

Implementing Algolia click-and-convert events for e-commerce



Implementing Algolia click-and-convert events for e-commerce

If you are a current customer of Algolia and you are not intimately familiar with the Algolia Analytics Dashboard, I guarantee you are leaving money on the table and customers with an experience with your brand that could be improved.

Algolia is a powerful search tool with highly advanced AI capabilities to provide customers with personalized experiences. It also provides site operators with the information they need to make decisions about the configuration, performance, and effectiveness of the platform. At the core of these powerful capabilities are click-and-convert Events which inform the AI tools and their human operators about how the platform is being used so that machines and people can make informed decisions about how to tune the discovery experience.



1. Dynamic Re-Ranking

These events allow Algolia to understand user interactions with search results. By tracking clicks, Algolia can learn which results are most relevant and adjust ranking based on actual user behavior.



2. Personalization

Conversion events, such as purchases or sign-ups, help Algolia fine-tune personalized recommendations. This leads to more accurate and tailored search results for each user based on what actions they take after searching.



3. Insights and analytics

Monitoring click and conversion data provides valuable insights into search performance. It shows which queries lead to engagement, and which don't, allowing data-driven decisions for search optimization.



4. A/B testing & search optimization

Click and conversion data are essential for running A/B tests. It helps in understanding which search configurations perform better, driving better user engagement and sales.



5. ROI measurement

By tracking conversions, you can tie search improvements directly to business outcomes, ensuring that search optimizations result in increased revenue or other key performance metrics.

By implementing click-and-convert events correctly, you can get the most out of the platform and drive increased conversions for your business.



InstantSearch default events and event validation

If you begin your implementation from the Algolia-provided Search UI libraries (InstantSearch), you will have inherited a few events by default. The functionality that tracks click ("Filter Applied", "Hit Clicked") and view ("Hits Viewed") events are included in the library.

[View Algolia documentation for default events](#)

To enable automatic event collection, you will need to either enable collection in the Algolia Dashboard or enable the insights option in your InstantSearch code.

[View Algolia enable events collection documentation](#)

If you are not leveraging Algolia Search UI libraries, it is up to you to implement the "search-insights" library yourself.

[View Algolia API documentation for search-insights](#)

Validate events are being collected

Ensuring the events are correctly implemented is a multi-step process. You can validate that your events are arriving by going to the [Algolia Events](#) Hub. By clicking on the "Debugger" tab, you can view events as they are sent.

Each event in the dashboard contains details about the content of the message that is received including the validity of the data for each feature in the platform, HTTP Data, the events that have been sent for this user in a timeline view, and details about the query if the query ID is included.

In your development or test environment, navigate through the site and monitor the debugger to ensure the correct events are being sent to Algolia with the correct data.

In addition to Algolia's logging tools, you can use your browser's developer tools to monitor the network requests sending data to the Insights API. Open the Network tab in your browser's developer tools and apply a filter with the string "insights" to display the requests that include Insights data.

Enable query ID tracking

You will also want to track the Query ID parameter as the user navigates across the site. Without this data, Algolia cannot effectively provide usage analytics or accurately provide re-ranking, personalization, and recommendations.

The HTTP client provides two different methods for events, one for actions that were performed after a query and one for events that are not associated with a query. Both events are important for analytics and Algolia functionality.

If your category listing page is driven by Algolia, you should still send the query ID for click and conversion events even if the page was not rendered as the result of a keyword search. You can obtain the Query ID by first ensuring your query includes setting the "clickAnalytics" parameter set to "true." This ensures the "queryID" parameter will be returned in the response and available to include in events.

It is recommended that you implement this tracking as a query parameter in the URL, though it is possible to leverage cookies or local storage. Be aware that this may introduce corner cases that may happen when the user has multiple browsers/windows open or other atypical use cases.

If you do choose not to use a query parameter, be sure to clear the QueryID when navigating away from the Product Listing / Search Results page so that you do not attribute unrelated queries to subsequent events.

When products are added to the cart, you should also store the query ID in the cart so that when the user checks out you can associate that query with the purchase. This is important for tracking and attributing revenue to search queries.

Each line item should be mapped to the queryID that resulted in the add-to-cart action, as a cart with multiple line items likely was built as the result of multiple queries. Depending on the commerce platform you are implementing Algolia for, you may have access to store this data on the line item itself, or you may need to manage an external map of line items to query ID's.

[View Algolia keeping track of query IDs documentation](#)

[View Algolia revenue transactions documentation](#)

Enable autocomplete events

If you have implemented Autocomplete in the search bar, you may also want to include the search events, clicks, and conversions.

Enabling events for Autocomplete is straightforward if you are using the Algolia autocomplete-js package. You can simply set the "insights" parameter to true.

[View Algolia insights events](#)

Enable events for AI personalization

For the AI Personalization feature of Algolia to work optimally, you will want to enable a few more View events.

You will want to enable events for:

- Product Viewed on Category Listing Page (Enabled by default with "insights: true")
- Product Viewed in Search Results (Enabled by default with "insights: true")
- Product Viewed on Product Detail Page

[View Algolia personalization configure events documentation](#)

Enable session tracking for personalization

To maximize the value of personalization, you should track users across sessions. The most efficient way to do this is to leverage your authentication platform or e-commerce platform to use the system-provided ID as a user token when sending events.

By default, Algolia's libraries set a cookie in the user's session with a generated token. By replacing this value with a token from your authentication system, you may be able to track authenticated customers across sessions and devices. Many platforms also maintain tokens for anonymous users across sessions providing better personalization for anonymous users.

[View Algolia set up personalization documentation](#)

Enable conversion events for e-commerce

Next, you will want to enable conversion events. The two primary events to track are cart additions and checkout.

Add-to-cart events should be sent from any location that a user may add to the cart. That may include the product detail page, but it is also common for cart additions to happen on

the product listing page, search results, autocomplete, or any product spotlight like recommendations or home page carousels.

You will also want to be aware of any query ID that is associated with the add-to-cart action. Similar to the implementation of add to cart in the product detail page, be sure to use the appropriate method so that the action is attributed correctly.

Re-test

After your implementation, be sure to retest all events and ensure that the correct data is arriving in Algolia. Leverage the debugger in your lower environment to ensure each action is met with the correct event and that the data sent meets your expectations.

First, ensure that the Algolia Object ID is being sent, not the product's SKU ID, product ID, or other business identifier. In some cases, the Algolia record may have been inserted with one of these business identifiers as the Object ID, but it is not advisable to rely on using a field other than the Object ID returned by Algolia as implementations may change and cause unintended side-effects.

Also, check the analytics dashboards and ensure that the data being provided matches your expectations. Leverage the Events Glossary in this document to perform a page-by-page audit of the implementation to ensure all events are in place and working as expected.

You can refer to the Algolia documentation on Object ID's here:

[View added to cart object IDs after search documentation](#)

In production

Once your implementation is in production, it will not take long for analytics data to start to appear. You can check in Data Sources -> Events Health to see if you are sending the correct data for each Algolia feature to function properly. In some cases, such as Dynamic Re-Ranking, it may take several days or weeks for the system to collect enough data for the solution to work properly.

Keep in mind that there are a minimum number of events that must be present in Algolia for various features of the platform for those features to be active. The more events sent, the more effective these features are. You can view the thresholds and timeframes in the Algolia documentation located here:

[View Algolia minimum number of events documentation](#)

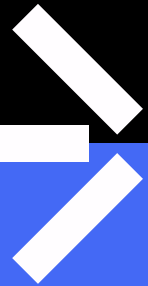
Conclusion

By following this guide, you can ensure that you have correctly implemented Algolia's Click and Conversion events, and therefore you can take full advantage of the powerful analytics and AI features of the platform. In our next guide, Icreon will explore how to maximize these capabilities and how to leverage analytics to make decisions and measure the effectiveness of your changes.

For additional assistance with Algolia Click-and-Conversion Events, please contact Icreon.



About Icreon



Icreon is a full-service digital experience consultancy that provides digital strategy and engineering at scale, with deep expertise in commerce, content, and cloud technologies. For over 24 years, we've partnered with leading Fortune 500, retailers and consumer goods companies to maximize business value by mapping customer lifecycles, designing optimal digital experiences, and powering them with composable content, commerce, and cloud solutions.

Headquartered in New York City, Icreon operates across 9 global offices, partnering with top brands like PepsiCo, Army Air Force Exchange Store, Ferrari, Slumberland, New York Road Runners, and Lincoln to fulfill their digital transformation needs. With 500+ experts in engineering, digital strategy, and UX design, Icreon is the preferred digital velocity partner for growth brands aiming to excel in a competitive marketplace.

[Learn more at Icreon.com](https://www.icreon.com)

Event Glossary

[Algolia insights API documentation](#)

Autocomplete

Event type	Event name	API function	Included In Instant search
Product Clicked	clickedObjectIDs	addedToCartObjectIDsAfterSearch	Y
Conversion	Product added to cart	convertedObjectIDs	N

Category listing

Event type	Event name	API function	Included In Instant search
Click	Product clicked	clickedObjectIDs	Y
Click	Product clicked after search	clickedObjectIDsAfterSearch	Y
Click	Clicked filter	clickedFilters	Y
View	Product viewed	viewedObjectIDs	N
View	Filter viewed	viewedFilters	N
Conversion	Product added to cart	convertedObjectIDs	N



Search results

Event type	Event name	API function	Included In Instant search
Click	Product clicked	clickedObjectIDs	Y
Click	Product clicked after search	clickedObjectIDsAfterSearch	Y
Click	Clicked filter	clickedFilters	Y
View	Product viewed	viewedObjectIDs	N
View	Filter viewed	viewedFilters	N
Conversion	Product added to cart	convertedObjectIDs	N
Conversion	Product added to cart after search	addedToCartObjectIDsAfterSearch	N

Product detail

Event type	Event name	API function	Included In Instant search
View	Product viewed	viewedObjectIDs	N
Conversion	Product added to cart	addedToCartObjectIDs	N
Conversion	Product added to cart after search	addedToCartObjectIDsAfterSearch	N

Order confirmation

Event type	Event name	API function	Included In Instant search
View	Product Purchased	purchasedObjectIDs	N
Conversion	Product Purchased After Search	purchasedObjectIDsAfter Search	N