Angular Working Group meeting notes

July 31, 2014

Who: Igor, Michałucas

* jQuery 2 update
  + <https://github.com/angular/angular.js/pull/7311>

July 17, 2014

Who: Karl, Lucas, Igor

- Landed:

<https://github.com/angular/angular.js/commit/86d55c1ded21a5be6091344493d70c6dc4194e43>

- application partitioning approaches:

- <https://github.com/geddski/overmind>

- <https://github.com/lgalfaso/angular-subapp>

July 8, 2014

**Who:** Igor, Michał, Karl, Lucas

**Agenda + Notes AI:**

* To Karl: Will something like <https://github.com/lgalfaso/angular-subapp> help you get out of the `regions` hack?
* A way to shutdown an angular app? (#8005)
* [Partial Digest](https://github.com/angular/angular.js/pull/8055)
  + how will this work with global applyAsync?
  + trigger root watchers in addition to partial digest
  + upgrade partial digest to global digest when we suspect that partial digest is not enough?
* jQuery 2.x upgrade progress
  + AI: update FAQ
  + AI: add info about which modules we use (for people with custom builds of jQuery)
* ngAnnotate <https://github.com/btford/ngmin/issues/93>
* svg support: <https://github.com/blesh/angular-jqlite-svg>
* review comments for: <https://github.com/angular/angular.js/commit/eec6394a342fb92fba5270eee11c83f1d895e9fb#commitcomment-6715248>
* Batarang? [https://github.com/angular/angularjs-batarang/issues/110#issuecomment](https://github.com/angular/angularjs-batarang/issues/110#issuecomment-45127935)

June 5, 2014

**Who:** Igor, Lucas, Ari, Karl

**Agenda + Notes AI:**

* bind-once [landed](https://docs.angularjs.org/guide/expression#one-time-binding) (more improvements on the way)
* benchmark automation
* $digest in progress discussion ([issue](https://github.com/angular/angular.js/issues/5549))
* 1.3 issue [hot list](https://github.com/angular/angular.js/issues?labels=high+priority&milestone=32&state=open)

May 22, 2014

**Who:** Igor, Michał, Dave

**Agenda + Notes AI:**

* bind-once update ([PR](https://github.com/angular/angular.js/pull/7486), [design doc](https://docs.google.com/document/d/1fTqaaQYD2QE1rz-OywvRKFSpZirbWUPsnfaZaMq8fWI/edit#), [example](http://plnkr.co/edit/rJOqa0oh35Y0xBhTYZ6w))
* jQuery update
* jQuery/jqLite data as bottleneck? ([example](http://plnkr.co/edit/rJOqa0oh35Y0xBhTYZ6w))
* lazy loading of ng modules - Dave. Working on example to share.
* pushState - possible to polyfill cheaply in IE9? (m\_gol to try)

May 8, 2014

**Who:** Igor, Lucas, Ari, Misko, Karl

**Agenda + Notes AI:**

* **Bind Once**
  + **simple case:**
  + **<p>{{: user.name }}</p>**
  + **complex case:**
  + **<table bind-notify-someid="users">**

**<tr ng-repeat="user in users"><td fast-bind="user.name"></td></tr>**

**</table>**

April 17, 2014

**Who:** Igor, Lucas, Karl, Michał, Dave

**Agenda + Notes AI:**

* Angular 1.3 WG member focus areas
  + Memory management in AngularJS (AI: igor - add a link)
  + Lucas: backlog is increasing
    - There are about ~130 issues marked for 1.3, what can the WG do to try to lower these numbers?
* Angular 2.0 WG member focus areas
  + Lucas: Interoperability
    - What form of interoperability should be expected between Angular.Dart and Angular.JS? What about between AngularJS 1.x and 2.x?
    - Should templates work in .dart and .js? (this would imply that the AST should use the same operators and follow the same rules)
  + Igor: [on being DOM](https://docs.google.com/document/d/1gBh7ka9QZXIN1b0n2Aj3X8-mLdNjWCyld3sd7cM3a8k/edit#)
  + Michał: should we disallow using ES5 constructs where there is a better ES6 alternative?
    - Could we enforce it via tooling? JSCS? Would we need experimental branches to be able to validate ES6 code?
    - Examples:
      * rest parameters instead of the arguments object
      * let, const instead of var - advantages of let/const disappear once someone uses var since we can no longer assume the block scope, also: interactions between var and let/const are not always trivial to fully understand
        + issue: Traceur’s implementation of that is low in performance because of using try-catch blocks.

could we change the Traceur implementation so that it matches what defs.js (described below) does?

* + - * + Alternative: [defs.js](https://github.com/olov/defs) by Olov Lassus

doesn’t introduce any changes besides changing const/let to var & optionally appending “$0”, “$1” etc. to variable names

no performance penalty

line numbers don’t change, completely non-destructive

An example: <https://github.com/olov/defs#example>

* potential replacement for ngResource <https://github.com/geddski/richmodels>
* Karl: bind-once discussion ([design doc](https://docs.google.com/a/google.com/document/d/1Ir3NdoKImWbIi_7nNybpaCo5XkyuPhe6RxYchzvLuQk/edit?disco=AAAAAJPfYgg))

Mar 12, 2014

**Who:**

Ari, Dave, Lucas, Igor, Karl, Michał, Miško

**Agenda + Notes AI:**

* Welcome Lucas Galfasó to the Working Group
  + see [recent activity](https://github.com/lgalfaso?tab=activity)
  + [pending PRs](https://github.com/angular/angular.js/pulls/lgalfaso)
* Angular 1.3 issues (review the most risky ones and find owners)
  + require.js support - let's prototype and decide if we want to do it in 1.3 or wait for 2.0 where it is being dealt with already via ES6 modules (owners: Michał and Dave)
  + upgrade to jQuery 2.0 - not too risky, but should be done early one because all apps using Angular and jQuery will need to update to 2.0 as well. (owner: Michał)
  + [Lazy/debounced ng-model updates](https://github.com/angular/angular.js/pull/2129) (on blur) (owner: Ari, buddy: Pete)
  + Stripping of trailing slashes by $resource ([#992](https://github.com/angular/angular.js/issues/992), [#5560](https://github.com/angular/angular.js/pull/5560)) (owner: Michał)
  + [$asyncApply](https://github.com/angular/angular.js/issues/5297) (owner: Karl)
  + change watches and listeners to use linked list like we do in AngularDart, do this instead of [$onRootScope](https://github.com/angular/angular.js/issues/4574) (owner: Lucas + Ari)
  + bind-once / bind-notify / digest regions - we need a design doc for this first because there are many ways to deal with this particular thing and we need to consider them all (owner: Karl + Lucas)
  + Get rid of all IE8 hacks and deprecated features
    - we said that we won't actively remove IE8 support, but as Michał it doesn't make sense to drop IE8 support and not remove the code because IE8 will be broken very quickly by silly ES3 vs ES5 differences
    - owner: TBD - Ari might have a candidate, we are looking for a "junior" person who'd like to grow by working 1:1 with someone from the core team, this is going to be a multi-week project most likely
  + ng-options updates not rendering in IE9, IE10 ([#2809](https://github.com/angular/angular.js/issues/2809))
    - ng-repeat needs to be able to handle all cases, once it does we can deprecate ng-options, we should work towards this
    - keep piece of ingredient is ng-value support for <option> elements
    - owner: Dave
* Angular 1.3 WG member focus areas
  + Karl: performance
    - Will work on some PRs in Q2
    - Happy to help others working on similar PRs
    - Need to chat about which areas to focus on and what bind-once and bind-on-notify directives to move into AngularJS (if any)
  + Lucas: performance, compiler
    - Should $watch be extended so the checks are performed only on notifications (and use this for bind-on-notify)?
    - Should there be a mechanism to define directives that start with a given prefix (where ng-attr-\* is only one of them)? This same mechanism can be used for bind-once-attr-\* and maybe even ng-class-foo
      * What the implications for 2.0 would be if this is implemented at 1.3
  + Dave:
    - Spike on lazy loading / require.js w/ Michał
  + Michał:
    - jQuery-specific issues (e.g. porting to 2.x)
    - Spike on lazy loading / require.js w/ Dave
    - build issues (checking dependencies, shrinkwrap)
* Angular 2.0 WG member focus areas
  + [angular 2 design docs](https://drive.google.com/#folders/0BxgtL8yFJbacQmpCc1NMV3d5dnM)
  + Ari: documentation
    - Would like to work on initiating documentation (not sure what’s necessary for this yet: TBD)
      * agreeable documentation generation format / potentially a documentation generator tool
  + Dave
    - Project Structure Standard
    - Rich Models/Persistence (work w Jeff)
* other stuff
  + Igor to create an invite for next Monday's triaging party
  + Next meeting in two weeks, 1:1/small focused meetings in the meantime
  + Igor to identify an angular buddies from the core team for everyone on WG
    - Ari - Pete
    - Dave - Vojta
    - Lucas - Caitlin
    - Karl - Tobias
    - Michał - Vojta

Feb 24, 2014

## Who is here!

* at Google: Igor Minar, Misko Hevery
* Community: Dave Geddes (at Domo), Michał Gołębiowski (Laboratorium EE, jQuery core), Karl Seamon (DoubleClick)

## Agenda:

* v1.3 update
  + [current issue tracker list](https://github.com/angular/angular.js/issues?milestone=32&state=open)
  + Planning on doing the branch but got held up. Hopefully we should be able to do it this week. Plan is to merge 1.3.x milestone which contains more risky/controversial changes.
  + Dave: ng-route make parseRoute public <https://github.com/angular/angular.js/pull/4192>
  + Implicit Namespacing? <https://github.com/angular/angular.js/issues/2767>
  + David: Lazy loading? Could we get it into 1.3? Igor: if you (dave) implement it.
* v2.0 update
  + Igor: we spent most of our time last week. Bunch of design docs.
  + Package manager
    - Igor: Package management for angular is ready for review. See: <https://drive.google.com/folderview?id=0B7Ovm8bUYiUDR29iSkEyMk5pVUk&usp=sharing>
      * [batmobile.js example project](https://github.com/pkozlowski-opensource/batmobile.js)
    - Igor: more research needed but npm is closer to the ideal solution than bower,
    - Igor: npm does have issues though, for example with multiple versions of the same package
    - Igor: discussing the issues of bower and why we have chosen NPM. See the [Package manager doc](https://docs.google.com/document/d/14t1u5bjJV0TDJxNHhJnZe6Qu8Tn0muWytwjfuVPdAdY/edit?userstoinvite=mscales@google.com#heading=h.xgjl2srtytjt) for more info on the discussion. Plan is to prototype more workflow with NPM and if no issues found we will go with NPM.
    - Misko: what will be the name conventions? Igor: ng-\* prefix for core and angular-\* for community packages
    - Michał: using NPM but not require is strange: Igor: ES6 may make this issue moot. Michał: npm folks plan to support ES6 modules in npm out-of-the-box (probably): <https://github.com/npm/npm/issues/4323>
  + Templating
    - Misko: We are researching the next-gen templating for Angular, we have a design doc in progress but need a bit more time to polish it before it's shared later this week.
    - Misko: There are flaws in how templating works in 1.x, we are trying to identify them and address them: e.g. directive api is too complex, event handling could be better, toolability is not good (which value in template is string vs expression?)
    - Misko: We have new change observation (dirty checking) implemented in AngularDart. We can do 20k checks in 1 ms. We are planning to port this to JS.
* ng-swarm planning
  + Dave: have done some thinking, don't have anything planned out.
    - duplicates
    - issues which have been resolved but not closed
    - Issues have plunker / simple repro case.
    - Single event or on going program.
    - Earn t-shirts by closing bugs.
    - Create a score ladder to create a competition. This would create a roadmap for people on how to grow to become a committer.

# Feb 6, 2014

## Agenda:

- meet and greet

- working group vision/goals

- 1.3 priorities planning

- update on Angular 2.0

- brainstorming about more efficient community participation

## Who is here!

* at Google: Igor Minar, Misko Hevery, Naomi Black
* Community: Ari Lerner (educator, evangelist), Dave Geddes (at Domo), Michał Gołębiowski (Laboratorium EE, jQuery core), Martin Probst (Google), Karl Seamon (DoubleClick)

## Vision/Goals for the working group (Igor)

* Make good decisions on behalf of the community
* Help us coordinate work, empower the community to take ownership, delegate issues while retaining a common vision and direction for angular
* bring new ideas to the table, help us find ways to implement them.

Set expectations about what's involved in participating; if this isn't what you bargained for or your role changes or you want to be less involved, please speak up. We value transparency (and also value your time!)

We also request that the working group members serve as role models for the rest of the community when it comes to any interaction with Angular or other communities.

## Priorities

Dave --

Domo are working on a really huge app. Experiencing large app pain points, and as a personal goal, really committed to understanding and improving the large app use case.

Igor -- absolutely. But we also want to be sure that we're not just solving the problems of big applications. Finding ways to scale our solutions from small apps to biggest apps.

Dave, Karl, Martin very embedded in large apps.

Ari has a foot in each camp (large customer) but spends a lot of time teaching Angular, which forces a focus on small apps and components.

Q. How do we know when we're successful?

Igor - community growth is one way to measure this. We can also try surveys and gathering feedback to see if we're doing things that represent the community well.

Dave proposes something like NPS - net promoter score "how likely are you to refer X to your friends?"

Ari - asking the students, or the mailing list. We have a pulse on how angular is doing in terms of rate of questions we get on the mailing list. Metric based on the activities of the newsletter

Igor - link in the meeting notes which would allow people to give feedback. 'do you think we represented you well' 'Would you recommend this working groups to your friends'

Dave - 'how likely are you to recommend angular to your friends'

Ari - need to be careful about how questions are asked. It is hard to fake answers to questions like "How many lines of angular code have you written." to please the researcher. Need to be deliberate.

Dave - Have goals: drive down # issue on github. Drive down # regressions. What are our goals.

Igor - Getting github under control is the biggest goal. We can't do this on our own (core team). We think that WG can help. If we can get other from the community involved into existing processes. Need a better way to submit PRs/issue.

1. Need to improve the process. Subject/description is not enough. Most people don't know what a useful bug report is. We are working on issue templates through Mary Poppins, this will guide people in submitting high quality bug reports. This will help us understand issues and act on them faster.
2. Need help with Triaging. We spent 1hr/day whole team, and we can't keep up. We need more people involved. We had 66 people signed up to help via the [i-want-to-help](http://angularjs.org/i-want-to-help) form. Most on Docs. We are reaching out to them. Some help on the new DI. But there are some interested in triaging and reviewing PRs.
3. Dave ng-swarm: Once a month/week get big group to find duplicates and closing them, give feedback etc. Igor likes this idea for backlog, but not for ongoing basis. Input needs to be processed in timely manner. Regression is an example where time matters. Karl, Github does not allow you to delegate power to others, which may but a dent to ng-swarm.

Github is not working for us. The ACL tools is to primitive. Github is awesome for small projects but not for large projects. Maybe Mary Poppins can help with this. Igor has also reached out to Github and is working on in with them. Github did visual refresh which fixed some of our pain points. In the meantime we may have to work around them for a bit. We could also build something to show them what we need and motivate them through actual code.

Reference app could have a better ACL model, which could then direct Mary Poppins to do the work for us. But this is a lot more complicated.

Igor: I would like the WG to have commit access to the github repos, but we should go through PR so that we can show others how it is supposed to work. We are very careful about our syncing process and try hard to keep code quality up. With some guidance and evidence for trust we will give out commit access.

Misko: what about having lieutenants? for each component - like the linux model.

Igor: Good model, but it would not work for Angular v1, but it would work a lot better for Angular v2 because it is already broken up into projects.

Misko: things we could still do this for v1.

Igor: Angular is modularized, but there are still many pieces which share code across the codebase. It is common to touch many modules for each change. Also, we need to work more on v2 as opposed to v1. We are happy to empower community and improve v1 where they feel the pain. We are very concerned that if v2 is not ready by the time web-standards are ready, we are going to miss out. We are cautions talking about v1.3, because v2 is more important. We feel responsibility to the community for v1 so that's why we need to balance the future vs current apps.

Misko: can the working group be the shepherds of v1.x?

Igor: the working group should have more responsibilities than just taking care of the v1 code.

Misko: we should turn this over to WG.

Igor: how can we ship 1.3 without sacrificing v2? we dropped ie8. We asked people to help and no one stepped to support ie8, which speaks for itself.

Karl: my priority is 1.x right now, but would like to make myself available for v2 in summer.

Igor: how do you separate the people who are serious from the people who just do drive by code dump?

Dave: Big goals: getting 1.3 out the door and getting v2.0 done.

Igor: describes what is planned for 1.3 : see github milestone.

Karl: Bind once module.

Igor: bidirectional support for angular.

Igor: bug fixes only on 1.2 new features on 1.3. Fork 1.2 into separate branch and start working on 1.3. Not sure what is the best way to maintain 1.2. Most likely only low maintenance/low overhead/big impact changes.

Michał: Concerned that it will be way too much work split into too many versions.

Igor: Only cut new 1.2 versions on critical issues.

Michał: what to do for PRs on 1.2.x milestone. Igor we killed it and turned it into backlog. We can pull stuff out of backlog and merge into 1.3 if it's ready.

Michał: what about ie9 for v2? Igor: no, evergreen browsers only.

David: is new DI from vojta going to 1.3? Igor: no, but if we have volunteers from community, we can support them in creating a port.

Michał: DI question? Annotations are not part of ES6. => Igor: we have our own, and make it possible to use without it as well. Effect the standards in this way.

David: What about Python decorators like system instead of annotations? => Misko: there are issue with them. See [this doc](https://docs.google.com/a/google.com/document/d/1XhAjrNCHkvTjS-WNDuGNUu9Uzh7ma3TJrWB3P_fIs6g/edit#heading=h.qbaubqkoiqds).

Action items:

* [\*] Review issues in v1.3 milestone and find owners. Add comments on github to get the discussion going.
* [\*] Could you help us organize the ng-swarm so that we can get through the github untriaged backlog and clean it up.
* [\*] Follow up meeting next week, and decide how often to meet on regular basis.