
Open edX Data Analytics API Version 0
Alpha
Release

September 24, 2015

1	Read Me	1
2	Change Log	3
3	edX Data Analytics API Overview	5
3.1	edX Data Analytics API Version 0, Alpha	5
3.2	edX Data Analytics API Capabilities	5
4	Set up the Data Analytics API Server	7
4.1	Get the Repository	7
4.2	Install Server Requirements	7
4.3	Run the Server	7
4.4	Load Sample Data	8
4.5	Test the Data Analytics API	8
5	edX Data Analytics API Authentication	11
5.1	Create a User and Token	11
6	edX Data Analytics API Endpoints	13
7	Course Information API	15
7.1	Get Weekly Course Activity	15
7.2	Get Recent Course Activity	16
7.3	Get the Course Enrollment	17
7.4	Get the Course Enrollment by Mode	18
7.5	Get the Course Enrollment by Birth Year	19
7.6	Get the Course Enrollment by Education Level	20
7.7	Get the Course Enrollment by Gender	21
7.8	Get the Course Enrollment by Location	21
7.9	Get the Course Video Data	23
8	Problem Information API	25
8.1	Get the Grade Distribution for a Course	25
8.2	Get the Answer Distribution for a Problem	26
8.3	Get the View Count for a Subsection	27
9	Video Data API	29
9.1	Get the Timeline for a Video	29

Read Me

The edX Data Analytics API documentation is created using [RST](#) files and [Sphinx](#). You, the user community, can help update and revise this documentation project on [GitHub](#):

<https://github.com/edx/edx-analytics-data-api/tree/master/docs/api/source>

To suggest a revision, fork the project, make changes in your fork, and submit a pull request back to the original project: this is known as the [GitHub Flow](#).

Change Log

Date	Change
5 Aug 2015	Updated <i>Get the Course Enrollment by Mode</i> to add the <code>cumulative_count</code> response value.
21 May 2015	Updated <i>Get the Course Video Data</i> to add the <code>users_at_start</code> and <code>users_at_end</code> response values.
18 May 2015	Added <i>Get the Course Video Data</i> and <i>Video Data API</i> .

edX Data Analytics API Overview

The edX Data Analytics API provides the tools for building applications to view and analyze student activity in your course.

The edX Platform APIs use REST design principles and support the JSON data-interchange format.

3.1 edX Data Analytics API Version 0, Alpha

The edX Data Analytics API is currently at version 0 and is an Alpha release. We plan on making significant enhancements and changes to the API.

The Data Analytics API uses key-based authentication and currently has no built-in authorization mechanism. Therefore third parties cannot currently use the Data Analytics API with edx.org data.

Open edX users can use the Data Analytics API with their own instances.

EdX plans to make the Data Analytics API available to partners in the future, and invites feedback.

3.2 edX Data Analytics API Capabilities

With the edX Data Analytics API, you can:

- *Get Weekly Course Activity*
- *Get Recent Course Activity*
- *Get the Course Enrollment*
- *Get the Course Enrollment by Mode*
- *Get the Course Enrollment by Birth Year*
- *Get the Course Enrollment by Education Level*
- *Get the Course Enrollment by Gender*
- *Get the Course Enrollment by Location*
- *Get the Course Video Data*
- *Get the Grade Distribution for a Course*
- *Get the Answer Distribution for a Problem*
- *Get the View Count for a Subsection*

- *Get the Timeline for a Video*

Set up the Data Analytics API Server

This chapter describes how to set up and test the edX Data Analytics API server:

1. *Get the Repository*
2. *Install Server Requirements*
3. *Run the Server*
4. *Load Sample Data*
5. *Test the Data Analytics API*

Also see *edX Data Analytics API Authentication*.

4.1 Get the Repository

You must get the [Data Analytics API repository](#) from GitHub.

From the terminal, enter:

```
git clone https://github.com/edx/edx-analytics-data-api
```

You may choose to get the repository in a virtual environment.

4.2 Install Server Requirements

From the terminal at the top level of the server repository, enter:

```
$ make develop
```

Server requirements are then installed.

4.3 Run the Server

From the terminal at the top level of the server repository, enter:

```
$ ./manage.py runserver
```

The server starts.

4.4 Load Sample Data

From the terminal at the top level of the server repository, enter:

```
$ make loaddata
```

4.5 Test the Data Analytics API

After you load sample data and run the server, you can test the API.

1. In a browser, go to: `http://<server-name>:<port>/docs/#!/api/`
2. Enter a valid key and click **Explore**.

See *edX Data Analytics API Authentication* for information on keys.

You see an interactive list of API endpoints, which you can use to get responses with the sample data. Expand the **api** section to see the available endpoints.

api		Show/Hide	List Operations	Expand Operations	Raw
POST	/api-token-auth/				
GET	/api/v0/courses/{course_id}/activity/			Get counts of users who performed specific activities in a course	
GET	/api/v0/courses/{course_id}/recent_activity/			Get counts of users who performed specific activities at least once during the most recently computed week	
GET	/api/v0/courses/{course_id}/enrollment/			Get the number of enrolled users	
GET	/api/v0/courses/{course_id}/enrollment/mode/			Get the number of enrolled users by enrollment mode	
GET	/api/v0/courses/{course_id}/enrollment/birth_year/			Get the number of enrolled users by birth year	
GET	/api/v0/courses/{course_id}/enrollment/education/			Get the number of enrolled users by education level	
GET	/api/v0/courses/{course_id}/enrollment/gender/			Get the number of enrolled users by gender	
GET	/api/v0/courses/{course_id}/enrollment/location/			Get the number of enrolled users by location	
GET	/api/v0/problems/{module_id}/sequential_open_distribution			Get the number of views of a subsection, or sequential, in the course	
GET	/api/v0/problems/{problem_id}/answer_distribution			Get the distribution of student answers to a specific problem	
GET	/api/v0/problems/{problem_id}/grade_distribution			Get the distribution of grades for a specific problem	

3. Expand the section for an endpoint:

GET
/api/v0/courses/{course_id}/enrollment/
Get the number of enrolled users

Implementation Notes

Get the number of enrolled users.

****Example request**:**

```
GET /api/v0/courses/{course_id}/enrollment/
```

****Response Values****

Returns a collection that contains the counts of users:

- * `course_id`: The ID of the course for which data is returned.
- * `date`: The date for which the enrollment count was computed.
- * `count`: The count of enrolled users.
- * `created`: The date and time the user enrolled in the course.

****Parameters****

You can specify the start and end dates for which to count enrolled users.

You specify dates in the format: Date format: YYYY-mm-dd; for example, ``2014-12-15``.

If no start or end dates are specified, the data for the previous day is returned.

Response Class

Model | Model Schema

```
CourseEnrollmentDailySerializer {
  course_id (string),
  date (date),
  count (integer),
  created (datetime)
}
```

Response Content Type application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type
<code>course_id</code>	<input style="width: 100%;" type="text" value="(required)"/>		path	string
<code>start_date</code>	<input style="width: 100%;" type="text"/>	Date after which enrolled students are counted (inclusive)	query	
<code>end_date</code>	<input style="width: 100%;" type="text"/>	Date before enrolled students are counted (exclusive)	query	

Try it out!

4. Enter parameters as needed and click **Try it out**. The response opens:

Try it out! [Hide Response](#)

Request URL

```
http://localhost:8000/api/v0/courses/edX%2FDemo%2FDemo_Course/enrollment/
```

Response Body

```
[
  {
    "course_id": "edX/DemoX/Demo_Course",
    "date": "2014-12-10",
    "count": 1892,
    "created": "2014-12-10T193146"
  }
]
```

Response Code

```
200
```

Response Headers

```
{"Date": "Fri, 12 Dec 2014 19:40:16 GMT", "Vary": "Accept, Cookie", "Server": "WSGIServer/0.1 Python/2.7.5", "Allow": "GET, HE
```

To get the sample enrollment data, use `edX/DemoX/Demo_Course` as the `course_id`.

edX Data Analytics API Authentication

The edX Data Analytics API uses the Django REST framework `TokenAuthentication`.

5.1 Create a User and Token

You create users and tokens in the Data Analytics API server. In the server terminal, enter:

```
$ ./manage.py set_api_key <username> <token>
```

edX Data Analytics API Endpoints

The edX Platform API allows you to view information about users and their course enrollments, course information, and videos and transcripts.

The following tasks and endpoints are currently supported.

To:	Use this endpoint:
<i>Get Weekly Course Activity</i>	/api/v0/courses/{course_id}/activity/
<i>Get Recent Course Activity</i>	/api/v0/courses/{course_id}/recent_activity/
<i>Get the Course Enrollment</i>	/api/v0/courses/{course_id}/enrollment/
<i>Get the Course Enrollment by Mode</i>	/api/v0/courses/{course_id}/enrollment/mode/
<i>Get the Course Enrollment by Birth Year</i>	/api/v0/courses/{course_id}/enrollment/birth_year/
<i>Get the Course Enrollment by Education Level</i>	/api/v0/courses/{course_id}/enrollment/education/
<i>Get the Course Enrollment by Gender</i>	/api/v0/courses/{course_id}/enrollment/gender/
<i>Get the Course Enrollment by Location</i>	/api/v0/courses/{course_id}/enrollment/location/
<i>Get the Course Video Data</i>	/api/v0/courses/{course_id}/videos/
<i>Get the Grade Distribution for a Course</i>	/api/v0/problems/{problem_id}/grade_distribution
<i>Get the Answer Distribution for a Problem</i>	/api/v0/problems/{problem_id}/answer_distribution
<i>Get the View Count for a Subsection</i>	/api/v0/problems/{module_id}/sequential_open_distribution
<i>Get the Timeline for a Video</i>	/api/v0/videos/{video_id}/timeline/

Course Information API

Section Contents

- *Get Weekly Course Activity*
- *Get Recent Course Activity*
- *Get the Course Enrollment*
- *Get the Course Enrollment by Mode*
- *Get the Course Enrollment by Birth Year*
- *Get the Course Enrollment by Education Level*
- *Get the Course Enrollment by Gender*
- *Get the Course Enrollment by Location*
- *Get the Course Video Data*

7.1 Get Weekly Course Activity

class `analytics_data_api.v0.views.courses.CourseActivityWeeklyView` (**kwargs)

Get counts of users who performed specific activities in a course.

Example request

```
GET /api/v0/courses/{course_id}/activity/
```

Response Values

Returns a list of key/value pairs for student activities, as well as the interval start and end dates and the course ID.

- `any`: The number of unique users who performed any action in the course, including actions not counted in other categories in the response.
- `attempted_problem`: The number of unique users who answered any loncapa-based problem in the course.
- `played_video`: The number of unique users who started watching any video in the course.
- `posted_forum`: The number of unique users who created a new post, responded to a post, or submitted a comment on any discussion in the course.
- `interval_start`: The time and date at which data started being included in returned values.
- `interval_end`: The time and date at which data stopped being included in returned values.
- `course_id`: The ID of the course for which data is returned.

- created: The date the counts were computed.

Parameters

You can specify the start and end dates for the time period for which you want to get activity.

You specify dates in the format: YYYY-mm-ddTttttt; for example, 2014-12-15T000000.

If no start or end dates are specified, the data for the week ending on the previous day is returned.

start_date – Date after which all data is returned (inclusive).

end_date – Date before which all data is returned (exclusive).

Example Response

```
HTTP 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
Allow: GET, HEAD, OPTIONS

[
  {
    "interval_start": "2014-12-08T000000",
    "interval_end": "2014-12-15T000000",
    "course_id": "edX/DemoX/Demo_Course",
    "any": 3013,
    "attempted_problem": 206,
    "played_video": 1049,
    "created": "2014-12-10T193104"
  }
]
```

7.2 Get Recent Course Activity

class analytics_data_api.v0.views.courses.**CourseActivityMostRecentWeekView** (**kwargs)

Get counts of users who performed specific activities at least once during the most recently computed week.

Example request

GET /api/v0/courses/{course_id}/recent_activity/

Response Values

Returns a list of key/value pairs for student activities, as well as the interval start and end dates and the course ID.

- activity_type: The type of activity counted. Possible values are:
 - any: The number of unique users who performed any action in the course, including actions not counted in other categories in the response.
 - attempted_problem: The number of unique users who answered any loncapa-based problem in the course.
 - played_video: The number of unique users who started watching any video in the course.
 - posted_forum: The number of unique users who created a new post, responded to a post, or submitted a comment on any discussion in the course.
- count: The number of unique users who performed the specified action.
- interval_start: The time and date at which data started being included in returned values.

- interval_end: The time and date at which data stopped being included in returned values.
- course_id: The ID of the course for which data is returned.

Parameters

You can specify the activity type for which you want to get the count.

activity_type – The type of activity: any (default), attempted_problem, played_video, posted_forum.

Example Response

```
HTTP 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
Allow: GET, HEAD, OPTIONS

{
  "interval_start": "2014-12-08T00:00:00Z",
  "interval_end": "2014-12-15T00:00:00Z",
  "activity_type": "any",
  "count": 3013,
  "course_id": "edX/DemoX/Demo_Course"
}
```

7.3 Get the Course Enrollment

class analytics_data_api.v0.views.courses.CourseEnrollmentView (**kwargs)

Get the number of enrolled users.

Example request

GET /api/v0/courses/{course_id}/enrollment/

Response Values

Returns the count of enrolled users:

- course_id: The ID of the course for which data is returned.
- date: The date for which the enrollment count was computed.
- count: The count of enrolled users.
- created: The date the count was computed.

Parameters

You can specify the start and end dates for which to count enrolled users.

You specify dates in the format: YYYY-mm-dd; for example, 2014-12-15.

If no start or end dates are specified, the data for the previous day is returned.

start_date – Date after which enrolled students are counted (inclusive).

end_date – Date before which enrolled students are counted (exclusive).

Example Response

```
HTTP 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
Allow: GET, HEAD, OPTIONS
```

```
[
  {
    "course_id": "edX/DemoX/Demo_Course",
    "date": "2014-12-10",
    "count": 1892,
    "created": "2014-12-10T193146"
  }
]
```

7.4 Get the Course Enrollment by Mode

class analytics_data_api.v0.views.courses.**CourseEnrollmentModeView** (**kwargs)
Get the number of enrolled users by enrollment mode.

Example request

```
GET /api/v0/courses/{course_id}/enrollment/mode/
```

Response Values

Returns the counts of users by mode:

- course_id: The ID of the course for which data is returned.
- date: The date for which the enrollment count was computed.
- count: The count of currently enrolled users.
- cumulative_count: The cumulative total of all users ever enrolled.
- created: The date the counts were computed.
- honor: The number of users currently enrolled in honor code mode.
- professional: The number of users currently enrolled in professional mode.
- verified: The number of users currently enrolled in verified mode.

Parameters

You can specify the start and end dates for which to count enrolled users.

You specify dates in the format: YYYY-mm-dd; for example, 2014-12-15.

If no start or end dates are specified, the data for the previous day is returned.

start_date – Date after which enrolled students are counted (inclusive).

end_date – Date before which enrolled students are counted (exclusive).

Example Response

```
HTTP 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
Allow: GET, HEAD, OPTIONS

[
  {
    "course_id": "edX/DemoX/Demo_Course",
    "date": "2014-12-10",
    "count": 1890,
    "cumulative_count": 1931,
```

```

    "created": "2014-12-10T193146",
    "honor": 945,
    "professional": 189,
    "verified": 756
  }
]

```

7.5 Get the Course Enrollment by Birth Year

class `analytics_data_api.v0.views.courses.CourseEnrollmentByBirthYearView` (**kwargs)
Get the number of enrolled users by birth year.

Example request

```
GET /api/v0/courses/{course_id}/enrollment/birth_year/
```

Response Values

Returns an array with a collection for each year in which a user was born. Each collection contains:

- `course_id`: The ID of the course for which data is returned.
- `date`: The date for which the enrollment count was computed.
- `birth_year`: The birth year for which the enrollment count applies.
- `count`: The number of users who were born in the specified year.
- `created`: The date the count was computed.

Parameters

You can specify the start and end dates for which to count enrolled users.

You specify dates in the format: YYYY-mm-dd; for example, 2014-12-15.

If no start or end dates are specified, the data for the previous day is returned.

`start_date` – Date after which enrolled students are counted (inclusive).

`end_date` – Date before which enrolled students are counted (exclusive).

Example Response

```

HTTP 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
Allow: GET, HEAD, OPTIONS

[
  {
    "course_id": "edX/DemoX/Demo_Course",
    "date": "2014-12-10",
    "birth_year": 1960,
    "count": 11,
    "created": "2014-12-10T193146"
  },
  {
    "course_id": "edX/DemoX/Demo_Course",
    "date": "2014-12-10",
    "birth_year": 1961,
    "count": 58,

```

```

    "created": "2014-12-10T193146"
  }
]

```

7.6 Get the Course Enrollment by Education Level

class `analytics_data_api.v0.views.courses.CourseEnrollmentByEducationView(**kwargs)`
 Get the number of enrolled users by education level.

Example request

```
GET /api/v0/courses/{course_id}/enrollment/education/
```

Response Values

Returns a collection for each level of education reported by a user. Each collection contains:

- `course_id`: The ID of the course for which data is returned.
- `date`: The date for which the enrollment count was computed.
- `education_level`: The education level for which the enrollment count applies.
- `count`: The number of users who reported the specified education level.
- `created`: The date the count was computed.

Parameters

You can specify the start and end dates for which to count enrolled users.

You specify dates in the format: YYYY-mm-dd; for example, 2014-12-15.

If no start or end dates are specified, the data for the previous day is returned.

`start_date` – Date after which enrolled students are counted (inclusive).

`end_date` – Date before which enrolled students are counted (exclusive).

Example Response

```

HTTP 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
Allow: GET, HEAD, OPTIONS

[
  {
    "course_id": "edX/DemoX/Demo_Course",
    "date": "2014-12-10",
    "education_level": "bachelors",
    "count": 634,
    "created": "2014-12-10T193146"
  },
  {
    "course_id": "edX/DemoX/Demo_Course",
    "date": "2014-12-10",
    "education_level": "doctorate",
    "count": 88,
    "created": "2014-12-10T193146"
  }
]

```


7.7 Get the Course Enrollment by Gender

class `analytics_data_api.v0.views.courses.CourseEnrollmentByGenderView` (**kwargs)
Get the number of enrolled users by gender.

Example request

```
GET /api/v0/courses/{course_id}/enrollment/gender/
```

Response Values

Returns the count of each gender specified by users:

- `course_id`: The ID of the course for which data is returned.
- `date`: The date for which the enrollment count was computed.
- `female`: The count of self-identified female users.
- `male`: The count of self-identified male users.
- `other`: The count of self-identified other users.
- `unknown`: The count of users who did not specify a gender.
- `created`: The date the counts were computed.

Parameters

You can specify the start and end dates for which to count enrolled users.

You specify dates in the format: YYYY-mm-dd; for example, 2014-12-15.

If no start or end dates are specified, the data for the previous day is returned.

`start_date` – Date after which enrolled students are counted (inclusive).

`end_date` – Date before which enrolled students are counted (exclusive).

Example Response

```
HTTP 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
Allow: GET, HEAD, OPTIONS

[
  {
    "course_id": "edX/DemoX/Demo_Course",
    "date": "2014-12-10",
    "female": 732,
    "male": 1155,
    "other": 435,
    "unknown": 0,
    "created": "2014-12-10T193146"
  }
]
```

7.8 Get the Course Enrollment by Location

class `analytics_data_api.v0.views.courses.CourseEnrollmentByLocationView` (**kwargs)
Get the number of enrolled users by location.

Location is calculated based on the user's IP address. Users whose location cannot be determined are counted as having a country.name of UNKNOWN.

Countries are denoted by their ISO 3166 country code.

Example request

```
GET /api/v0/courses/{course_id}/enrollment/location/
```

Response Values

Returns counts of genders specified by users:

- course_id: The ID of the course for which data is returned.
- date: The date for which the enrollment count was computed.
- country: Contains the following fields:
 - alpha2: The two-letter country code.
 - alpha3: The three-letter country code.
 - name: The country name.
- count: The count of users from the country.
- created: The date the count was computed.

Parameters

You can specify the start and end dates for which to count enrolled users.

You specify dates in the format: YYYY-mm-dd; for example, 2014-12-15.

If no start or end dates are specified, the data for the previous day is returned.

start_date – Date after which enrolled students are counted (inclusive).

end_date – Date before which enrolled students are counted (exclusive).

See [ISO 3166 country codes](#) for more information.

Example Response

```
HTTP 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
Allow: GET, HEAD, OPTIONS

[
  {
    "date": "2014-12-10",
    "course_id": "edX/DemoX/Demo_Course",
    "country": {
      "alpha2": "CA",
      "alpha3": "CAN",
      "name": "Canada"
    },
    "count": 264,
    "created": "2014-12-10T193146"
  },
  {
    "date": "2014-12-10",
    "course_id": "edX/DemoX/Demo_Course",
    "country": {
```

```

    "alpha2": "CN",
    "alpha3": "CHN",
    "name": "China"
  },
  "count": 416,
  "created": "2014-12-10T193146"
}
]

```

7.9 Get the Course Video Data

class `analytics_data_api.v0.views.courses.VideosListView` (**kwargs)
Get data for the videos in a course.

Example request

```
GET /api/v0/courses/{course_id}/videos/
```

Response Values

Returns a collection of video views and metadata for each video. For each video, the collection the following data.

- `video_id`: The ID of the video.
- `encoded_module_id`: The encoded module ID.
- `duration`: The length of the video in seconds.
- `segment_length`: The length of each segment of the video in seconds.
- `users_at_start`: The number of viewers at the start of the video.
- `users_at_end`: The number of viewers at the end of the video.
- `created`: The date the video data was updated.

Example Response

```

HTTP 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
Allow: GET, HEAD, OPTIONS

[
  {
    "pipeline_video_id": "UniversityX/UX.3.01x/1T2015|i4x-UniversityX-
      UX_3_01x-video-02874e0ae0c74ae7b16faa5d6fdc8085",
    "encoded_module_id": "i4x-UX-UT_3_01x-video-
      02874e0ae0c74ae7b16faa5d6fdc8085",
    "duration": 142,
    "segment_length": 5,
    "users_at_start": 2,
    "users_at_end": 0,
    "created": "2015-04-15T214158"
  },
  {
    "pipeline_video_id": "UniversityX/UX.3.01x/1T2015|i4x-UniversityX-
      UX_3_01x-video-03454e0ae0c72ae7b16fab3d6fdc2143",
    "encoded_module_id": "i4x-UX-UT_3_01x-video-

```

```
    "03454e0ae0c72ae7b16fab3d6fdc2143",  
    "duration": 66,  
    "segment_length": 5,  
    "users_at_start": 1044,  
    "users_at_end": 0,  
    "created": "2015-04-15T21:41:58"  
  },  
]
```

Problem Information API

Section Contents

- *Get the Grade Distribution for a Course*
- *Get the Answer Distribution for a Problem*
- *Get the View Count for a Subsection*

8.1 Get the Grade Distribution for a Course

class `analytics_data_api.v0.views.problems.GradeDistributionView` (**kwargs)
Get the distribution of grades for a specific problem.

Example request

```
GET /api/v0/problems/{problem_id}/grade_distribution
```

Response Values

Returns a collection for each unique grade given to a specified problem. Each collection contains:

- `course_id`: The ID of the course for which data is returned.
- `module_id`: The ID of the problem.
- `grade`: The grade being counted in this collection.
- `count`: The number of times the grade in this collection was given.
- `max_grade`: The highest possible grade for this problem.
- `created`: The date the count was computed.

Example Response

```
HTTP 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
Allow: GET, HEAD, OPTIONS

[
  {
    "module_id": "i4x://edX/DemoX/Demo_Course/problem/97fd93e33a18495488578e9e74fa4cae",
    "course_id": "edX/DemoX/Demo_Course",
    "grade": 1,
```

```

    "max_grade": 2,
    "count": 5,
    "created": "2014-09-12T114957"
  },
  {
    "module_id": "i4x://edX/DemoX/Demo_Course/problem/97fd93e33a18495488578e9e74fa4cae",
    "course_id": "edX/DemoX/Demo_Course",
    "grade": 2,
    "max_grade": 2,
    "count": 256,
    "created": "2014-09-12T114957"
  }
]

```

8.2 Get the Answer Distribution for a Problem

class `analytics_data_api.v0.views.problems.ProblemResponseAnswerDistributionView` (**kwargs)
Get the distribution of student answers to a specific problem.

Example request

GET /api/v0/problems/{problem_id}/answer_distribution

Response Values

Returns a collection for each unique answer given to specified problem. Each collection contains:

- `course_id`: The ID of the course for which data is returned.
- `module_id`: The ID of the problem.
- `part_id`: The ID for the part of the problem. For multi-part problems, a collection is returned for each part.
- `correct`: Whether the answer was correct (`true`) or not (`false`).
- `count`: The number of times the answer in this collection was given.
- `value_id`: The ID of the answer in this collection.
- `answer_value`: An answer for this problem.
- `problem_display_name`: The display name for the specified problem.
- `question_text`: The question for the specified problem.
- `variant`: For randomized problems, the random seed used. If problem is not randomized, value is null.
- `created`: The date the count was computed.

Example Response

```

HTTP 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
Allow: GET, HEAD, OPTIONS

[
  {
    "course_id": "edX/DemoX/Demo_Course",

```

```

    "module_id": "i4x://edX/DemoX/Demo_Course/problem/
      268b43628e6d45f79c52453a590f9829",
    "part_id": "i4x-edX-DemoX-Demo_Course-problem-
      268b43628e6d45f79c52453a590f9829_2_1",
    "correct": false,
    "count": 9,
    "value_id": "choice_0",
    "answer_value_text": "Russia",
    "answer_value_numeric": null,
    "problem_display_name": "Multiple Choice Problem",
    "question_text": "Which of the following countries has the largest
      population?",
    "variant": null,
    "created": "2014-12-05T225026"
  },
  {
    "course_id": "edX/DemoX/Demo_Course",
    "module_id": "i4x://edX/DemoX/Demo_Course/problem/
      268b43628e6d45f79c52453a590f9829",
    "part_id": "i4x-edX-DemoX-Demo_Course-problem-
      268b43628e6d45f79c52453a590f9829_2_1",
    "correct": true,
    "count": 15,
    "value_id": "choice_1",
    "answer_value_text": "Indonesia",
    "answer_value_numeric": null,
    "problem_display_name": "Multiple Choice Problem",
    "question_text": "Which of the following countries has the largest
      population?",
    "variant": null,
    "created": "2014-12-05T225026"
  }
]

```

8.3 Get the View Count for a Subsection

class `analytics_data_api.v0.views.problems.SequentialOpenDistributionView` (**kwargs)
 Get the number of views of a subsection, or sequential, in the course.

Example request

```
GET /api/v0/problems/{module_id}/sequential_open_distribution
```

Response Values

Returns a collection that contains the number of views of the specified problem. The collection contains:

- `course_id`: The ID of the course for which data is returned.
- `module_id`: The ID of the subsection, or sequential.
- `count`: The number of times the subsection was viewed.
- `created`: The date the count computed.

Example Response

```
HTTP 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
Allow: GET, HEAD, OPTIONS

[
  {
    "module_id": "i4x://edX/DemoX/Demo_Course/sequential/5c6c207e16dd47208c29bd8d3e68861e",
    "course_id": "edX/DemoX/Demo_Course",
    "count": 23,
    "created": "2014-09-12T114838"
  }
]
```

Video Data API

Section Contents

- *Get the Timeline for a Video*

9.1 Get the Timeline for a Video

class `analytics_data_api.v0.views.videos.VideoTimelineView` (**kwargs)
Get the counts of users and views for a video.

Example Request

```
GET /api/v0/videos/{video_id}/timeline/
```

Response Values

Returns viewing data for each segment of a video. For each segment, the collection contains the following data.

- `segment`: The order of the segment in the video timeline.
- `num_users`: The number of unique users who viewed this segment.
- `num_views`: The number of views for this segment.
- `created`: The date the segment data was computed.

Example Response

```
HTTP 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
Allow: GET, HEAD, OPTIONS

[
  {
    "segment": 0,
    "num_users": 472,
    "num_views": 539,
    "created": "2015-05-13T050419"
  },
  {
    "segment": 1,
```

```
"num_users": 450,  
"num_views": 510,  
"created": "2015-05-13T050419"  
},  
{  
  "segment": 2,  
  "num_users": 438,  
  "num_views": 493,  
  "created": "2015-05-13T050419"  
}  
]
```

C

- CourseActivityMostRecentWeekView (class in analytics_data_api.v0.views.courses), 16
- CourseActivityWeeklyView (class in analytics_data_api.v0.views.courses), 15
- CourseEnrollmentByBirthYearView (class in analytics_data_api.v0.views.courses), 19
- CourseEnrollmentByEducationView (class in analytics_data_api.v0.views.courses), 20
- CourseEnrollmentByGenderView (class in analytics_data_api.v0.views.courses), 21
- CourseEnrollmentByLocationView (class in analytics_data_api.v0.views.courses), 21
- CourseEnrollmentModeView (class in analytics_data_api.v0.views.courses), 18
- CourseEnrollmentView (class in analytics_data_api.v0.views.courses), 17

G

- GradeDistributionView (class in analytics_data_api.v0.views.problems), 25

P

- ProblemResponseAnswerDistributionView (class in analytics_data_api.v0.views.problems), 26

S

- SequentialOpenDistributionView (class in analytics_data_api.v0.views.problems), 27

V

- VideosListView (class in analytics_data_api.v0.views.courses), 23
- VideoTimelineView (class in analytics_data_api.v0.views.videos), 29