Common Built-In Services

Brice Wilson

@brice_wilson | www.BriceWilson.net

Overview

Promises and the $q service

Angular routing

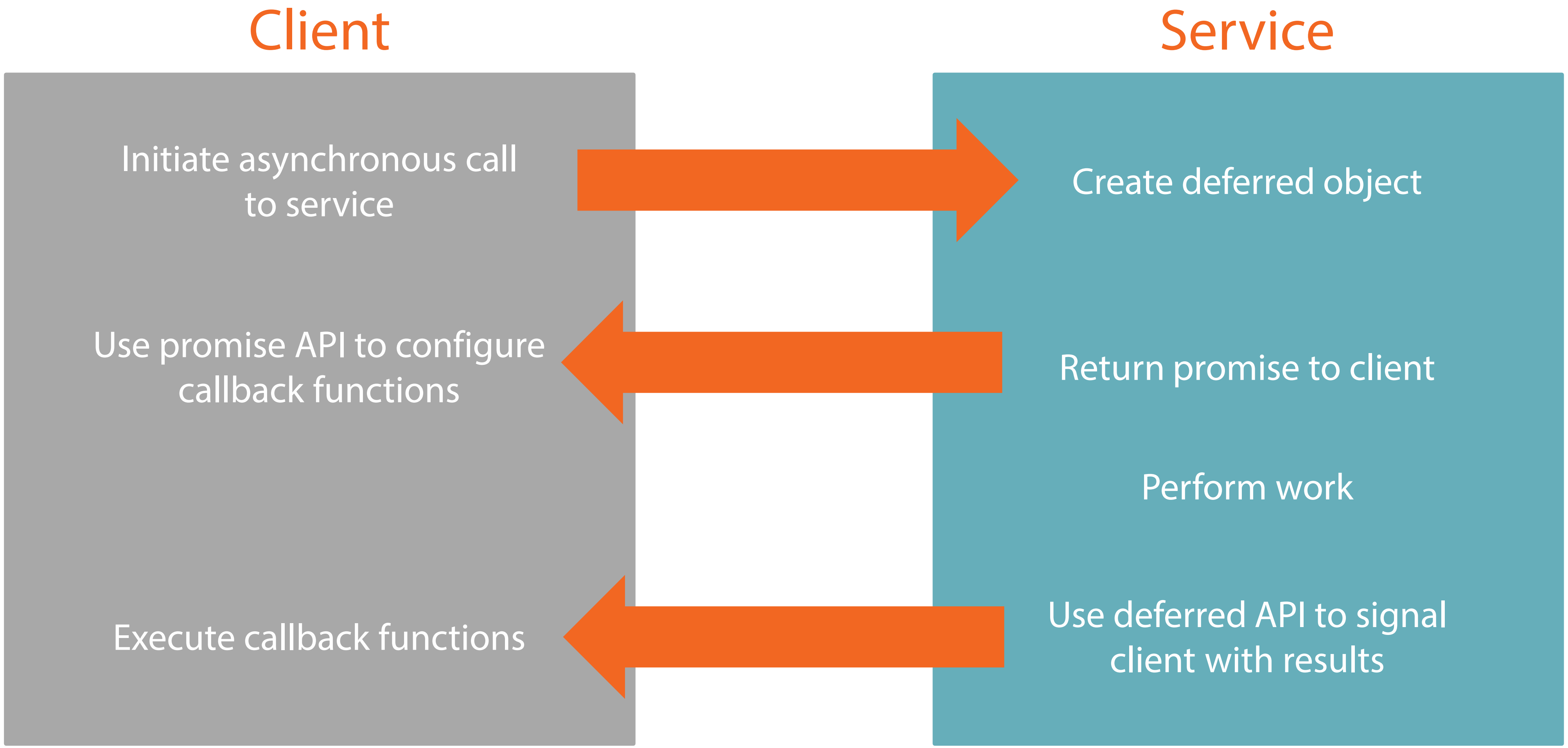
Using $cookies, $cookieStore and $log

> In Javascript, promises are objects which represent the pending result of an asynchronous operation.

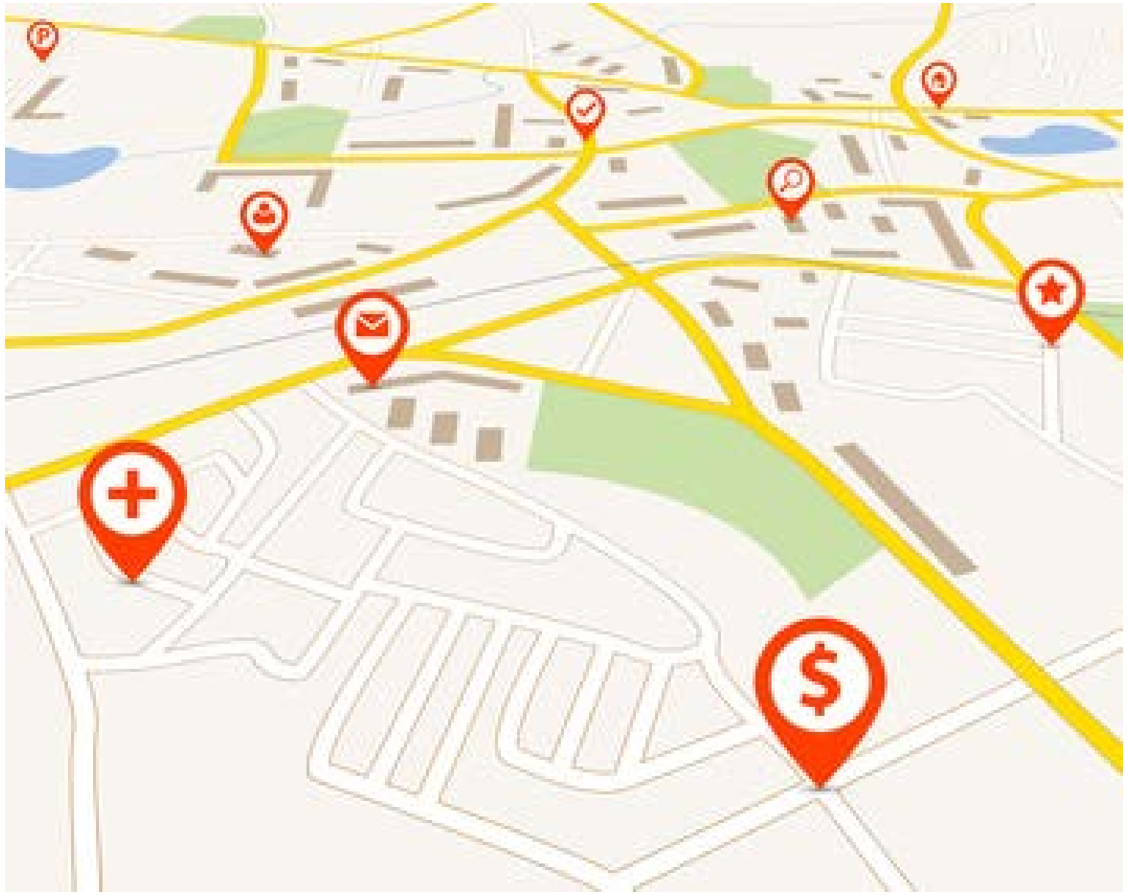
— Martin Fowler

http://martinfowler.com/bliki/JavascriptPromise.html

Asynchronous Calls with $q


Client
Service
Initiate asynchronous call to service
Create deferred object
Use promise API to configure callback functions
Return promise to client
Perform work
Execute callback functions
Use deferred API to signal client with results

Routing



Use $routeProvider to configure mapping of controllers and views to URLs

$route watches the URL and directs browser to the correct route

Use $routeParams to capture URL parameters

```
var ngRouteModule = angular.module('ngRoute', ['ng']).
                         provider('$route', $RouteProvider)
```

$route and $routeProvider Configuration

- $route created with “provider” function
- Service is configurable via the underlying $routeProvider

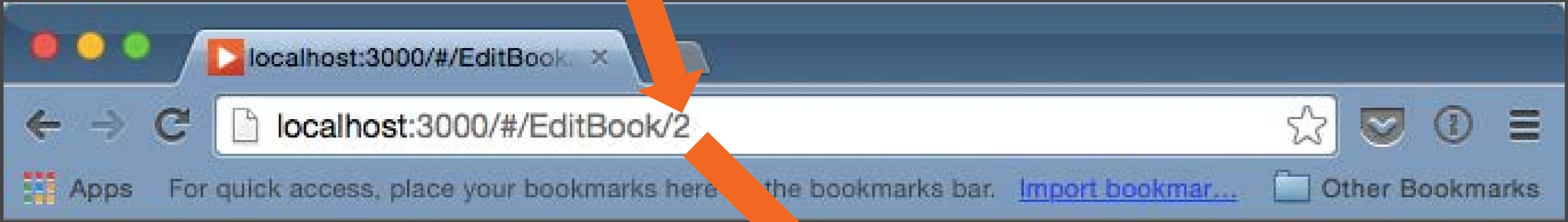
```
app.config(['$routeProvider', function ($routeProvider) {
    $routeProvider
        .when('/', {
            templateUrl: '/app/templates/books.html',
            controller: 'BooksController',
            controllerAs: 'books'
        })
        .when('/AddBook', {
            templateUrl: '/app/templates/addBook.html',
            controller: 'AddBookController',
            controllerAs: 'addBook'
        })
        .otherwise('/');
}]);
```

Configuring Routes

- Inject $routeProvider into config()
- Define a route using when() specifying a template and controller
- Use otherwise() to define a default route

$routeParams

```
.when('/EditBook/:bookID', {
    // ...
})
```



localhost:3000/#/EditBook ×
Apps
For quick access, place your bookmarks here
the bookmarks bar.
Import bookmar...
Other Bookmarks

localhost:3000/#/EditBook/2

```
function EditBookController($routeParams) {
    var myParam = $routeParams.bookID;
    console.log(myParam);
}
```

Resolve Property

```
.when('/EditBook/:bookID', {
    templateUrl: '/app/templates/editBook.html',
    controller: 'EditBookController',
    controllerAs: 'bookEditor',
    resolve: {
        books: function (dataService) {
            return dataService.getAllBooks();
        }
    }
})


angular.module('app')
    .controller('EditBookController', ['books', EditBookController]);

function EditBookController(books) {
    // configure controller here
}
```

$route Events

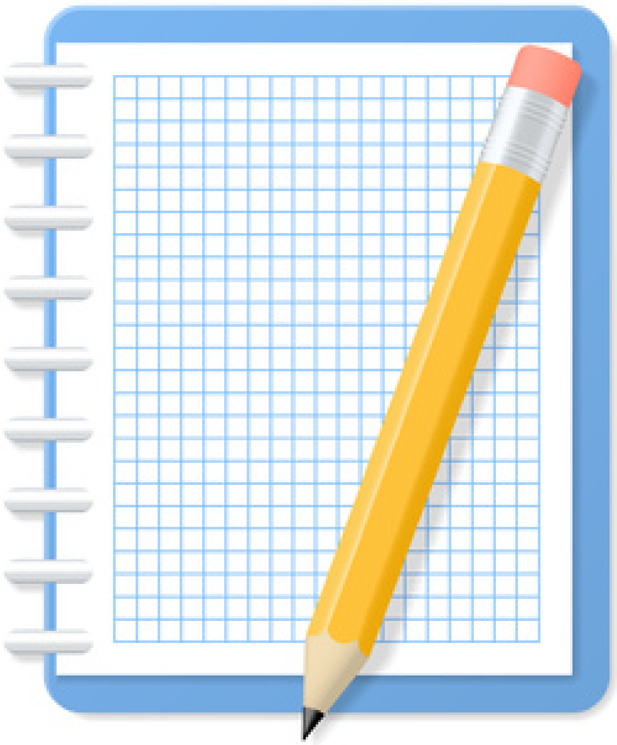
- Events related to route changes broadcast on $rootScope
- Inject $rootScope to define event handlers
- Use $rootScope.$on() to specify event types to listen for
- $route defines four events
  - $routeChangeStart
  - $routeChangeSuccess
  - $routeChangeError
  - $routeUpdate

Using $cookies and $cookieStore

- Available in the ngCookies module
- Use $cookies to read/write simple strings as properties on the service
- Use $cookieStore to read/write objects
  - Serialized/deserialized to and from JSON
  - get(key);
  - put(key, value);
  - remove(key);

Using $log

- Simple wrapper around common console functions
  - log();
  - info();
  - warn();
  - error();
  - debug();
- Output with debug() is configurable with $logProvider



Summary

- Lots of built-in services
- Conceptually no different custom services
- Routing around the SPA
- Using $q to manage asynchronous calls