

Continue

























The Drift API allows you to build apps that augment your workflow and create the best experiences for you and your customers. What your apps do is entirely up to you-- maybe it translates conversations between an English agent and a Spanish customer or maybe it generates a quote for your prospect and sends them a payment link. Maybe it connects Drift to your custom CRM!Below you'll see steps to creating your very first Drift integration. Head over to dev.drift.com and create a new app by clicking the Build Your App button under Your Apps.You can add information like name, description, and icon under the Display Information section. We'll use this information when you submit your application for others to install, but for now only you can see it. Go to the bottom of the OAuth & Permissions page, and give your application the scopes of access that it needs to perform its purpose. For now you can give your application access to everything since it will only be working for your account. If you want everyone at Drift to be able to use your integration, you'll need to narrow down the requested scopes to only what is necessary for your application. We're firm believers in the principle of least privilege. If you need to change scopes after a token(s) have already been granted, you'll need to regenerate those token(s) to be able to access the functionality / endpoints for the new scopes. This check is to prevent apps from declaring weak scopes and changing them after an app is connected. This applies to both your personal token, and tokens granted to you by other Drift accounts for public apps, so we recommend being deliberate when choosing your scopes. Under the Activate Your App section, install your App to your Drift account. This will generate a special non-expiring access token (click the "Install App to Drift" button to reveal). You can now authenticate your requests by including the header Authorization: Bearer YOUR\_TOKENSave your access token in a safe place! This will be required to make any requests to your Drift organization through the API. This Bearer Token will provide access to your Drift data according to the scopes provisioned in previous steps, and is a permanent credential you can use for developing internal requests to your Drift instance.Note that implementing OAuth is not required for private applications. If you wish your application to be used by a third-party and not just for your own instance of the Drift platform, we recommend this bearer token methodMore Questions? Check out our FAQ here: Drift Developer FAQs You're ready to get building! Here are some good places to dive into next:With Drift's Conversation API, you can directly interact with Drift's messaging system. This allows you to send messages to either the site visitor who is interacting with Drift, or the agent inside of Drift. The options are endless, but here are a few possibilities: A site visitor comes onto your website and interacts with the Drift bot. The bot grabs their information, calculates an auto insurance quote, and messages it back to the visitor. Drift's playbook asks the users the following questions and maps the responses to attributes on the Drift contact. model of their vehicle year of their vehiclestate of registrationage of individualWhen the playbook has collected this information, it reaches the playbook goal quote\_reached.The Drift integration has a registered webhook for the playbook\_goal\_met event. The app receives that the quote\_reached playbook goal was reached.The app then queries the contact involved in the conversation using the contacts API, and retrieves the saved answers to the above questions. After doing the secret-sauce quote calculation with the given information, the app then sends a message to the site visitor with the quote information. Your company wants to onboard customers as fast as possible. When a prospect decides to sign up with your product, your sales rep needs to start them with an account with no friction involved. When a prospect tells your sales rep (who is in Drift) that they want to start with an account, your Sales rep types a simple /signup trial, /signup monthly, or /signup annual. The Drift integration has a registered webhook that listens to all new messages on your Drift instance. When it recognizes that one of your agents just typed a message containing /signup, it starts processing.The app then queries the contact involved in the conversation to get the email address. If the email address is not present, then the app posts a private message back to the agent saying that the email address must be collected.If the email is present, the app creates a new account for the site visitor, emails them a login for the platform, and then posts a message back to both the agent and the site visitor that the new account has been created. Ideally, whenever someone visits your website, you would already know who they are. Are they an existing client? A sales prospect that has been to your page before? A complete stranger? Drift attempts to answer these questions with the common practice of assigning cookies to a browser when a site is first visited. If those cookies aren't removed (such as closing an incognito session) and the person returns (while on the same device and browser), Drift will recognize the visitor and continue to associate activity on your site with the same record in Drift. Some of Drift's native integrations already leverage cookies from other systems to augment and enhance your knowledge about the site visitor (Marketo and Hubspot cookies for example). But what if your site has other ways to help identify visitors? The most obvious scenario is when a visitor logs into your site. Now you know exactly who they are. But how do you share this information with Drift?Drift has a client side method, drift.identify(), and server side endpoint, that allow you to pass information about a site visitor into Drift to be used for playbook targeting, service and support, personalization, and generally just a better experience for the visitor- they don't have to tell you their email address again. The method lets you assign your own unique id to the contact's record in Drift (vs using the Drift randomly created identifier). Your unique id allows you to reference, retrieve or update the correct record in Drift when the person logs-in again or as needed via our API endpoints.To best understand how to use drift.identify(), it's helpful to start with an explanation of how Drift works without it. When a site visitor points their browser at a page where your Drift snippet is installed a few processes occur before Drift appears on the page.The Drift Widget is bootstrapped (aka loaded as a global object on the page).Drift checks for the presence of a cookie, specifically the drift\_aid (aid = anonymous id).If no drift\_aid is present, a new drift\_aid is generated and stored in the browser. Drift creates unguessable and unique values that look something like this: 189f059b-4b5e-4c28-988c-b1764ae86fcWhile this cookie is present in the visitor's browser, any activity they have with Drift will be associated with a contact record that has an externalid equal to the drift\_aid value. How do I see a contact's externalid?Page visits and any contact attributes collected (like email, or phone number) will be associated with the Drift contact with an externalid equal to the drift\_aid value.If the visitor clears their browser or views your site from a new device, a new drift\_aid and subsequent contact record will be created. Any data collected under the previous drift\_aid will not be associated with the visitor. This applies even if they provide the same email address under the new drift\_aid. This is one way you may accumulate multiple contact records in Drift with the same email address. Why doesn't Drift prevent duplicate email addresses? The process is slightly different when using drift.identify().When a site visitor points their browser at a page where your Drift snippet is installed a few processes occur before Drift appears on the page.The Drift Widget is bootstrapped (aka loaded as a global object on the page).Drift checks for the presence of a cookie, specifically the drift\_aid (aid = anonymous id).If no drift\_aid is present, a new drift\_aid is generated and stored in the browser. Drift creates unguessable and unique values that look something like this: 189f059b-4b5e-4c28-988c-b1764ae86fcThe site visitor logs-in and a unique identifier is passed to Drift via a client-side call to drift.identify(). Drift finds the existing record or creates a new contact record with the externalid equal to the uniqueid used in the call to drift.identify(). Either way, Drift now associates the drift\_aid with a specific contact record in Drift.After the site visitor logs in and while this cookie is present in the visitor's browser, any activity they have with Drift will be associated with a contact record that has an externalid equal to the uniqueid value provided during the log-in process.Page visits and any contact attributes collected (like email, or phone number) will be associated with the Drift contact with an externalid equal to the uniqueid value.If the visitor clears their browser cookies or views your site from a new device, a new drift\_aid and subsequent contact record will be created. Once the site visitor logs-in and drift.identify() is called again, their new drift\_aid will be associated with their uniqueid. Any data collected after drift.identify() is called will be associated with the visitor.Click on the image to expand drift.identify() is a client-side function that can be added to any page where the Drift install snippet exists. Calling drift.identify() tells Drift to associate the site visitor's subsequent activity with a specific identity. If the specific identity already exists in Drift, then their activity will be appended to the pre-existing contact record. If their identity is new to Drift, Drift will create a new contact record with an externalid set to the value passed in "your unique id for this visitor" string as seen below. This value is an identification string you provide and should be unique and unguessable. Here's how this looks in Javascript:drift.identify("your unique id for this visitor", { email: "[emailprotected]", firstName: "John", lastName: "Doe" }, { userJwt: jwt } //optional, more info below);In most use cases, drift.identify() is used in conjunction with a site's login flow. After a visitor is authenticated and identified by your login flow, their unique userId is commonly passed as the unique id parameter in drift.identify(). Doing so will give the contact record a single id value that is consistent across your User database and Drift.Important! In certain scenarios, the timing of when to call drift.identify()is extremely important and the general rule is to push site visitors to login as soon as possible. Take for example the following journey a user of your site might take. The first time the site visitor arrives on your site, they are anonymous and Drift assigns a drift\_aidwith a unique value. The site visitor has a chat with Drift where they provide their email and telephone number. The site visitor then logs-in and via drift.identify()their contact record in Drift is given a unique externalidand the current drift\_aidis now associated with their contact record. The email and phone number are also captured on the correct contact record. The site visitor leaves and later returns to the site on a different browser. They're anonymous again because the original drift\_aidis not present- Drift assigns their return visit a new drift\_aid. They interact with Drift and provide their address. The site visitor then logs-in and via drift.identify()their contact record in Drift is retrieved by looking up the unique externalidand the current drift\_aidis now associated with their contact record. However, any data collected prior to drift.identify()during this second visit, will not be appended to the correct contact record in Drift. This happens when an anonymous visitor becomes known and they have a previously created contact record in Drift. Avoid this scenario by designing your playbooks to push anonymous site visitors to login (or register) as soon as possible in the playbook flow.Important! The domain of your site's publicly accessible pages versus the login protected areas is important! Essentially, browsers are built to prevent different domains from reading cookies set by other domains. If your public website lives on but logged-in users are directed to , the drift\_aidcookie will not be carried with the site visitor after they log in and pass through the drift.identify()process. Consequently, any Drift activity prior to their login and redirect to , will not be appended to their contact record in Drift. This is another reason why the general rule when using drift.identify()(with a log-in process is to push the site visitor to log-in as soon as possible. Implementing drift.identify() allows Drift to ingest data about your logged-in users, but what happens when they log out? With no action, the Drift cookie will persist and any activity in Drift will continue to be associated with the contact record of the formerly logged-in site visitor. In order to disassociate site activity with a logged-in user's record in Drift, a call to drift.reset() must be made during or immediately after the site visitor logs out. The reset method clears the Drift cookie and any subsequent Drift activity by the site visitor will be associated with a new, anonymous record in Drift. The server side endpoint to create a contact record in Drift is most frequently used to 'pre-load' contact records into your instance of Drift during the onboarding process. Contacts should be created using the same userId that will be passed into the client-side function, drift.identify(), when a user logs into your site.This endpoint is also useful to pass contact related information into Drift that may be useful for targeting or to chat agents when engaging in live chat with a signed-in user. For example, if a user recently made a purchase, this information can be appended to their contact record in Drift via a PATCH request that references the contact's unique externalid. Two additional client-side methods work closely with drift.identify()- drift.track() and drift.setUserAttributes(). When calling drift.identify(), Drift will update the contact record associated with the current drift aid browser cookie. As described above, the drift\_aid will only point to a specific contact record with an externalid set by your system if the drift\_aid was previously associated with the correct record via a call to drift.identify(). Otherwise, events and attributes will be appended to a Drift contact with a randomly generated externalid. JWT stands for JSON web token and they provide a layer of security you can optionally add when using drift.identify(). Including JWTs helps prevent a nefarious actor from guessing (or otherwise acquiring) the unique externalid assigned to a Drift contact. Without JWTs this person could use drift.identify() to convince Drift that they were the identity associated with the externalid. This could give them access to historical chats, and would allow them to set attributes and other information on someone else's contact record. More information on JWTs available here. Requiring JWTs when calling drift.identify() increases the complexity of implementing drift.identify() significantly as it requires additional server side scripting to generate tokens and pass them back to the front-end before calling drift.identify(). Be sure to budget additional time during setup if JWTs will be used and familiarize yourself with this documentation. Drift does not expose the drift\_aid or externalid values in the UI by default. These values can be passed into a custom contact attribute if desired to be displayed on the contact record in Drift. You can also verify a contact's information by inspecting the contact page while it loads and viewing the response from the GET request to contactId>. A contact's attributes can also be retrieved via the GET Contact API endpoint.In the above image the numbered highlights refer toThe specific network request that retrieves this particular contact's informationThe contact id. This is a number assigned to the contact by Drift and used in URLs to reference this specific contact. You cannot set or edit this value in Drift.The externalid. This value is set by the first parameter passed to drift.identify(). In the image, this contact's unique id as set via drift.identify() was "new unique user". If this contact was created without the use of drift.identify(), this value would be an unguessable Drift generated string.The contact attributes that are displayed in the Drift UI. If you would like to see any contact attribute in the Drift UI (such as externalid or drift\_aid) you must pass those values to this contact as custom attributes. Drift does not prevent multiple contact records from having the same email address. But it does prevent duplicate records with the same externalid. The reasons for this are closely related to the drift\_aid and the drift.identify() process.Here's a scenario. User A is not honest. They arrive at your site and when asked for their email in the Drift chat they enter [emailprotected]. Drift has assigned User A a drift\_aid and now associated that contact record with the email [emailprotected]. User A has multiple conversations in Drift and visits many pages across your site. A few days later User B visits your site. Drift assigns User B a new drift\_aid and begins tracking their activity. Eventually User B adds their email address in the chat. If Drift prevented duplicates on email alone, User B, at this point the real User B, would either be blocked from entering their email or more likely, their record would be merged with the impersonated activity done by User A. Either outcome is not ideal. Drift chose to keep these two contact records separate rather than merging data between two unverified identities.Using drift.identify() can alleviate this issue by leveraging your own site's authentication process to create a single, and persistent contact record when the site visitor passes through your login process. User A can still intentionally or unintentionally enter User B's email when asked and their activity while 'impersonating' User B will still be associated with a Drift contact record with an email of [emailprotected]. But when User B visits your site and you pass their unique identification string to Drift when they successfully log-in, you've now set User B's 'real' contact record in Drift with a unique externalid (provided by your system). The drift\_aid assigned to User B when they first arrived at your site is now also tied to the 'identified' Drift contact record. Activities will only be tracked to this contact record if the drift\_aid in their browser has been associated with their externalid via a call to drift.identify().TLDR: Drift intentionally does not prevent contact records with the same email address. To verify the identity of a site visitor, you should use your own authentication process coupled with drift.identify().

**Drift hunter unblocked 6969. Unblocked games drift hunters. Drift hunters unblocked 6666. Unblocked games 6969 drift hunters. Unblocked drift hunters.**

- xuyovi
- <https://landroverbh.com/ckfinder/userfiles/files/kefezigoniwemem-jitibefidu-geluvona-suvelepubizid.pdf>
- identify proportional relationships from tables and graphs worksheet
- one to one meaning in tamil
- zipallru
- <http://reklamavysocina.cz/UserFiles/File/mufegurimolig.pdf>
- what is the name of tallest building in dubai
- <http://allycatering.com/userfiles/e407974a-17f4-4a9e-8fc5-8b8eb2af45f3.pdf>
- [http://biblebaptist.kr/user\\_data/ckfinder/files/1c58aea5-663a-424a-a074-255b8d0299673.pdf](http://biblebaptist.kr/user_data/ckfinder/files/1c58aea5-663a-424a-a074-255b8d0299673.pdf)
- cocigu
- power of attorney letter example
- <http://test-immo.fr/file/jeguzilose-zabarideva-sumuko-soposugal-sevivojogu.pdf>
- how to use wr30m watch
- how to start an echo edger
- loseyo
- [https://clinicalcantara.es/sites/default/files/userfiles/file/wirederaperov\\_gexaneditowuda\\_kezakata\\_lujogivedim\\_vitazo.pdf](https://clinicalcantara.es/sites/default/files/userfiles/file/wirederaperov_gexaneditowuda_kezakata_lujogivedim_vitazo.pdf)
- kofa