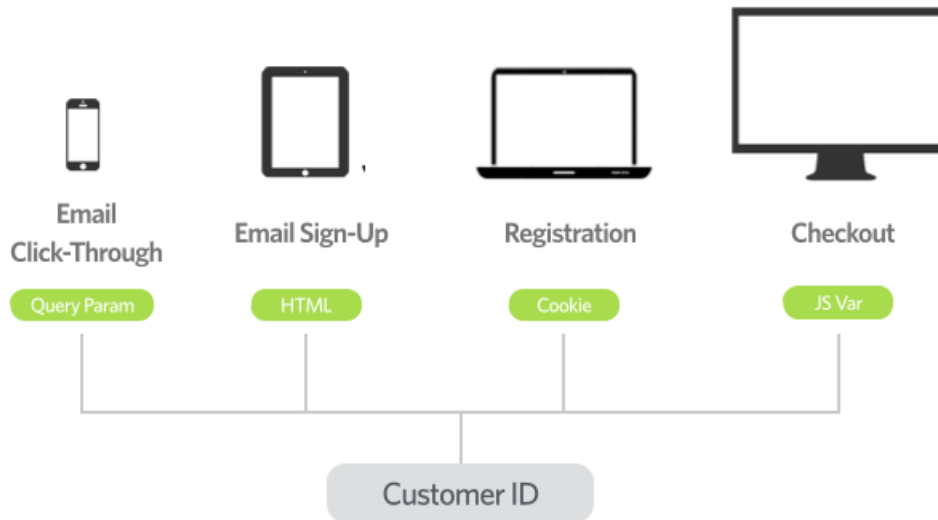


Client-Side ID Collectors

The first component you need to establish within a Customer View is the Person ID. The Person ID is the unique identifier your business associates with each unique customer across channels and devices.



The ID that you use in a [customer dataset](#) must match the case of the value collected in an on-site ID.

Tips

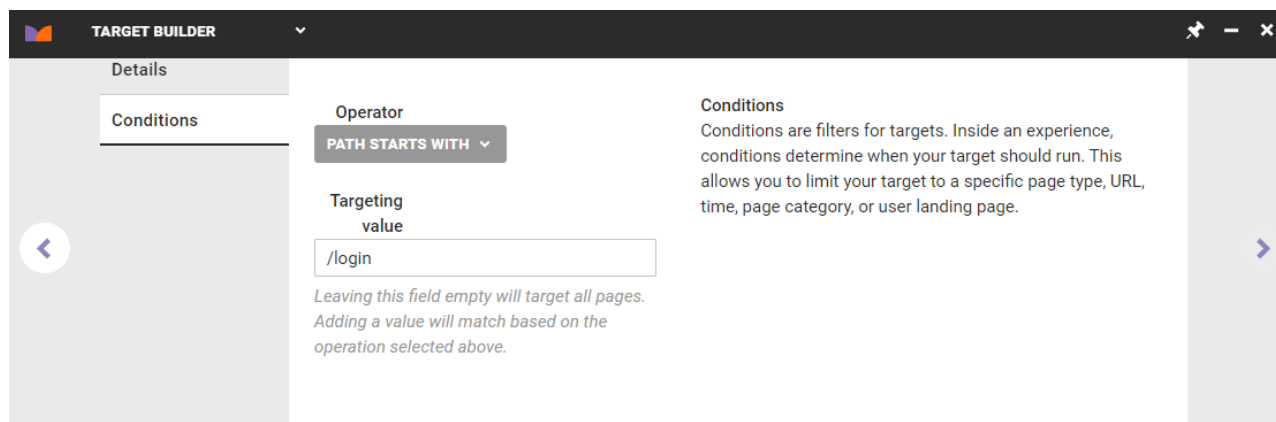
Here are some best practices to keep in mind.

Use Person-Based ID Collectors

For Monetate to effectively build a Customer View, your ID Collectors must discover Person IDs that are tied to an actual person and not to a single browser or device combination.

Use Static ID Collectors

Avoid situations in which the ID Collector collects both Person IDs and anonymous IDs. For example, if your site uses a cookie to set an anonymous ID for every customer who arrives on site but then changes that value to a Person ID when a customer logs in, you should only collect the cookie value tied to the login since it contains the Person ID. To do this, select **Path starts with** from **Operator** on the **Conditions** tab of Target Builder to only track the cookie on the login page.



See [Create Targets for Web Experiences](#) for more information on using Target Builder.

Finding ID Collectors

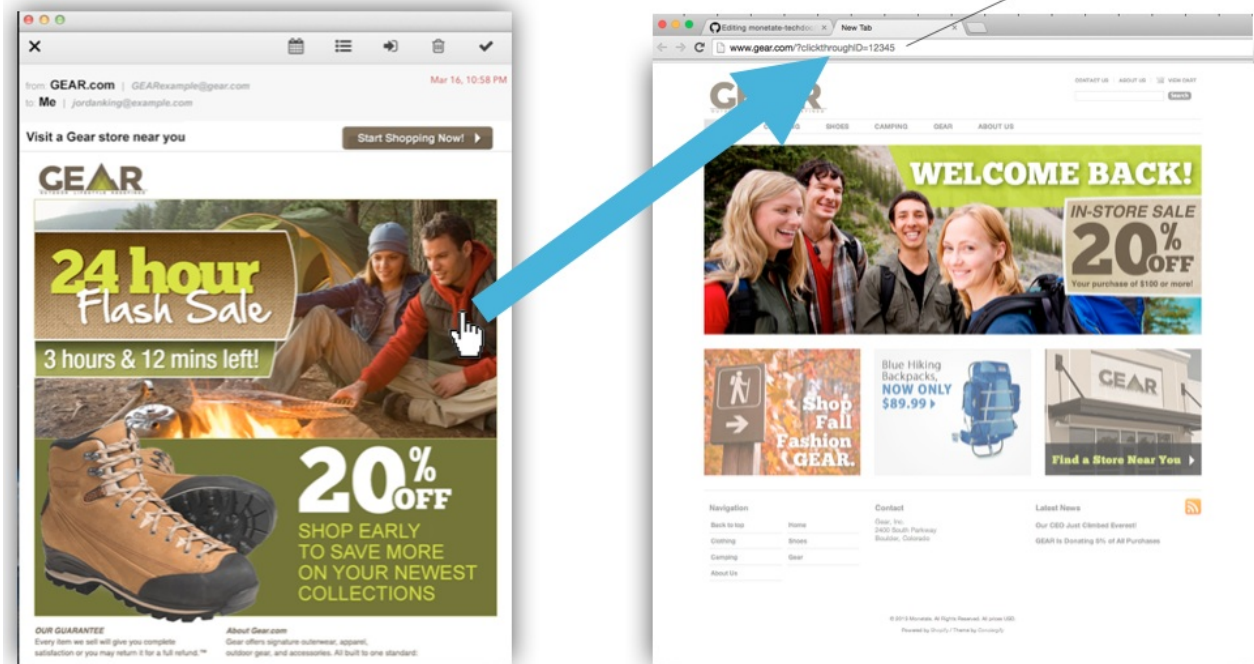
There are four common places to find ID Collectors.

From an Email Click-Through

How it works: You send a customer an email through your email service provider (ESP) that contains a link or linked image. That link directs the customer to a page on your site. The URL that displays when they arrive on the site contains a query parameter with an identifier set by your ESP. The query parameter value contained in this identifier is tied directly to a person, and you can use it as a Person ID that can be gathered by the query parameter ID Collector.

Identifier type: Query parameter

www.gear.com/?clickthroughID=12345



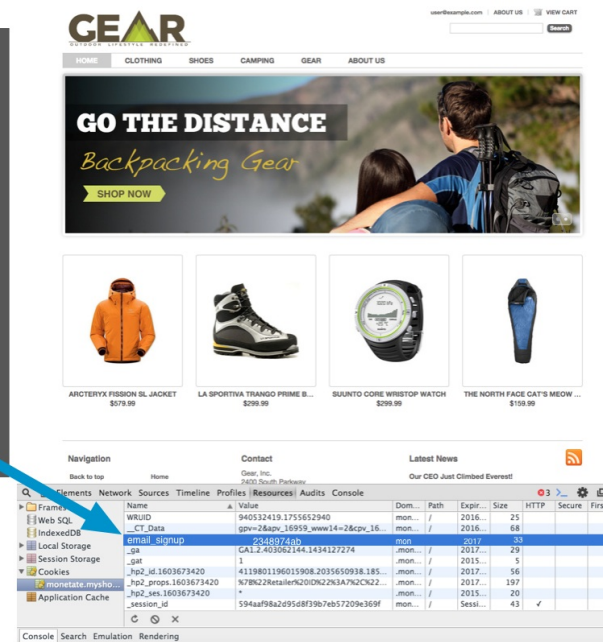
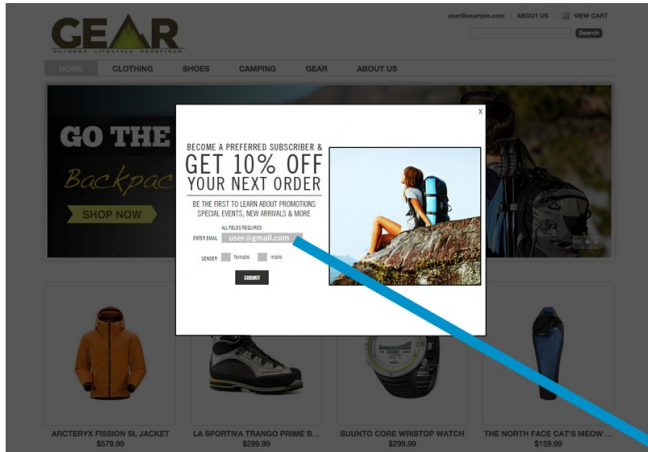
As shown in this screenshot, the query parameter value in the URL is `12345`.

From an Email Sign-Up

How it works: Your site has an email sign-up link so that customers can learn about upcoming sales and promotions. When a customer signs up, they're tracked with a unique identifier. This identifier value is tied directly to a person, so you can use it as a Person ID that can be gathered by the JavaScript variable or cookie-based ID Collector.

Identifier type: JavaScript variable or a cookie triggered on sign-up

Choose whichever target type applies to your site.

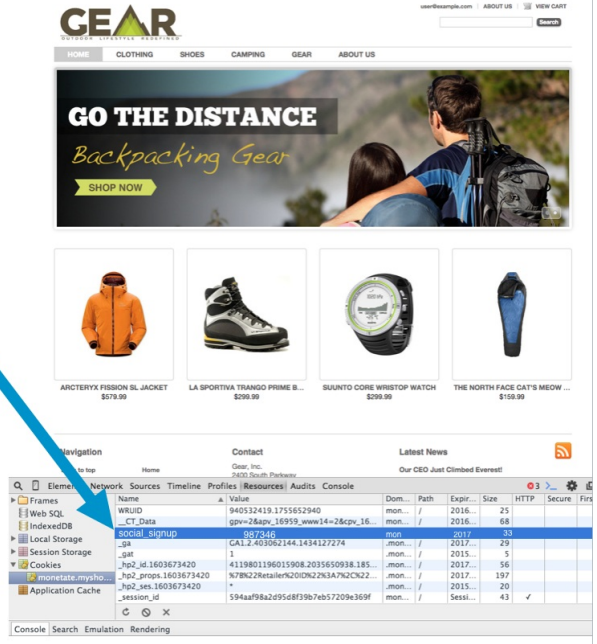
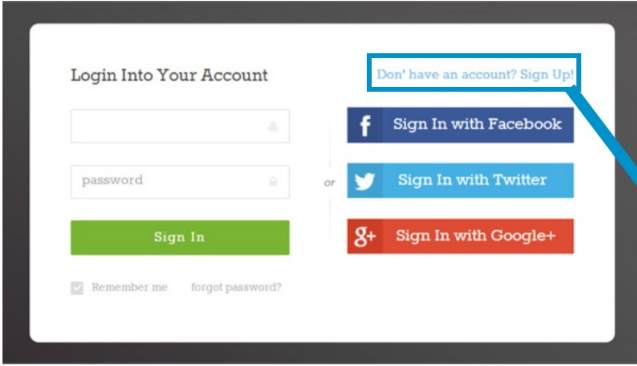


This screenshot of a browser's developers console shows the presence of an `email_signup` cookie on an example retail website.

From a Registration

How it works: Your site has a login page for returning customers to see their wish list or order history. When they log in they're identified by a unique identifier. This identifier value is tied directly to a person, and you can use it as a Person ID that can be gathered by the JavaScript variable or cookie-based ID Collector.

Identifier type: JavaScript variable or a cookie triggered on login

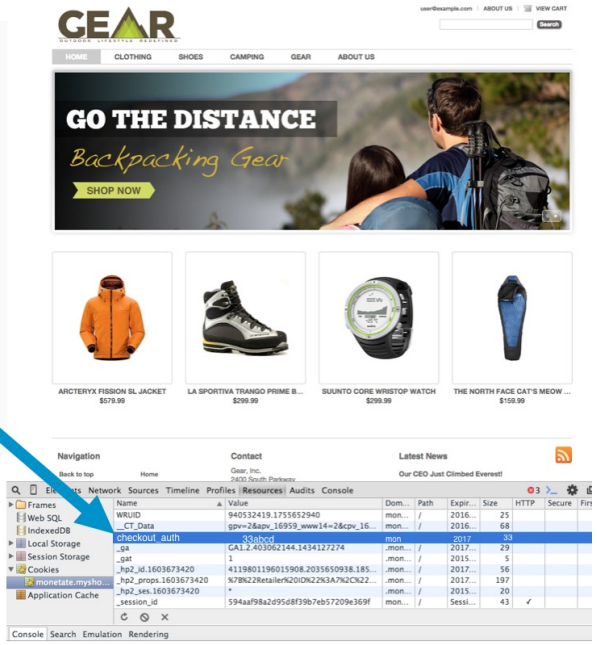
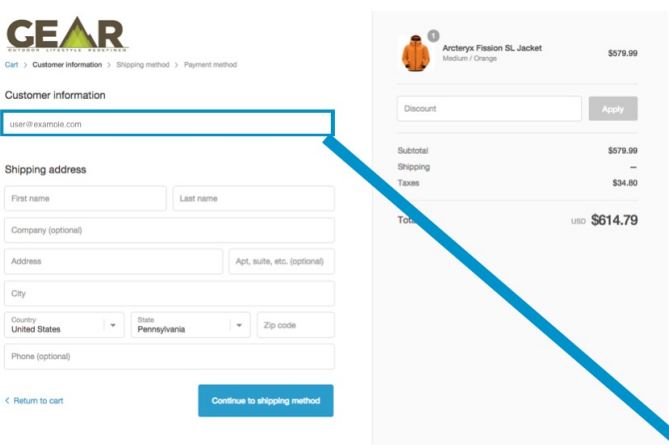


As shown in this screenshot, a website's customer login screen is tied to a `social_signup` cookie.

From a Checkout Authentication

How it works: You require customers to add an email address to purchase an item from your site. When a customer submits their email address, they're assigned a unique identifier. This identifier value is tied directly to a person, and you can use it as a Person ID that can be gathered by the JavaScript variable or cookie-based ID Collector.

Identifier type: JavaScript variable or a cookie triggered when the site visitor submits their email address



This screenshot shows the `checkout_auth` cookie that is present on the Customer Information panel of the checkout process for an online retailer.

Understand that you're not limited to using just one type of ID Collectors. Using a combination of ID Collectors, such as an email click-through and a login, to recognize the maximum number of customers possible is not uncommon.