HWDB validation tool(s)

- Want to run unit tests.. probably once a day
  to test various REST api commands.
  We will likely setup cron jobs on one of the U of M Physics dept.
  servers… under who's account should such cron-jobs be run?
  Mine? or c/should we have a dedicated account to run such jobs?
- Besides the available api commands,
  we also want to test the actual functions employed by our apps.

- Urbas Ekka has been developing Unit Tests for the Python API.
  The next few pages show lists of the already available Tests.

Get:

Component:

Swagger method	Unit test	description
/api/v1/components/**{part_id}**	Test_normal_item	Checks if response is expected response for given part id
/api/v1/components/**{part_id}**	Test_broken_item	Checks if api throws Server Error when run with a broken item
/api/v1/components/**{part_id}**	Test_invalid_item	Checks if api throws Error when given invalid # for part_id
/api/v1/components/**{part_id}/images**	test_image_by_part_id	Checks if response is expected response for given part id with associated image

Misc:

Swagger method	Unit test	description
/api/v1/**countries**	Test_countries	Checks if list of countries is the same as expected
/api/v1/**users/whoami**	Test_whoami	Checks if response is the same as alex wagner
/api/v1/**institutions**	test_institutions	Checks if list of institutions is the same as expected
/api/v1/**manufacturers**	Test_manufacturers	Checks if list of manufacturers is the same as expected
/api/v1/**projects**	Test_projects	Checks if list of projects is the same as expected
/api/v1/**roles**	Test_roles	Checks if list of roles is the same as expected
/api/v1/**users**	Test_users	Checks if list of users is the same as expected

/api/v1/**users/{user_id}**	Test_userid	Checks if response based on specific user_id is as expected
/api/v1/**role/{role_id}**	Test_role_id	Checks if response based on specific role_id is as expected
/api/v1/**subsystems/{project_id}/{system_id}**	Test_subsys_by_projid_sys (+checking that it throws error/empty list when appropriate)	1. Checks if response is expected given specific project_id and system_id 2. Checks if it returns an empty list when given invalid project_id/system_id
/api/v1/**subsystems/{project_id}/{system_id}/{subsystem_id}**	Test_subsys_by_projid_sys_subsys (+checking that it throws error/empty list when appropriate)	1. Checks if response is expected given specific project_id, system_id, and subsystem_id 2. Checks if it throws an Error when given invalid project_id/system_id/susbsystem_id
/api/v1/**systems/{project_id}**	Test_projid (+checking that it throws error/empty list when appropriate)	1. Checks if response is expected given specific project_id 2. Checks if it returns an empty list when given invalid project_id
/api/v1/**systems/{project_id}/{system_id}**	Test_projid_sysid (+checking that it throws error/empty list when appropriate)	1. Checks if response is expected given specific project_id, system_id, and subsystem_id 2. Checks if it throws an Error when given invalid project_id/system_id/susbsystem_id

Component Types:

Swagger method	Unit test	description
/api/v1/component-types/**{part_type_id}**	Test_part_type_id	Checks if response is expected for given part_type_id
/api/v1/component-types/{part_type_id}/**components**	Test_part_type_id_components	Checks if response is expected for given part_type_id
/api/v1/component-types/{part_type_id}/**connectors**	Test_part_type_id_connectors	Checks if response is expected for given part_type_id
/api/v1/component-types/{part_type_id}/**specifications**	Test_part_type_id_specifications	Checks if response is expected for given part_type_id
/api/v1/component-types/**{project_id}/ {system_id}**	Test_component_type_by_proj_sys	Checks 1st page of component types for given project_id and system_id
/api/v1/component-types/**{project_id}/ {system_id}/{subsystem_id}**	test_component_type_by_proj_sys_subsys	Checks component types for given project_id, system_id, and subsystem_id

Test:

Swagger method	Unit test	description
/api/v1/components/**{part_id} /test-types**	Test_part_type_id_test_types	Checks test_types of given part_type_id for expected response
/api/v1/component-types/**{part_type_id}**/test-types/**{test_type_id}**	Test_part_type_id_test_types_id	Checks component type based on part_type_id and test_type_id
/api/v1/component-tests-types/**{oid}**	test_comp_test_by_oid	Checks component types based on given id

- He will add tests for POST and PATCH.

Where are we going from here?

- Will determine the results of these tests more carefully.
  And will present them at this meeting (soon?)

- Will start to run them as cron jobs.

- He will then start to develop similar Test Units for the iPad app.
  Hopefully he could reach this point by mid October.