## example.js

```
function getTemplate(templateName, callback) {
   require.ensure([], function(require) {
       callback(require("../require.context/templates/"+templateName)());
   });
}
getTemplate("a", function(a) {
    console.log(a);
});
getTemplate("b", function(b) {
    console.log(b);
});
dist/output.js
/*****/ (() => { // webpackBootstrap
           var __webpack_modules__ = ({});
/* 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/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/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__ = {};
!*** ./example.js ***!
  /*! unknown exports (runtime-defined) */
/*! runtime requirements: __webpack_require__.e, __webpack_require__.*
function getTemplate(templateName, callback) {
    __webpack_require__.e(/*! require.ensure */ 577).then((function(require) {
       callback(__webpack_require__(1)("./"+templateName)());
   }).bind(null, __webpack_require__)).catch(_webpack_require__.oe);
```

```
getTemplate("a", function(a) {
   console.log(a);
getTemplate("b", function(b) {
   console.log(b);
});
/*****/ })()
dist/577.output.js
(self["webpackChunk"] = self["webpackChunk"] || []).push([[577],[
/* 0 */,
/* 1 */
!*** ../require.context/templates/ sync ^\.\/.*$ ***!
 /*! default exports */
/*! exports [not provided] [no usage info] */
/*! runtime requirements: module, __webpack_require__.o, __webpack_require__ */
/***/ ((module, __unused_webpack_exports, __webpack_require__) => {
var map = {
   "./a": 2,
   "./a.js": 2,
   "./b": 3,
   "./b.js": 3,
   "./c": 4,
   "./c.js": 4
};
function webpackContext(req) {
   var id = webpackContextResolve(req);
   return __webpack_require__(id);
}
function webpackContextResolve(req) {
   if(!__webpack_require__.o(map, req)) {
       var e = new Error("Cannot find module '" + req + "'");
       e.code = 'MODULE_NOT_FOUND';
       throw e;
   }
   return map[req];
}
```

```
webpackContext.keys = function webpackContextKeys() {
   return Object.keys(map);
};
webpackContext.resolve = webpackContextResolve;
module.exports = webpackContext;
webpackContext.id = 1;
/***/ }),
/* 2 */
!*** ../require.context/templates/a.js ***!
 /*! unknown exports (runtime-defined) */
/*! runtime requirements: module */
/*! CommonJS bailout: module.exports is used directly at 1:0-14 */
/***/ ((module) => {
module.exports = function() {
   return "This text was generated by template A";
/***/ }),
/* 3 */
!*** ../require.context/templates/b.js ***!
 /*! unknown exports (runtime-defined) */
/*! runtime requirements: module */
/*! CommonJS bailout: module.exports is used directly at 1:0-14 */
/***/ ((module) => {
module.exports = function() {
   return "This text was generated by template B";
}
/***/ }),
/* 4 */
!*** ../require.context/templates/c.js ***!
 /*! unknown exports (runtime-defined) */
/*! runtime requirements: module */
/*! CommonJS bailout: module.exports is used directly at 1:0-14 */
/***/ ((module) => {
module.exports = function() {
```

```
return "This text was generated by template C";
}
/***/ })
]]);
```

## Info

## Unoptimized

```
asset output.js 8.96 KiB [emitted] (name: main)
asset 577.output.js 2.23 KiB [emitted]
chunk (runtime: main) output.js (main) 266 bytes (javascript) 4.98 KiB (runtime) [entry] [re
  > ./example.js main
 runtime modules 4.98 KiB 6 modules
  ./example.js 266 bytes [built] [code generated]
    [used exports unknown]
    entry ./example.js main
chunk (runtime: main) 577.output.js 457 bytes [rendered]
  > ./example.js 2:1-4:3
  dependent modules 240 bytes [dependent] 3 modules
  ../require.context/templates/ sync ^\.\/.*$ 217 bytes [built] [code generated]
    [no exports]
    [used exports unknown]
    cjs require context ./example.js 3:11-64
webpack 5.51.1 compiled successfully
```

## Production mode