

Public REST API and Developer Portal Roadmap

12/13/2023

Mark O'Neil and Camilo Dominguez



Forward-Looking Statement

Statements regarding our product development initiatives, including new products and future product upgrades, updates, or enhancements, represent our current intentions but may be modified, delayed, or abandoned without prior notice, and there is no assurance that such offering, upgrades, updates, or functionality will become available unless and until they have been made generally available to our customers.

Public APIs

Completed (Oct-Jan)

- Gradebook: add dates to results and searches
- Allow changing column formulas /
 Formula validation
- Courses include copy_from_identifier in response
- Calendar: Enable non-3LO calendar management
- LTI context and resource history
- Institutional Hierarchy: Move child nodes to new parent node
- Course Content: Implement resource/x-bb-blankpage content handler
- Add support for User secondary Ids and firstname lastname searches
- Access assessments passwords

Next (Jan-Mar)

- Rubrics:
 - Create/Update Rubrics
 Evaluations
 - Create/Update/Delete Rubric
 Associations
 - Create/Update/Delete Rubrics (maybe – currently being scoped)
- Announcements:
 - createdBy:userId on Course and System Announcements
 - Add create and modified date search (and filter on availability)

Later (release TBD)

- Create/Update/Delete Rubrics
- Group Attempts CRUD
- Learning Modules
- Progress Tracking Read/Update student status for non-participatory items
- Groups: Manage Group Tool Availability
- Enhance Attendance API with timestamp data
- Enhance gradebook API to support Exceptions
- GET Logs
- And more...

Public APIs: and more... (not prioritized)

ΑΡΙ	Description
Activity Stream	Provide ability to manage Activity Stream content. (CRUD)
Assessment APIs	Support Assessment CRUD Support QTI 3.0
Attendance API	General refactoring of Service APIs
Batch API requests	Enable submitting multiple 'like' operations in a single request.
Content Collection	Create/Update/Delete Content Collection Resource Content Collection - Ability to upload zip files
Course Quotas	Provide APIs for Course Quota CRU
Data Sources	Improve searches
Group Grading	Group attempt grading CRUD
Multiple Search Criteria	Add ability to use multiple search criteria
OAuth: Tokens	Ability to revoke and force refresh OAuth tokens
System Endpoint	Add System stats to System endpoint (quotas: disk, DB, etc)
Terms	Search option on Name
UltraStatus management	Course Ultra Status management - set ultra status on courses
Users	Add Add user emails to the list of secondary Ids

Developer Portal

Completed (Oct-Dec)

 Support improvement: Anthology Staff Portal Administrators may manage (CRUD) Accounts and password initiation.

Next (Jan-Mar)

- Token Bucket Request Limiting...
- Change Audit trail
- In Portal API Usage
 Reporting

Later (release TBD)

- Portal Accreditation (FEDRAMP & IL4)
- Improve integration deployment for administrators
- Improve OpenAPI docs
- JWT Service for LTI and REST
- Regional Portal Deployment
- Restrict 'cross-domain' application registration
- Restrict deployment of non-production applications

Rewriting my story...

After 14 wonderful years, December 22nd will be my last day at Anthology.

I am retiring (AKA "work sabbatical") to spend time with my family, work on my 1848 Greek Revival home, continue restoration of my 1970 C10 and 1941 Indian, and return to my artistic roots in my silkscreen studio. I'll likely continue to dabble on some coding as well.

Next week, I'll post more info in Techies where I hope to continue to hang out with all of you.

It has been great being able to serve, and I genuinely **Thank You** for the comradery and friendship over the years!



New Token Bucket Rate_Limiting Model

Token Bucket stores some number of tokens not more than bucket capacity¹. Tokens are requested before a request may be processed. If there are enough tokens, bucket capacity is reduced by one and the request processed. Otherwise, the request is managed per a defined throttling model.

A Token Bucket is refilled with N tokens at a defined time (BFRT)². This controls number of allowed actions over time. This allows for handling traffic bursts, depending on the number of available tokens.

When the Token Bucket is depleted, experiences sustained requests beyond the bucket's capacity, or exceeds the allotted contractual request count, limiting enters an overage/throttling phase.³ During the overage/throttling phase requests are conditionally processed in a manner dictated by the volume and duration of the burst/usage. Developers and stakeholders are informed of throttling cases for proactive adjustment/outreach.

In all cases the integration continues to function albeit with potential perceived performance degradation in the latter case.

Token Bucket Algorithm for Rate Limit



¹ bucket capacity corresponds to existing limits – e.g. 10K for primary groups.
 ² BFRT is the refill rate and corresponds to the current refresh rate, however in our new model T is in milli-seconds vs 24 hours.
 ³ throttling phase slows the request processing queue and provides an opportunity for collecting fees re API use.
 Note: we will continue to support portal admin overrides of settings on individual groups.

New Token Bucket Rate_Limiting Model

Guiding principles behind Token Bucket Rate Limiting:

- 1. Integrations never stop due to rate-limiting
- 2. Contracted Limits set the number of allowed requests during a specified time period *per Group and shared across applications.*
- 3. Contracted limits set the point at which integrations enter throttling based on exceeding usage during the specified Contracted period. E.g, 24 hrs.
- 4. Throttling throttling *may* occur at three points in an application life cycle:
 - 1. When the bucket capacity is exceeded this is a mild throttling with notification to developer/group owner.
 - When the contracted-limit is exceeded this is a mild throttling, with notification to developer/group owner. This is where we would collect overage fees.
 - 3. When all bucket capacity, soft-limit, and hard–limits are exceeded. This typically occurs from a DDOS attack, runaway app, or intentional overuse and is a degrading throttling. Notification is sent to both Portal admins and developer. Note we never completely stop processing requests the integration will display performance degradation.

Scenario:

- 1. Time period: 31 days
- 2. Your bucket is 10K/day
- 3. Your contracted limit is 310K/month
- 4. Your contracted hard limit is 500K/month
- 5. The bucket replenish rate (BRR) is 20/second
- 6. We only allow max of 200 requests per second under all conditions (RPS)
- If you stay within your BRR, you are fine
- If you stay within your BRR while under contracted limits, you are fine.
- In all cases throttling is invoked when:
 - you exceed your BRR while under contracted-limits
 - you exceed your contracted limit
 - you exceed our RPS

New Token Bucket Rate_Limiting Model: A Visual



All non-auth result headers contain limit information: https://www.ietf.org/archive/id/draft-polli-ratelimit-headers-02.html

New Token Bucket Rate_Limiting Model in action...



Improve integration deployment for administrators

- 1. Remove requirement for Admins to create integration users and roles.
 - Admins may choose to create user and role based on #2 and #3.
- Provide integration information to Admins ala previous integration installation processes.
- 3. Require Developers to provide an integration manifest for Production release.





