# Status of the Hardware Database

## What (else) needs to happen in order to release the production version?

1. Run a daily cron-job that checks popular REST API endpoints and observe stable conditions for a certain period of time.
   (the rest of today's report)
2. A test user uploads realistic data to both dev. and pro. versions successfully.
   We'll have a zoom meeting with him sometime this week (on Thu or Fri?).
   We like to clear this task by early next week.

For the 2nd item above, the setup on the pro. version is synced to the dev.,
so the production version should be ready, as long as the dev. is ready.

# Daily cron-job

- It started to run on my laptop at the end of the last November.
  And it's been running everyday since then, except two days.
- It is executed once a day, at noon (on my laptop) and
  sends out the results to us (me, Alex, and Urbas) through emails.
  (the noon execution time could change, of course.
  It runs at noon for now since it runs on my laptop,
  though our Test Units produces a minimal stress on the DB side
  so "what time to run" shouldn't be a concern)

- We'll soon start to run this job on a Linux box of the MN Physics Department.

- In the future, we like to start to run the job somewhere at FNAL.

# Daily checks

- At the beginning (until ~before the Christmas?),
  we kept receiving all sorts of “expected” errors while we were tuning the tests.

- In the last couple of weeks, it’s been stable, except two occasions:
  - On the new year’s day: All tests were failed.
    Found out that its proxy server was down so that we couldn’t even connect to the HWDB.
  - This past Saturday & Sunday (the 7th): A large fraction of the tests were failed.
    The proxy server was overloaded.

# Daily checks

- Check if one can:
  - POST/GET information of Component Types
  - POST/GET information of Items
  - POST/GET information of Test Types
  - POST/GET information of Tests
  - POST/GET information of Images

- Checks are done in two ways:
  - Have a template and compare when the expected response is known.
    I.e., check a posted test result that we post by ourselves.
  - When a response is expected to change (i.e., # of HWDB users),

  ➡ check the expected # of fields

  ➡ and their datatypes (string? int?..)

# Countries/Institutions/Manufacturers

(the actual endpoint is in blue)

- Check if one can GET a list of available Countries :
  /countries
- Check if one can GET a list of available Institutions :
  /institutions
- Check if one can GET a list of available Manufacturers :
  /manufacturers

# Users/Roles

(the actual endpoint is in blue)

- Check if one can GET info of a specific user :
  /users/<user ID>
- Check if one can GET a list of available users :
  /users
- Check if one can GET info of yourself :
  /users/whoami
- Check if one can GET a list of users & Component Types for a specific Role:
  /roles/<role ID>
- Check if one can GET a list of available Roles:
  /roles

# Projects/Systems/Sub-systems

(the actual endpoint is in blue)

- Check if one can GET a list of available Projects: /projects
- Check if one can GET a list of available Systems: /systems/<Project ID>
- Check if one canNOT GET a list of available Systems from a non-existent Project: /systems/s
- Check if one can GET info of a specific System: /systems/<Project ID>/<System ID>
- Check if one canNOT GET info of a non-existent specific System: /systems/X/80
- Check if one can GET a list of available Sub-systems: /subsystems/<Project ID>/<System ID>
- Check if one can GET info of a specific Sub-system: /subsystems/<Project ID>/<System ID>/<Sub-system ID>
- Check if one canNOT GET info of a non-existent specific Sub-system: /subsystems/Z/1/9

# Component Types

(the actual endpoint is in blue)

- Check if one can GET a list of Types for a given Project/System IDs:
  /component-types/<Project ID>/<System ID>
- Check if one can GET a list of Types for a given Project/System/Sub-system IDs:
  /component-types/<Project ID>/<System ID>/<Sub-system ID>

# Item - 1

(the actual endpoint is in blue)

- Check if one can GET a list of Items for a given Component Type IDs:
  /component-types/<Component Type ID>/components
- Check if one can GET def. of Component spec:
  /component-types/<Component Type ID>/specifications
- Check if one can GET info of an Item:
  /components/<EID>
- Check if one canNOT GET info of non-existent Item:
  /components/Z99999999999-99999
- Check if one can POST a new Item:
  /component-types/<Component Type ID>/components
- Check if one can BULK-POST new Items (we post two new items daily):
  /component-types/<Component Type ID>/bulk-add

# Item - 2

(the actual endpoint is in blue)

- Check if one can PATCH an existing Item (changing a serial #):
  /components/<EID>
- Check if one can PATCH multiple Items at once (changing serial #/specs/ manufacturer):
  /component-types/<Component Type ID>/bulk-update
- Check if one canNOT POST an Item with an extra field in its spec:
  /component-types/<Component Type ID>/components
- Check if one canNOT POST an Item with a missing field in its spec:
  /component-types/<Component Type ID>/components

# Sub-components

(the actual endpoint is in blue)

- Check if one can GET def. of sub-components:
  /component-types/<Component Type ID>/connectors
- Check if one can PATCH to Enable an Item (to be a sub-component candidate):
  /components/<EID>/enable
- Check if one can PATCH to Disable an Item (basically UNDO the above):
  /components/<EID>/enable
- Check if one can PATCH to Enable & Disable multiple Items at once:
  /components/bulk-enable
- Check if one can PATCH to create a sub-component link to another Item:
  /components/<EID>/subcomponents
- Check if one can POST to create a sub-component link to another Item:
  /component-types/<Component Type ID>/components
- Check if one can PATCH to remove the sub-component link we just created:
  /components/<EID>/subcomponents

# Tests

(the actual endpoint is in blue)

- Check if one can GET def. of a specific Test Type:
  /component-types/<Type ID>/test-types/<Test Type ID>
- Check if one can GET a list of Test Types for a given Component Type ID:
  /component-types/<Component Type ID>/test-types
- Check if one canNOT POST a Test with an extra field in Test spec:
  /components/<EID>/tests
- Check if one canNOT POST a Test with a missing field in Test spec:
  /components/<EID>/tests
- Check if one canNOT POST a Test with a mis-matched (invalid selection) Test spec:
  /components/<EID>/tests
- Check if one can POST a Test:
  /components/<EID>/tests

# Images

(the actual endpoint is in blue)

- Check if one can GET a list of stored images for a given Type ID:
  /component-types/<Type ID>/images
- Check if one can GET one of the images there:
  /img/<image ID>
- Check if one can GET a list of stored images for a given EID:
  /components/<EID>/images
- Check if one can GET one of the images there:
  /img/<image ID>
- Check if one can GET a list of stored images for a given record of Tests:
  Needs to be added.
- Check if one can POST an image for a specific Item:
  /components/<EID>/images
- Check if one can POST an image for a specific Component Type:
  /component-types/<Component Type ID>/images ….. will be added soon.

# Summary

- The Test Units have been running great.
  We will add a couple more tests next week.
  And we understand we'll just have to keep improving the tests by its nature
  at least for awhile.

- As for individual tests in the Test Units,
  you are very much welcome to request to add Tests that we have not covered!!

- To release the production version,
  we need to have successful uploads to the both dev/pro versions by a test user.
  Hopefully this would happen within a week as well.