



Newrow Integration with LTI - V1.1

1. What is LTI:

LTI - Learning Tools Interoperability™ is developed by IMS Global Learning Consortium, to allow remote tools and content to be integrated into a Learning Management System (LMS). This document brings a subset of that specification forward for standardization. [This document](#) precedes the full IMS LTI v1.0 specification set.

2. Newrow_ LTI integration:

newrow_ supports Basic LTI that exposes a single destination as a Tool Provider system. Newrow_ exposes the live room as single destination page.

Basic LTI will do for you:

- Create Newrow_ user - if the user does not already exist
- Create Newrow_ Course / Room - if the course/room does not already exist
- Append user to course attendees with the specific role
- Redirect the user to the room

The procedure for establishing a request to this single destination is simple:

1. Create your newrow account (if you do not already have one)
2. Send an email to sales@newrow.com that includes the following information:
Email address associated with newrow_ account, name of your company/institution, name of the LMS you are using. Upon clarification of needs, you will be provided your LTI_KEY & LTI_SECRET.
3. Setup newrow_ as a tool provider in your LMS using the provided key, secret and set the launch url to be: <https://smart.newrow.com/backend/lti/course>

If you are using an LMS that has built in LTI integration, you need to find their documentation on how to configure LTI tool providers.

For example:

- Brightspace / D2L - [How to](#)
- Schoology - [How to](#)
- Moodle - [How to](#)

If you have your own system and you want to integrate newrow_ LTI - find the php example below and implement it in your system!

For more info [click here](#).

3. PHP Example

Code example for secured LTI Request.

** In the example below you will find the mandatory fields that Newrow use.*

```
<?php
// -----
// CONFIGURATION SECTION
// -----
$launch_url = "https://smart.newrow.com/backend/lti/course";

$key = "LTI_KEY_GIVEN_BY_NEWROW";
$secret = "LTI_SECRET_GIVEN_BY_NEWROW";

$launch_data = array(

    // Consumer identifier (Vendor name like: desire2learn, moodle, etc.)
    "tool_consumer_info_product_family_code" => "newrowTest",

    //-----
    // User details
    //-----
    // Internal authenticated user id in your system
    "user_id" => "2928321264",

    // User role ('Administrator','Instructor', 'Student')
    "roles" => "Student",

    // User email
    "lis_person_contact_email_primary" => "sam@school.edu",

    // User first name
    "lis_person_name_given" => "Sam",

    // User last name
    "lis_person_name_family" => "Thompson",

    //-----
    // Room details
    //-----
    // Internal Room identifier
    "context_id" => "4512313",

    // Room name
    "context_title" => "Design of Personal Environments",

);
```

```

// -----
// OAUTH CONFIGURATION
// -----

$now = new DateTime();

$launch_data["lti_version"] = "LTI-1p0";

# Basic LTI uses OAuth to sign requests
# OAuth Core 1.0 spec: http://oauth.net/core/1.0/

$launch_data["oauth_callback"] = "about:blank";
$launch_data["oauth_consumer_key"] = $key;
$launch_data["oauth_version"] = "1.0";
$launch_data["oauth_nonce"] = uniqid('', true);
$launch_data["oauth_timestamp"] = $now->getTimestamp();
$launch_data["oauth_signature_method"] = "HMAC-SHA1";

# In OAuth, request parameters must be sorted by name
$launch_data_keys = array_keys($launch_data);
sort($launch_data_keys);

$launch_params = array();
foreach ($launch_data_keys as $key) {
    array_push($launch_params, $key . "=" . rawurlencode($launch_data[$key]));
}

$base_string = "POST&" . urlencode($launch_url) . "&" . rawurlencode(implode("&",
$launch_params));
$secret = urlencode($secret) . "&";
$signature = base64_encode(hash_hmac("sha1", $base_string, $secret, true));

?>

<form id="ltiLaunchForm" name="ltiLaunchForm" method="POST" action="<?php
printf($launch_url); ?>">
    <?php foreach ($launch_data as $k => $v ) { ?>
        <input type="hidden" name="<?php echo $k ?>" value="<?php echo $v ?>">
    <?php } ?>
    <input type="hidden" name="oauth_signature" value="<?php echo $signature ?>">
</form>

<script language="javascript">
    document.getElementById("ltiLaunchForm").style.display = "none";
    document.ltiLaunchForm.submit();
</script>

```

* Best practice is to load this page in iframe in your App.

Lets says that the php example available in your app at:

<https://www.yourapp.com/newrow/ltiRoomLancher>

To load this launch url in your iframe, add the following:

```
<--> Site Header Section <-->
```

```
<iframe allow="microphone *; camera *; speakers *; usermedia *;"  
allowfullscreen src="https://www.yourapp.com/newrow/ltiRoomLancher"  
></iframe>
```

```
<--> Site Footer Section <-->
```

4. LTI Roles:

Newrow_ has 3 user roles.

1. Admin - Can join to each room and manage company
2. Instructor - Can host courses and manage course content
3. Student - Can join courses

Newrow_ support LTI roles:

```
$roles = [  
  ROLE_ADMIN => [  
    'urn:lti:sysrole:ims/lis/SysAdmin',  
    'urn:lti:sysrole:ims/lis/SysSupport',  
    'urn:lti:sysrole:ims/lis/Creator',  
    'urn:lti:sysrole:ims/lis/AccountAdmin',  
    'urn:lti:sysrole:ims/lis/Administrator',  
    'urn:lti:instrole:ims/lis/Staff',  
    'urn:lti:instrole:ims/lis/Administrator',  
    'Administrator'  
  ],  
  ROLE_INSTRUCTOR => [  
    'urn:lti:instrole:ims/lis/Faculty',  
    'urn:lti:instrole:ims/lis/Instructor',  
    'urn:lti:instrole:ims/lis/Mentor',  
    'urn:lti:role:ims/lis/TeachingAssistant',  
    'Instructor'  
  ],  
  ROLE_STUDENT => [  
    'urn:lti:instrole:ims/lis/Student',  
    'urn:lti:instrole:ims/lis/Member', // Member of a course, can be a student  
    'urn:lti:instrole:ims/lis/Learner',  
    'urn:lti:instrole:ims/lis/Alumni', // Alumni likely means former student  
    'urn:lti:instrole:ims/lis/Other',  
    'urn:lti:instrole:ims/lis/ProspectiveStudent',  
    'urn:lti:instrole:ims/lis/Guest',  
    'urn:lti:instrole:ims/lis/Observer',  
    'urn:lti:instrole:ims/lis/None',  
    'Learner',  
    'Student'  
  ]  
];
```

5. Embedded Iframe - Parent Window Functionality

Newrow_ expose to Parent Window the ability to take some actions in Newrow_ Iframe. Actions can be passed by window.postMessage JS function.

Object structure to pass to function is:

```
{
  type: String,
  action: String
}
```

Each type is different module in Newrow, and for each type there are allowed actions.

Allowed types:

- Newrow Top Menu Events - '**newrowTopBarEvents**'
Allowed actions:
 1. Video ON/OFF - '**videoToggle**'
 2. Audio ON/OFF - '**audioToggle**'
 3. Chat Show/Hide - '**chatToggle**'
 4. Participants list Show/Hide - '**participantsListToggle**'

Example for Participants list Show/Hide :

```
window.postMessage({
  type: 'newrowTopBarEvents',
  action: 'participantsListToggle'
}, '*');
```