



AiiDA 1.0.0 Plugin Testing Guide

Leopold Talirz & Giovanni Pizzi

plugin migrations in progress

Links to "work in progress" branches

- [aiida-cp2k](#)
- [aiida-zeopp](#)
- [aiida-plugin-cutter](#)
 - aiida-diff (default output) [release_v1.0.0a1 branch](#)
- [aiida-vasp](#)
- [aiida-wannier90](#)
- [aiida-fleur](#)

aiida-diff - a minimalist example

aiida-diff

- plugin fixtures from `aiida.manage.fixtures`
 - automatically set up a postgres cluster
 - automatically set up a (test) profile
 - loads the dbenv
 - take care of resetting database in between tests
- aiida-diff uses `pytest`
 - we recommend `pytest` for testing plugins
 - `pytest` will *also* run tests written using `unittest` (aiida-diff contains [an example](#) of this)
 - if you want to stay with `unittest` you'll need to use a custom test runner for the plugin fixtures (see [manage.py in aiida-zeopp](#) for an example)

aiida-diff - a minimalist example

[aiida-diff](#)

- let's look at the code

Setting up Travis-CI

Using minimal [aiida-diff](#) example

- add `travis.yml` (see [here](#), [travis log](#))
- register on [travis.ci.org](#)

-  aiida-zeopp



 Settings

The next commit to your repo will trigger the build.

To do list / Discussions

1. fixtures: [allow use of existing test profile](#)
(saves a few seconds to set up postgres cluster)
2. [expose fixtures via entry point](#)
 - include: `get_computer`, `get_code`, ...
 - return test helper class
3. provide official docker container to test plugins?
(no longer need to configure postgres, rabbitmq, install aiida-core, ...)