*****/
                __webpack_require__.r = (exports) => {
                    if(typeof Symbol !== 'undefined' && Symbol.toStringTag) {
/*****/
/*****/
                        Object.defineProperty(exports, Symbol.toStringTag, { value: 'Module
/*****/
/*****/
                    Object.defineProperty(exports, '__esModule', { value: true });
/*****/
                };
/*****/
            })();
/*****/
/*****/
            /* webpack/runtime/publicPath */
/*****/
/*****/
                __webpack_require__.p = "dist/";
/*****/
            })();
/*****/
/*****/
            /* webpack/runtime/jsonp chunk loading */
/*****/
            (() => \{
/*****/
                // no baseURI
/*****/
/*****/
                // object to store loaded and loading chunks
/*****/
                // undefined = chunk not loaded, null = chunk preloaded/prefetched
/*****/
                // [resolve, reject, Promise] = chunk loading, 0 = chunk loaded
/*****/
                var installedChunks = {
/*****/
                    179: 0
/*****/
                };
/*****/
/*****/
                __webpack_require__.f.j = (chunkId, promises) => {
/*****/
                        // JSONP chunk loading for javascript
/*****/
                        var installedChunkData = __webpack_require__.o(installedChunks, chu
/*****/
                        if(installedChunkData !== 0) { // O means "already installed".
```

```
/*****/
/*****/
                            // a Promise means "currently loading".
/*****/
                            if(installedChunkData) {
/*****/
                                promises.push(installedChunkData[2]);
/*****/
                            } else {
/*****/
                                if(true) { // all chunks have JS
/*****/
                                    // setup Promise in chunk cache
/*****/
                                    var promise = new Promise((resolve, reject) => (install
/*****/
                                    promises.push(installedChunkData[2] = promise);
/*****/
/*****/
                                    // start chunk loading
/*****/
                                    var url = __webpack_require__.p + __webpack_require__.u
/*****/
                                    // create error before stack unwound to get useful stac
/*****/
                                    var error = new Error();
/*****/
                                    var loadingEnded = (event) => {
/*****/
                                        if(__webpack_require__.o(installedChunks, chunkId))
/*****/
                                            installedChunkData = installedChunks[chunkId];
/*****/
                                            if(installedChunkData !== 0) installedChunks[ch
/*****/
                                            if(installedChunkData) {
/*****/
                                                var errorType = event && (event.type === '1
/*****/
                                                var realSrc = event && event.target && even
/*****/
                                                error.message = 'Loading chunk ' + chunkId
/*****/
                                                error.name = 'ChunkLoadError';
/*****/
                                                error.type = errorType;
/*****/
                                                error.request = realSrc;
/*****/
                                                installedChunkData[1](error);
/*****/
                                            }
/*****/
                                        }
/*****/
                                    };
/*****/
                                    __webpack_require__.l(url, loadingEnded, "chunk-" + chu
/*****/
                                } else installedChunks[chunkId] = 0;
/*****/
                            }
/*****/
                        }
/*****/
                };
/*****/
/*****/
                // no prefetching
/*****/
/*****/
                // no preloaded
/*****/
/*****/
                // no HMR
/*****/
/*****/
                // no HMR manifest
/*****/
/*****/
                // no on chunks loaded
/*****/
/*****/
                // install a JSONP callback for chunk loading
```

```
/*****/
               var webpackJsonpCallback = (parentChunkLoadingFunction, data) => {
/*****/
                   var [chunkIds, moreModules, runtime] = data;
/*****/
                   // add "moreModules" to the modules object,
                   // then flag all "chunkIds" as loaded and fire callback
/*****/
/*****/
                   var moduleId, chunkId, i = 0;
/*****/
                   if(chunkIds.some((id) => (installedChunks[id] !== 0))) {
/*****/
                       for(moduleId in moreModules) {
/*****/
                           if(__webpack_require__.o(moreModules, moduleId)) {
/*****/
                              __webpack_require__.m[moduleId] = moreModules[moduleId];
                           }
/*****/
/*****/
                       }
/*****/
                       if(runtime) var result = runtime(__webpack_require__);
/*****/
/*****/
                   if(parentChunkLoadingFunction) parentChunkLoadingFunction(data);
/*****/
                   for(;i < chunkIds.length; i++) {</pre>
/*****/
                       chunkId = chunkIds[i];
/*****/
                       if(__webpack_require__.o(installedChunks, chunkId) && installedChun
/*****/
                           installedChunks[chunkId][0]();
/*****/
                       }
/*****/
                       installedChunks[chunkIds[i]] = 0;
/*****/
                   }
/*****/
/*****/
               }
/*****/
/*****/
               var chunkLoadingGlobal = self["webpackChunk"] = self["webpackChunk"] || [];
/*****/
               chunkLoadingGlobal.forEach(webpackJsonpCallback.bind(null, 0));
/*****/
               chunkLoadingGlobal.push = webpackJsonpCallback.bind(null, chunkLoadingGloba
/*****/
           })();
/*****/
              ********************
var __webpack_exports__ = {};
// This entry need to be wrapped in an IIFE because it need to be isolated against other more
(() => \{
/*!*****************!*\
  !*** ./example.js ***!
  /*! unknown exports (runtime-defined) */
/*! runtime requirements: __webpack_require__.*
__webpack_require__.e(/*! import() / chunk-foo */ 930).then(__webpack_require__.bind(__webp
    console.log('foo:', foo);
})
__webpack_require__.e(/*! require.ensure | chunk-foo1 */ 930).then((function(require) {
   var foo = __webpack_require__(/*! ./templates/foo */ 2);
   console.log('foo:', foo);
```

```
}).bind(null, __webpack_require__)).catch(__webpack_require__.oe);

var createContextVar = "r";
    __webpack_require__(1)("./ba" + createContextVar).then(function(bar) {
        console.log('bar:', bar);
})

})();

/******/ })()
:
```

Info

Unoptimized

```
asset output.js 11.3 KiB [emitted] (name: main)
asset 548.output.js 858 bytes [emitted] (name: chunk-bar-baz2)
asset 791.output.js 858 bytes [emitted] (name: chunk-bar-baz0)
asset 930.output.js 858 bytes [emitted] (name: chunk-foo)
chunk (runtime: main) output.js (main) 565 bytes (javascript) 5.54 KiB (runtime) [entry] [re
  > ./example.js main
 runtime modules 5.54 KiB 8 modules
  dependent modules 160 bytes [dependent] 1 module
  ./example.js 405 bytes [built] [code generated]
    [used exports unknown]
    entry ./example.js main
chunk (runtime: main) 548.output.js (chunk-bar-baz2) 38 bytes [rendered]
  > ./baz ./templates/ lazy ^\.\/ba.*$ chunkName: chunk-bar-baz namespace object ./baz
  > ./baz.js ./templates/ lazy ^\.\/ba.*$ chunkName: chunk-bar-baz namespace object ./baz.j:
  ./templates/baz.js 38 bytes [optional] [built] [code generated]
    [exports: default]
    [used exports unknown]
    import() context element ./baz ./templates/ lazy ^\.\/ba.*$ chunkName: chunk-bar-baz na
    import() context element ./baz.js ./templates/ lazy ^\.\/ba.*$ chunkName: chunk-bar-baz
chunk (runtime: main) 791.output.js (chunk-bar-baz0) 38 bytes [rendered]
  > ./bar ./templates/ lazy ^\.\/ba.*$ chunkName: chunk-bar-baz namespace object ./bar
  > ./bar.js ./templates/ lazy ^\.\/ba.*$ chunkName: chunk-bar-baz namespace object ./bar.js
  ./templates/bar.js 38 bytes [optional] [built] [code generated]
    [exports: default]
    [used exports unknown]
    import() context element ./bar ./templates/ lazy ^\.\/ba.*$ chunkName: chunk-bar-baz na
    import() context element ./bar.js ./templates/ lazy ^\.\/ba.*$ chunkName: chunk-bar-baz
```

```
chunk (runtime: main) 930.output.js (chunk-foo) 38 bytes [rendered]
> ./templates/foo ./example.js 1:0-62
> ./example.js 5:0-8:16
./templates/foo.js 38 bytes [built] [code generated]
    [exports: default]
    [used exports unknown]
    import() ./templates/foo ./example.js 1:0-62
    cjs require ./templates/foo ./example.js 6:11-37
webpack 5.51.1 compiled successfully
```

Production mode

```
asset output.js 2.45 KiB [emitted] [minimized] (name: main)
asset 548.output.js 130 bytes [emitted] [minimized] (name: chunk-bar-baz2)
asset 791.output.js 130 bytes [emitted] [minimized] (name: chunk-bar-baz0)
asset 930.output.js 130 bytes [emitted] [minimized] (name: chunk-foo)
chunk (runtime: main) output.js (main) 565 bytes (javascript) 5.54 KiB (runtime) [entry] [re
  > ./example.js main
  runtime modules 5.54 KiB 8 modules
  dependent modules 160 bytes [dependent] 1 module
  ./example.js 405 bytes [built] [code generated]
    [no exports used]
    entry ./example.js main
chunk (runtime: main) 548.output.js (chunk-bar-baz2) 38 bytes [rendered]
  > ./baz ./templates/ lazy ^{..}ba.*$ chunkName: chunk-bar-baz namespace object ./baz
  > ./baz.js ./templates/ lazy ^\.\/ba.*$ chunkName: chunk-bar-baz namespace object ./baz.js
  ./templates/baz.js 38 bytes [optional] [built] [code generated]
    [exports: default]
    import() context element ./baz ./templates/ lazy ^\.\/ba.*$ chunkName: chunk-bar-baz na
    import() context element ./baz.js ./templates/ lazy ^\.\/ba.*$ chunkName: chunk-bar-baz
chunk (runtime: main) 791.output.js (chunk-bar-baz0) 38 bytes [rendered]
  > ./bar ./templates/ lazy ^\.\/ba.*$ chunkName: chunk-bar-baz namespace object ./bar
  > ./bar.js ./templates/ lazy ^\.\/ba.*$ chunkName: chunk-bar-baz namespace object ./bar.js
  ./templates/bar.js 38 bytes [optional] [built] [code generated]
    [exports: default]
    import() context element ./bar ./templates/ lazy ^\.\/ba.*$ chunkName: chunk-bar-baz na
    import() context element ./bar.js ./templates/ lazy ^\.\/ba.*$ chunkName: chunk-bar-baz
chunk (runtime: main) 930.output.js (chunk-foo) 38 bytes [rendered]
  > ./templates/foo ./example.js 1:0-62
  > ./example.js 5:0-8:16
  ./templates/foo.js 38 bytes [built] [code generated]
    [exports: default]
    import() ./templates/foo ./example.js 1:0-62
    cjs require ./templates/foo ./example.js 6:11-37
```

webpack 5.51.1 compiled successfully