

JavaScript + Angular

Part 6

services

\$document

```
angular.module('documentExample', [])  
.controller('ExampleController', function($scope, $document) {  
  $scope.title = $document[0].title;  
  $scope.windowTitle = angular.element(window.document)[0].title;  
});
```

\$cacheFactory

```
let cache = $cacheFactory( 'cacheId' );  
$cacheFactory.get( 'cacheId' );  
$cacheFactory.get( 'noSuchCacheId' );  
  
cache.put( "key", "value" );  
cache.put( "another key", "another value" );
```

\$controller

```
$controller(constructor, locals);  
  
let MyController = $controller('MyController', {  
    $http,  
    MyService  
});
```

\$interpolate

```
let $interpolate = ...; // injected
let exp = $interpolate('Hello {{name | uppercase}}!');
expect(exp({name: 'Angular'})).toEqual('Hello ANGULAR!');
```

\$interval & \$timeout

```
$interval(fn, delay, [count], [invokeApply], [Pass]);
```

```
$timeout([fn], [delay], [invokeApply], [Pass]);
```

```
let stopTime = $interval(updateTime, 1000);
```

```
$interval.cancel(stopTime);
```


\$filter

```
angular.module('filterExample', [])  
.controller('MainCtrl', function($scope, $filter) {  
  $scope.originalText = 'hello';  
  $scope.filteredText = $filter('uppercase')($scope.originalText);  
});
```




00:00

1 Min

1 Second

10 Min

10 Second

Start/Stop

Reset

Tasks

- Create kitchen timer

\$templateCache

```
<script type="text/ng-template" id="templateId.html">  
  <p>This is the content of the template</p>  
</script>
```

```
$templateCache.put('templateId.html', 'This is the content of the template');
```

\$q

```
let promise = $q(function(resolve, reject) {
  setTimeout(function() {
    if (okToGreet(name)) {
      resolve('Hello, ' + name + '!');
    } else {
      reject('Greeting ' + name + ' is not allowed.');
```



```
    }
  }, 1000);
});

promise.then(function(greeting) {
  alert('Success: ' + greeting);
}, function(reason) {
  alert('Failed: ' + reason);
});
```

\$http

```
$http({method: 'GET', url: '/my-profile'}).then((response) => {
  // this callback will be called asynchronously
  // when the response is available
}, (response) => {
  // called asynchronously if an error occurs
  // or server returns response with an error status.
});

$http.get('/someUrl', config).then(successCallback, errorCallback);
$http.post('/someUrl', data, config).then(successCallback, errorCallback);

/**
 * $http.get
 * $http.head
 * $http.post
 * $http.put
 * $http.delete
 * $http.jsonp
 * $http.patch
 */
```

\$http

```
$httpProvider.interceptors.push((...) => {  
  return {  
    'request': function(config) { ... },  
  
    'response': function(response) { ... },  
  
    'requestError': function(rejection) { ... },  
  
    'responseError': function(rejection) { ... }  
  };  
});  
  
$httpProvider.interceptors.push('myHttpInterceptor');
```

\$httpParamSerializer

```
{'foo': 'bar'} results in foo=bar
```

```
{'foo': Date.now()} results in foo=2015-04-01T09%3A50%3A49.262Z  
(toISOString() and encoded representation of a Date object)
```

```
{'foo': ['bar', 'baz']} results in foo=bar&foo=baz  
(repeated key for each array element)
```

```
{'foo': {'bar': 'baz'}} results in foo=%7B%22bar%22%3A%22baz%22%7D  
(stringified and encoded representation of an object)
```

Tasks

- Show statistic of timer usage (json of usage should be on server)
- Catch all server requests and show error message on error call
- Show debug information with all requests

i18n and l10n

```
import "angular-translate";

var app = angular.module('app', ['pascalprecht.translate']);

app.config(function ($translateProvider) {

    $translateProvider.translations('en', {
        TITLE: 'Hello',
    });
    $translateProvider.translations('es', {
        TITLE: 'Hola'
    });
    $translateProvider.preferredLanguage('en');
});

<h1>{{ 'TITLE' | translate }}</h1>
```

i18n and l10n

```
npm install angular-translate --save
```

\$sce

```
<textarea ng-model="html"></textarea>  
<div ng-bind-html="trustedHtml"></div>
```

```
angular.module('mySceApp', ['ngSanitize'])  
  .controller('AppController', function($sce, $scope) {  
    var self = this;  
    $scope.$watch('html', (oldVal, newVal) => {  
      $scope.trustedHtml = $sce.trustAsHtml($scope.html);  
    });  
  
  });
```

Tasks

- Make i18n for timer
- Create smiles filter with html support

\$location

```
// url(), protocol(), host(), port(), path(), search(), hash()  
  
let absUrl = $location.absUrl();  
  
$location.replace('/hello');
```

users services

```
//provider style, full blown, configurable version
myApp.provider('helloWorld', function() {

    this.name = 'Default';

    this.$get = function() {
        var name = this.name;
        return {
            sayHello: function() {
                return "Hello, " + name + "!"
            }
        }
    };

    this.setName = function(name) {
        this.name = name;
    };
});
```



```
angular.service('myService', function () {  
    // use new myService() for create new service instance  
    return function myService () {}  
});  
  
angular.factory('myFactory', function () {  
    return {  
        // inject myFactory and call myFactory.myMethod();  
        'myMethod': () => {...}  
    }  
});
```





Pet Project