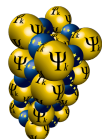




Getting started with writing AiiDA plugins

Group B

Leopold Talirz



swissuniversities



Outline

Slides / demos:

1. aiida-diff: let's look at the code!
2. **aiida-siesta**: Intro & plugin structure
3. aiida-plugin-cutter
4. get your new plugin registered
5. pre-commit hooks & IDEs
6. setting up CI **tests**

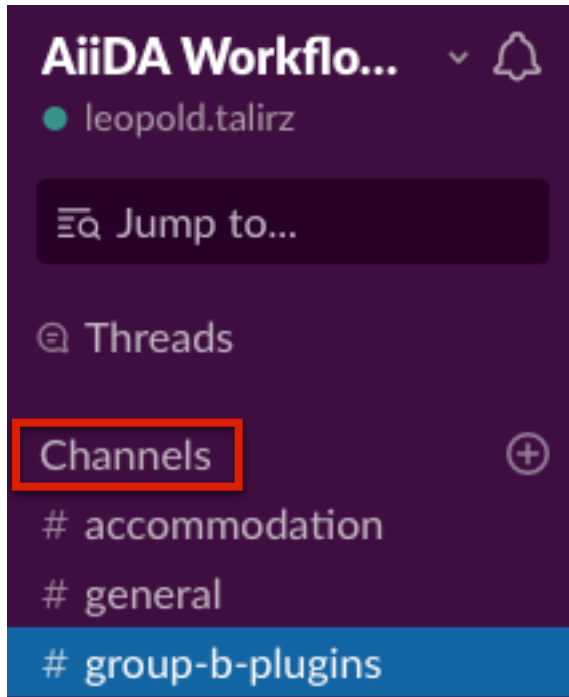
Tutors

- Leopold Talirz
- Oscar Arbelaez
- Daniele Tomerini
- Alberto Garcia

Afterwards

- Assistance for AiiDA setup on your hardware
- Get started with writing AiiDA plugins!

slack channel



#group-b-plugins

Going to look through code!

Links will be posted there

aiida-diff

Branch: master ▾ [aiida-diff](#) / [aiida_diff](#) /

Create new file Upload files Find file History

 **Italirz** transform get_computer and get_code into fixtures (#22) ... Latest commit 2ae3ac1 2 hours ago

..

data	improve tests of data cli (#21)	a day ago
tests	transform get_computer and get_code into fixtures (#22)	2 hours ago
__init__.py	transform get_computer and get_code into fixtures (#22)	2 hours ago
calculations.py	transform get_computer and get_code into fixtures (#22)	2 hours ago
cli.py	improve tests of data cli (#21)	a day ago
helpers.py	transform get_computer and get_code into fixtures (#22)	2 hours ago
parsers.py	improve tests of data cli (#21)	a day ago

aiida-siesta

aiida-plugin-cutter

README.md

AiiDA plugin cutter

Cookie cutter recipe for [AiiDA](#) plugins.

The fastest most convenient way for getting started with developing AiiDA plugins.

For the default output of the plugin cutter, see the [aiida-diff](#) demo plugin.

Note: This cuts plugins for `aiida-core>=1.0.0`. See the `support/aiida-0.x` branch for cutting plugins for older versions of `aiida-core`.

Usage

```
pip install cookiecutter yapf
cookiecutter https://github.com/aiidateam/aiida-plugin-cutter.git
```

DEMO - register a new plugin

AiiDA registry of plugins

[\[View on GitHub/register your plugin\]](#)

Global summary of the AiiDA plugin registry

Total number of entries: 36

Calculations	62 plugins in 25 entries
Parsers	55 plugins in 25 entries
Data	24 plugins in 13 entries
Workflows	53 plugins in 11 entries
Other	40 plugins in 12 entries

Available plugins (alphabetically sorted)

[alloy: aiida-alloy package](#)

Current state: development

[Plugin code homepage](#) (hosted on github.com)

Documentation: Documentation not provided by the plugin author

[Show plugin details](#)

[ase: aiida-ase package](#)

The official AiiDA plugin for ASE

Current state: stable

[Plugin code homepage](#) (hosted on github.com)

Documentation: [Go to plugin documentation](#)

Calculations 1 Parsers 1

[Show plugin details](#)

pre-commit hooks

- run a program every time you make a git commit
- aiida-diff comes with
 - formatter (yapf): automatic code styling
 - linter (prospector/pylint):
static code analysis & code
 - python-modernize: ensure "automatically" that code
is python 2 & 3 compatible

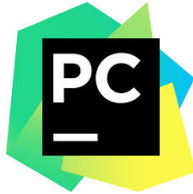
- Enable using

```
pip install -e .[pre-commit]  
pre-commit install
```


IDEs

- vim and emacs are highly efficient & flexible text editors
- **BUT** today's IDEs offer tons of convenience features:
 - TAB-auto-complete names of functions/classes/...
 - jump to definition of function X/ class Y with one click
 - flag unused imports
 - flag programming errors via static code analysis
 - set breakpoints with one click & run debugger, inspect values of variables, ...
 - ...

IDEs - two examples



pycharm

- python only
- free community edition / pro edition free for students
- can be resource hungry & takes some time to open
- niceties
 - "search everywhere"
 - vim emulation mode

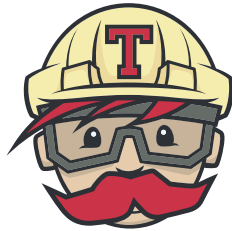


Visual Studio Code

- multi-programming-language (but lots of python features)
- free and open source
- less resource hungry & fast to open
- niceties
 - show git commit on hover
 - click on test to run

There's more: SublimeText, Atom, ...

Continuous integration tests on Travis



How to enable #1: Activate on Travis-CI

Travis CI

Blog Status Help

Giovanni Pizzi



Giovanni Pizzi
Repositories 3

Token: 

Beta Features

See what's new!

Organizations



AiiDA team
Repositories 5



Quantum ESPRESSO
Repositories 12



Wannier90 developers
Repositories 1

Is an organization missing?
[Review and add your authorized organizations.](#)

Giovanni Pizzi

Sync account

We're only showing your public repositories. You can find your private projects on [travis-ci.com](#).

1



Flick the repository switch on

2



Add .travis.yml file to your repository

3



Trigger your first build with a git push

☐

 giovannipizzi/aiida_core

☒

 giovannipizzi/seekpath

☒

 giovannipizzi/wannier90

Just click here

to enable integration between your GitHub repo and Travis-CI

 12

Transition to aiida-core 1.0

- Many API changes from aiida 0.12 to 1.0
- AiiDA API **should remain stable** from aiida-core 1.0.0b2

Videos

- [aiida-core 1.0.0b2 \(plugin migration lectures\)](#)
- [aiida-core 0.9.0 \(lectures\)](#)
- [aiida-core 0.8.0 \(short videos\)](#)
- [aiida-core 0.6.0 \(lectures\)](#)

aiida-tutorials.readthedocs.io

Useful resources

- [AiiDA release roadmap](#)
- [AiiDA plugin registry](#)
- [Git cheatsheet](#)
- [How to migrate from BitBucket to GitHub](#)
- [RST tutorial for GitHub](#)
- [AiiDA 1.0 plugin migration guide](#)

github.com/aiidateam/aiida_core/wiki

Plugin design guidelines

1. **Start simple.**

Use existing classes like `Dict`, `SinglefileData`, ...

Write only what is necessary to pass information from/to AiiDA.

2. **Don't break data provenance.**

Store *at least* what is needed for full reproducibility.

3. **Parse what you want to query for.**

Make a list of which information to:

1. parse into the database for querying (`Dict`, ...)
2. store in files for safe-keeping (e.g. `SinglefileData`, ...)
3. leave on the remote computer (`RemoteData`, ...)

4. **Expose the full functionality of your code.**

Don't artificially limit the power of a code you are wrapping.

If the code can do it, there should be *some* way to do it with your plugin.

