

Navigating through the code ft. Open edX Repositories

Azan & Dawoud
Open edX Conference 2022
Lisbon, Portugal

Agenda

- Speakers Introduction
- Talk Details
- Breaking Down Code Structures
- Deconstructing Integrations and Communications
- Using Tools for quick code navigation
- Summary
- Q/A

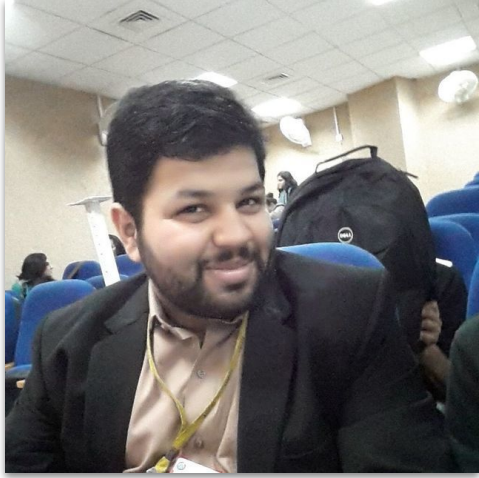
Speakers Introduction - Azan Bin Zahid



- ~1.5 Years with edX/2U as part of Arbisoft team
- Worked with various parts of open edX and edx-platform, including but not limited to:
 - Feature enhancements across various repositories
 - Consumer Driven Contract Testing Pilot
 - Internal Support Tools
 - Catalog Integration
 - XSS Fixes

- <https://www.linkedin.com/in/azanbinzahid>
- <https://github.com/azanbinzahid>
- abinzahid@2u.com
- azanbinzahid@gmail.com

Speakers Introduction – Syed Dawoud Sheraz



- ~3 Years with edX/2U as part of Arbisoft team
- Worked with various parts of open edX and edx-platform, including but not limited to:
 - MathJax a11y updates
 - Feature enhancements across various repositories
 - Video Encode Manager development
 - Consumer Driven Contract Testing Pilot
 - Internal Support Tools
 - Catalog Integration

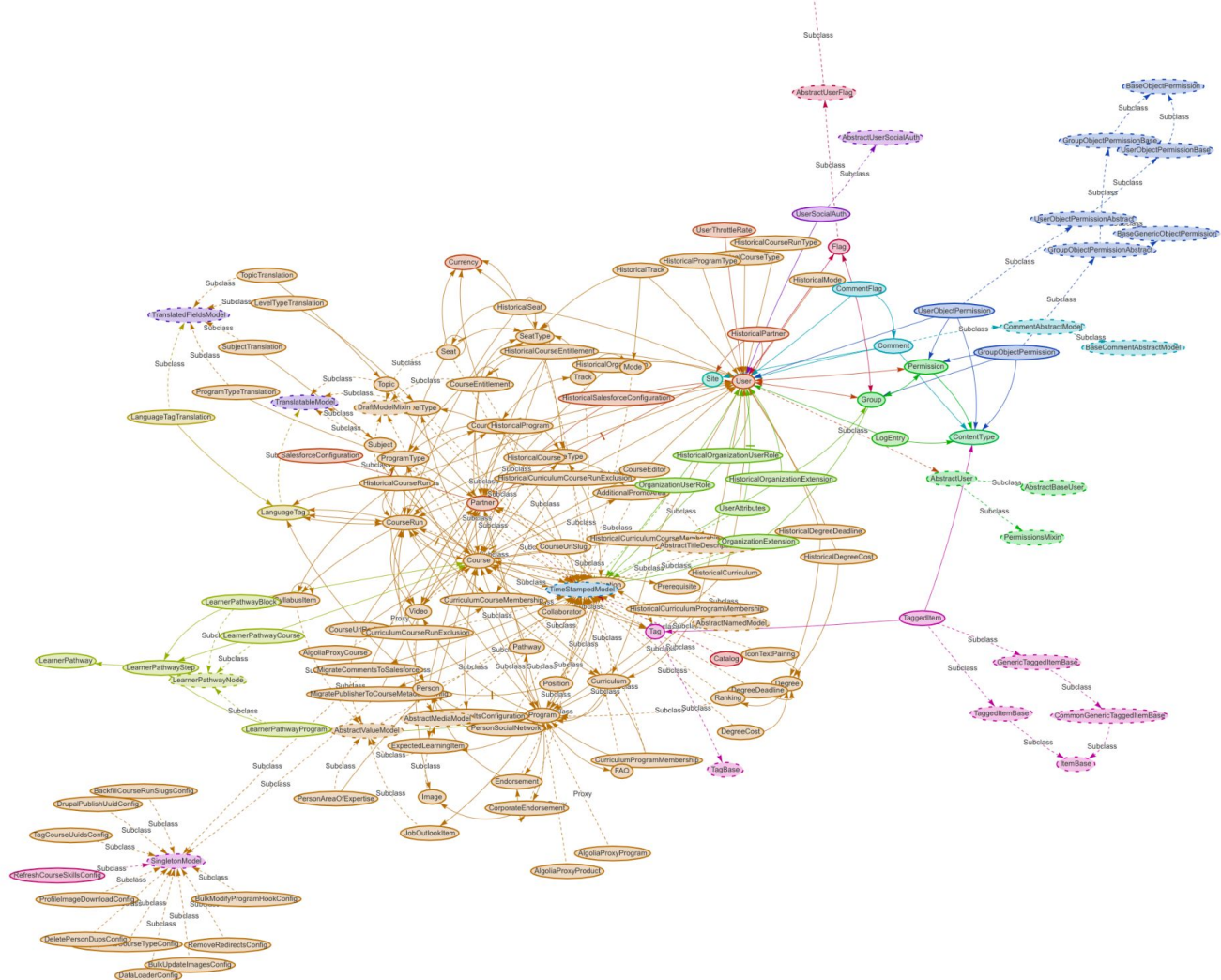
- [LinkedIn](#)
- [Github](#)
- dsheraz@2u.com
- dawoud.sheraz@gmail.com

Problem Statement

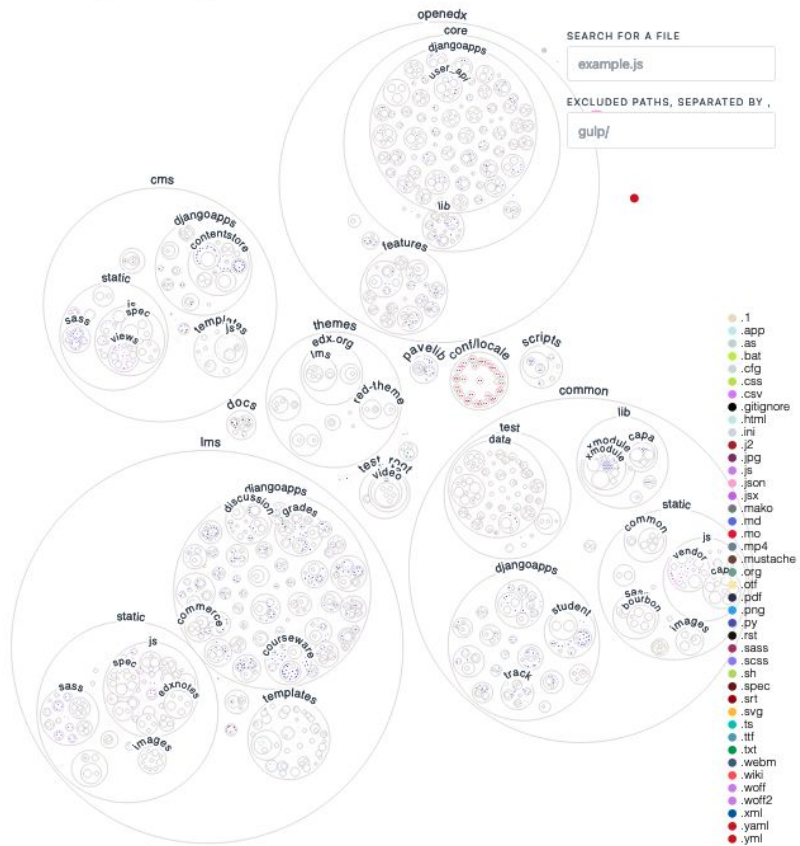
- Open edX Github → 188 open source repositories
- Various type of repositories
 - Edx-platform
 - Plugins
 - Independently Deployed Application(IDA)
 - Micro Frontend(MFE)
 - Utility(i18n)
- Edx-platform, one if not the largest open source repository
 - ~175k lines of code

Problem Statement

- Services communicate with one another in Consumer-Provider spectrum
- No single view to identify which service(IDA, MFE, or platform) relies on what service(s)
- Once a provider service is identified, locating the view/endpoint consumed by consumer



openedx/edx-platform



Example - LMS Initiated Payment flow

- Scrolling the courses on edx.org
- Learner selects and enrolls in a course
- Selects the Verified Track
- Redirected to Basket Page
- Redirected to LMS after payment

Example - Adding a course via Publisher

- List all courses or filter based on types
- Create a new course into Discovery service
- Fill detailed page, update, and review course
- Review sends course details into Studio and Ecommerce
- Ecommerce then sends mode and SKU information to LMS

Breaking Down Code Structures

Independently Deployed Applications(IDA)

- All the related code contained within the repository([Cookiecutter](#))
- Follows Django best [coding practices](#)
- Deployed separately on their own, as the name implies
- Using OAuthClient to connect with other IDAs and/or platform
- Prominent IDAs
 - [License Manager](#)
 - [Notes API](#)
 - [Ecommerce](#)
 - [Registrar](#)
 - [Credentials](#)

IDA Cookiecutter Code Structure

- Pre-cookiecutter IDA repos do not necessarily follow the below format
- IDA or Repo name/
 - apps/
 - api/ ← Important w.r.t exposed Restful endpoints
 - V0
 - urls.py
 - views.py
 - V1
 - urls.py
 - views.py
 - ...
 - urls.py
 - ..continued

IDA Cookiecutter Code Structure

- IDA or Repo name/
 - apps/
 - core ← Core application logic
 - models.py
 - serializers.py
 - admin.py
 - api_client.py/<service>_interface.py ← **Holds OAuthClient for communicating with other services, can be multiple service specific files**
 - ...
 - ... Non-core helper applications

matthugs feat: manage_programs now creates organizations if need be (#449) ...

✓ 6869489 on 23 Nov 2021 History

| | | |
|-----------------------|---|---------------|
| .. | | |
| management | feat: manage_programs now creates organizations if need be (#449) | 5 months ago |
| migrations | BOM-1983 : Post Python 3.8 changes (#373) | 12 months ago |
| tests | style: fix string formatting due to pylint errors (#444) | 6 months ago |
| __init__.py | Introduce boilerplate & basic project structure | 3 years ago |
| admin.py | feat: add group users to program and group model admin (#427) | 9 months ago |
| api_client.py | style: fix string formatting due to pylint errors (#444) | 6 months ago |
| apps.py | feat: Unpin edx-lint and pylint (#410) | 12 months ago |
| auth_checks.py | Consolidate and optimize AuthZ logic | 2 years ago |
| constants.py | style: fix string formatting due to pylint errors (#444) | 6 months ago |
| context_processors.py | Introduce boilerplate & basic project structure | 3 years ago |
| csv_utils.py | style: fix string formatting due to pylint errors (#444) | 6 months ago |
| discovery_cache.py | MST-380 Sync with Discovery on Orgs and Programs (#355) | 2 years ago |
| fiestore.py | style: fix string formatting due to pylint errors (#444) | 6 months ago |
| jobs.py | feat: Unpin edx-lint and pylint (#410) | 12 months ago |
| models.py | style: fix string formatting due to pylint errors (#444) | 6 months ago |
| permissions.py | Consolidate and optimize AuthZ logic | 2 years ago |
| rest_utils.py | Clean up constraints file and remove pin for edx_rest_api_client (#417) | 11 months ago |
| signals.py | style: fix string formatting due to pylint errors (#444) | 6 months ago |
| tasks.py | style: fix string formatting due to pylint errors (#444) | 6 months ago |
| views.py | Merge common+core apps; use relative imports (#246) | 2 years ago |

edx-requirements-bot chore: Updating Python Requirements (#469)

✗ e53488f 12 days ago History

| | | |
|----------------|---|---------------|
| .. | | |
| internal | BOM-1983 : Post Python 3.8 changes (#373) | 12 months ago |
| tests | style: fix string formatting due to pylint errors (#444) | 6 months ago |
| v1 | chore: Updating Python Requirements (#469) | 12 days ago |
| v2 | BOM-1285 : Updgrade edx-lint in registrar (#268) | 2 years ago |
| v3 | added v3 endpoint for program list pagination (#356) | 2 years ago |
| __init__.py | Introduce boilerplate & basic project structure | 3 years ago |
| apps.py | BOM-1285 : Updgrade edx-lint in registrar (#268) | 2 years ago |
| constants.py | Remove deprecated LEGACY_PERMISSION_QUERY_PARAMS | 2 years ago |
| exceptions.py | BOM-1983 : Post Python 3.8 changes (#373) | 12 months ago |
| mixins.py | MST-316 Add the context associated with these events for tracking | 2 years ago |
| segment.py | style: fix string formatting due to pylint errors (#444) | 6 months ago |
| serializers.py | Consolidate and optimize AuthZ logic | 2 years ago |
| urls.py | added v3 endpoint for program list pagination (#356) | 2 years ago |
| utils.py | BOM-1285 : Updgrade edx-lint in registrar (#268) | 2 years ago |

Django Plugins

- Reusable Django packages
- Mostly installed and exposed within edx-platform
- Enhancing the features provided by platform
- Prominent plugins
 - [edx-proctoring](#)
 - [edx-val](#)
 - [edx-ace](#)
 - [Completion](#)
 - [edx-when](#)

Django Plugins - General Code Structure



















- Code files distributed as per Django coding standards
- Plugin app Directory/
 - models.py
 - admin.py
 - views.py ← Restful Endpoints
 - api.py ← Internal API utility methods utilized by the installer application
 - urls.py ← Urls of views.py restful endpoints

master edx-val / edxval /

Go to file

Add file

...

| | | |
|---|---|--|
|  arbabkhalil chore: add requirements for reserved keywords linter | a5912b3 on 2 Mar |  History |
| .. | | |
|  config | fix: upgrade usage of edx_toggles.toggles | 15 months ago |
|  migrations | refactor: Remove squashed migrations. | 12 months ago |
|  pacts | chore: cleanup django32 | 2 months ago |
|  settings | chore: add requirements for reserved keywords linter | 2 months ago |
|  tests | chore: cleanup django32 | 2 months ago |
|  <code>__init__.py</code> | chore: cleanup django32 | 2 months ago |
|  <code>admin.py</code> | Python code cleanup by the cleanup-python-code Jenkins job. | 15 months ago |
|  <code>api.py</code> | chore: cleanup django32 | 2 months ago |
|  <code>exceptions.py</code> | Fix pylint issues. | 2 years ago |
|  <code>models.py</code> | refactor: quality fixes | 8 months ago |
|  <code>serializers.py</code> | refactor: quality fixes | 8 months ago |
|  <code>transcript_utils.py</code> | fix: allow conversion of Latin-1 encoded transcripts. | 13 months ago |
|  <code>urls.py</code> | chore: cleanup django32 | 2 months ago |
|  <code>utils.py</code> | Python code cleanup by the cleanup-python-code Jenkins job. | 15 months ago |
|  <code>views.py</code> | Python code cleanup by the cleanup-python-code Jenkins job. | 15 months ago |
|  <code>wsgi.py</code> | refactor: add pact provider verification basic structure | 12 months ago |



Code Architecture - Platform

- Core platform logic mainly divided into:
 - lms
 - cms (aka Studio)
- Monolith created initially to contain everything:
 - a huge number of django applications
 - imports many plugins / xblock
 - but what about scalability?

Code Architecture – Platform

- Unit tests
 - Dedicated **/tests** folder for all related unit tests for each level
 - Test utils imported from plugins
 - Frontends has their own suite of tests
- Frontends
 - Jinja2 templates
 - Mako templates
 - Reacts apps
 - Static html pages

Code Architecture – MFEs

- Mostly ReactJS based micro-frontends bootstrapped with [openedx/frontend-platform](#) using both:
 - Component Based Architecture (CBA)
 - Model View Control (MVC) Architecture
- Theme and UI kits are imported from [openedx/paragon](#)
- Communicates with IDAs and Platform via REST APIs
- Deployed Independently
- Prominent MFEs
 - [frontend-app-publisher](#)
 - [frontend-app-learning](#)

MFEs Structure

- src/
 - components/
 - data/ ← api calls communicating with IDA/LMS
 - sub-component/index.js
 - containers/ ← UI
 - Header
 - Navigation
 - data/ ← redux store (where applicable)
 - ...

Deconstructing the integrations and communications

Communication and Integration Pathways

- Deconstructing the communication linkage
- Following the code structure described in the previous section
- Items to be discussed
 - IDA - (IDA / MFE)
 - Platform - Plugin

IDA - IDA: Consumer - Provider

- Consumer: Consuming the data through the endpoints exposed by Provider
- Provider: Provides the data to be consumed by the consumer via API endpoints
- An IDA can be both a consumer and a provider
- Example
 - LMS(Provider) - Discovery(Consumer)
 - [Course Loading in Discovery using LMS Course API](#)
 - [API Access Checks](#)
 - Registrar(Consumer) - Discovery(Provider)
 - [Program Fetch](#)

IDA - IDA: Consumer OAuth Configuration

- OAuthApiClient from edx-rest-api-client
- Commonly used file or directory patterns across IDAs
 - api_client.py
 - api_client/resource.py
 - {resource}_interface.py
- Houses the following
 - Base url of the resource
 - URL(s) of the resource' consumed endpoints
- The above pattern is religiously being followed post cookiecutter

IDA - MFE: Locating backend from API call

- auth from frontend-platform
- Commonly used file or directory patterns across MFEs
 - data/api.js
- Houses the following
 - Base url of the resource
 - URL(s) of the resource' consumed endpoints

IDA - IDA: Locating Provider View from URL

- Consumer using a complete url in the format “base/api/v{0,1,...n}/{resource}”
- Searching for post-fix URL in provider would not yield result, unless specified in the view docstring
- Re-calling IDA API app structure: “ida/apps/api/v{}/urls.py”
- Resource name and corresponding view will be mentioned in said versioned api url.py

Deconstructed

- Registrar: Discovery base/api/v1/programs/uuid
 - Discovery Viewset Identification: course discovery/apps/api/v1/views/programs.py
- Platform: notes api base/api/v1/annotations
 - Notes API ViewSet: notesapi/v1/urls.py
- Discovery: lms base/api/courses/v1/courses
 - Platform API View: lms/djangoapps/course api/urls.py
- License Manager: LMS base/enterprise/api/v1/
 - Enterprise API: enterprise/api/v1/urls.py

github.com/openedx/registrar/blob/master/registrar/apps/core/api_client.py

master registrar / registrar / apps / core / api_client.py / <> Jump to -

bseverino style: fix string formatting due to pylint errors (8444) ✓ Latest commit 8d663b8 on 15 Oct 2021 History

3 contributors

92 lines (75 sloc) 2.51 KB

```
1  """
2  Utility objects for communicating with other web services
3  """
4  import logging
5  from posixpath import join as urljoin
6
7  from django.conf import Settings
8  from requests.exceptions import HTTPError
9
10 from .rest_utils import get_all_paginated_results, make_request
11
12
13 logger = logging.getLogger(__name__)
14 DISCOVERY_API_TPL = 'api/v1/{}/{}'
15
16
17 class DiscoveryServiceClient:
18     """ The Discovery service API client to make the RESTful API call """
19     @classmethod
20     def get_program(cls, uuid):
21         """
22         Fetch a JSON representation of a program from the Discovery service.
23
24         Returns None if not found or other HTTP error.
25
26         Arguments:
27             * uuid (UUID)
28
29         Returns: dict|None
30         """
31         url = urljoin(
32             settings.DISCOVERY_BASE_URL,
33             DISCOVERY_API_TPL.format('programs', uuid)
34         )
35         try:
36             return make_request('GET', url, client=None).json()
37         except HTTPError:
38             logger.exception(
39                 "Failed to load program with uuid %s from Discovery service.",
40                 uuid,
41             )
42         return None
43
```

Platform - Plugin

- The simplest integration/communication pathway
- One way communication; Platform → Plugin
- Installed plugins used in conjunction with platform flows
- Platform imports and uses the utility methods exposed in plugin/api.py
 - Achieves abstraction
 - Plugin logic contained within the plugin

Platform - Plugin: Examples

- edxval/api.py
 - Video module: Video Profile info, xml export
 - Contentstore Video Views: Creating/Updating video entries, Transcript Preferences, Transcript languages
- edx_proctoring/api.py
 - Contentstore proctoring module: Studio configuration of proctored exams
 - Courseware: Exam Attempt summary

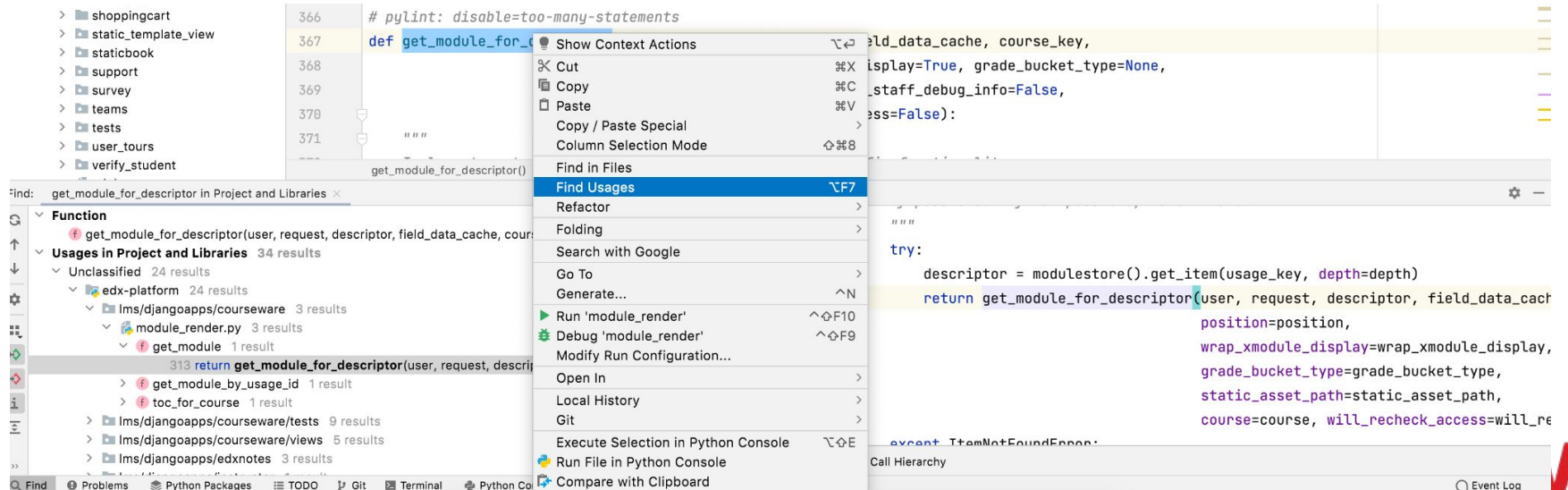
Using Tools for Quick Navigation

PyCharm

- Listing Code Usages
- Utilizing Python Interpreter configuration
- Search Everywhere

PyCharm: Listing Code Usages

- Find all the instances of a method or a class used within a repo's codebase
- Helps identify the risk and severity of the changes in a module



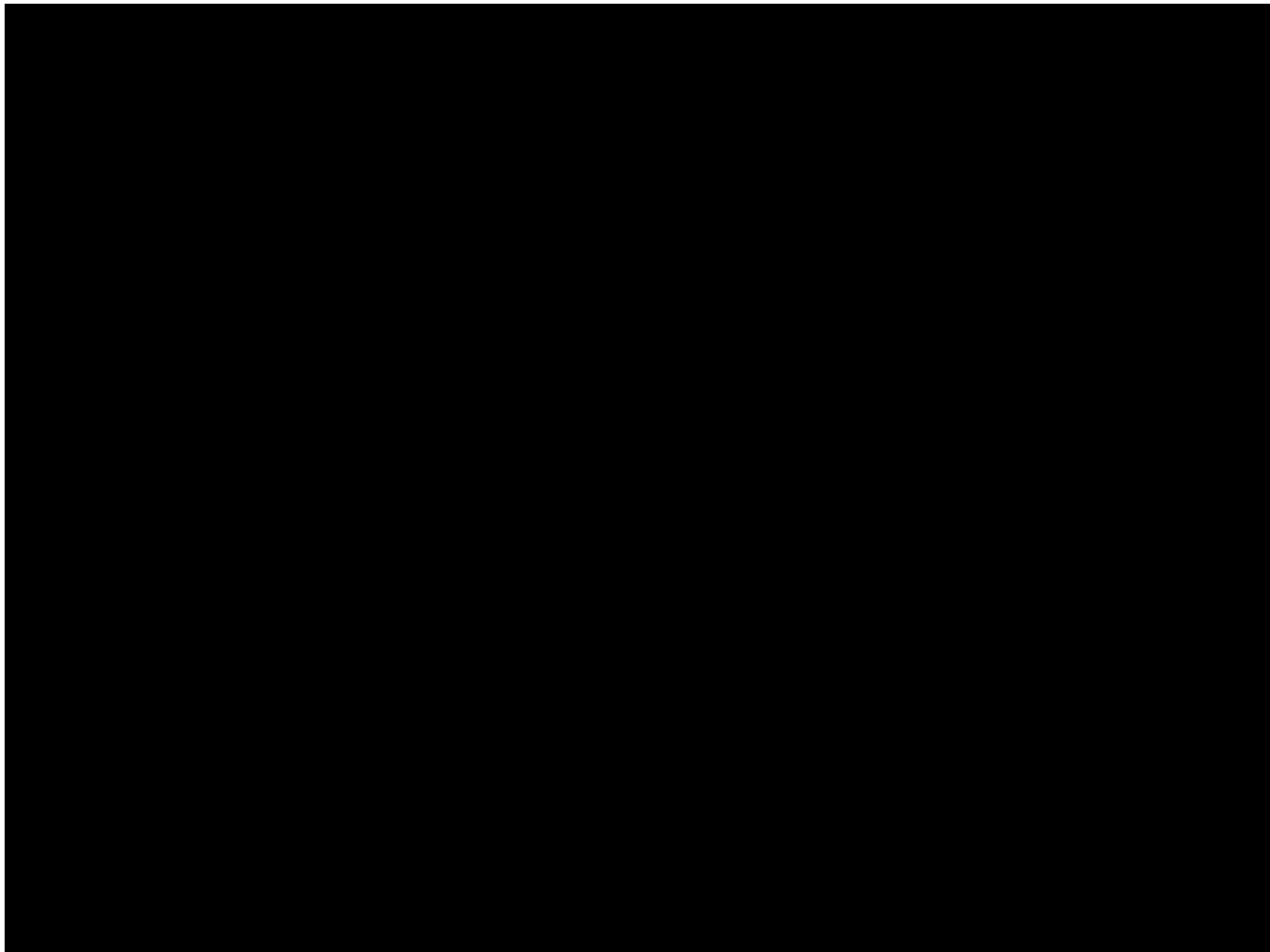
PyCharm: Interpreter configuration

- Configure an appropriate Python version interpreter with each project
- Install the repo's requirement within the configured interpreter
 - Jump into an installed package/plugin codebase with just one click



PyCharm: Search Everywhere

- One of the powerful features; ability to search for string in entire codebase in mere seconds
- Different filter criteria
 - File masks; include or exclude certain file types or file name patterns
 - Limiting search within a scope or directory
 - Case limiting the string for an even narrower search



VS Code – Code Navigation

- Listing code search similar to PyCharm
- Custom key-bindings
- Equally useful in developing frontend web applications
- In-built features:
 - Breadcrumbs
 - Go to type definitions, implementation, references
 - Peek
- Useful Extensions
 - Git lens
 - Language specific tools

Github Search

- Search-based navigation
- Precise code navigation (supported for Python)
- Using open source [tree-sitter](#) and [stack-graphs](#) library under the hood
- Search Across Organization
- Use git blame to find PR that added that code

Summary

The screenshot shows the GitHub repository page for `openedx/edx-platform`. The browser is Firefox, and the URL is `https://github.com/openedx/edx-platform`. The repository is public and has 419 watchers, 3.3k forks, and 6k stars. The main content area displays a list of files and their commit history. The files listed are:

- `.github`: build: Remove empty coursegraph module (and the pylint reference L...
- `.tx`: fix: transifex migration to new client (#30084)
- `cms`: fix: Updated course live docs url (#30293)
- `common`: Merge pull request #30297 from openedx/kiram15/ENT-5498
- `conf/locale`: chore(i18n): update translations
- `docs`: fix: remove a couple unused learning MFE serializer fields
- `lms`: Merge pull request #30303 from openedx/mikix/outline-children
- `openedx`: fix: Updated course live docs url (#30293)
- `pavelib`: build: Removing un-used django versions from tox and other places. ...
- `requirements`: chore: upgrade edx-name-affirmation to 2.3.3 for new api docs (#30...
- `scripts`: fix: some tests are skipped due to duplicate names (#29959)

The right sidebar contains the "About" section, which describes the repository as "The Open edX LMS & Studio, powering education sites around the world!". It also includes links to `openedx.org`, a README, the AGPL-3.0 License, the Code of conduct, and a list of contributors (613 total, with 602 contributors shown).

Thank you for your attention!

We are open to Q/A