**On-site Agenda**

* Standards (Jonathan, Anders, Ron, Misko, Igor)
  + Decorators and metadata
  + Types
* What's blocking angular.io (Mohamed, Vlad, Naomi, David)
  + Single File Compilation
  + Moving samples over to TS
* Dev tools for JS (All Eng)
  + Packaging
  + System.js
  + Hacking compiler
    - broccoli
    - preserving comments on transpilation
  + Minification
  + Include paths
* Angular on TS (Martin, Alex, Misko, Mohamed, Vlad)
  + Angular on TS
  + TS output for Dart
  + RTTS
  + Run-time type checking
* Build talk (Anders, Brad, Misko)
  + //build

**-----------------------**

**Overall list of topics to discuss**

* Angular → TS ([conversion plan](https://docs.google.com/document/d/14RJLhu6uuv7NchFkAb6PKzOOO0L7l3Z507eKWzkEUhQ/edit))
* Angular.io (quickstart examples) conversion AtScript → TS
* Hacking the compiler (for Dart output, etc)
  + BroccoliJS, development mode serving
  + (More) language services servers (running in broccolijs, interaction with IDE/editor)
  + Can our transpiler get access to the resolved AST, symbol table, etc?
  + Preserving doc comments on transpilation (Closure types?)
* Will TypeScript support the System.js shim?
* Decorator/annotation design
* Async/await and Angular promises
* What is the right way to deliver typings to users? Hook tsd into node module 'postinstall' hook? A new node module for each DefinitelyTyped package? Should all users commit typings into VCS or should they be fetched and cached locally like the npm model?
* minification/obfuscation story
  + Development Mode for minification/obfuscation
  + Interop with Closure Compiler and our Google internal tooling -- TypeScript->Closure Compiler
    - include path support,
    - dev mode
    - overall compatibility of TypeScript with Closure's optimizations and compilation
* Observables interface for ES2016 and a combinator library for it.
* Compiling in the browser
* Run-time type checking (not needed for Beta)
* Brad or Miško on stage with Anders for //build? (<http://www.buildwindows.com/>)
* Brad and Polita sync on telemetry
* ES6 modules and System.import

**Co-work - Areas to attack?**

* Migrating Angular to TypeScript
  + ~~Getting Dart unblocked~~
  + Reflection API use
* Compiling in the browser

Notes

* Single-file compilation (almost ready)
  + **[P0] TS**: Finishing up now
* System.js and jspm
  + **[P0] TS**: We could support System.register as another module output format
    - (est: roughly a few days to a week)
  + **[??] TS**: System register for packaging.
    - (est: roughly a few days to a week)
    - Alternative approach is to have a separate bundling step
    - Or, the compiler can handle bundling, and takes care of .d.ts output
* Internal compiler API
  + **[P2] TS/Martin**: Explore pass or internal API
  + Rewriters need to be able to maintain source maps
* Minification
  + **[P2] Martin**: exploring a pipeline into “JS compiler”
  + Q2 = April start
* Broccoli (the build tool, gulp replace)
  + **[✓] Mohamed/Igor/Steve Lucco**: Best path seems to hook into TS server
  + (will circle back)
* Doc generation and preserving comments
  + **[P1] TS**: Allow way to attach additional comments to nodes
* Include Paths
  + Seems like adding an include path is the simpler, more straightforward fix
  + **[P1] TS/Martin/Igor**: Need to explore designs for include path (loop in Martin and Igor)
* Angular wants to standardize on NPM
* Decorators
  + **[P0] TS, Angular, Yehuda**: Let’s put together a test suite
  + **[P1] TS**: Let’s split off decorators and the Reflect api, and we’ll emit the Reflect api calls
    - Angular: Will polyfill Reflect api
    - Current workaround to “kick can down the road”
  + For RTTS, current workaround is to implement classes, and also use classes in the RTTS
* Module loaders
  + Angular needs to work with pretty much everything
  + Igor would like everything to use ‘system.js’