All Day TA

LTI/API Scoping Documentation

LTI 1.3 Integration

Overview

All Day TA implements LTI 1.3 (Learning Tools Interoperability) to provide seamless integration with Learning Management Systems. This allows institutions to embed All Day TA directly within their LMS and enable grade passback functionality from the All Day TA feature Intelligent Quiz.

Supported LTI Features

1. Dynamic Registration

- **Endpoint**: `/lti/register`
- **Method**: GET, POST
- **Description**: Automated platform registration using OpenID Connect discovery
- **Required Parameters**:
  - `openid_configuration`: OpenID configuration URL from the platform
  - `registration_token`: Platform-provided token for authorization
  - `org_id`: (Optional) Organization ID for hierarchical access
  - `faculty_id`: (Optional) Faculty ID for hierarchical access
  - `department_id`: (Optional) Department ID for hierarchical access

2. OIDC Login Flow

- **Endpoint**: `/lti/login`
- **Method**: GET, POST
- **Description**: Initiates the LTI login flow with the platform
- **Required Parameters**:
  - `iss`: Platform issuer URL

    - `login_hint`: User identifier from the platform
    - `target_link_uri`: Target launch URL
    - `client_id`: OAuth client ID
    - `lti_message_hint`: (Optional) Additional launch context

3. LTI Launch

- **Endpoint**: `/lti/launch`
- **Method**: GET, POST
- **Description**: Handles authenticated LTI launches and redirects users appropriately
- **Supported Launch Types**:
    - Resource link launches (course navigation)
    - Deep linking requests (assignment selection)
    - Deep linking content item launches (graded assignments)

4. Deep Linking

- **Endpoint**: `/lti/deep-linking/response`
- **Method**: POST
- **Description**: Creates content items for assignment integration
- **Required Parameters**:
    - `issuer`: Platform issuer
    - `client_id`: OAuth client ID
    - `deployment_id`: LTI deployment ID
    - `context_id`: LMS course/context ID
    - `deep_link_return_url`: Platform return URL
    - `module_id`: All Day TA module ID to link

5. Assignment and Grade Services (AGS)

- **Capability**: LTI 1.3 Grade Passback
- **Scope Required**: `https://purl.imsglobal.org/spec/lti-ags/scope/score`
- **Features**:
    - Automatic grade synchronization
    - Progress tracking (InProgress, Completed)
    - Grading status (FullyGraded)
    - Audit logging of all grade submissions

All Day TA

LTI Claims Utilized

Standard LTI Claims

None

```
https://purl.imsglobal.org/spec/lti/claim/context
https://purl.imsglobal.org/spec/lti/claim/resource_link
https://purl.imsglobal.org/spec/lti/claim/roles
https://purl.imsglobal.org/spec/lti/claim/deployment_id
https://purl.imsglobal.org/spec/lti/claim/message_type
https://purl.imsglobal.org/spec/lti/claim/version
https://purl.imsglobal.org/spec/lti/claim/custom
```

Assignment and Grade Services Claims

None

```
https://purl.imsglobal.org/spec/lti-ags/claim/endpoint
```

Deep Linking Claims

None

```
https://purl.imsglobal.org/spec/lti-dl/claim/deep_linking_setting
s
https://purl.imsglobal.org/spec/lti-dl/claim/content_items
https://purl.imsglobal.org/spec/lti-dl/claim/data
```

LTI Scopes Required

From Platform Configuration

JSON

```
{
  "scopes_supported": [
```

```
    "openid",
    "https://purl.imsglobal.org/spec/lti-ags/scope/lineitem",

"https://purl.imsglobal.org/spec/lti-ags/scope/result.readonly",
    "https://purl.imsglobal.org/spec/lti-ags/scope/score"
  ]
}
```

LTI Placements

Course Navigation

JSON

```
{
  "type": "LtiResourceLinkRequest",
  "label": "All Day TA",
  "icon_uri":
"https://alldayta.com/static/media/logo.65cf0bb930020eb0f0f8f1e09
f1feeb8.svg",
  "placements": ["course_navigation"]
}
```

Assignment Selection (Deep Linking)

JSON

```
{
  "type": "LtiDeepLinkingRequest",
  "label": "Intelligent Quiz",
  "icon_uri":
"https://alldayta.com/static/media/logo.65cf0bb930020eb0f0f8f1e09
f1feeb8.svg",
  "placements": ["assignment_selection"]
}
```

JWKS Endpoint

- **Endpoint**: `/lti/jwks/<platform_id>`
- **Method**: GET
- **Description**: Provides public key for JWT validation
- **Response Format**: JWK Set (RFC 7517)

Role Detection

The system detects user roles using the LTI roles claim and grants appropriate permissions:

Instructor Roles (Full Access)

- Instructor
- Teaching Assistant
- Administrator
- Content Developer

Student Roles (Limited Access)

- All other roles default to student permissions

Canvas API Integration

Overview

All Day TA integrates with Canvas LMS using OAuth 2.0 for authentication and Canvas REST API for content import.

OAuth 2.0 Configuration

Authorization Flow

- **Authorization Endpoint**: `{canvas_base_url}/login/oauth2/auth`

- **Token Endpoint**: `{canvas_base_url}/login/oauth2/token`
- **Scopes Required**: `url:GET|/api/v1/courses url:GET|/api/v1/users/self`

OAuth Endpoints

- **Login Initiation**: `/api/oauth/login/canvas`
- **Callback Handler**: `/api/oauth/callback/canvas`

Canvas API Endpoints Used

1. User Information

- **Endpoint**: `/api/v1/users/self`
- **Method**: GET
- **Purpose**: Retrieve authenticated user details
- **Required Data**: User ID, name, email

2. Course List

- **Endpoint**: `/api/v1/courses`
- **Method**: GET
- **Purpose**: Fetch courses accessible to the authenticated user
- **Filters**: Active courses, instructor/admin access

3. Course Content

- **Endpoint**: `/api/v1/courses/{course_id}`
- **Method**: GET
- **Purpose**: Retrieve course modules, files, and assignments
- **Includes**: Modules, files, pages, assignments

API Scopes

```JavaScript
{
  "scope": "url:GET|/api/v1/courses url:GET|/api/v1/users/self"
```

```
}
```

Integration Setup Requirements

Required Configuration

1. **Client ID**: Provided by Canvas administrator
2. **Client Secret**: Provided by Canvas administrator
3. **Base URL**: Canvas instance URL (e.g., `https://canvas.university.edu`)
4. **Redirect URI**: `https://yourdomain.com/api/oauth/callback/canvas`

Database Storage

Python

```
{
  "provider": "canvas",
  "client_id": "<canvas_client_id>",
  "client_secret": "<canvas_client_secret>",
  "base_url": "<canvas_instance_url>",
  "redirect_uri": "<redirect_uri>",
  "is_active": true,
  "org_id": <organization_id>,
  "faculty_id": <faculty_id>,  # Optional
  "department_id": <department_id>  # Optional
}
```

D2L (Brightspace) API Integration

Overview

All Day TA integrates with D2L Brightspace using OAuth 2.0 for authentication and D2L Valence API for content access.

All Day TA

OAuth 2.0 Configuration

Authorization Flow

- **Authorization Endpoint**: `https://auth.brightspace.com/oauth2/auth`
- **Token Endpoint**: `https://auth.brightspace.com/core/connect/token`
- **API Base URL**: `{d2l_instance_url}` (institution-specific)
- **Scopes Required**: `core:*:* users:userdata:read`

OAuth Endpoints

- **Login Initiation**: `/api/oauth/login/d2l`
- **Callback Handler**: `/api/oauth/callback/d2l`

D2L API Endpoints Used

1. User Information

- **Endpoint**: `/d2l/api/lp/1.50/users/whoami`
- **Method**: GET
- **Purpose**: Get current user identifier
- **Endpoint**: `/d2l/api/lp/1.50/users/{user_id}`
- **Method**: GET
- **Purpose**: Retrieve detailed user information
- **Required Data**: User ID, first name, last name, email

2. Course List (Not currently released to production)

- **Endpoint**: `/d2l/api/lp/{version}/enrollments/myenrollments/`
- **Method**: GET
- **Purpose**: Fetch user's enrolled courses
- **Filters**: Active enrollments, instructor/admin roles

All Day TA

API Scopes

```javascript
{
  "scope": "core:*:* users:userdata:read"
}
```

Integration Setup Requirements

Required Configuration

1. **Client ID**: Provided by D2L administrator
2. **Client Secret**: Provided by D2L administrator
3. **Base URL**: D2L instance URL (e.g., `https://brightspace.university.edu`)
4. **Redirect URI**: `https://yourdomain.com/api/oauth/callback/d2l`

Database Storage

```python
{
  "provider": "d2l",
  "client_id": "<d2l_client_id>",
  "client_secret": "<d2l_client_secret>",
  "base_url": "<d2l_instance_url>",
  "redirect_uri": "<redirect_uri>",
  "is_active": true,
  "org_id": <organization_id>,
  "faculty_id": <faculty_id>,  # Optional
  "department_id": <department_id>  # Optional
}
```

All Day TA

OAuth Provider Integrations

Google OAuth 2.0

Configuration

- **Provider**: Google
- **Server Metadata URL**: `https://accounts.google.com/.well-known/openid-configuration`
- **Scopes**: `openid email profile`

Endpoints

- **Login**: `/api/oauth/login/google`
- **Callback**: `/api/oauth/callback/google`

User Data Retrieved

- Email address (primary identifier)
- Full name
- Google user ID
- Email verification status

OAuth Status Check

Endpoint

- **URL**: `/api/oauth/status`
- **Method**: GET
- **Description**: Check which OAuth providers are configured and active

Response Example

```json
{
  "providers": {
```

```
    "google": true,
    "canvas": true,
    "d2l": false
  },
  "auto_create_users": false
}
```

OAuth Unlinking

Endpoint

- **URL**: `/api/oauth/unlink/<provider>`
- **Method**: POST
- **Description**: Disconnect OAuth provider from user account
- **Requirements**: User must have alternative authentication method (password)

Security and Best Practices

LTI Security

1. State and Nonce Management

- State and nonce values are generated using `os.urandom(16).hex()`
- Stored in database with 10-minute expiration
- Validated during callback to prevent CSRF attacks
- Automatically cleaned up after use or expiration

2. JWT Validation

- All ID tokens validated using platform's JWKS endpoint
- Signature verification using RS256 algorithm
- Claims validation includes:

    - Issuer (iss)
    - Audience (aud)
    - Nonce
    - Expiration with 60-second leeway

3. Platform Registration

- Supports dynamic registration via OpenID Connect
- Generates unique RSA key pairs per platform
- Keys stored securely in the database
- JWKS endpoint exposes only public keys

4. Grade Passback Security

- Uses client_credentials grant with JWT assertion
- Private key JWT authentication (not client_secret)
- Access tokens obtained per-request (short-lived)
- All grade submissions logged for audit trail

OAuth Security

1. Token Management

- Access tokens stored in server-side sessions
- Tokens never exposed to client-side JavaScript
- Session-based authentication after the OAuth flow
- Integration ID linked to the session for provider selection

2. State Parameter

- Generated using `secrets.token_urlsafe(32)`
- Validated on callback to prevent CSRF
- Expires after single use

3. HTTPS Enforcement

- All OAuth redirects are forced to HTTPS in production
- Local development supports HTTP for testing

All Day TA

API Security

1. Authorization Checks

- All API endpoints protected with `@admin_required()` decorator
- Feature flag validation for LTI and gradebook endpoints
- Hierarchical access control based on org/faculty/department ownership

2. Access Token Validation

- OAuth tokens validated on each API request
- Provider-specific token refresh if supported
- Automatic token expiration handling

3. Rate Limiting

- Gradebook submissions rate-limited per platform
- Failed attempts logged and monitored
- Retry logic with exponential backoff

Data Privacy

1. User Data

- Minimal PII collection (email, name only)
- User consent obtained during OAuth flow
- Email verification status tracked
- Option to auto-create users or require pre-registration

2. LTI User Identifiers

- Platform-specific user IDs stored separately
- No cross-platform user correlation
- Anonymous quiz session tokens for privacy
- LTI context isolated per platform/deployment

3. Session Management

- Sessions tied to specific organizations
- Database-tracked sessions for audit
- Automatic session cleanup on logout
- Last activity timestamps for session expiration

Audit and Compliance

1. Grade Passback Audit Log

- Every grade submission logged with:
  - Practice session ID
  - LTI quiz session ID
  - Score given/maximum
  - Platform information
  - Success/failure status
  - Error details (if failed)
  - Timestamp

2. Access Logs

- OAuth authentication events logged
- LTI launch events logged
- Integration creation/modification logged

3. Data Retention

- OIDC states expire after 10 minutes
- Old states automatically cleaned up
- Session activity tracked for expiration
- Audit logs retained per institutional policy

Best Practices for Institutions

1. LTI Deployment

- Use separate deployments per institution/faculty/department

- Generate unique RSA keys per platform
- Enable the institutional management feature flag
- Test in the sandbox environment first

2. OAuth Integration

- Use institution-specific client credentials
- Configure appropriate scopes (minimal required)
- Set up proper redirect URIs
- Test authentication flow thoroughly

3. Access Control

- Assign ownership at appropriate hierarchy level
- Use faculty/department scoping when possible
- Regularly audit integration access
- Monitor gradebook submission logs

4. User Onboarding

- Pre-register users when possible (auto-create disabled by default)
- Provide clear instructions for LTI/OAuth setup
- Test with pilot group before full rollout
- Monitor error logs during initial deployment

Changelog

Version 1.0 (Current)

- LTI 1.3 dynamic registration support
- Canvas and D2L OAuth integration
- Google OAuth providers
- Assignment and Grade Services (AGS) implementation
- Deep Linking support
- Hierarchical access control
- Comprehensive audit logging