Integration: Client Side vs Server Side Integration

Overview

There are currently two different ways to integrate RichRelevance products into your site; Client Side or Server Side. Both options have benefits and drawbacks.

Intended Audience

This topic is intended to help Retailers who are implementing RichRelevance on their websites understand the two implementation options, Server Side or Client Side. A thorough knowledge of the merchandiser's web site and business model is helpful.

What are the benefits of this feature?

Understanding the way each of the two implementation methods works is important, no matter which product or products from the RichRelevance lineup you are implementing.

What are the drawbacks of this feature?

Retailers will want to work closely with their Rich Relevance team to get the most out of their implementation. While many features are self-service, taking advantage of the RR team and the training provided will help avoid hiccups in implementation.

How It Works

RichRelevance ingests data from the retailer in various Feeds that it uses to recommend products, show content or return in search/browse results for end users in placements on the retailer's website, using a complex mix of strategies, rules and an engine that continuously works to display the most relevant recommendations in real time.

When a shopper visits a site that uses a RichRelevance product, the site sends information about the shopper and the site to the Relevance Cloud and receives personalized content in return. This call to the Relevance Cloud and the response can be made from the shopper’s browser (which we call a client-side request) or from the merchant’s server (a server-side request).

The RR data center and the RR back end communicate on a regular basis, bringing logs of user activity to the back end and updates to the runtime to the data centers. Each data center has an exact copy of the current runtime, so it doesn’t
matter which data center is called from the browser.

More data comes to the RR back end from feeds (catalog updates, offline purchase data, etc.). The Relevance Cloud dashboard also configures the back end. Changes from these sources propagate to the data centers through regular updates. (For example, a catalog update that came in through the feed will affect the recommendations in the runtime.)

**Client Side vs. Server Side**

**Client Side**

In a Client Side implementation, the shopper visits the client’s web site and the merchant’s server delivers a page that includes JavaScript code or an API that calls the RichRelevance server at one of our data centers. This call includes information about the user, which the Relevance Cloud uses to choose the best personalized information to show. The RichRelevance server sends this information back (including recommendations, promotions, and sorted catalog data) to the shopper’s browser, either as HTML or JSON.

In a client-side setup, pages call the RichRelevance server from the shopper’s browser, either through our JavaScript integration or a client-side API call:
Server Side

On the other hand, in a server-side setup, pages call the RichRelevance server from the retailer's server:

Cookies

On client-side implementations, the RichRelevance server creates cookies through the shopper's browser to record the shopper's views, purchases, recent search terms, and other information about the shopper's behavior. With every call to the RR server, this information is read from the cookie and used to personalize the shopper's experience.

Because cookies are features of the shopper’s browser, cookies aren’t inherently part of server-side implementations. The RR server can’t create cookies if it’s not in direct contact with a browser.

Cookie Proxy

In server-side setups, merchants can effectively give the RR server access to the browser’s cookies by acting as a cookie proxy (or cookie pass-through). When the RR server requests a cookie, the merchant’s server requests the cookie through the shopper’s browser. When the shopper’s browser requests a page from the merchant site, the merchant's server reads the RR cookies and then passes this information to the RR server.

https://help.richrelevance.com/Integration_Guide/0010_Integration_Overview/0010_Integration%3A_Client_Side_vs_Server
Vanity URLs

Cookies are domain-specific in a browser, so if the merchant website is called SuperStore.com, the cookies it creates will be available only to SuperStore.com. When recs.richrelevance.com creates cookies, those cookies are only available to richrelevance.com. By default, client-side calls to the RR server will not be able to see cookies created by the merchant server, and vice versa.

When these cookies are not shared between the two types of integrations, logging and reporting will be incorrect, and everything that the Relevance Cloud learns from user behavior can suffer as a result.

To share the cookies, you need to set up a subdomain on your site (like recs.SuperStore.com), a CNAME reference, and the RichRelevance Vanity URLs feature. The RichRelevance cookies will be created under your site’s domain, and they can be accessed from your server-side as well as your client-side implementations. See Vanity URLs.