

# **Interfacing with the Loader Spec**

(discussed this week by Dave Herman & Allen Wirfs-Brock)

*Current draft spec*

# ModuleEvaluationJob

⇒ ParseModuleAndImports

⇒ HostGetSource

⇒ HostNormalizeModuleName

⇒ LinkModules

⇒ ModuleDeclarationInstantiation

⇒ ResolveExport

⇒ GetExportedNames

⇒ ModuleEvaluation

## Module Records

|                           |                     |
|---------------------------|---------------------|
| [[SourceCodeId]]          | String              |
| [[ImportedModules]]       | List<String>        |
| [[ECMAScriptCode]]        | parse result        |
| [[ImportEntries]]         | List<ImportEntry>   |
| [[LocalExportEntries]]    | List<ExportEntry>   |
| [[IndirectExportEntries]] | List<ExportEntry>   |
| [[StarExportEntries]]     | List<ExportEntry>   |
| [[Environment]]           | Lexical Environment |
| [[Namespace]]             | Object              |
| [[Evaluated]]             | Boolean             |

## Module Records

|                           |                     |
|---------------------------|---------------------|
| [[SourceCodeId]]          | String              |
| [[ImportedModules]]       | List<String>        |
| [[ECMAScriptCode]]        | parse result        |
| [[ImportEntries]]         | List<ImportEntry>   |
| [[LocalExportEntries]]    | List<ExportEntry>   |
| [[IndirectExportEntries]] | List<ExportEntry>   |
| [[StarExportEntries]]     | List<ExportEntry>   |
| [[Environment]]           | Lexical Environment |
| [[Namespace]]             | Object              |
| [[Evaluated]]             | Boolean             |

## Module Records

|                           |                     |
|---------------------------|---------------------|
| [[SourceCodeId]]          | String              |
| [[ImportedModules]]       | List<String>        |
| [[ECMAScriptCode]]        | parse result        |
| [[ImportEntries]]         | List<ImportEntry>   |
| [[LocalExportEntries]]    | List<ExportEntry>   |
| [[IndirectExportEntries]] | List<ExportEntry>   |
| [[StarExportEntries]]     | List<ExportEntry>   |
| [[Environment]]           | Lexical Environment |
| [[Namespace]]             | Object              |
| [[Evaluated]]             | Boolean             |

## Realm

...

[[modules]]                      List<String>

...

Let  $m$  be ModuleAt( $realm.[[modules]]$ ,  $id$ ).

...



*Needs for the Loader spec*



```
Reflect.Loader.hook("instantiate", (key, s) => {  
    if (key === "some/commonjs/module")  
        return new Reflect.Module(...);  
});
```

```
var realm = window.open(...);  
realm.Reflect.Loader.import(url, mod => {  
    Reflect.Loader.install(url, mod);  
});
```

~~ParseModuleAndImports~~

***Change: Eliminate [[SourceCodeId]]***

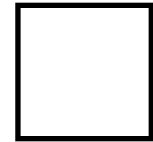
# *Interfacing with Loader: Resolution*

- ~~HostGetSource, HostNormalizeModuleName~~
- ~~[[SourceCodeId]]~~
- HostResolveImport: (Module, String) → Module

```
import $ from "jquery";  
import _ from "underscore";  
...
```

HostResolveImport

"jquery"



"underscore"



***Change: Refactor Module Record***



# *Interfacing with Loader: Dynamic Modules*

- Module Record
- Normal Module Record <: Module Record
- Reflective Module Record <: Module Record



## Module Records

|                 |                     |
|-----------------|---------------------|
| [[Realm]]       | Realm               |
| [[Environment]] | Lexical Environment |
| [[Namespace]]   | Object              |
| [[Evaluated]]   | Boolean             |

## Normal Module Records

|                           |                   |
|---------------------------|-------------------|
| [[SourceCodeId]]          | String            |
| [[ImportedModules]]       | List<String>      |
| [[ECMAScriptCode]]        | parse result      |
| [[ImportEntries]]         | List<ImportEntry> |
| [[LocalExportEntries]]    | List<ExportEntry> |
| [[IndirectExportEntries]] | List<ExportEntry> |
| [[StarExportEntries]]     | List<ExportEntry> |

# *Interfacing with Loader: Dynamic Modules*

- ~~ModuleDefinitionInstantiation, ModuleEvaluation, GetExportNames, ResolveExport~~
- Module Record
  - :: Instantiate
  - :: Evaluate
  - :: GetExportNames
  - :: ResolveExport

***Change: Split ParseModuleAndImports***

# TopLevelModuleEvaluationJob

⇒ ParseModule

⇒ HostResolveModule

⇒ LinkModules

⇒ Module::Instantiate

⇒ Module::ResolveExport

⇒ Module::GetExportedNames

⇒ Module::Evaluate