

TypeScript Modules



Deborah Kurata

@deborahkurata | blogs.msmvps.com/deborahk/

Immediately Invoked Function Expression (IIFE)

TypeScript Transpiled Code (.js)

```
var ProductListCtrl = (function () {  
    function ProductListCtrl () {  
        ...  
    }  
})();  
  
angular  
    .module("productManagement")  
    .controller("ProductListCtrl",  
                ProductListCtrl);
```

Hand-Written JavaScript (.js)

```
(function () {  
    angular  
        .module("productManagement")  
        .controller("ProductListCtrl",  
                    ProductListCtrl);  
  
    function ProductListCtrl(pResource) {  
        ...  
    }  
})();
```

TypeScript Module

Module Overview

What?

How?

TypeScript Modules

- Encapsulate variables, interfaces, and classes
- Define unique namespaces
 - `System.Data`
 - `System.Data.Common`
 - `java.io`
- Organize symbols and identifiers into a logical namespace hierarchy

Two Types



Internal

External

TypeScript Module Conventions

Around each code file:

Angular modules

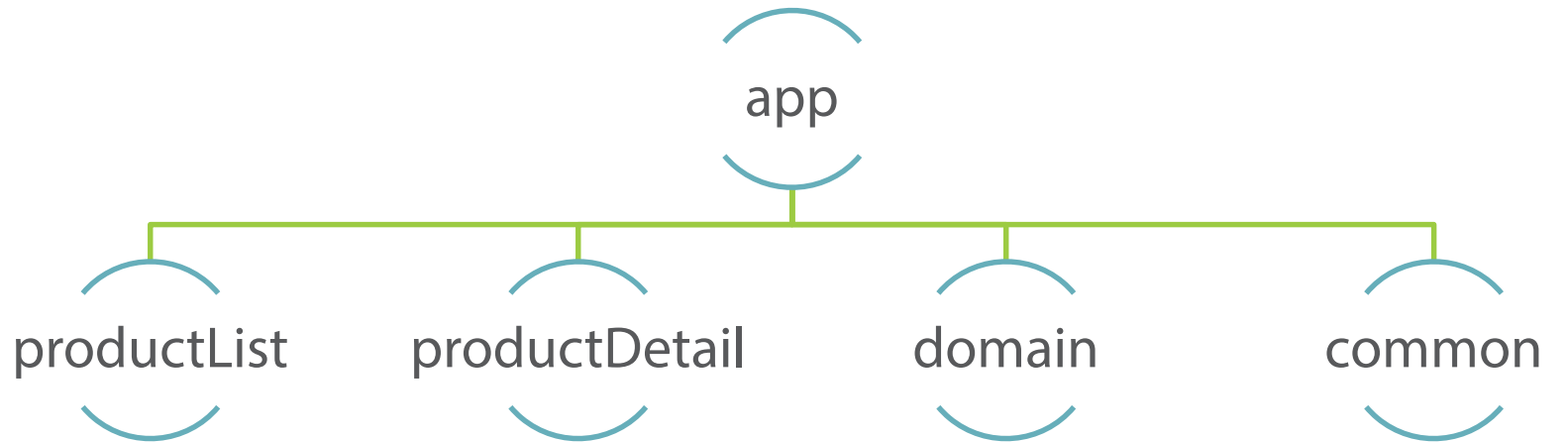
Angular controllers

Naming conventions

"app" as root namespace

sub-namespaces for:

- each feature
- domain objects
- common code



TypeScript Module Conventions

- Multiple files can use the same module name
 - `app.common`
- One file can contain multiple modules

TypeScript Module Syntax

```
module app.productList {  
    ...  
}
```

Encapsulation

Interfaces and classes within a module are local to that module

Expose an interface or class with the **export** keyword

Export not required for Angular modules and controllers

```
module app {  
  x = new app.a.A();  
  y = new app.b.B();  
}
```

```
module app.a {  
  class A {}  
}
```

```
module app.b {  
  export class B {}  
}
```

This Module Covered



TypeScript Modules

TypeScript module !== Angular module

Similar to namespaces/packages

Transpiles to IIFE

Follows the JavaScript module pattern

Encapsulates the code

