LISTENING FOR ERRORS

Clint Checketts
@checketts

SILENCE = BLINDNESS

Fear the silence

- Has anything broken?
- Who was affected?
- What specifically failed?

3 POINTS

- Client side exceptions - send to server
- Server side - publish failure to the user
- Forms - show that something is missing

Client side errors

- JS bugs
- Unit testing should catch the worst
- But what if one slips through?


OH NOES!
Intensify forward firepower!


ONE DOES NOT SIMPLY
INTENSIFY FORWARD FIREPOWER
quickmeme.com

`$exceptionHandler` service

By default logs uncaught exceptions to console

Extending $exceptionHandler

- Create your own implementation (.service/.factory)
- Extend an existing one (.provide)

```
app.config(function ($provide) {

    $provide.decorator("$exceptionHandler", function ($delegate, $injector) {
        return function (exception, cause) {
            var rootScope = $injector.get("$rootScope");
            var HttpLoggingService = $injector.get("HttpLoggingService");
            if (rootScope) {
                HttpLoggingService.sendLog(exception + '\n\n' + exception.stack);
                rootScope.$broadcast('ccAppError',
                    {
                        type: 'danger',
                        msg: 'There was an error making a request. The support team has
been notified. If necessary, please refresh your page and try again.',
                        details: exception + '\n\n' + exception.stack
                    }
                );
            }

            $delegate(exception, cause);
        };
    });
});
```

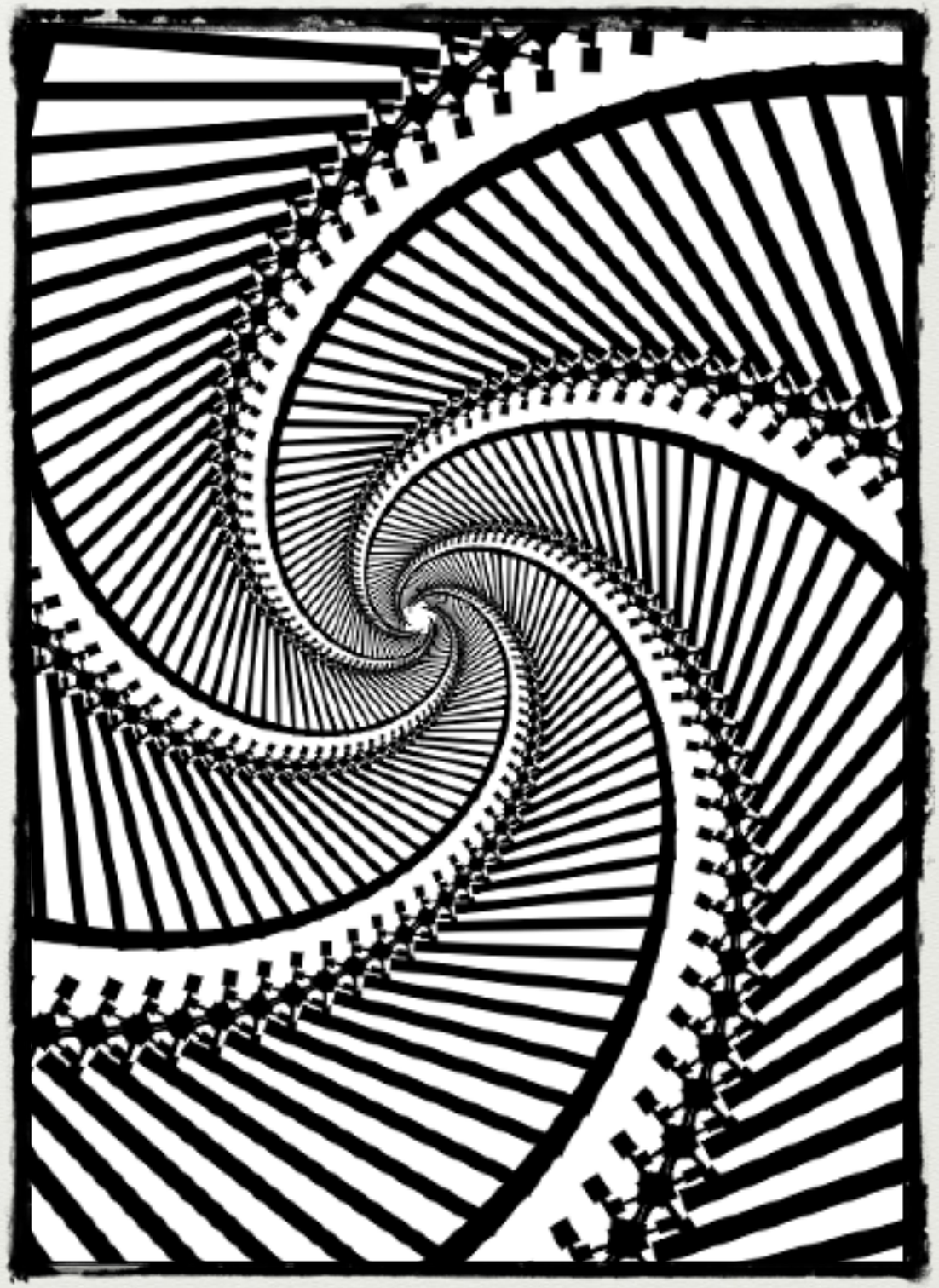
Key Points

- config
- $provide.decorator
- $injector
- HttpLoggingService
- $broadcast

Key Obstacle

Don’t make the error worse:

error -> $http -> $apply ->
error -> $http -> $apply ->
error -> $http -> $apply ->
error -> $http -> $apply ->



Skipping $http

```
app.service('HttpLoggingService', function ($httpBackend) {
    return {
        sendLog: function (msg) {
            $httpBackend('POST', '/logError', msg,
                function (status, resp, headerString) {
                },
                {"Content-Type": "application/json"}
            );
        }
    };
});
```

DEMO

https://github.com/checketts/angularExceptionHandler

Interceptor

```
app.factory('ccHttpInterceptor', function ($q, $rootScope, $interpolate, $log, HttpLoggingService)
{
    var createLogMessage = function (response) {
        if (response.status) {
            return $interpolate('Http Failure: Status: {{status}}, Url: {{config.url}} Method:
{{config.method}}')(response);
        } else {
            return angular.toJson(response);
        }
    };

    return {
        response: function (response) {
            HttpLoggingService.returnToService(response.config.url);
            return response || $q.when(response);
        },
        responseError: function (rejection) {
            $log.error('ccHttpResponseInterceptor response error', rejection);
            var logMessage = createLogMessage(rejection);
            HttpLoggingService.sendLog(logMessage);

            if (rejection.status === 403) {
                $rootScope.$broadcast('ccAppError', {type: 'danger', msg: 'Access Denied'});
            } else {
                $rootScope.$broadcast('ccAppError',
                    {
                        type: 'danger',
                        msg: 'There was an error making a request. The support team has been
notified. If necessary, please refresh your page and try again.'
                    });
            }
            return $q.reject(rejection);
        }
    };
});
```

```
<ng-form name="name" class="form-group" ng-class="{'has-error':name.$invalid &&
(name.$dirty)}">
    <label class="control-label" for="input">Name</label>
    <div><input id="input" required name="input" ng-model="someName" /></div>
    <div class="error-text small text-danger" ng-show="name.input.
$error.required && name.input.$dirty" >Required</div>
</ng-form>
```

```
<cc-form-group name="test" label-text="'Name'"  error-text="'Required'">
    <input required name="input" ng-model="otherName" />
</cc-form-group>
```

Credit

- Dice- https://www.flickr.com/photos/jeremybrooks/4146565830
- Dynamite- https://flic.kr/p/6ryW5r
- Recursion - http://payload125.cargocollective.com/1/10/328621/4804967/Recursive%20Drawing-7.png
- Question box - https://flic.kr/p/fhRZKU