

# Technical Specifications

---



# Infrastructure

## Unlimited Ubuntu servers

A full working Ubuntu server that starts in a couple of seconds. Each user can have any number of servers.

## 512KB thru 2GB RAM

Our default memory allocation is fine for 99% of use cases. However, if you need more for memory hungry applications, you've got it.

## Sudo privileges

You can install any programming language, database or component just like you can on a regular machine.

## Storage

Each project can have up to 5GB of storage space with more available on request.

## Scaling

The Codio infrastructure serves every student in every state and country, so we are constantly dealing with very large numbers of users already and more and more every day.

## GUI Applications

Teaching GUI based application development works perfectly with the Virtual Desktop.

## Always On Boxes

Codio servers automatically shut down when they are not in use and start on demand in a couple of seconds. If you don't want your server to shut down when not in use, you can use our optional Always On Box allocation.



# Online IDE

## Any modern browser

Runs on all modern browsers such as Chrome, Firefox, IE, Edge, Safari and more.

## Multiple code panels

Like most powerful IDEs, you can create and destroy panels for code editing, terminal access and web page previews.

## Command line access

Terminal window providing access to the command line with sudo level privileges.

## Debugger

Full visual debugger with support for Java, C, C++ and NodeJS. Other languages can be added on request.

## Virtual Desktop

Create GUI based applications.

## File tree

The file tree is fully functional and supports drag and drop, uploading and downloading of files.

## Manual grading

Access manual grading features, including grading rubric and grading moderation within the IDE while reviewing student code.

## Remote deployment targets

Configure remote deployment targets and deploy your code via FTP, SFTP, SSH or rsync.

## SSH In

SSH into your Box from your local PC.

## SSH Out

Configure remote SSH servers for easy access from the command line.

## Search and Replace

Full search and replace functionality including regex support.

## Code beautifier

Automatically format and indent your code (some languages only).

## Command bar

Access any IDE feature by typing in a search string rather than locating through menus.

## Share access

Give read/write/admin access to your server to other users.

## Fork projects

Fork another user's project, including the server stack, into your own account.

## Server restart

Reboot the server, which takes only 10 seconds from within the IDE.

## Export code

Download a single file or the entire project to your local PC.

## Upload files

Files can be uploaded from your desktop directly into the file tree.

## QR code

For web applications, generate a QR code so students can try their project out on a smartphone without typing in long URLs.

## Git

Git is preinstalled on every Codio server. We also offer some menu shortcuts for simple operations as well as displaying the current branch name in the file tree. Your user SSH public key can also be uploaded to your Git remote using a menu.

## Syntax highlighting

Supported for almost every language.  
Additional languages easily added.

## Code complete

Language dependent

## Intellisense

Javascript has full intellisense but limited support currently for other languages.



## Emmet

Powerful support for HTML, CSS and Javascript keyboard shortcuts.

## Themes

We offer a number of light and dark themes to suit your preferences.

## Install software shortcuts

A dialog driven approach to installing commonly used components with a single click without having to install yourself from the command line. We take requests for additional items.

## Custom menus

To prevent students from having to use the command line, add commands to a dropdown menu.

## Custom settings

A list of over 70 settings that customize the code editing experience.

## Codio support

Contact Codio support directly from within the IDE. You can also track all support conversations, current and past.

# Grading & Auto-Grading

## Standard auto graded code tests

Dialog driven, this approach lets you create tests based on argument and stdin inputs and match expected output. You can create several test cases for each assessment to ensure full code coverage. Caters for at least 80% of cases.

## Advanced auto graded code tests

This gives you complete control over how your tests execute. Use any language you like to script the tests and use any framework or library such as JUnit to help.

## Multiple choice questions

Single and multiple response MCQs. Supports negative points for incorrect answers.

## Fill in the blank

An autograded assessment type that requires students to complete words or phrases in a sentence by entering text or selecting from a dropdown list.

## Free text answer

A manually graded assessment type that requires a student to enter a free text answer to a question. Grading is done from the teacher dashboard.

## Manual grading

Allows a teacher to review a project assignment and assign a manual grade. See below for additional related features.

## Grading rubric

A user definable two dimensional matrix with grading criteria on the side and scores across the top. Weighting can be applied to both axes. This can then be displayed during the manual grading process as a reference or can be clicked in to automatically calculate the grade.

## Grading moderation

Secondary faculty members can review the grading and assign a parallel grade. These can then be reviewed to ensure grading consistency.

## Grade -> LMS gradebook transfer

If a class is LMS enabled, grades (auto graded and manual) can be transferred back to the LMS gradebook for the matching unit.

## Releasing grades

By default, grades are not shown to students or passed to an LMS platform (if used). Releasing grades ensures that all grades are shown when the teacher decides.

## End of coding assignment auto grading

Run an auto grading script when the student has completed the assignment unit. Can also be triggered manually by the teacher to process an entire class.



## Class management

Create classes of students and assign content and projects.

## LMS integration

Although Codio is a superb LMS for coding in its own right, we appreciate the need to integrate with LMS platforms. Our LTI integration gives your LMS full control and covers account creation, single sign on, LMS unit mapping and grade passthrough.

## Adding students

You can add students to your class using either an invitation URL or using your LMS. No need to deal with complex exports and imports.

## Adding content

You can assign any server, project or piece of tutorial content to your class with just a couple of clicks. Each student will instantly have their own cloned copy to work with exactly as you configured it.

## Teaching assistants

You can add any number of teaching assistants to a class so they can help reviewing student code, grading and other tasks.

## Class dashboard

Our dashboard allows you to monitor all aspects of student progress. You can see time spent, unit scores and scores for individual assessments.

## CSV export

You are able to export all class data to a CSV file.

## Instant access to student code

This is one of Codio's most beloved features. With a single click, you can instantly access any student's code. You have full access to the project, so you can review, compile, run and grade the students work. It spells an end to emailing or painful aggregation of student code.

## **Plagiarism detection**

A plagiarism detection capability that was specially developed at Princeton University to ensure the best possible results for student programming assignments.

Detection is launched across all students with a single click. No need to painstakingly aggregate code upload it for processing.

## **Pin unit**

Allows you to pin a unit of teaching content to the top of the student dashboard for easy location.

## **Controlled assessments**

Specify a start and end date and time for a teaching unit. The unit will not be accessible outside these times. Auto grading scripts can be automatically triggered upon completion.

## **Grading rubric mapping**

Map a grading rubric to a teaching unit to support manual grading.

## **Gigabox assignment**

Specify that a Gigabox (larger memory allocation) should be used for a given unit.

## **Release grades**

Allows a teacher to release grades to the students and to an LMS platform, if one is being used.

# Curriculum Authoring

## ◆ Codio Guides

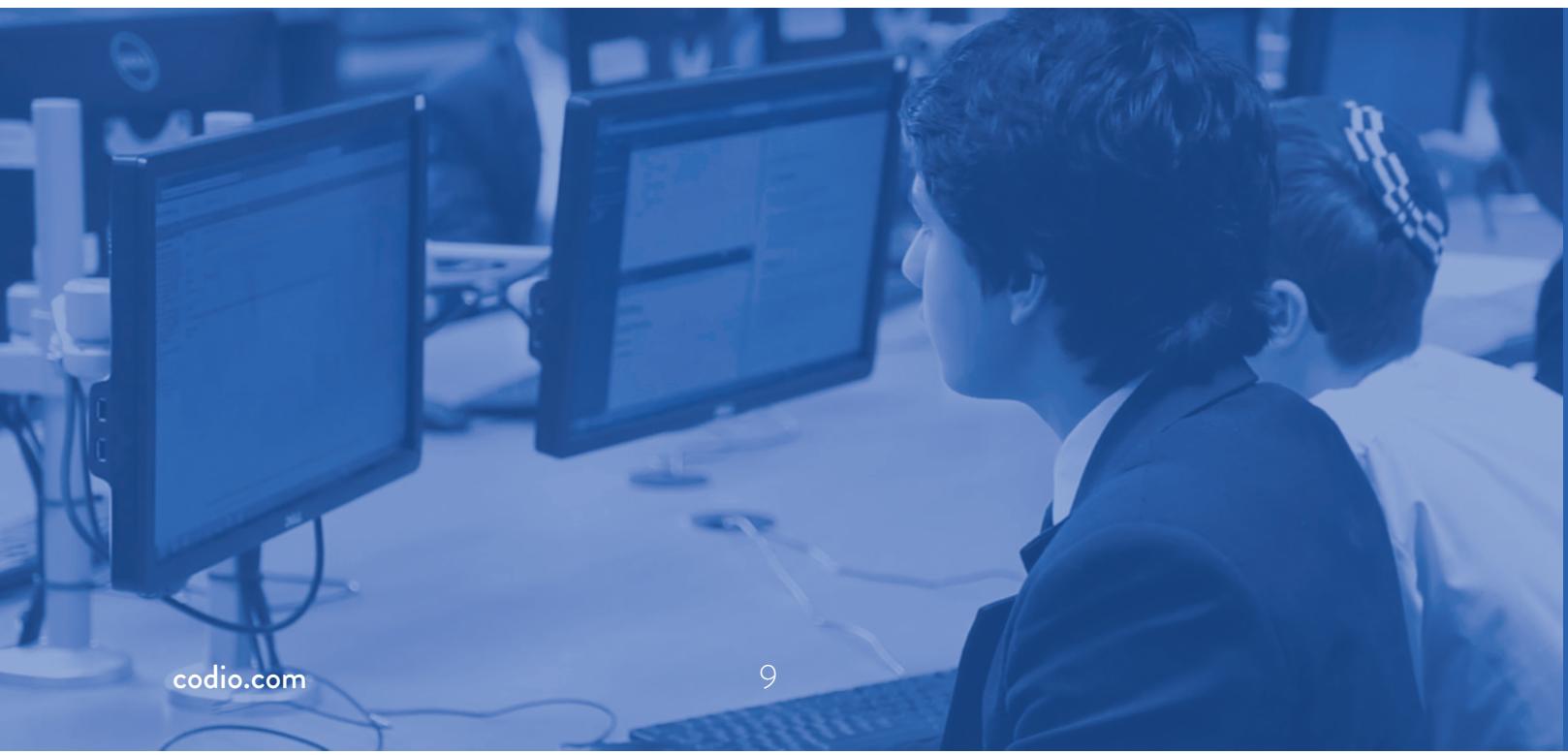
A content and curriculum authoring tool that controls the IDE as a student navigates through the content.

## ◆ Assessment creation

All assessment types are authored in Guides. See above section for a complete list of assessment types.

## ◆ Curriculum course management

A course is comprised of teaching assignment units that are organized into a simple hierarchy. Course materials can be assigned to a class.



# Templating

## Stacks

A template library that allows a snapshot to be taken of an underlying server configuration for any project. The snapshot can then be applied to new projects or replace the stack on an existing one.

## Starter Packs

A combination of a Stack (see above) and code. Most often used as a way of providing students instant access to a working project as a suitable starting point for the remainder of a coding exercise.



29 Wood Street, Stratford-upon-Avon,  
CV376JG, UK

Twitter: @CodioHQ

[www.codio.com](http://www.codio.com)

Copyright © 2015 Codio  
All rights reserved