Git Auto-creation

This is a simple process that does some heavy lifting for your course and otherwise stays out of the way.

How it Works

To get started, read The Introduction, and the rest of this document. Email `techstaff@cs.uchicago.edu` and indicate which course you would like to use it with. We can start creating project structures immediately.

For each course that is enabled for automation, by default:

- A dedicated Namespace is created in Gitlab, using course numbers by default.
- Repo and Project resources are refreshed from official rosters each night. Any customizations you make are refreshed and merged with the upstream data sources once per hour.
- Each student is given an individual Project beneath the main project (git repository).
- Nominated Graders get appropriate access roles to Projects and the enclosing namespace.
- Subsequent customization is ignored or respected by the automation (add, drop).
- An administrative project is established for Instructors and TAs to optionally continue to modify the provisioner (refresh rate is once per hour).
- The tool is flexible and can be modified to suit different situations.

Accessing Repositories

The Gitlab server emails each individual at the time they are granted access to a resource. If you have not received an email with repository details and think that you should have, please write Techstaff and let us know.

Hints for finding a repository:

- The current server is named Proj , and you can login and look around
- The entire namespace path is identical to the Course identifier, e.g., `https://proj.cs.uchicago.edu/mpcs-53001-aut-20`
- Use Gitlab built-in search if necessary

Controlling The Rosters

The overall automation tool is controlled by Techstaff. However, online customization of repository and project details are available to instructors and other course staff.

Internal Repo-based Configuration

Before establishing any repositories, the system can be optionally configured to read data that is published in **your own** secure Gitlab repository. Note that the automations can only read from certain known Gitlab servers.

Every hour, the automations bot will checkout the head of your `main` or other nominated branch and scan a directory for YAML files containing course and enrollment data.

Because of Roster Config Merging, you typically only need to *augment* the Registrar's enrollment data. You are able to add and drop students, and elevate or modify other roles. See Advanced Usage.

It is also possible to specify partial information, or split data across multiple files in your repository. All configuration files are merged with others having the same course identifier.

```
CMSC-3456-aut-2020:
  display_name: Optimal Data Structures 2020
  memberships:
    student:
      - tdobes
    grader:
      - chudler
      - ctopper
```

An example to create course without any associated Registrar data is the same, but includes more memberships

```
# sample config to be checked into a file rosters.d/sample.yaml
#
my-globaly-unique-id-1234-aut-2020:
  display_name: my-special Custom Course Taking Place in 2020
  memberships:
    instructor:
      - rdb
    student:
      - kauffman
      - tdobes
    ta:
      - chudler
    grader:
      - ctopper
```

Configuration Merging

Techstaff will augment any configuration you provide with roster data from the University Registrar. The union of memberships is considered, and scalar values are overridden by your values.

More Roster Configuration Examples