

Basic implementation:

- ☒ Future-proof `findImportedModuleIdentifiers` for real `import(...)`
- ☒ Source maps in development
- ☒ Debugger stops at reasonable points in dev tools
- ☒ Open another WebSocket? NO
- ☒ Make `import(...)` work on the server
- ☒ Modules are minified but not merged in production
 - ☒ Wrap modules with function to enable better minification
- ☒ Babel transform from `import(...)` to `module.importAsync(...)`
- ☒ Local module caching.
 - ☒ Prototype with `localStorage`.
 - ☒ Reimplement using `indexedDB` (much larger size limits).
- ☒ Compact `previous` state representation
- ☒ Improve dependency resolution in `packages/dynamic-import/server.js`
- ☒ Report static import/syntax/etc. errors for async files
- ☒ What about old/new versions of code?
- ☒ What about package pseudo-globals (imports)?
- ☒ What about dynamic stubs?
- ☒ Avoid creating dynamic files on the server.
- ☐ ~~`__dynamicImport` could be implemented without the fallback on the server if we were sure the server had no dynamic files.~~
- ☒ Make sure client-only reloads work (revisit `_read` caching).
- ☒ Make sure path manipulation is Windows-safe.
- ☒ Install dynamic modules with correct `meteorInstall` options.
- ☒ Tests!

Future work:

- ☐ Batch multiple `__dynamicImport` method calls?
- ☐ Detect modules unevaluated during page load and recommend importing them dynamically.
- ☐ Quantify the impact of using `import(...)`.
- ☐ Report initial bundle sizes.
- ☐ Warn about `import(...)` calls before `Meteor.startup`, since they should probably be static.
- ☐ Analyze module graph to suggest dynamic cut points (e.g. in router callbacks).
- ☐ Warn if dynamically imported modules are imported statically elsewhere (killing the benefit of the dynamic import).
- ☐ Use `Cache-Control: immutable` for the initial bundle.
- ☐ Upgrade caching to `ServiceWorker` and `Cache` in supporting browsers (if actually faster!).
- ☐ Preload modules that are often dynamically imported, when page becomes idle.
- ☐ Allow the client to overfetch soon-to-be-needed modules to avoid waterfalls.
- ☐ Write [Meteor Guide](#) article about techniques for optimizing page load times.
 - ☐ Inlining imports.
 - ☐ Making eager modules in apps and packages lazy.

- ☐ Using dynamic `import(...)` in the right places.