AngularJS 1.4

Planning Doc

# Introduction

This document is a place to capture ideas for features, breaking changes and refactorings that could go into AngularJS 1.4.

# Planning Meeting 11 December 2014

We had a planning meeting last week to decide what will go into AngularJS 1.4. Here is a summary of what happened. You can watch the video of the meeting on YouTube, see below.

## Release Schedule

It was agreed that the first big release (1.4) will be in Spring 2015 - coinciding with [ng-conf](http://www.ng-conf.org/) on 5th March 2015.

## 2015 Work Planning

Lucas prepared a [spreadsheet](https://docs.google.com/spreadsheets/d/1ihM8ORf8v38qAbnzJM4TiCqdKeCRz_1VO9JPjSpPWKU/edit?usp=sharing) with a list of suggested items for 1.4. This focussed on large work items or things that would require breaking changes or considerable new API.

The bulk of the meeting was going through these items, deciding whether they should appear in Angular 1.x and, if so, which release and who would own the item.

## 1.4 Targets

There is a [tracking spreadsheet](https://docs.google.com/spreadsheets/d/1F4JmM25GeaVWLv7oaJrmfFZoE8ioepqNZQKd6xx_Jc4/edit?usp=sharing) for the items that are scheduled for 1.4.

The **main themes** for 1.4 are as follows:

* **Router** - *Brian* - a new router for Angular 1 and 2 - [Progress](https://docs.google.com/document/d/1-DBXTHaeec6XH5qx2tKVrgrjiILy76_lSrjgJv95RJ4/edit#)
* **I18N** - *Chirayu* - provide a first class internationalization story for Angular - [Design Doc](https://docs.google.com/document/d/1mwyOFsAD-bPoXTk3Hthq0CAcGXCUw-BtTJMR4nGTY-0/edit?usp=sharing)
* **Forms** - *Martin* - a fresh look at parsing/formatting/validation to simplify usage and maintenance (while fixing numerous outstanding issues) - [Design Doc](https://drive.google.com/open?id=1-MhomULgCFOXVsqAGrI7l-cXYMYmJM4BMzgy_anC93o&authuser=0)
* **HTTP** - *Pawel* - improvements to the $http service, such as serialization, JSON parsing, testing mock DSL
* **Parser** - *Lucas* - performance improvements to $parse service
* **Documentation** - *Caitlin* - redesign the look of the docs app to use Material Design

In addition there are some notable or breaking changes to expect such as

* **$injector** - *Brian* - throw an error if you redefine a module, to help identify bugs faster ([#1779](https://github.com/angular/angular.js/issues/1779))
* **$compile** - *Igor* - provide extra new module.component helper for defining component type directives more easily ([#10007](https://github.com/angular/angular.js/issues/10007)).
* **$compile** - *Caitlin* - throw an error if non-optional isolated scope mappings are missing their attributes ([#9216](https://github.com/angular/angular.js/pull/9216)).
* **Project layout** - *Pete* - further partition angular.js into smaller optional modules/files to reduce the non-optional core file size (useful for mobile use cases).

## Github Milestones and Labels

Finally we will start developing 1.4.x on the master branch soon. To support this here are the new labels/milestones for on-going development:

### Milestones:

**1.4.x** - use this for issues and pull requests that are accepted and scheduled to be in 1.4

### Labels:

**branch: 1.2.x** (replaces stable: yes)

**branch: 1.3.x** (replaces stable: no)

**branch: 1.4.x** (replaces 1.4 - for triaging 1.4.x issues and PRs)

**Primary Focus**: (new for items that we are focussing on for 1.4 - i.e. the stuff in the tracking spreadsheet)

## Other Versions and Backporting

The master branch (i.e. 1.4.x) will now receive the majority of the focus. AngularJS 1.4 will continue to support all ever-green browsers and Internet Explorer 9 and above.

The 1.3.x branch will receive version specific fixes backported from master.

The 1.2.x branch will now only receive fixes for security issues and major regressions.

## Video

In our continued efforts to be transparent and open about our design and development activities we recorded this meeting: <https://www.youtube.com/watch?v=Uae9_8aFo-o>



# 

# Previous Discussions Below

## New Router

Owner: Brian

## Better I18N Support

Owner: Chirayu

## Smaller Core More Modules

Owner: ??

It would be nice to move some non-core stuff out of ng and put it into auxiliary modules.

**Candidates are**: filters; ngSwitch; ngModel/Form/input/select??;

**Disadvantages**: less able to share private helpers and minify as well; more files to deal with for package management and deployment

**Advantages**: no need to include unused directives/filters/services in application downloads

## Remove Cruft

Owner: ??

Remove unnecessary code - particularly IE8 related stuff

## Better Forms

Owner: Martin

See <https://docs.google.com/document/d/1-MhomULgCFOXVsqAGrI7l-cXYMYmJM4BMzgy_anC93o/edit?usp=sharing>

## $http improvements

Owner: Pawel

There are number of recurring problems with $http that IMO can’t be solved by simple bug fixes and merit some stepping back / better design.

### Request params serialization

The current way of serializing request params doesn’t work for all the backend (seems to be particularly painful for people using Rails / PHP). The basic idea here is to extract URL params serialization (Object -> request params string) logic into a dedicated service so people can override it and community can create services dedicated to the request params format used by a given back-end (BTW: similar issue exists for $location). Examples of issues that would be solved by introducing this new service:

* <https://github.com/angular/angular.js/issues/3311>
* <https://github.com/angular/angular.js/issues/3740>
* <https://github.com/angular/angular.js/issues/6039>
* <https://github.com/angular/angular.js/pull/7409>
* <https://github.com/angular/angular.js/issues/7429>
* <https://github.com/angular/angular.js/issues/9224>

### Expose XMLHttpRequest factory as part of the “public” API

Currently creation of XMLHttpRequest object instances is hard-coded (<https://github.com/angular/angular.js/blob/9474ec120acc425103516f1018e63ae39d7d629b/src/ng/httpBackend.js#L3>) so people can’t pass any constructor args / set any properties on it. This is apparently limiting to several people and would allow us to tackle issues like those outside of core:

* <https://github.com/angular/angular.js/issues/2318>
* <https://github.com/angular/angular.js/pull/7903>
* <https://github.com/angular/angular.js/issues/7669>
* <https://github.com/angular/angular.js/pull/7689>
* <https://github.com/angular/angular.js/pull/8160>

### Progress events notifications

Request for those seem to be very popular:

* <https://github.com/angular/angular.js/issues/1934>
* <https://github.com/angular/angular.js/pull/3606>
* <https://github.com/angular/angular.js/pull/7995>

The question is if we can do it in a way that it works reasonably in all the supported browsers (or at least doesn’t do any harm in non-supported browsers).

### Better mocking story

I believe that there was an attempt at it already and there is / was some work scheduled for 2.0 in this respect. Would be keen on reviving this discussion.  
<https://docs.google.com/document/d/1XZ4lq-AWcpbpJQ_p4_hzxrin6J19fTahluzMuD7rXD8/edit?usp=sharing>

### Respect Content-Type response header

Do not try to parse responses as JSON, unless the Content-Type is set to “application/json”. This will avoid issues such as [#10357](https://github.com/angular/angular.js/pull/10357).

## Real-time Communication

Owner: Jeff???

There was some work for 2.0 on this by Jeff - perhaps some could be backported

<https://docs.google.com/document/d/1DMacL7iwjSMPP0ytZfugpU4v0PWUK0BT6lhyaVEmlBQ/edit?usp=sharing>

## limitTo filter with from / to args

Owner: Pawel

The current limitTo filter isn’t enough to one of the most common scenarios - collection pagination. While it is true that people can easily write a custom filter it is kind of sad that we don’t have a solution for such a common use-case in core. Especially that it is just a couple of lines of code. Example of issues / PRs raising this:

* <https://github.com/angular/angular.js/issues/5355>
* <https://github.com/angular/angular.js/pull/6399>
* <https://github.com/angular/angular.js/pull/6625>

## Notable PRs

Owner: ??

## Materialize the Docs

Owner: Caitlin

## Simpler Component Definition

Owner: Matias?

John Lindquist has some thoughts on this...

<https://github.com/angular/angular.js/issues/10007>

## Attribute Mapping:

<https://gist.github.com/matsko/389c297e60857ab4d218> (Lucas? Matias?)

## Other notable defects and PRs

<https://docs.google.com/spreadsheets/d/1ihM8ORf8v38qAbnzJM4TiCqdKeCRz_1VO9JPjSpPWKU/edit#gid=0>