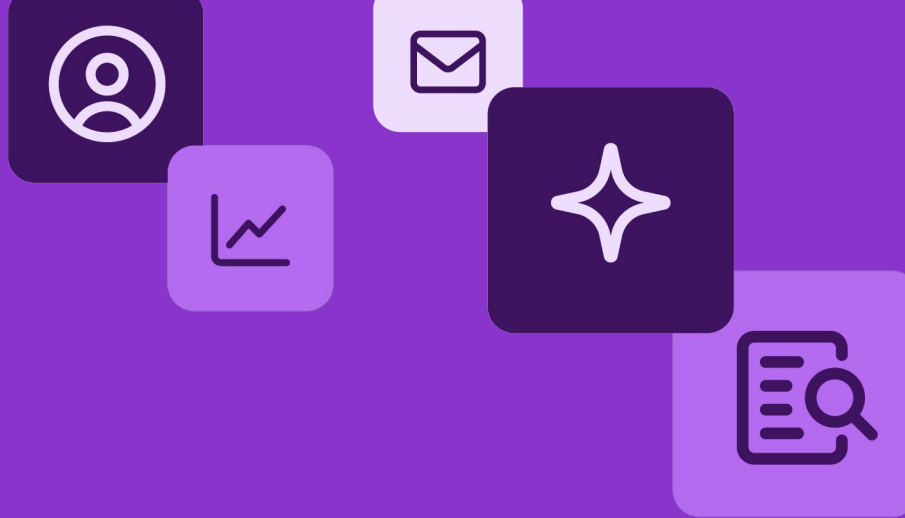




Developer Program Updates

What we shipped, what is changing, and how we are building with our community

Shruti Hulmani, Lead Product Manager – DECO

Agenda

Three connected workstreams driving the eBay Developer Program forward.

1

Developer Experience

Improving how developers discover, learn, build, and troubleshoot with eBay APIs.

2

Program Updates

Strategic updates, roadmap communication, and program governance.

3

Developer Outreach & Engagement

Understanding developer needs, gathering feedback, and strengthening community.

01

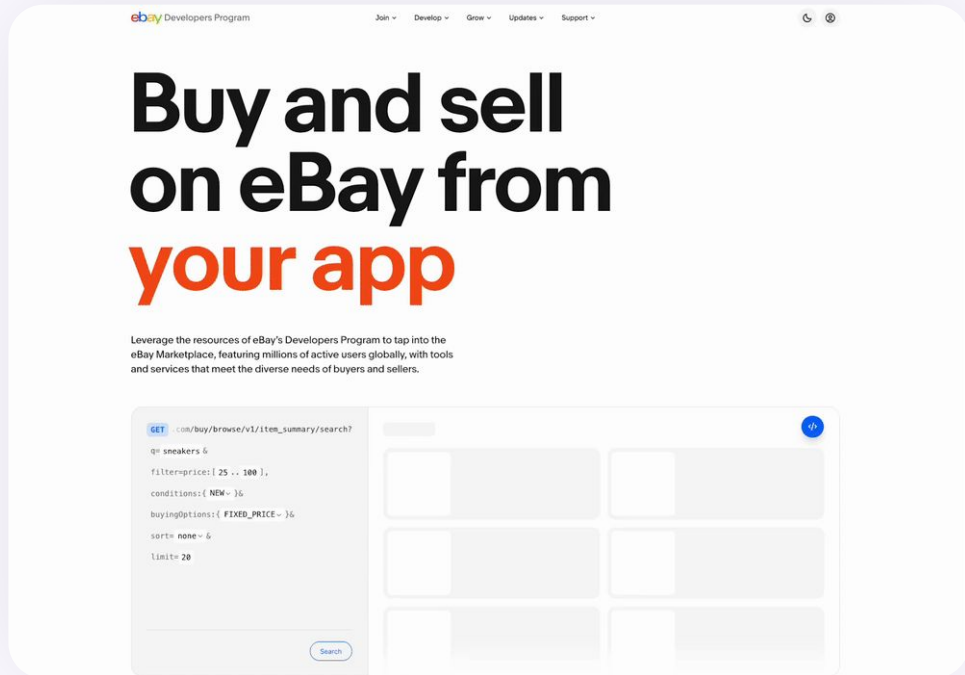
Developer Experience

Improving how developers discover, learn, build, and troubleshoot with eBay APIs.

Developer Portal

Reimagined look & feel

A simpler, smarter experience that helps you find APIs and resources faster – so users can build with confidence and quality.



Minimize clicks

Simplified navigation

Use-case clarity

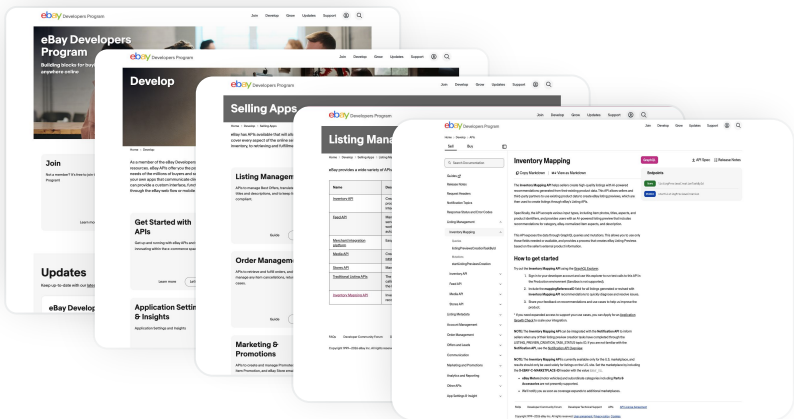
End-to-End guidance

UI Simplicity: From ~~5 clicks~~ to 3 clicks

✗ Old Flow

5 clicks

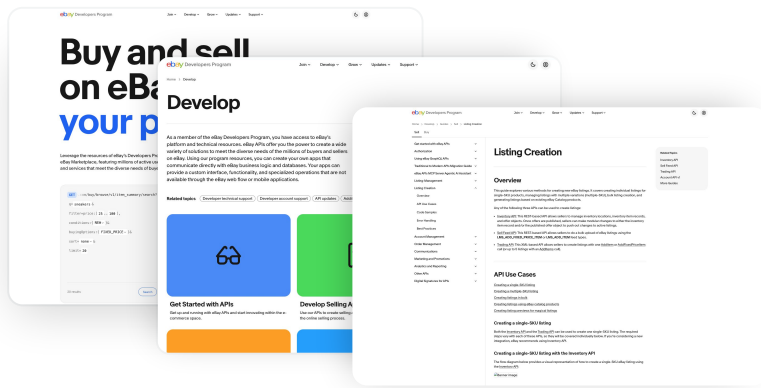
A tedious path spread out across multiple links and screens



✓ Updated Flow

3 Clicks

A simple path directly leading you to the screen you are looking for.



A simpler way to build Guides and APIs, together

Pick your journey – then navigate between Guides and APIs without losing your place.

Develop **Selling** Applications

Use our APIs to create selling applications tailored to each step of the selling process.

Guide

APIs



Develop **Buying** Applications

Create Buying apps with eBay APIs can enhance the shopping experience for users

Guide

APIs

Categorization built around Use cases

APIs

Technical reference and endpoints to power applications.

The screenshot shows the 'Browse API' page in the eBay Developers Program. The page is divided into a left sidebar with navigation links (Home, Develop, APIs, Sell, Buy, etc.), a main content area, and a right sidebar with 'Endpoints' and 'Release Notes'. The main content area is titled 'Browse API' and includes a 'Copy Markdown' button and a 'View as Markdown' link. Below this, there is a section titled 'The Browse API has the following resources:' which lists several resources, including 'item_summary' and 'item'. The 'item_summary' resource is described as allowing shoppers to search for specific items by keyword, GTIN, category, charity, product, image, or item aspects and refine the results by using filters, such as aspects, compatibility, and fields values, or UI parameters. The 'item' resource is described as allowing shoppers to retrieve the details of a specific item or all items in an item group, which is an item with variations such as color and size and check if a product is compatible with the specified item, such as if a specific car is compatible with a specific part. Below this, there is a section titled 'This resource also provides a bridge between the eBay legacy APIs, such as the Finding, and the RESTful APIs, which use different formats for the item IDs.' and a note that the 'item_summary', 'search_by_image', and 'item' resource calls require an Application access token. A 'Show more' button is located at the bottom of the main content area.

Find the right API and guides for your journey – whether building buyer or seller applications.

Guides

Step-by-step workflows and best practices for every use case.

The screenshot shows the 'Inventory Discovery and Refresh Guide' page in the eBay Developers Program. The page is divided into a left sidebar with navigation links (Home, Develop, APIs, Sell, Buy, etc.), a main content area, and a right sidebar with 'Related topics'. The main content area is titled 'Inventory Discovery and Refresh Guide' and includes a 'Copy Markdown' button and a 'View as Markdown' link. Below this, there is a section titled 'Overview' which describes the Inventory Discovery APIs as allowing users to discover eBay listings through query strings and other advanced search criteria and retrieve information about eBay listings. In addition, the Inventory Refresh APIs allow users to subscribe to push notifications that indicate when tracked listings have changed price, availability, or have been added or removed from a Promoted Listings campaign. Below the overview, there is a section titled 'API Use Cases' which lists several use cases: 'Discover and retrieve eBay items using the Browse API', 'Check product compatibility with a specified item', 'Downloading listings from a top-level (L1) category', and 'Subscribe to push notifications to track listings'. Below these use cases, there is a section titled 'Discover and retrieve eBay items using the Browse API' which describes the Browse API as allowing users to search for eBay items by various criteria, such as keywords, categories, or even images, as well as retrieve the details of a specific item or items. Below this, there is a section titled 'Below is the basic flow used to discover and retrieve items using the Browse API' which includes a diagram showing the flow from 'Discover items to search for the eBay items by keyword, category, product, image, or item aspects' to 'Use search or searchByImage to retrieve items based on query criteria' to 'Accept the filtered view of all items to which you have access' to 'Use one of the gotten methods to retrieve detailed information about the item'.

Integration workflow

Steps, samples, best practices

Everything you need to integrate – organized by use case with step-by-step workflows, samples, best practices.

Use cases

Explore common API use cases and find the right solution for our needs.

API Use Cases

[Creating a single-SKU listing](#)

[Creating a multiple-SKU listing](#)

[Creating listings in bulk](#)

[Creating listings using eBay catalog products](#)

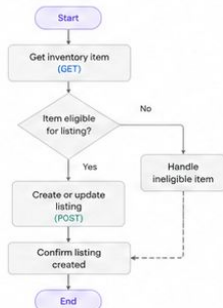
[Creating listing previews for magical listings](#)



Workflows

Visual workflows that show the end-to-end process at a glance.

Creating a single-SKU listing



Workflow steps

Detailed step-to-step instructions for each part of the workflow.

1. Create the location and location type using the `createInventoryLocation` method.
2. Define the item using the `createOrReplaceInventoryItem` method.
3. SKU: the seller defined stock keeping unit (sku) is passed in as a path parameter.
4. Title and description: the listing title and listing description are passed in under the product container.
5. Product aspects: the product aspects (aka item specifics) of the item are passed in as name-value pairs.

[View all steps →](#)

Samples

Real API request and response examples to help you get started fast.

Request Response Copy

```
{  "itemsBySearch": {    "items": [      {        "itemId": "v1|36654212112|...",        "title": "NEW IN BOX! Men's Brooks Adrenaline GTS 25",        "price": {          "value": "78.00",          "currency": "USD"        }      }    ]  }}
```

[See full sample →](#)

Best practices

Proven tips and recommendations for successful integrations

Best Practices

- Use a location object, an inventory item group, and an offer object before creating a new listing.
- Validate item specifics before publishing.
- Use appropriate categories and condition values.
- Test in Sandbox before going live.

[View all best practices →](#)

Developer Tools Reimagined around AI

Tools that provide everything a developer is looking for — with AI at the center.

AI-powered tools & support

Get instant answers from our AI Chatbot.
Free support when self-serve isn't enough.
Managed support for high-touch partners.

My Tickets **+ AI-Assisted Support** Application Growth Check

 **How can I help you today?**

Submit

AI-assisted support is an experiment that may return inaccurate or inappropriate responses. View our [User Agreement](#) and [User Privacy Notice](#). Do not share personal information.

AI in your environment

AI tools that integrate directly with the developer's environment for easy discoverability.

MD Content for AI-native consumption

MEDIA.md

 Copy Markdown |  View as Markdown

Guides: Listing creation

LISTING_CREATION.md

 Copy Markdown |  View as Markdown

MCP – discover and call GET operations & queries

GET /sell/inventory/v1/item/{item_id} **Try it** 

What's coming next?

Agent Skills

Complete your migration with the help of agent skills that help with discovery and mapping of API operations.

SDKs by use case

SDKs organized around use case – with detailed schemas, OAS files, client code for each one.

ebay Developers Program

Home > Develop > SDKs and Widgets

SDKs and Widgets

SDKs

ebay provides the following SDKs to make it easier for you to integrate with eBay's APIs.

SDK	Supported APIs	Description
Digital Signature SDK (Client)	Key Management	Use this eBay Open Source Java SDK to simplify the process of generating digital signature headers and also provide a method to validate the digital signature headers.
Digital Signature SDK (PHP/JS)	Key Management	Use this eBay Open Source Node.js SDK to simplify the process of generating digital signature headers and also provide a method to validate the digital signature headers.
Event Notification SDK (Python)	Event Notifications	Use this eBay Open Source Python SDK for the latest eBay notifications that use ECC signatures and JSON payloads.
Event Notification SDK (Client)	Event Notifications	Use this eBay Open Source Java SDK for the latest eBay notifications that use ECC signatures and JSON payloads.
Event Notification SDK (PHP)	Event Notifications	Use this eBay Open Source PHP SDK for the latest eBay notifications that use ECC signatures and JSON payloads.
Event Notification SDK (NET)	Event Notifications	Use this eBay Open Source .NET SDK for the latest eBay notifications that use ECC signatures and JSON payloads.
Event Notification SDK (C#)	Event Notifications	Use this eBay Open Source C# SDK for the latest eBay notifications that use ECC signatures and JSON payloads.

Developer Tools

Tools that help you test from the Developer Portal

API Explorer

Build a request against the eBay REST APIs, attach an OAuth token, and inspect the live response.

The API Explorer interface shows a GET request to the eBay REST API. The environment is set to Production with Keyset1. The App ID / Client ID is TWL397NL-test-PRD-38f5e75c8-Sac88864. The request is a GET to https://api.ebay.com/buy/browse/v1/item_summary/search?gdrnd&limit=3&offset=0. The response is a 200 OK with a time of 728 ms. The response body is a JSON object containing a list of items. The interface includes a left sidebar with filters for Site ID, API, Operation, Parameters, Query, and Samples. The main area shows the request details, parameters, and the response body.

Upcoming

GraphQL Explorer

The GraphQL Explorer interface shows a mutation request. The environment is set to Production with Keyset1. The App ID / Client ID is ABC39CBA-keyset-PRD-b8c6188ba-12c8ca3. The request is a mutation to startListingPreviewsCreation. The response is a JSON object containing the result of the mutation. The interface includes a left sidebar with filters for Environment, App ID / Client ID, and OAuth tokens. The main area shows the request details, variables, and the response body.

02

Program Updates

Strategic updates, roadmap communication, and program governance.

Sell API Updates

APIs updates to support every eBay-wide initiative — so sellers and buyers always get the latest capabilities.

Platform parity and Seller Capabilities

- Inventory Mapping API now open to all developers
- Refurbished Program expanded to 2200+ leaf categories
- Coin Condition Grading across Coins & Salvage
- Buyer Protection fees enablement across UK and AU markets

Motors, Marketing APIs

- Ads campaigns: The campaign budget minimum requirement was changed for PLP campaigns
- New annotation keys are available in ad report tasks allowing access to more and certain data points based on the dimensions.
- Multi-warehouse Support in Fulfillment and Inventory APIs

Compliance and Regulatory

- Energy labeling, product aspects, Common Charger Directive (CDD), materials sourcing, market-specific shipping rules, and new financial transaction types for regulatory.

Modernization

- New Metadata Shipping Methods have been released
- New Messaging and Feedback APIs released
- New webhook Notifications for various flows

Buy, Developer API and License Updates

APIs updates to support every eBay-wide initiative — so sellers and buyers always get the latest capabilities.

Buy / Developer API updates

- Buy Marketing API: getMostWatchedItems & getSimilarItems
- New Messaging and Feedback API
- New Notification events: Offer, Auction and Feedback
- New CBT feeds — inventory from bigger markets (US, UK, DE) available to smaller markets (e.g. AT)
- Geo expansion of the PL market for affiliates
- Geo expansions for Order APIs to support more markets: IT and corridors to enable markets
- Charity details included in the Browse getItem family

Developer API updates

- Developer Analytics API now clearly returns the count of the calls made

API license updates

- Data Protection
 - **Restricted APIs** — with tighter usage boundaries
 - **AI content usage** — new restrictions you need to review if you're using API data returned by AI in your workflows
- Data Transfers

Addressing critical feature gaps to provide seamless execution of flows and a better user experience — making sure sellers and buyers get the latest capabilities.

API Deprecations & Decommissions

As part of modernization and the general API lifecycle, the following operations will be decommissioned. See the API deprecation status page for the full Post-Order API list.

API	Operation(s) / Field(s)	Date	Replacement(s)
Product API	All methods	2026/08/15	The compatibilities methods of the Metadata API.
Product API	getProductCompatibilities	2026/08/15	getProductCompatibilities method of the Metadata API.
Trading API	UploadSiteHostedPictures	2026/09/30	Redundant — triggered when a seller manually marks an item as paid.
Trading API	GetShippingDiscountProfiles, SetShippingDiscountProfiles	2027/01/19	Migrate to the Combined Shipping Rules methods of the Account API v2.
Trading API	GetCategoryFeatures	TBD	Multiple methods of the Metadata API. See the GetCategoryFeatures migration guide.

<https://developer.ebay.com/develop/get-started/api-deprecation-status>

03

Developer Outreach & Engagement

Understanding developer needs, gathering feedback, and strengthening community relationships.

Developer Surveys & Insights

Announcing the annual survey — our most important tool for improving experience and building the roadmap.

Annual survey

Super important for helping us improve the experience and build the product roadmap.

Mid-Aug

Next annual survey

Feedback & inputs

- Feedback on all pages — understanding content and experience issues
- Dedicated feedback on Sandbox pages for Sandbox-specific issues



Community Outreach & Communication

From precise, targeted emails to broad newsletters and release communications.

Targeted emails

For specific mandates and issues that impact specific developers, so they take action — which is why having the right emails on file is critical.

Keep your email address up to date

Broader strategy

Blast emails to all subscribers:

- newsletters
- API license updates

Subscribe to receive these emails

Developer Programs & Engagement

The surfaces we use to reach developers and go deeper on what matters.



Release Notes

Detailed release updates

Inventory Mapping API

Release Version	Release Date	Description of Release
1.0.0	2025-08-11	The initial version of the Inventory Mapping API

[← Back To Inventory Mapping API Docs](#)

[Show more](#)

Charity API

Release Version	Release Date	Description of Release
1.2.1	2023-01-16	The <code>getCharityOrgByLegacyId</code> method has been deprecated and removed.
1.2.0	2020-08-14	Added postal code and country to the Address container of the Charity API.
1.1.0	2020-05-15	Added a new <code>getCharityOrgByLegacyId</code> resource method.
1.0.0	2020-04-20	The initial version of the Charity API.

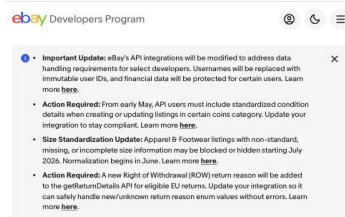
[← Back To Charity API Docs](#)

[Show more](#)



Banners

For important updates that need visibility.



Blog Posts & Landing Pages

New features and updates that need more information.

Blog

Size Standardization for eBay Fashion Listings

We're introducing size standardization for Apparel & Footwear listings on eBay – a step toward improving listing quality, buyer trust...

Explore the Inventory Mapping API

The Inventory Mapping API is now available for developer testing.

Open-Source SDK for Taxonomy Metadata

eBay's Category Aspects metadata, which has given definition to our marketplace listings, is evolving at a fast pace. With these rapid...



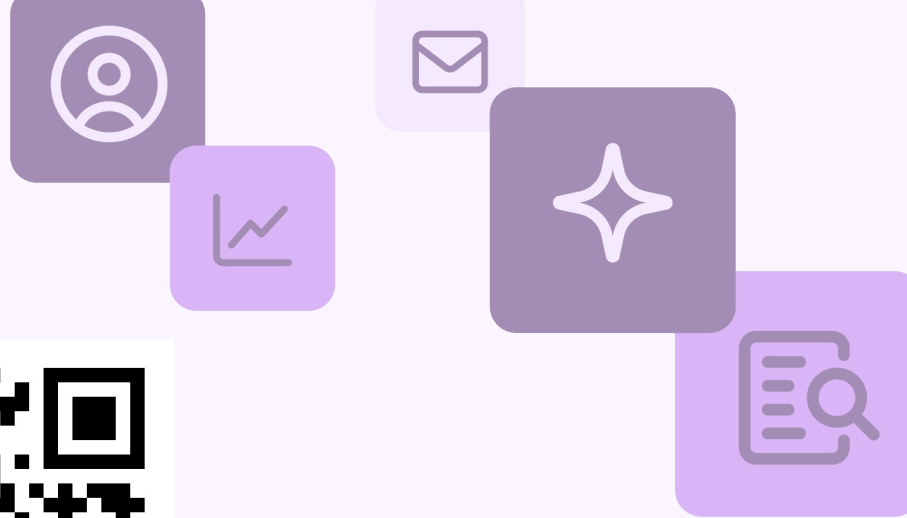
Future Roadmap

RSS / webhook subscriptions

- Release Notes
- Blog Posts

Regular Sync-up with interested developers

Thank You





New GraphQL APIs

Ramesh Ramachandran, Lead Technical Architect – DECO

Why Migrate?

eBay is modernizing its entire platform API surface

3

Legacy API Families
Being Replaced

2

New Tech Stacks
REST + GraphQL

7+

New API
Families

100%

Sell + Buy Flows
Covered

What's Going Away

- Trading API (SOA / XML-based)
- Post-Order API
- Legacy Platform Notifications

What's Replacing It

- GraphQL APIs (new, purpose-built)
- REST APIs (expanded coverage)
- Modern event/notification model

Key Benefits

- Request only the data you need
- Consistent OAuth 2.0 auth
- Better developer experience

GraphQL Operations: 91 (34 queries, 57 mutations)

Listing Management APIs

Traditional AddItem/Reviseltem/EndItem family → Listing API

Order Management APIs

GetOrders / GetSellerTransactions → Order API + Item Commitment API

Key Concept:

Orders are now split into ORDERS (paid) and ITEM COMMITMENTS (unpaid). TransactionID is replaced by orderLineItemId.

Post-Order: Cancellations

Post-Order Cancellation API → Cancellation API

Buyer-facing operations NOT migrated.

Post-Order: Order Inquiries (INR)

Item Not Received inquiries & cases → Order Inquiry API

Post-Order: Returns

Post-Order Return API → Return API

Separate Case entity eliminated.

Why GraphQL ?

No Over-fetching

Clients request only specific fields. This prevents downloading unnecessary data and saves network bandwidth.

No Under-fetching

Clients request nested, related resources simultaneously, eliminating the need for multiple successive API calls.

Single Endpoint

All operations run through one URL (via HTTP POST). This simplifies backend routing and client configuration.

Strongly-Typed Schema

A strict schema acts as a contract between client and server, catching data errors early during development.

Built-in Introspection

The API is inherently self-documenting. Developers can query the schema directly to discover available fields.

Smooth API Evolution

Fields can be added or deprecated gradually, eliminating the need for strict API versioning like /v1/ or /v2/.

Native Real-Time Support

It features built-in "Subscriptions" which allow out-of-the-box data streaming over WebSockets.

GraphQL Explorer Demo

The screenshot displays the GraphQL Explorer interface on the eBay Developers Program website. The browser address bar shows the URL: `developer.ebay.com/my/graphql_explorer?index=0&env=production`. The page header includes the eBay logo and navigation links: Join, Develop, Grow, Updates, Support, and Hi rameshpoti. The main heading is "GraphQL Explorer".

Below the heading, the environment is set to "Production" (indicated by a radio button). The selected environment is "ramesh-test". The App ID/Client ID is "rameshra-rameshte-PRD-02feb288b-e4e501f9 (OAuth Scopes)". The OAuth Application Token and OAuth User Token are also displayed.

The interface is divided into three main sections:

- Explorer:** A list of available queries, including `query SellerListingsItemTitle`, `query BuyerNegotiationHistory`, `query ContractCancellation`, `query ItemBiddingActivity`, `query ItemCommitments`, `query ItemCommitmentsByIds`, `query ListingPreviewsCreationTaskById`, `query OrderInquiry`, `query OrderLineItemsByIds`, `query OrderReturn`, `query Orders`, `query OrdersByIds`, `query PurchaseQuoteLineItemsByIds`, `query PurchaseQuotes`, `query PurchaseQuotesByIds`, `query ReturnRuleListingGroup`, `query SellerContractCancellations`, `query SellerListingSearch`, and `query SellerListings`. The `query SellerListings` is selected.
- Query Editor:** A text area for writing the GraphQL query. The query is:

```
1 query SellerListingsItemTitle {
2   sellerListings(input: {listings: {listin
3     ... on ListingsSuccess {
4       __typename
5       listings {
6         ... on ListingSuccess {
7           __typename
8           listing {
9             listingId
10            sellerProduct {
11              title
12            }
13          }
14        }
15      }
16    }
17  }
18 }
```
- Variables:** A section for defining variables. The variables are:

```
1 {
2   "Authorization": "Bearer v^1.1#1^1#1^3#r^0#f^0
3 }
```
- Result:** The JSON response for the query. The result is:

```
{
  "data": {
    "sellerListings": {
      "__typename": "ListingsSuccess",
      "listings": [
        {
          "__typename": "ListingSuccess",
          "listing": {
            "listingId": "127879680361",
            "sellerProduct": {
              "title": "New Balance MF791MM Men's Mid
              Top Lace-Up Plastic Cleats Sneaker White/Silver"
            }
          }
        }
      ]
    }
  }
}
```



Thank you

Developer Programs & Engagement

The surfaces we use to reach developers and go deeper on what matters.

Release Notes

Detailed release updates

Upcoming

RSS / webhook subscriptions

Banners

For important updates that need visibility.

Blog posts

About new features and updates that need more information.

Upcoming

RSS / webhook subscriptions

Landing pages

Standalone pages linked from banners and blogs.