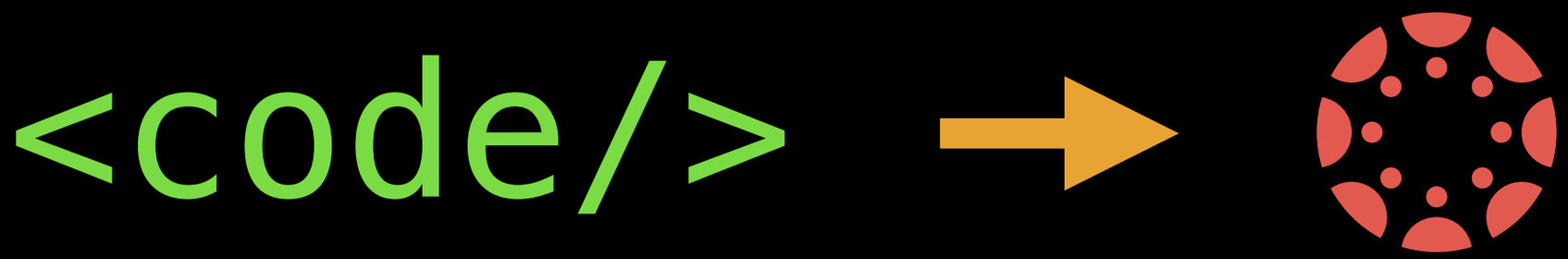



# Do It Yourself!

## Contributing Code Back to Canvas

# But why?

# Turnaround

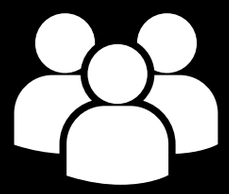
# Maintainability

# Everyone Benefits

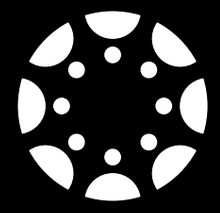
# Getting Started

# The Canvas Community

Freenode
#canvas-lms



Google Groups
canvas-lms-users



Discussion
Forums

# Prerequisites

Familiar with Ruby on Rails

Familiar with Git (and GitHub)

Able to run Canvas locally

Sign their Contributor License Agreement

# Setting up Your Environment

# Quick Start

**phaggood** edited this page on Apr 29 · 30 revisions

# Quick Start

This page will explain how to get a working version of Canvas LMS running as quickly as possible. Note that you probably don't want to use these instructions for getting a production-ready system running; for that, please see Production Start.

If you need help installing Canvas or troubleshooting your installation, your best bet is to join the community mailing list or the IRC channel (see the wiki) and ask specific questions there. It's likely that somebody else has already tackled the same problem.

## Prerequisites

This tutorial is targeting POSIX-based systems (like Mac OS X and Linux). This tutorial was written and tested using Ubuntu's latest LTS 10.04.1, Mac OS X 10.6 (Snow Leopard), and Debian Squeeze 6.0. If you have a different system, consider setting up a server or virtual machine running the latest Ubuntu LTS. We'll assume you've either done so or are familiar with those working parts enough to do translations yourself.

A note about Debian - you will have much more luck if you use Debian Squeeze (6.0) or newer. You'll probably have better luck if you go ahead and create a 'canvas' user as sudoer.

Pages 12

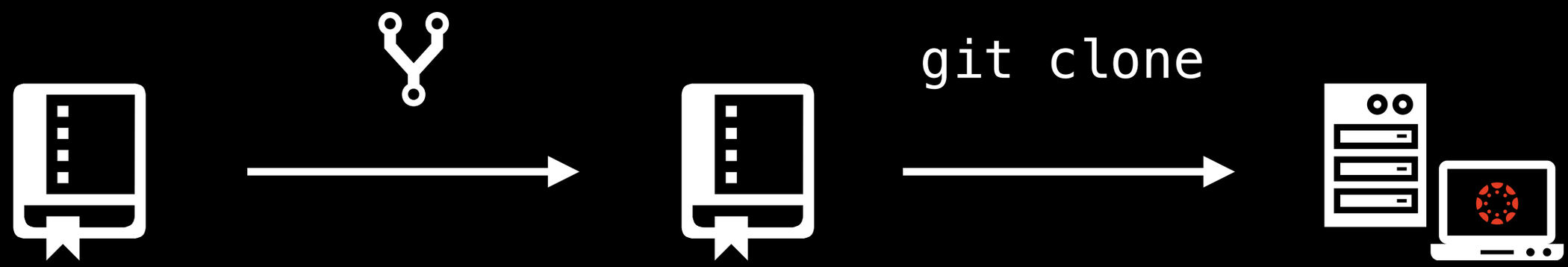
Canvas Integration
Coding Guidelines
Ember Components
FAQ
Home
I18n
Kaltura setup instructions
Production Start
Quick Start
REE Instructions
Troubleshooting
Upgrading

Clone this wiki locally
https://github.com/instructure
Clone in Desktop

https://github.com/instructure/canvas-lms/wiki/Quick-Start

gotta get git +

# Fork, then Clone




instructure/
canvas-lms

your-fork/
canvas-lms

# Install Dependencies

- Ruby 1.9.3 — use RVM or rbenv
- Other Canvas dependencies

# Prepare Canvas

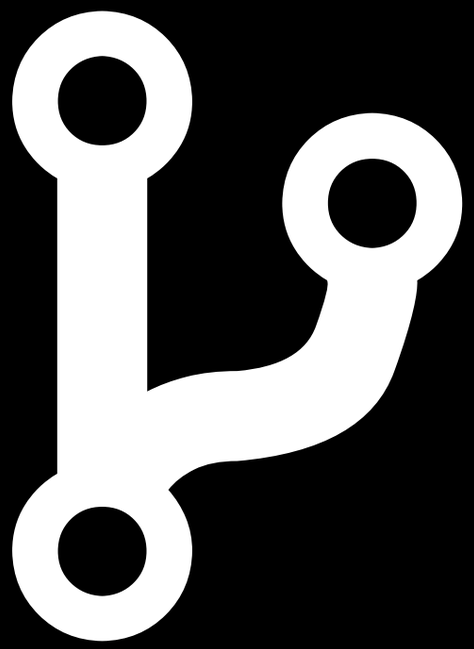
- Set up configuration files from samples
- Initialize databases
- Run tests

```
> script/server
```

http://localhost:3000

# Writing Your Code

master



feature / bug fix branch

# Coding Guidelines

**leiAscham** edited this page on Mar 30 · 4 revisions

Getting code folded into Canvas core is not trivial. You should take a second right now and see if maybe your change would be better implemented as a separate service using the API or LTI. We have a high quality standard at Instructure and if you're not familiar with both Ruby and Rails then getting your code accepted may be difficult. We're happy to give suggestions on how to improve commits, but we're not going to teach you how to write Ruby code when you issue a pull request.

Every commit for Canvas is reviewed by at least one Instructure engineer (this is all true of Instructure-created code as well, btw). Every line is read and the reviewer is responsible for checking out the commit and testing it in their local environment. As such, commit messages need to provide enough information that the engineer knows what has changed and what should be tested.

The following checklist is worked through for every commit:

- Check out and try the changeset.
- Ensure that the commit message has a test plan.
- Ensure that the tests and test plan cover all necessary cases.
- Ensure that the code follows the language coding conventions.
- Ensure that the code is well designed and architected.
- Ensure that all user-facing strings/dates/times/numbers are internationalized.

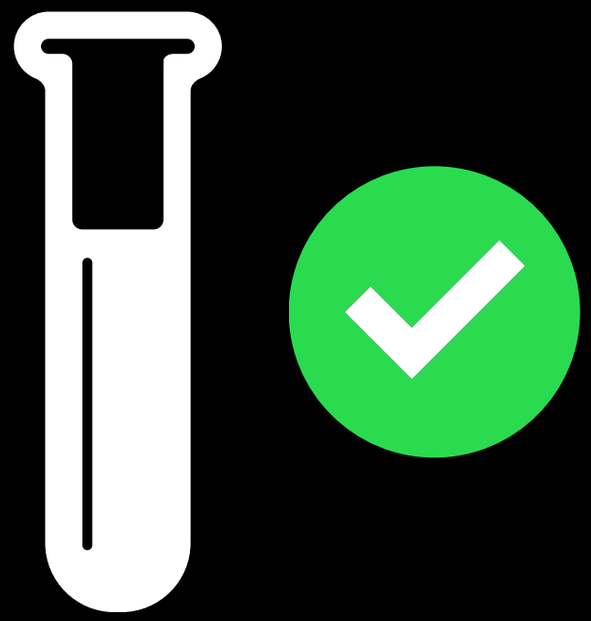
Other factors that should be considered:

- Must remain performant under heavy load.

Pages 12
Canvas Integration
Coding Guidelines
Ember Components
FAQ
Home
I18n
Kaltura setup instructions
Production Start
Quick Start
REE Instructions
Troubleshooting
Upgrading

Clone in Desktop

https://github.com/instructure/canvas-lms/wiki/Coding-Guidelines



Test Your Code

Ask Questions

# Submitting Your Code

# Tidy Up

```
5001eb Implement awesome feature
9a278a Write tests
b2d9b2 Fix typo “awsome”
fa50ea Add big green button
ba0786 More pandas
1fad4e One more thing...
```

# Tidy Up

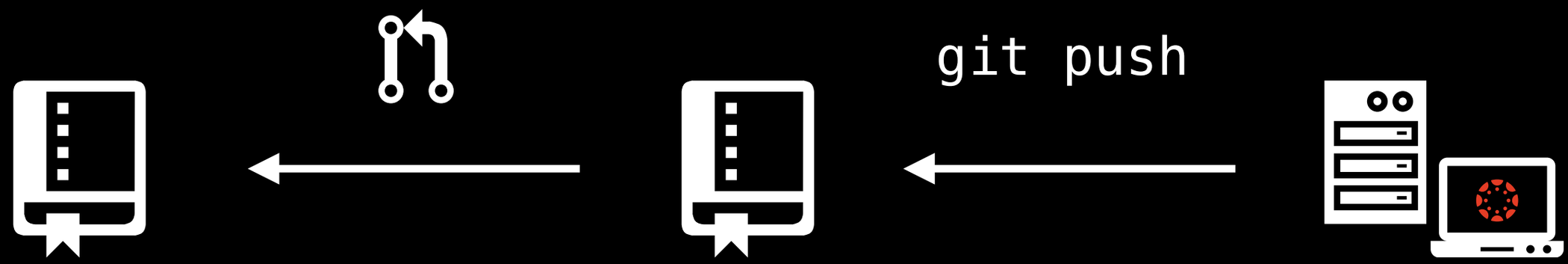
```
1985aa Implement awesome feature
```

# Write a Commit Message

```
Implement awesome feature

Test plan:
- As a teacher, select any active course
- Ensure that the course page looks normal
- As a hero, select the same course
- Click on the big green button
- Experience a 215% increase in pure awesomeness
```

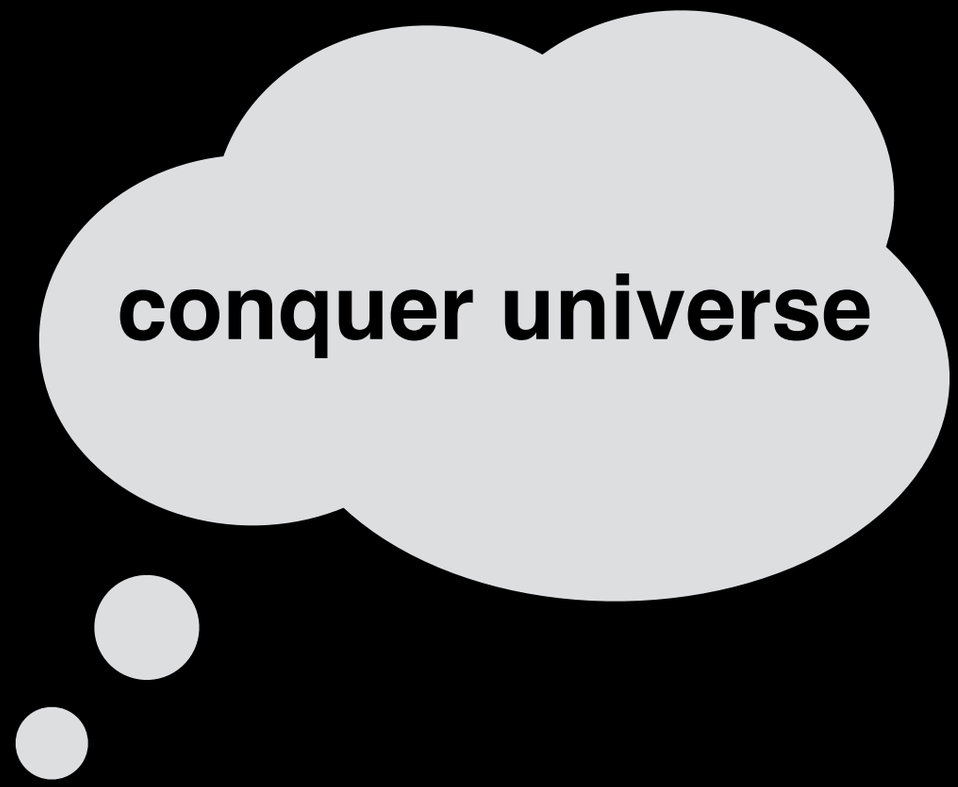
# Send Pull Request




instructure/
canvas-lms

your-fork/
canvas-lms

# Tips




hello world

LTI · API · separate service?

# Internationalization

# Make UI Accessible

# Examples

# Basic File Uploader Bug

Sandbox Course

## Sandbox Course

Add File | Add Folder

Nothing in this Folder

Storage Used: 0 Bytes out of 5 GB

see files for all your courses/groups

Sandbox Course

Sandbox Course

Add File | Add Folder

Choose File no file selected

Add File Cancel

Storage Used: 0 Bytes out of 5 GB

see files for all your courses/groups

# Alerts Bug

Home
Announcements
Assignments
Discussions
Grades
Pages
Files
Syllabus
Outcomes
Quizzes
Modules
Collaborations
Library Reserves
Import WebCT Content
People
Settings



# Alerts

An alert is generated for each student that meets all of the criteria. They are checked every day, and notifications will be sent to the student and/or the teacher until the triggering problem is resolved.

**Trigger an alert when:**

No Teacher Interaction ⊕ Add Trigger

**Send a notification to:**

The Student ⊕ Add Recipient

**Repetition:**

- (•) Do not re-send alert
- ( ) Re-send alert every _____ days until resolved

Cancel | Save Alert

⊕ Add Alert

```
authorized_action(@context, @current_user, :manage_alerts)
```

```
authorized_action(@context, @current_user, :manage_alerts)
:manage_alerts
```

```
authorized_action(@context, @current_user, :manage_alerts)
:manage_alerts

:manage_interaction_alerts
```

# Enrollment Terms in Course Autocomplete

## Find a Course

math1 | Go

MATH100 D100 Precalculus

MATH100 D100 Precalculus

MATH100 D200 Precalculus

MATH100 D200 Precalculus

MATH113 D100 Euclidean Geometry

MATH130 D100 Geometry for Computer Graphics

MATH150 C100 Calculus I with Review

MATH150 C100 Calculus I with Review

MATH150 D100 Calculus I with Review

MATH150 D100 Calculus I with Review

MATH150 D100 Calculus I with Review

MATH150 D200 Calculus I with Review

## Find a Course

math1 | Go

- MATH100 D100 Precalculus
- MATH100 D100 Precalculus
- MATH100 D200 Precalculus
- MATH100 D200 Precalculus
- MATH113 D100 Euclidean Geometry
- MATH130 D100 Geometry for Computer Graphics
- MATH150 C100 Calculus I with Review
- MATH150 C100 Calculus I with Review
- MATH150 D100 Calculus I with Review
- MATH150 D100 Calculus I with Review
- MATH150 D100 Calculus I with Review
- MATH150 D200 Calculus I with Review

## Find a Course

math1 | Go

- MATH100 D100 Precalculus
  *Spring 2014*
- MATH100 D100 Precalculus
  *Fall 2013*
- MATH100 D200 Precalculus
  *Fall 2013*
- MATH100 D200 Precalculus
  *Spring 2014*
- MATH113 D100 Euclidean Geometry
  *Spring 2014*
- MATH130 D100 Geometry for Computer Graphics
  *Spring 2014*
- MATH150 C100 Calculus I with Review
  *Summer 2014*

# Names in Users CSV Import

```
user_id, login_id, first_name, last_name, status
13834, jdoe03, John, Doe, active
```

```
user_id, login_id, first_name, last_name, status
13834, jdoe03, John, Doe, active
```

```
user_id, login_id, first_name, last_name,
short_name, active
13834, jdoe03, John, Doe, “Johnny D”, active
```

# Thank You!

andrewleung@sfu.ca

va7map