

edunext.co



www.edunext.co



OPEN/edX[®]

AT HOME

Conference _____ 2023
hosted by MIT





Reasons to integrate

- Streamline operations
- Synchronize data
- Automate processes
- Adapt to specific business rules
- Extend the capabilities of your platform
- Improve the learner experience

Custom built integration

- Plugins
- Xblocks
- IDAs
- Fork (Hopefully not)

```
16  
17 # using BeautifulSoup to parse the response object  
18 soup = BeautifulSoup(response.content, "html.parser")  
in the soup ("title": "Post 1000")
```


Custom built integration

Build

- Product design
- Implementation
- Testing
- Documentation

Maintain

- Maintaining compatibility with Open edX
- Product changing needs
- Changes in the external applications

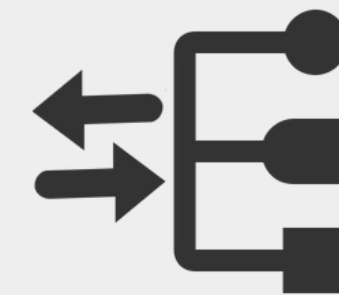
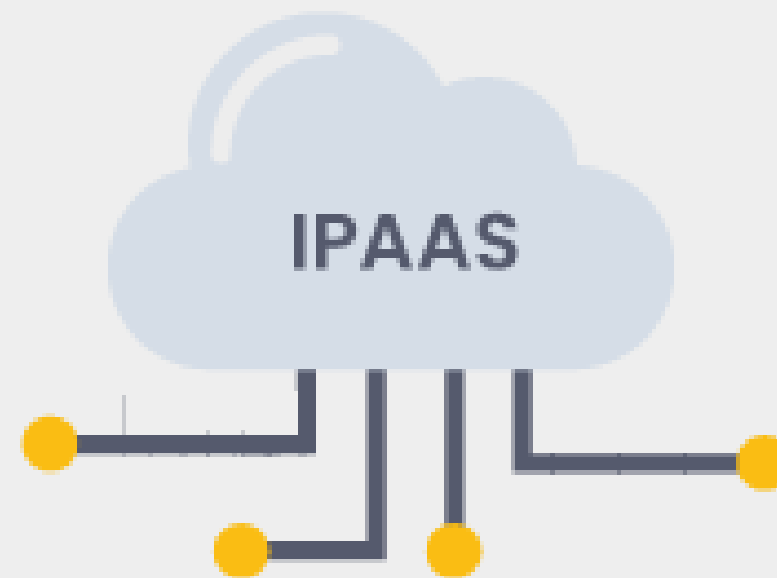
Technical expertise
Time
Budget

```
16  
17 # using BeautifulSoup to parse the response object  
18 soup = BeautifulSoup(response.content, "html.parser")  
in the soup ("title": "Post 1000")
```


PROPOSED ALTERNATIVE

**Integration
broker**

**Open edX
adapters**





INTEGRATION PLATFORM AS A SERVICE



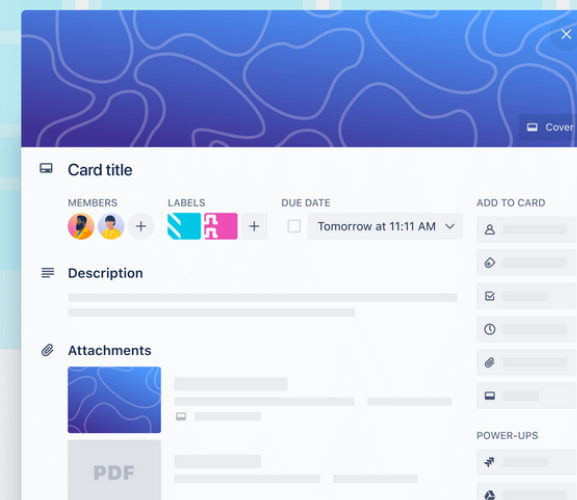
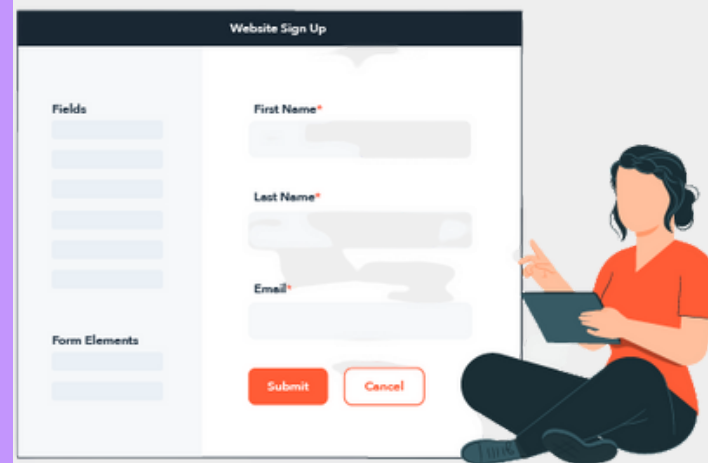


INTEGRATION PLATFORM AS A SERVICE

- Listens to (polls) platform A for a **Trigger** event
- Captures the event **payload**



- **Authenticates** into platform B
- Launches the **Action** in platform B





INTEGRATION PLATFORM AS A SERVICE



INTEGRATION PLATFORM AS A SERVICE



1.

CREATE A ZAP

2.

SET UP TRIGGER

- Select hubspot
- Connect my hubspot account
- Select the event
- Select the form id

3.

SET UP ACTION

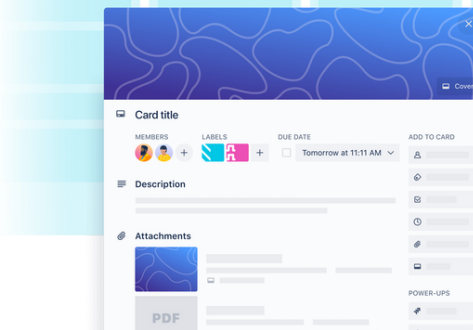
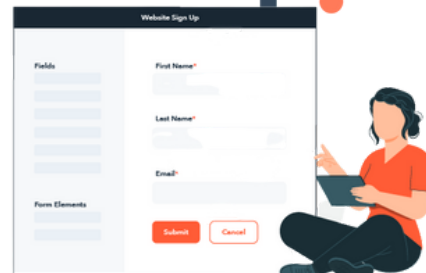
- Select trello
- Connect my trello account
- Select the action
- Select the card
- Configure the content of the card

4.

PUBLISH

TEST IT

TEST IT



INTEGRATION PLATFORM AS A SERVICE


Advanced tools



1.

CREATE A ZAP

2.

SET UP TRIGGER

- Scheduler
- Email parser
- Expose Webhooks

3.

MULTI STEP ACTIONS

- Delays
- If then
- Loops
- Storage
- Send email
- Send SMS
- Publish RSS

4.

PUBLISH

INTEGRATION PLATFORM AS A SERVICE

 zapier

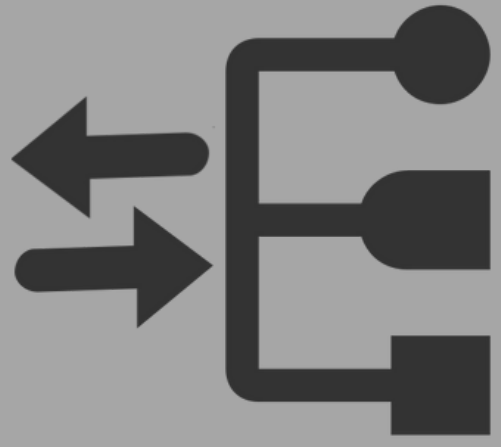
Advantages

- Large # of apps with native integrations
- Intuitive interface
- Useful Built-in tools
- Great flexibility - more complex workflows
- Free to start - kind of
- Auditing, error logging, notifications

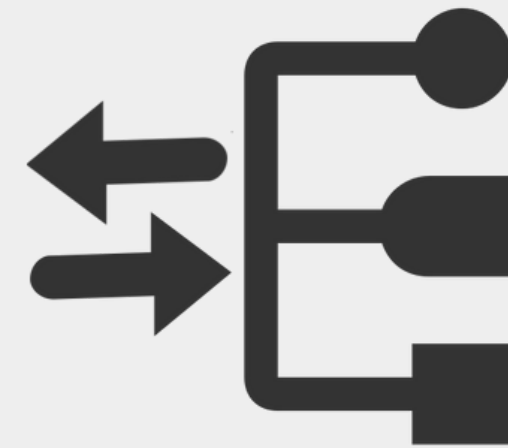
- Powerful developer tools to build a native integration



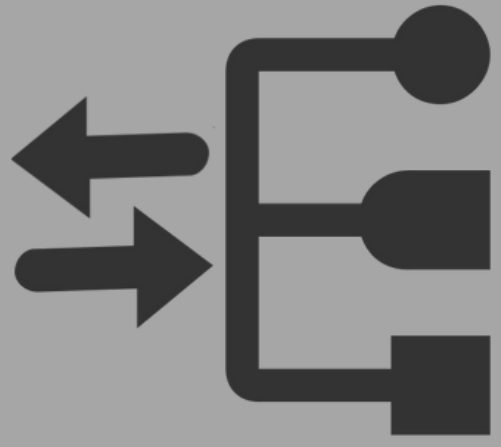
IPaaS



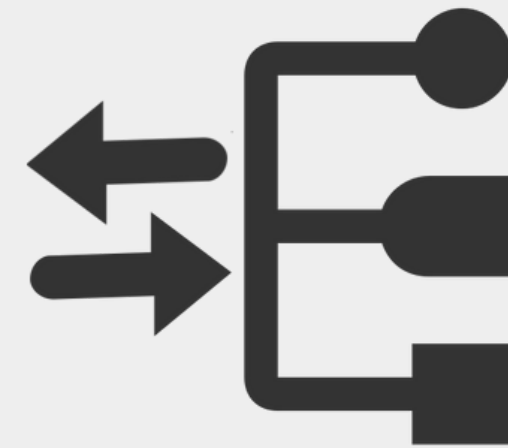
OPEN EDX ADAPTERS



OPEN EDX ADAPTERS



#1 - Inbound workflow



#2 - Outbound workflow

#1 - Inbound workflow

Whenever someone registers in my **Salesforce** form

a user account is created in Open edX

Whenever a request is marked as approved in **Notion**

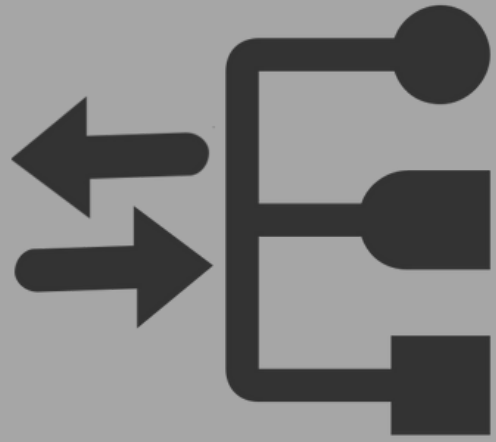
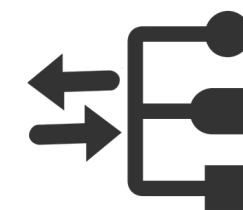
course creation permissions are granted to the user

Whenever a purchase is made in **Paypal**

an enrollment to the Open edX course is done

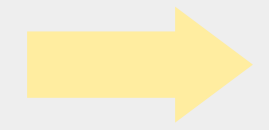
Whenever the user submits a specific type of requests in **Zendesk**

he/she are unenrolled from the course



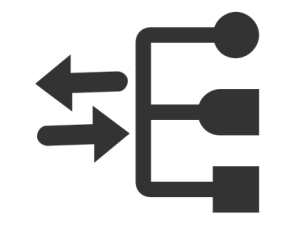
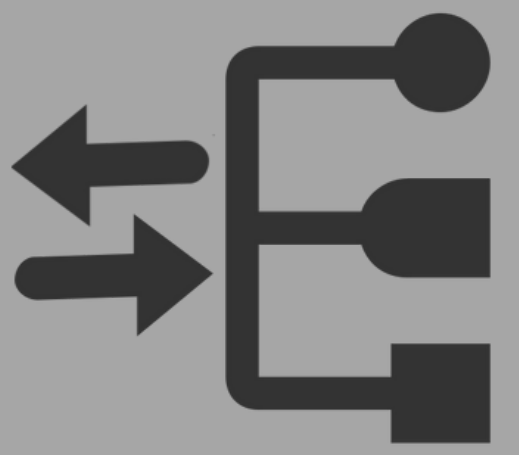
#1 - Inbound workflow

Custom HTTP request



Administrative APIs

- GET
- POST
- PATCH
- PUT
- DELETE



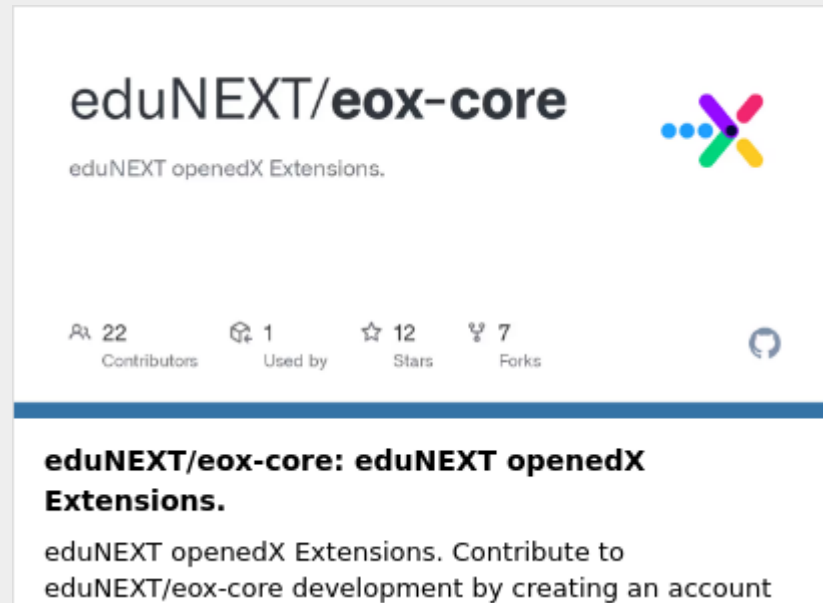
OPEN EDX ADAPTERS

#1 - Inbound workflow

EOX_CORE

Release 8.1.0

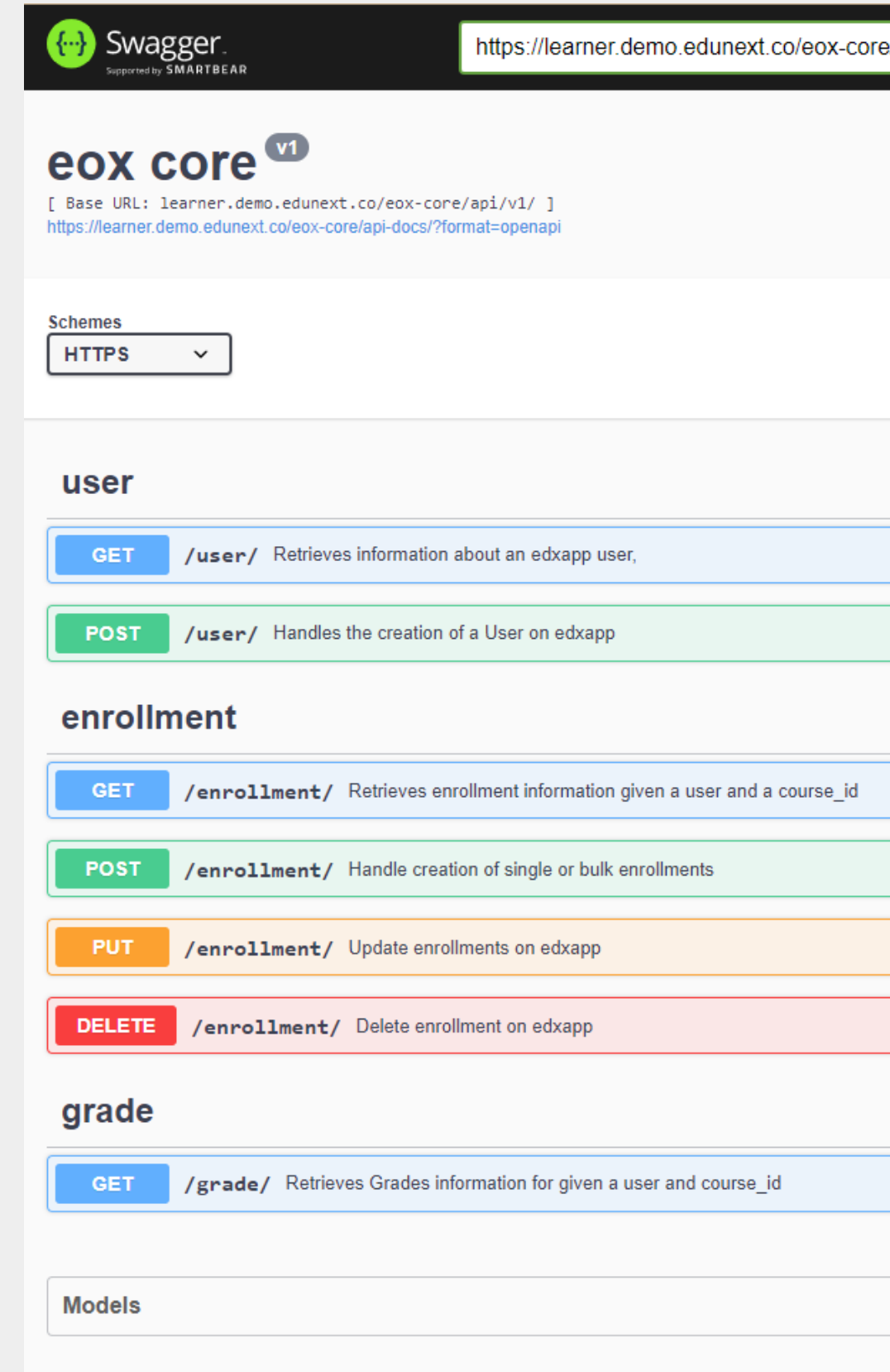
<https://github.com/eduNEXT/eox-core>



eduNEXT/eox-core
eduNEXT openedX Extensions.

22 Contributors, 1 Used by, 12 Stars, 7 Forks

eduNEXT/eox-core: eduNEXT openedX Extensions.
eduNEXT openedX Extensions. Contribute to eduNEXT/eox-core development by creating an account



Swagger
Supported by SMARTBEAR
<https://learner.demo.edunext.co/eox-core>

eox core ^{v1}
[Base URL: learner.demo.edunext.co/eox-core/api/v1/]
<https://learner.demo.edunext.co/eox-core/api-docs/?format=openapi>

Schemes
HTTPS

user

- GET /user/ Retrieves information about an edxapp user,
- POST /user/ Handles the creation of a User on edxapp

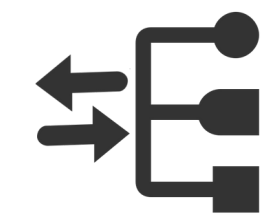
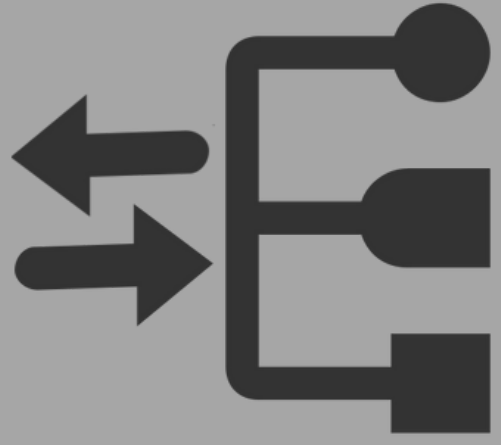
enrollment

- GET /enrollment/ Retrieves enrollment information given a user and a course_id
- POST /enrollment/ Handle creation of single or bulk enrollments
- PUT /enrollment/ Update enrollments on edxapp
- DELETE /enrollment/ Delete enrollment on edxapp

grade

- GET /grade/ Retrieves Grades information for given a user and course_id

Models



Whenever a purchase is made in
Paypal



an enrollment to the Open edx course
is done

1.

CREATE A ZAP

2. SET UP TRIGGER

- Select Paypal
- Connect your paypal account
- Select the event "Successful sale"
- Provide the Webhook URL to paypal IPN (instant payment notification)
- Test (make a purchase)
- Verify the payload

3.

SET UP ACTION

- Select "Webhooks by Zapier"
- Select the action "Custom Request"
- Provide all the details for the API endpoint
- Provide the custom data package that the API endpoint needs using the learner email from step 2.
- Provide the authentication token headers
- Test the API

4.

PUBLISH

EXAMPLE


#1 - Inbound workflow

Whenever a purchase is made in
Paypal




an enrollment to the Open edx course
is done

3. SET UP ACTION

 Action
2. Custom Request in Webhooks...

App & event ✓

 Webhooks by Zapier **Premium** Change

* Event (required)

Custom Request

This is performed when the Zap runs.

Continue

* Method (required)

POST

The HTTP method we'll use to perform the request.

* URL (required)

https://mydomain.com/eox-core/api/v1/enrollment/

Any URL with a querystring will be re-encoded properly.

Data

```
{
  "email": "1. Payer Email: testaccount@zapier.com",
  "course_id": "course-v1:edX+DemoX+Demo_Course",
  "mode": "honor",
  "force": "True",
  "is_active": "False"
}
```

Place raw data for the request here. Could be JSON, XML, etc. Nothing is parsed fo... [more](#)

Headers

authorization

Bearer vX41YpmiO00E0qLy5fEuMkJekUXkJs

×

+

Key, value pairs to be added as headers in all requests.

Refresh fields

Continue

Whenever a purchase is made in
Paypal



an enrollment to the Open edx course
is done

1.

CREATE A ZAP

2. SET UP TRIGGER

- Select Paypal
- Connect your paypal account
- Select the event "Successful sale"
- Provide the Webhook URL to paypal IPN (instant payment notification)
- Test (make a purchase)
- Verify the payload

3.

SET UP ACTION

- Select "Webhooks by Zapier"
- Select the action "Custom Request"
- Provide all the details for the API endpoint
- Provide the custom data package that the API endpoint needs using the learner email from step 2.
- Provide the authentication token headers
- Test the API

4.

PUBLISH

#2 - Outbound workflow

Send a them welcome **email**

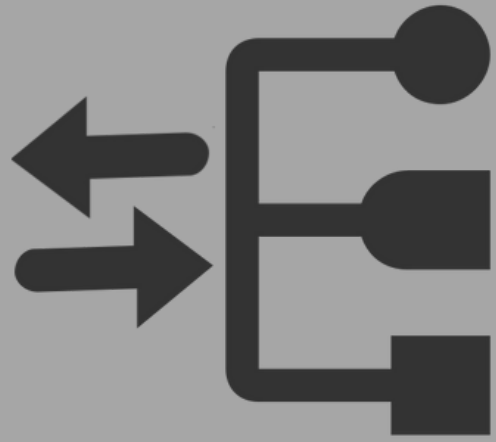
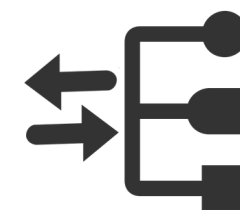
Whenever a learner registers to the Open edX platform

Invite them to a specific channel in **slack**

Whenever the user enrolls in a course

Request the generation of a badge in **credly**

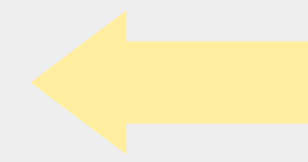
Whenever a learner gets a certificate



OPEN EDX ADAPTERS

#2 - Outbound workflow

Zapier
Webhook

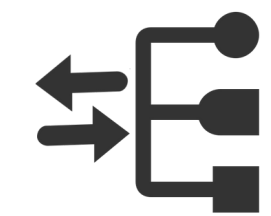
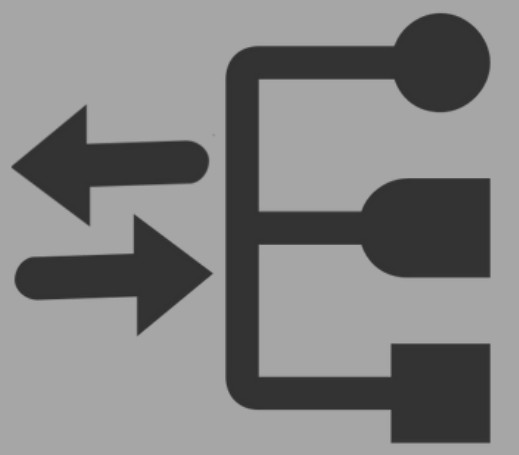


-catches signal
-packs payload



Signal

Open edX Hooks
OEP-50
<https://github.com/openedx/openedx-events>
<https://github.com/openedx/openedx-filters>



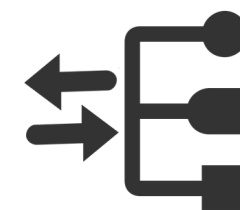
#2 - Outbound workflow

EOX_HOOKS

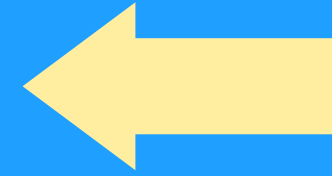
Release 4.0

<https://github.com/eduNEXT/eox-hooks>

POST_REGISTER
POST_ENROLLMENT
POST_UNENROLLMENT
POST_COHORT_CHANGE
PERSISTENT_GRADE_SUMMARY_CHANGED
POST_CERTIFICATE_CREATION
POST_CERTIFICATE_REVOKE
POST_SESSION_LOGIN



Invite them to a specific channel in
slack



Whenever the user enrolls in a course

1.

CREATE A ZAP

2.

SET UP TRIGGER

- Select "Webhooks by Zapier"
- Select the event "Catch Hook"
- Get the webhook URL
- Configure the post_enrollment hook in EOX_HOOKS to call the webhook URL
- Test the webhook (perform a registration)
- Verify the payload

3.

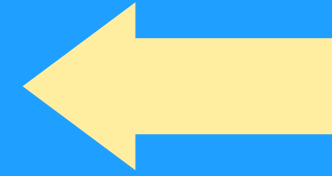
SET UP ACTION

- Select Slack
- Connect my slack account
- Select the action "Invite user to Channel"
- Select the Channel
- Select the user using the info received in step 2.

4.

PUBLISH

Invite them to a specific channel in
slack



Whenever the user enrolls in a course

2. SET UP TRIGGER

Trigger
1. Catch Hook in Webhooks by Z...

App & event ✓

Webhooks by Zapier **Premium** Change

* Event (required)
Catch Hook ↕

This is what starts the Zap.

Continue

Trigger
1. Catch Hook in Webhooks by Z...

App & event ✓

Trigger ✓

Test (skipped) ✓

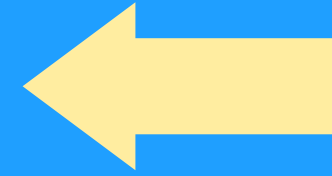
Your webhook URL
You'll need to configure your application with this Zap's webhook URL.

<https://hooks.zapier.com/hooks/catch/2...> Copy

We've generated a custom webhook URL for you to send... [more](#)

```
lms.env.yml  
  
EOX_HOOKS_DEFINITIONS:  
  post_enrollment:  
    action: "post_to_webhook_url"  
    config:  
      fields:  
        course_id: "course_key"  
        learner_email: "user.email"  
        learner_name: "user.profile.name"  
    url: "https://hooks.zapier.com/hooks/cat  
fail_silently: false  
module: "eox_hooks.actions"
```

Invite them to a specific channel in
slack



Whenever the user enrolls in a course

1.

CREATE A ZAP

2.

SET UP TRIGGER

- Select "Webhooks by Zapier"
- Select the event "Catch Hook"
- Get the webhook URL

- Configure the post_enrollment hook in EOX_HOOKS to call the webhook URL

- Test the webhook (perform a registration)
- Verify the payload

3.

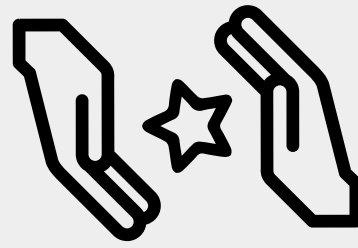
SET UP ACTION

- Select Slack
- Connect my slack account
- Select the action "Invite user to Channel"
- Select the Channel
- Select the user using the info received in step 2.

4.

PUBLISH

Native OpenedX integration for Zapier?



integrations editor UI

The screenshot shows the Zapier Platform interface for configuring a 'New Issue Trigger' for a GitHub repository. The left sidebar contains navigation options like 'Dashboard', 'BUILD', 'Integration Home', 'Authentication', 'Triggers', and 'Actions'. The main area is titled 'New Issue Trigger' and includes tabs for 'Settings', 'Input Designer', and 'API Configuration'. The 'Form Editor' section shows a 'Next up' message and a 'Create a Zap' button. Below this is a table for defining trigger fields:

Label	Key	Sort
Repo	repo	↑ ↓
Owner of Repo	owner	↑ ↓
State	state	↑ ↓

Below the table is an 'Add User Input Field' button. On the right, a 'ZAP EDITOR PREVIEW' shows the form fields: 'Repo' (required), 'Type in the name of the repo', 'Owner of Repo' (required), and 'State' (dropdown).

Zapier Platform CLI



```
maguaybook:oauth2-example maguay$ zapier versions
All versions of your app "Wistia CLI Test" listed below.



| Version | Platform | Users | Deployment     | Deprecation Date | Timestamp           |
|---------|----------|-------|----------------|------------------|---------------------|
| 1.0.1   | 7.2.2    | 0     | non-production | null             | 2019-01-29T00:44:40 |



maguaybook:oauth2-example maguay$ zapier push
Preparing to build and upload your app.

✓ Copying project to temp directory
✓ Installing project dependencies
✓ Applying entry point file
✓ Building app definition.json
✓ Validating project
✓ Zipping project and dependencies
✓ Testing build
✓ Cleaning up temp directory
✓ Uploading version 1.0.1

Build and upload complete! You should see it in your Zapier editor at https://zapier.com/app/editor now!

There's a newer version of zapier-platform-core available.
Consider updating the dependency in your
package.json (7.2.2 → 7.6.1) and then running zapier test.

maguaybook:oauth2-example maguay$
```

Thank you!

Integrating Open edX with more than
5000 web applications

MARCH 2023



Juan Camilo Montoya
edunext.co

**SHAPING
THE FUTURE
OF EDUCATION
TOGETHER**
Boston

