Webhooks | Skillable TMS

**Target Skillable Personas:** Integration Teams
**Target Customer/Partner Personas:** API Integrators - TMS
**Target Timeframe:** Q2 2023

As strategic customers continue leaning on API integration with Skillable Connect we are working on ways to simplify and expand on the solution offering to minimize the load on our backend while offering a more robust method to engage at certain stages based on event driven model.

Integration with Skillable RESTful APIs are great ways to get or update data and work fantastic. The one challenge with this approach is that the remote site is required to constantly poll for updates of our Lab Instance data, for example. This constant polling adds undo load on our API framework. To eliminate the use of constant Rest API polling strategies it is best to implement a Webhook framework that will initiate a call back with information triggered by a specific event. This method minimizes the required amount of call backs.

* Skillable TMS does not support any webhooks and is definitely lacking in providing better remote programmability for lifecycle actions.

Linked Aha ideas:

<https://lods.aha.io/ideas/ideas/IDEA-I-377>

## Fundamentals

**Skillable TMS**

Customer Configuration

* Provide a similar form-based approach as Skillable Studio to see/edit configuration at Org level
* Provide a similar API approach to webhook management

Class/Subscription Webhooks

* Have a way to keep remote system updated when user activity progresses through the Course States
	+ Class/Course Feedback – when created send webhook
	+ Course Assignment completion status
		- Started – when the user actually starts
		- Completed
		- Expired and status Incomplete/Not Started ( user did not complete within allotted timeframe)
* Class updates – keeping remote systems in sync when class status changes
	+ Scheduled = 0
	+ Guaranteed To Run = 1
	+ Cancelled = 2
	+ Rescheduled = 3
	+ Tentative = 4
	+ Requested = 5
	+ Denied = 6
	+ Completed = 7

Enrollment Webhooks

* Enrollments have a lot of changing status and would benefit with webhooks to keep remote systems in sync
	+ class enrollment Status updates
		- 0 = Request
		- 10 = Cancel
		- 20 = Waitlist
		- 30 = Approve
		- 40 = Deny
		- 50 = Audit
		- 60 = Enroll

User Webhooks

* Keep remote systems in sync for user record management
	+ User record updated

Learning Path

* Be able to have the Learning Path completion trigger a webhook

Achievement

* When use receives an achievement a webhook can be triggered

Voucher redemption

* Trigger a webhook upon voucher redemption

Course Assignment

* Trigger a webhook when course assignment has more then a and less then y days before expiration