Advanced Features of code.seas

Overview

Once you've successfully started storing your data in the SEAS Code Repository, as outlined in the page Getting Started with code.seas, you can start using the “meta” features of the service. This page touches lightly on these capabilities.

Table of Contents:

- Overview
- Invite Collaborators
  - Manage Read Access
  - Manage Individual Collaborators
  - Teams
    - Defining a Team
    - Using Teams
      - Projects and Teams
      - Repositories and Teams
- Using the Wiki
- Browse Source and Diffs
- Source Bundles

Invite Collaborators

Then point of the SEAS Code Repository is invite and support collaboration between individuals. The service provides the ability for individuals to define their own teams and apply these teams to both projects and repositories. Teams defined as groups, or in a ad hoc fashion, and can be applied separately to projects or repositories.

Manage Read Access

To add an individual or a team as a collaborator, you need to add him to the “Manage access” list of the private project and/or the “Manage read access” list of the private repository.

For private project : “Manage access” is a button on the right panel of the project editor.
For private repository: “Manage Read access” is a button on the right panel of the repository editor.
To start, you can invite individuals to work with you on a repository directly, without needing to define a team. When you do this, you can retain "ownership", but give the other individual access to your repositories.

Browse in the GUI to a repository that you own. Click the button on the right labeled "Manage Collaborators".

What does the different permissions mean?

Viewers:
- Can see the repository

Reviewers:
- Can update merge request statuses
- Will receive notifications on new merge requests

Committers:
- Can push commits to the repository

Administrators:
- Can set the repository name, description and settings
- Can manage collaborators
- Can delete the repository

Note that the permissions do not cascade down, meaning that for instance a committer is not automatically a reviewer

From here, click "Add Collaborators" on the right, and take a look at the form on the following page
Here you can specify that a user has a set of roles on that repository; the text on the page explains the scope of each role. For close collaborators, often “View, Commit, Review” is the set of roles you’ll want to add, but for external collaborators – those you don’t want to be able to make core changes – you may just want “View” permissions.

Once you’ve selected the desired roles, hit “Add as a collaborator”.

Teams

Instead of adding collaborators individually, you can instead define “teams”, which are groups of users that work together. To share access and ownership of projects, the use of teams is required, but is also an option for managing collaborators on repositories as well.

Defining a Team

To define a team, click on the “Teams” main link just below the banner on the right.
From there you'll see a list of teams, and can browse them. From here, click the button on the right sidebar "Create a new team."

Create a new team

This form page lets you define a team, with an optional description and team logo image.

Once the team has been created, you can click on it to see its status page. From here you can add members to the team, change member roles, and edit the team's name, logo, etc.

Using Teams

Once you have a team created, you can manage control of projects and repositories using the teams.

Projects and Teams

For projects, the main mechanism for doing this is by changing the project ownership from an individual (you) to the team. Then the team members will have administrative access to the project.
Repositories and Teams

For repositories, teams can be used to specify collaborations in the same way that individual roles are added. Refer to the docs above under the heading "Manage Individual Collaborators," and instead of adding a user, add a team as a collaborator. The resulting role permissions will be applied to all team members at once.

Using the Wiki

Any project can have a wiki associated with it, for basic documentation.

Simulating Nasal Hair Growth

An example project intended for the SEAS Code Repository HowTo at:
* https://ac.seas.harvard.edu/display/USERDOCS/Getting+Started+with+the+SEAS+Code+Repository

Browse to the page for a project that you own, then click the "Open Wiki" button on the sidebar on the right.
This will start off by letting you edit the "Home" page of the wiki, using Markdown notation. In particular, if you add text to link to a new page of the syntax `[[NewWikiPage]]`, then this will create a link to a new wiki page, and clicking on it in the saved page will let you create this new wiki page (in typical wiki fashion).

You can also browse page history and see the index of pages in the wiki from the righthand sidebar. Lastly, the wiki contents can itself be treated like a git repository, and checked out using GIT from the buttons on the sidebar while browsing Wiki contents.

Browse Source and Diffs

From the main page for the repository, click "Browse Source."

<table>
<thead>
<tr>
<th>File</th>
<th>Last Modified</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hello.c</td>
<td>19 Oct 16:05</td>
<td>some basic code</td>
</tr>
<tr>
<td>Makefile</td>
<td>19 Oct 16:05</td>
<td>some basic code (default)</td>
</tr>
<tr>
<td>README</td>
<td>19 Oct 16:02</td>
<td>added extra line of text</td>
</tr>
</tbody>
</table>
From this interface you can browse the source tree through the web interface.

If instead you click “Commit Log” from that page, or the same button in the sidebar on other pages, you can browse the source tree again, but by version.

**Commits in simulating-nasal-hair-growth/growth:master**

<table>
<thead>
<tr>
<th>Time</th>
<th>Author</th>
<th>Revision</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>16:11</td>
<td>example-user</td>
<td>8cd213a</td>
<td>committed improvements</td>
</tr>
<tr>
<td>16:05</td>
<td>example-user</td>
<td>f9e53f6</td>
<td>committed some basic code</td>
</tr>
<tr>
<td>16:02</td>
<td>example-user</td>
<td>0fabc10</td>
<td>committed added extra line of text</td>
</tr>
<tr>
<td>15:46</td>
<td>sample-instructor</td>
<td>d58fffc</td>
<td>committed A first step</td>
</tr>
</tbody>
</table>

From here, you can click on a revision hash (i.e. something like “8cd213a”) to go to an page where you can edit comments and find diffs between that and the previous version:

Commit 8cd213a4788130c5144218d317e14d77747622c5

NB! This page relies heavily on JavaScript, enable it for a better view.

**Diffs**

**Comments**

**Add a new comment:**

Comment (Use Markdown for formatting)
If you don't need to download (or publish) every change, but instead want to code of the latest version, you can simply pull down a compressed bundle of the code directly.

<table>
<thead>
<tr>
<th>Makefile</th>
<th>Diff rendering mode: inline side by side</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>1 hello: hello.c</td>
<td>1 hello: hello.c</td>
</tr>
<tr>
<td>2 $(CC) -o hello hello.c -I.</td>
<td>2 $(CC) -o hello hello.c -I.</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4 clean:</td>
<td>clean:</td>
</tr>
<tr>
<td>5 rm -f hello.o hello</td>
<td>5 rm -f hello.o hello</td>
</tr>
</tbody>
</table>

Hello.c

```c
4
5 main();
6 {
7 printf("Hello World");
8 
9 }
```

Browse to the page of the desired repository, and then click on "Source tree". Once this page you can explore the source code, but on the right hand side is a button "Download as tar.gz". Clicking on this will request the server to generate a "tarball" of the code int he current branch, and to download it through your browser to your local system.