

## Git Auto-creation

This is intended to be a simple process that does heavy lifting and stays out of the way.

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

- A dedicated Namespace is created in Gitlab
- All students are given individual Projects (git repository)
- Nominated Graders, and others, get appropriate access roles to Projects and the enclosing namespace
- Customization is respected by the automation
- Registrar data (rosters) are processed nightly, and data is refreshed every 1 hour

## Controlling The Rosters

The overall automation tool is controlled by Techstaff. However, certain levels of customization are available to instructors and other course staff.

### Internal Repo-based Configuration

Before establishing any repositories, the automations server can be optionally configured to read configuration 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/ Registrar data.

It is also possible to specify partial information, or split data across multiple files. All configuration files are merged with others having the same 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-globally-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](#)

From:

<https://howto.cs.uchicago.edu/> - **How do I?**

Permanent link:

[https://howto.cs.uchicago.edu/vcs:gitlab\\_roster\\_usage?rev=1593431608](https://howto.cs.uchicago.edu/vcs:gitlab_roster_usage?rev=1593431608)

Last update: **2020/06/29 06:53**

