Butch:

* Yesterday, setting up organization and our project in github.
* Setting up production. Cloned integration to AWS EC2. Need to determine appropriate name for production.
* One issue: Jenkins did not run node. Is that misconfigured? Ran node manually, but Jenkins doesn’t seem to be running node automatically. Jonathan to send email with other process that needs to be included which will include this.
* 18fchallenge.tcg.com
* Next up: Continuous monitoring with Nagios.
* Next up: Reviewing docker and getting smart on containerization.
* Next up: notifications from redmine.

Jonathan:

* Ticket 48. Gets back counts, and committed so Matt can run against.
* Complications for counting. There is a large amount of parsing.
* Blocker: New ticket either refine or pick a new approach.

Matt:

* Form setup and map working.
* Putting a calendar picker for dates and things like that.
* Matt to review the endpoint and start wiring it up.

Paul:

* Nothing.

Scott:

* Reviewed human centered design principles.
* Putting together questionnaires
* No blockers

Alex:

* UX stuff
* Cleaning up document
* 2 tickets for jonathan.

Post standup:

Core premise of the app is that we have a distribution of enforcement data and we can plot it.

List of example string is all over the board. We can get mostly there using the FDA endpoints. They’re not going to be 100%. E.g. Distributed in all states where Whole Foods exists. Very difficult to parse.

Can we do this in real time?

Can’t completely download all the data. Can only pull 5100 records. Some of them have 2000 – 3000 that we can’t even get.

Run a cron job to keep polling the data and build a database?

Device, food, drug enforcement. Date.

Filters affect counts. Use their data, too? To ensure that data is accurate.

Why maintain a hybrid of data? We’ve got an inconsistency problem.

Technical simple: 1) We need a mapping that we can execute. 2) Our side is, when you say New Jersey, we mean these 50 things.

Are we proposing to pull the data and normalize it? We have to normalize it. Or the reverse? Someone puts in NJ and then we search for Nationwide, NJ, New Jersey, into query.

Have unnormalized data > use external data to normalize it. No external data source to normalize.

Knee jerk would be to have a developer normalize.

Can we have the app be to normalize it?

Handler for a new class of string.

Application that shows unhandled events.

Can we use part of our application to normalize (i.e. crowd source)?

1. Display the data.
2. Help us make the data better. State fields that haven’t been assigned. Turn them into a state assignment. Turk: Have three people do it and average. Thumbs up, thumbs down.

Drop comments. #3 UI component lets a user give us feedback on one of the unknown things.

Real-time vs. storing a normalized version. Acknowledge that what we have is not 100%. Best educated guess given the counts that the API returns.

JP: Just using their counts is 80%. Standing up our own Elasticsearch and parsing is difficult.

Matt: Should we stand up a mapping FROM us to FDA.

JP: I go through and pick out the counts—one call for all 50 states. Matt’s way would be 50 searches, one for each state.

Two pronged approach:

Assumption is that mapping approach is still good.

1. Attack using Jonathan’s approach using API query approach that will get us partway there.
2. Paul to work on the backend of pulling down data and building a normalize version of the states to get closer to something we can query.
3. Matt is going to continue working on the map.
4. Jonathan is jumping to saving search criteria.