
registry

Release 0.3.0

September 05, 2016

1	Getting Started	3
1.1	Wildcard Power	3
2	Indices and tables	5

Registry is a JS utility library designed to assist with registering units of functionality (classes, modules, plain old functions and objects) and then retrieving them at a later stage using wildcard selectors.

Getting Started

Registry can be a little hard to get your head around at first, so let's start with a simple example. In the example below we will create a *welcomer* module which exposes a single method *sayHello* which writes a greeting on the console.

```
registry.module('welcomer', function() {
  return {
    sayHello: function() {
      console.log('hi');
    }
  };
});

registry('welcomer').sayHello();
```

In the example above, we use the *registry.module* function to define a module which will be invoked as an **IIFE** which is equivalent to calling *registry.define* passing an IIFE yourself. Shown below is an example of the *welcomer* definition using the basic define functionality. It looks a little messy, which is why the module helper was introduced.

```
registry.define('welcomer', (function() {
  return {
    sayHello: function() {
      console.log('hi');
    }
  };
}))();

registry('welcomer').sayHello();
```

1.1 Wildcard Power

While the examples above show how you can use registry in a simple way, they really don't demonstrate why you would use registry. For this, we will create a slightly more complicated *welcomer* example.

In this example we will create two variants of a *welcomer*, both which provide a *sayHello* method:

```
registry.module('welcomer.polite', function() {
  return {
    sayHello: function() {
      console.log('hi, how are you?');
    }
  };
});
```

```
registry.module('welcomer.rude', function() {  
    return {  
        sayHello: function() {  
            console.log('um, what do you want?!');  
        }  
    };  
});  
  
registry('welcomer').sayHello();
```

As you can see, when we ask registry for an instance of a welcomer we don't ask it for a specific type of welcomer, but rather we let it decide which one will be supplied. In this case, the *welcomer.polite* instance is returned and used as it was the first declared. If you explicitly wanted to invoke the rude welcomer, you simply need to specify *welcomer.rude* in the registry call.

NOTE: As with *eve* the wildcard matching implementation (provided by the *wildcard* module) allows the request text of *welcomer* to match both *welcomer.polite* and *welcomer.rude* by inferring a pattern of *welcomer.**.

Indices and tables

- `genindex`
- `modindex`
- `search`