



Migrating from REST to GraphQL APIs

Tags: **Skills**

Pluralsight Skills' APIs provide the most efficient way to unlock Skills data and extend the platform into your internal systems and workflows. Skills' newest APIs are powered by **GraphQL**, a technology that gives you exactly what you need intuitively and efficiently.

	Who can use this?				
	<u>Std</u>	<u>Prem</u>	<u>Strt</u>	<u>Pro</u>	<u>Ent</u>
<u>Learners:</u>					
<u>Managers:</u>					
<u>Admins:</u>			✓	✓	✓

While Skills' previous REST APIs offered a limited dataset focused on courses, usage, and user management, GraphQL APIs expand the datasets to channels, roles, skills, and more. GraphQL also allows you to replace multiple REST queries with a single query which includes explicit data requirements. This article contains sample code and developer resources that will help you migrate from REST to GraphQL.

In this article

[How GraphQL differs from REST](#)

[Migrating from REST to GraphQL](#)

[Content completion APIs](#)

[License management APIs](#)

How GraphQL differs from REST

A single GraphQL query can fetch the complete array of data per the relationships defined in the schema, instead of multiple requests with REST.

GraphQL also allows for mutations, allowing you to perform, create, update, and delete operations, thereby replacing CRUD functionality in REST APIs.

GraphQL has a greater number of API fields available, and queries are simpler and more precise. See the complete Skills GraphQL Schema within the [Developer Portal \(opens in new tab\)](#) for fields and definitions.

[back to top](#)

Migrating from REST to GraphQL

As a plan admin, you have the ability to use GraphQL APIs, create API keys, and submit queries. You also have

access to the [Developer Portal \(opens in new tab\)](#), the primary developer resource for working with our APIs. If you're unsure whether you have plan admin permissions, visit [Manage Keys \(opens in new tab\)](#) to see if you can create a new GraphQL API key.

Tip: If you're a developer or analyst, you can still obtain data through APIs by having your plan admin generate and share their API key with you.

If you want to start fresh with GraphQL, use the Developer Portal to complete these steps:

1. Generate an [API key \(opens in new tab\)](#).
2. Learn how to build your [First GraphQL API Query \(opens in new tab\)](#).
3. Copy one of the sample code snippets from the tables below to begin testing your Queries in the built-in [Playground \(opens in new tab\)](#). Consult the API documentation in the Developer Portal to add or remove fields as required.

If you'd rather modify your existing REST queries, see examples of how to modify your queries below.

Course Daily Usage API

For applicable filters for the GraphQL Content Usage API, please see [courseDailyUsage \(opens in new tab\)](#).

See [Content usage \(opens in new tab\)](#) if you're looking to migrate from the REST Content Usage API.

REST API field	GraphQL API field	Sample code
UserId	user { id }	<pre> { courseDailyUsage { nodes { user { id firstName lastName email team { name } note isOnAccount } } course { id slug title } rollupDate clipViewSeconds } } </pre>
FirstName	user { firstName }	
LastName	user { lastName }	
Email	user { email }	
TeamName	user { team { name } }	
Note	user { note }	
IsOnAccount	user { isOnAccount }	
ContentType	N/A	
ContentId	course { id }	
ContentSlug	course { slug }	
ContentName	course { title }	
ViewDate	rollupDate	
UsageInSeconds	clipViewSeconds	

Filters

REST API Filter example	GraphQL API filter example	Example description
<code>&startDate=2019-12-04 &endDate=2019-12-04</code>	<pre>filter: { startDate: "2019-12-04" endDate: "2019-12-04" }</pre>	Use this filter to pull delta reports (example daily, weekly or biweekly reports). GraphQL API is based on the field <code>lastActivityOn</code> . Note: There could be a 24-hour data lag, so to get data from the 24th, for example, pull data on the 26th.

User API

Use this if you're looking to migrate from the [REST Users API \(opens in new tab\)](#).

For applicable filters for the GraphQL Users API, please see [Users \(opens in new tab\)](#).

REST API field	GraphQL API field	Sample code
UserId	id	<pre>{ users { nodes { id firstName lastName email startedOn note team { name } } } }</pre>
Email	email	
FirstName	firstName	
LastName	lastName	
StartDate	startedOn	
Note	note	
TeamName	team	
N/A	isOnAccount	
N/A	removedOn	
N/A	ssoidentifiers	
N/A	ssoEnabled	
N/A	planId	
N/A	createdOn	
N/A	additionalEmails	

Note: By default, the Users API includes all current as well as historical licensed users on the plan. Use `isCurrent` [userFilter \(opens in new tab\)](#) to extract current users only.

[back to top](#)

Content completion APIs

If you're looking to migrate from the [Content Completion REST API \(opens in new tab\)](#), there are two options available:

1. [CourseProgress API \(opens in new tab\)](#) is currently in General Release and only provides data from [Video courses \(opens in new tab\)](#).
2. [ContentProgress API \(opens in new tab\)](#) provides data from all content types like videos, guides, paths, interactive courses and projects. This API is currently in Beta.

Course Progress API

The GraphQL [Course Progress API \(opens in new tab\)](#) includes courses that are in progress.

REST API field	GraphQL API field	Sample code
dir="ltr">UserId	user { id }	<pre>{ courseProgress { nodes { user { id firstName lastName email team { name } note isOnAccount } course { id title slug } firstViewedClipOn percentComplete isCourseCompleted completedOn } } }</pre>
FirstName	user { firstName }	
LastName	user { lastName }	
Email	user { email }	
TeamName<	user { team { name } }	
Note	user { note }	
IsOnAccount	user { isOnAccount }	
ContentType	N/A	
ContentId	course { id }	
ContentSlug	course { slug }	
ContentName	course { title }	
FirstActivityDate	firstViewedClipOn	
PercentComplete	percentComplete	
CompletionStatus	isCourseCompleted	
CompletionDate	completedOn	

For applicable filters for the Course Progress API, please see [Course progress filter \(opens in new tab\)](#).

Content Progress API

For applicable filters for the [Content Progress API \(opens in new tab\)](#), please see [Content progress filter \(opens in new tab\)](#).

REST API field	GraphQL API field	Sample code
----------------	-------------------	-------------

REST API field	GraphQL API field	Sample code
UserId	user { id }	<pre> { contentProgress { nodes { user { id firstName lastName email team { name } note isOnAccount } } course { id title slug } firstViewedClipOn percentComplete isCourseCompleted completedOn } } </pre>
FirstName	user { firstName }	
LastName	user { lastName }	
Email	user { email }	
TeamName	user { team { name } }	
Note	user { note }	
IsOnAccount	user { isOnAccount }	
ContentType	ContentProgressType { ContentType }	
ContentId	course { id }	
ContentSlug	course { slug }	
ContentName	course { title }	
FirstActivityDate	firstViewedClipOn	
PercentComplete	percentComplete	
CompletionStatus	isCourseCompleted	
CompletionDate	completedOn	

Course Daily Usage API

Use the [Course Daily Usage API \(opens in new tab\)](#) if you're looking to migrate from the [REST course usage API \(opens in new tab\)](#).

For applicable filters for the GraphQL Course Daily Usage API, please see [Course daily usage filter \(opens in new tab\)](#).

REST API field	GraphQL API field	Sample code
UserId	user { id }	<pre> { courseDailyUsage { nodes { user { id firstName lastName email team name } } note } } </pre>
FirstName	user { firstName }	
LastName	user { lastName }	
Email	user { email }	
TeamName	user { team { name } }	
Note	user { note }	
IsOnAccount	user { isOnAccount }	
ContentType	N/A	

REST API field	GraphQL API field	Sample code
ContentId	course { id }	<pre> isOnAccount } course { id slug title } rollupDate clipViewSeconds } } } </pre>
ContentSlug	course { slug }	
ContentName	course { title }	
ViewDate	rollupDate	
UsageInSeconds	clipViewSeconds	
CompletionStatus	isCourseCompleted	
CompletionDate	completedOn	

[back to top](#)

License management APIs

Member Invite API

Use the [Member Invite API \(opens in new tab\)](#) if you're looking to migrate from the [REST Get All Invites API \(opens in new tab\)](#).

Note: The REST API returns a note field by default. If you need the same field in GraphQL, call Users API and input the redeemedByUserId value in the filter as shown in the sample code.

REST API field	GraphQL API field	Sample code
None	inviteId	<pre> { memberInvites { nodes { inviteId planId sentToEmail teamId teamId sentOn expiresOn redeemedByUserId status } } } </pre>
-	planId	
-	sentToEmail	
-	teamId	
-	user { note }	
-	sentOn	
-	expiresOn	
Filters		
REST API filters	GraphQL API filters	<pre> { users(filter: {ids: "<Input redeemedByUserId value here>"}) { nodes { id </pre>
note	Not available; Pending enhancement	
email	sentToEmail	

REST API field	GraphQL API field	note	Sample code
teamId	Not available; Pending enhancement		<pre> } } } </pre>

The response fields available for this GraphQL API closely match the response fields from the REST API.

REST API field	GraphQL API field
id	inviteId
email	sentToEmail
teamId	teamId
note	user { note }
sendDate	sentOn
expiresOn	expiresOn

Invite Member API—Mutation or Write API

Use the [Invite Member API—Mutation or Write API \(opens in new tab\)](#) if you're looking to migrate from the [REST Invite new user API \(opens in new tab\)](#).

REST API field	GraphQL API field	Sample code
email	email	<pre> mutation { inviteMember(input: { email: "<insert email here>", teamId: <Insert teamId here>, note: "<Insert note here>") { inviteId email teamId note } } </pre>
teamId	teamId	
note	note	

Cancel Invite—Mutation or Write API

Use the [Cancel Invite API \(opens in new tab\)](#) if you're looking to migrate from the [REST Cancel invite API \(opens in new tab\)](#).

REST API field	GraphQL API field	Sample code
id	inviteId	<pre> mutation { cancelInvite(input: { inviteId: "<Insert InviteID here>") { inviteId } } </pre>

Users API

Use the [Users API \(opens in new tab\)](#) if you're looking to migrate from the [REST Get all Users API \(opens in new tab\)](#).

REST API field	GraphQL API field	Sample code
-	id	<pre>{ users { nodes { id email additionalEmails firstName lastName startedOn createdOn note teams { edges { node { id } } } isOnAccount removedOn currentSsoldentifier } } }</pre>
-	additionalEmails	
-	createdOn	
-	teams { id }	
-	firstName	
-	lastName	
-	email	
-	note	
-	startedOn	
-	isOnAccount	
-	removedOn	
-	currentSsoldentifier	
Filters		
REST API filters	GraphQL API filters	
firstName	Not available	
lastName	Not available	
email	emails	
note	notes	
teamId	Not available; Pending enhancement	
Not available	ssolidentifiers	
Not available	onlyCurrent	
	ids	

The response fields available for this GraphQL API closely match the response fields from the REST API.

REST API field	GraphQL API field
id	id

REST API field	GraphQL API field
teamId	teams { id }
firstName	firstName
lastName	lastName
email	email
note	note

Get user by ID API

Use the [Get user by ID API \(opens in new tab\)](#) if you're looking to migrate from the [REST Get user by ID API \(opens in new tab\)](#).

REST API field	GraphQL API field	Sample code
-	id	<pre>{ users (filter: {ids:"<Insert user ID here>"}) { nodes { id email additionalEmails firstName lastName startedOn createdOn note teams { edges { node { id } } } isOnAccount removedOn currentSsoldentifier } } }</pre>
-	additionalEmails	
-	createdOn	
-	teams { id }	
-	firstName	
-	lastName	
-	email	
-	note	
-	startedOn	
-	isOnAccount	
-	removedOn	
-	currentSsoldentifier	

The response fields available for this GraphQL API closely match the response fields from the REST API.

REST API field	GraphQL API field
id	id
teamId	teams { id }
firstName	firstName
lastName	lastName

REST API field	GraphQL API field
email	email
note	note

Edit user API—Mutation or Write API

Use the [Edit user API—Mutation or Write API \(opens in new tab\)](#) if you're looking to migrate from the [REST update user API \(opens in new tab\)](#).

REST API field	GraphQL API field	Sample code
teamId	Use a separate API moveMemberToTeam	<pre>mutation { editUser (input: { userId: "<insert user ID>", firstName: "<Insert first name here>", lastName: "<insert last name>", email: "<Insert email Id here>", remoteUserId: "<Insert remote user ID here>") { userId firstName lastName email remoteUserId } }</pre>
note	Not available	
email	email	
Not available	userId	
Not available	firstName	
Not available	lastName	
Not available	remoteUserId	

Remove user API—Mutation or Write API

Use the [Remove user API—Mutation or Write API \(opens in new tab\)](#) if you're looking to migrate from the [REST Remove user API \(opens in new tab\)](#).

REST API field	GraphQL API field	Sample code
id	userId	<pre>mutation { removeUser(input:{userId:"<Insert user id here>"}){ userId } }</pre>

Teams API

Use the [Teams API \(opens in new tab\)](#) if you're looking to migrate from the [REST Get all teams API \(opens in new tab\)](#).

REST API field	GraphQL API field	Sample code
id	id	{
name	name	
Not available	planId	

REST API field	GraphQL API field	Sample code
Not available	description	<pre>teams { nodes { id name planId groupId description } }</pre>
Not available	groupId	
Filters		
REST API filters	GraphQL API filters	
name	name	
Not available	description	

If you need help, please email support@pluralsight.com for 24/7 assistance.