

CONSENT PREFERENCE

Introduction to Consent Preference Management

What is Consent Preference?

We provide Consent Preference solutions that consolidate your information and prioritize your customer relationships by unifying collected consent, and your communication for marketing compliance with multiple data privacy regulations worldwide.

In privacy compliance, the consent preference is important to maintain a comprehensive record of consent especially when concerned with GDPR. These records must contain the way to identify the user, proof of consent, and a record of the consent actions.

We provide a [Consent Preference SDK](#) that enables the following features for your website or web portal:

- Keep user consent and privacy preferences in one place.
- Store multiple preferences from each user (e.g., if you have multiple newsletters or opt-ins).
- Store whether a user is verified with double opt-in.
- Get the history of consent from each user.
- User-friendly dashboard for analyzing consent records.
- REST API and JavaScript SDK to give you complete control and how and when consent is stored.
- An easy-to-implement JS widget to simplify the consent tracking received through online forms.
- For each consent, track the legal notice or privacy notice that the user has accepted.
- Integrate with Mailchimp Webhooks (other webhooks integrations are in pipeline).
- Update consent preferences across your third-party tools (such as Mailchimp, Intercom, and Marketo).
- Can watch the session recording that is saved in Adzapier Portal when a user gives his consent.

We also provide a configurable [Subscription Consent Banner](#) for you to create consent widgets for Newsletter subscriptions and product updates subscriptions, and also keep and manage the consent records in the Adzapier Portal.

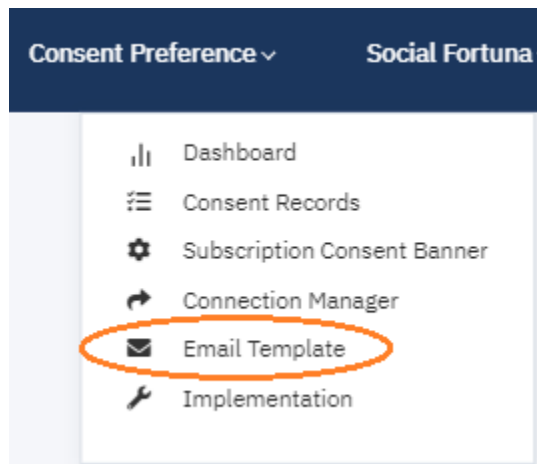
With this application, you can configure a fully customizable banner for subscription consent, implement it on your website, and record your users' consent to the Adzapier portal.

Configure Confirmation Email

You can configure the confirmation email that your website user receives when he/she provides consent through the consent widget that you have implemented using Consent Preference SDK and Subscription Consent Banner.

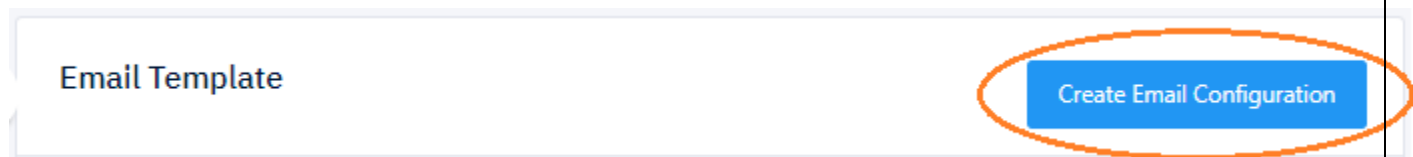
Steps to configure the Email...

1. In Consent Preference, click Email Template.



The Email Template page appears.

2. To configure a custom email, click Create Email Configuration.



Email configurator appears.

Email Sender *

-- Select Sender --

Email Subject *

Adzapier Consent Verification

Email Body

B I U H1 H2

x₂ x² Normal

Normal A Sans Serif

Hey {full_name},

Please verify your email.

[Verify email](#)

Cancel Submit

3. Create sender - You can configure multiple senders and select any one of them.
 - Click Create a new sender.
 - The Create sender form appears.

Create a new Sender

Nick Name *

Enter nick name Ex: John

From Email *

Enter sender email Ex: John@example.com

From Name *

Enter sender name Ex: Johnson

Reply Email *

Enter reply email Ex: John@example.com

Address *

Enter your address

City *

Enter your city

Country *



Enter your country

Submit

- In From Email, enter an email id of an active organization member of your Adzapier account.
 - Or, if you enter a non-member email id, you must verify that email id via the confirmation email that receives after the form submission.
 - Enter all other form fields, and click Submit.
 - In the same way, you can create more senders and select the senders from the list.
4. Edit Subject - type your email subject.
 5. Email Body - in Email Body, {full name} and Verify Email link are dynamic values, and you can customize other texts.

6. After customization, click Submit.

7. The configured email appears in the list. If you want to manage it, click .

Email Template					
			Email verification configuration <input type="checkbox"/> SDK <input checked="" type="checkbox"/> Banner		Create Email Configuration
Sender Name	Sender Email	Email Subject	Status	Created	Action
lipuu	aaa@yopmail.com	Hello User	Inactive	Dec 20, 2022	
Siyali	abc3@yopmail.com	Hello User	Active	Dec 20, 2022	

Note: Whenever you make changes to the banner or email configuration, you must publish the consent widget again, to reflect the changes on your website.

Enable/Disable Verification Email for SDK or Banner

- If You want to send the verification email to your visitors after getting their consent, you can enable the verification configuration email. You can enable email configuration for SDK or Banner by selecting their respective checkbox.
- If you don't want to send emails to your visitors, then disable the verification email.

Email Template	
Email verification configuration <input type="checkbox"/> SDK <input type="checkbox"/> Banner	

Note: Even though you've configured the email, you can't send the verification email to your visitors unless you select the Email verification configuration check box.

Connection Manager

You can integrate the Adzapier Subscription Consent widget or Consent Preference module with email service systems and database systems to maintain a contacts database and track the consent records of the website users including the new consents and the updates on the existing consents.

To integrate the subscription consent banner of your website with a system, Adzapier's (source) Consent Preference API, Consent Preference JavaScript, and Consent Banner widget are connected to the destination systems which will happen in the backend when you establish the connection through the portal. You can easily configure multiple systems to be connected with your source system.

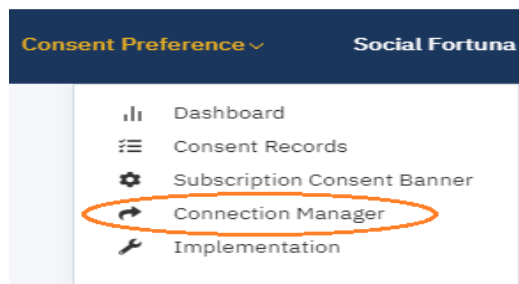
Note:

- For integrating the Subscription Consent Banner with the destination system, you must create a system connection through the [Setup System Integration](#) wizard.
- To learn how to set up a connection, see [Systems Connection](#).

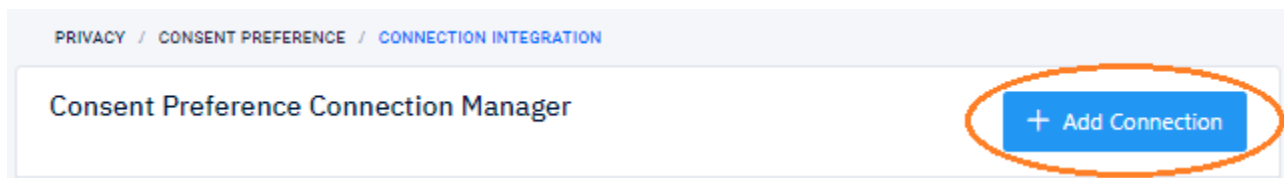
Configure Connection Integration

Once you set up the system connection successfully through the Setup System Integration wizard, follow the procedure to configure the integration of consent preference to the systems.

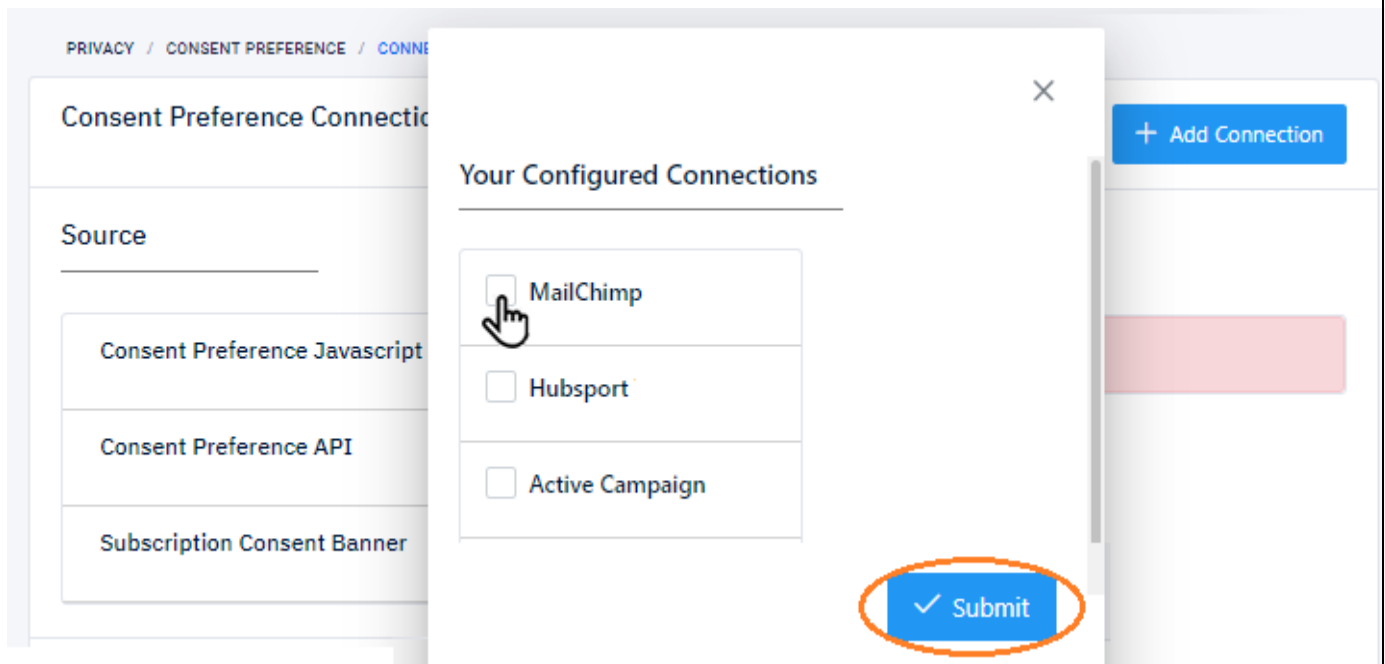
1. In the property selection menu, select the desired property.
2. In the Consent Preference menu, click Connections Integrations.



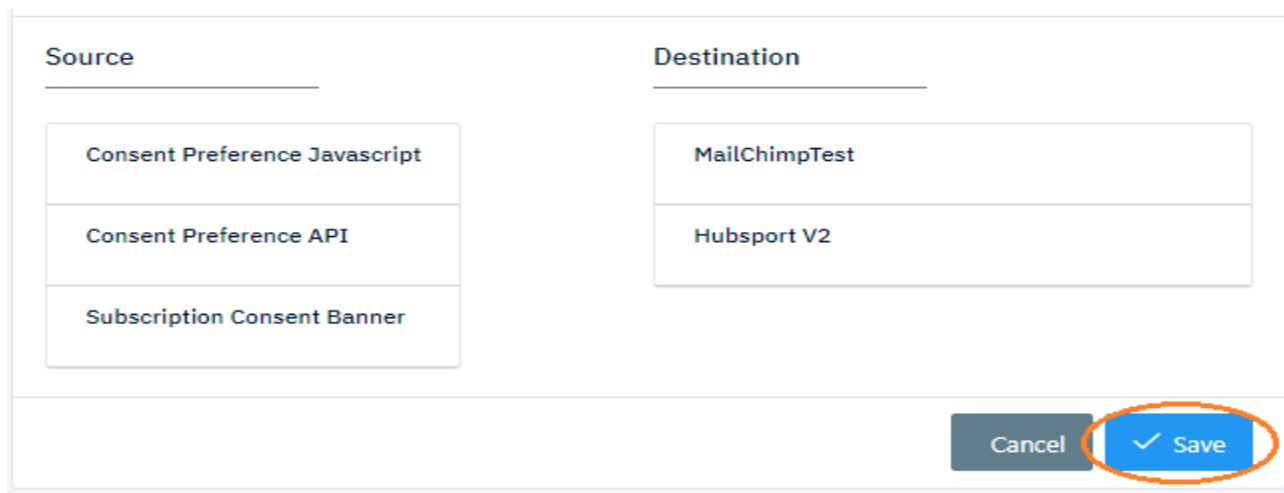
The Consent Preference Connections Manager page appears.



3. Click Add Connection.
A list of Configured Connections appears that shows the connections you have set up already.

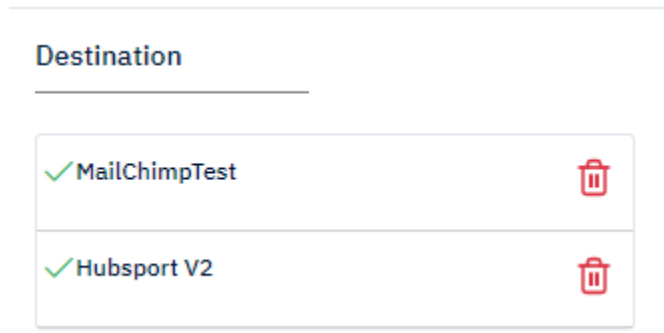


4. Select the connections that you want to integrate and click Submit.
The selected connections appear in the Destination list.




5. Click Save.

It validates and establishes the connections. If the connection is established successfully, a green tick appears right to each destination system.



Note: Once you publish a Subscription Consent Banner on your website, the above integration enables the consent and contact record to the connected system.

Remove Destination Systems

- If you want to remove any connection on the destination side, click  , and then click Save.

Subscription Consent Banner

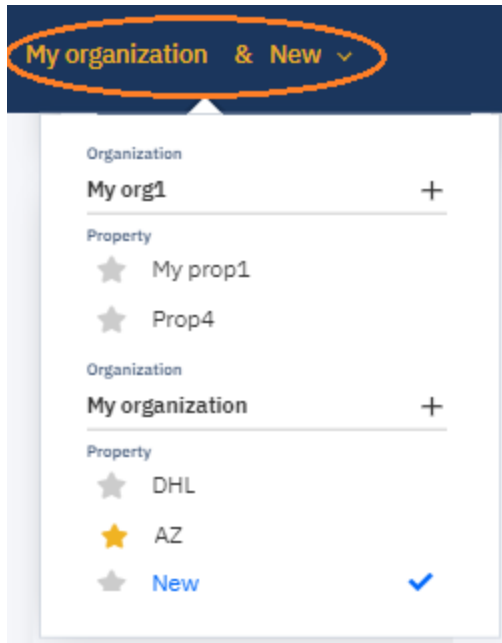
Adzapier's Subscription Consent Banner module offers you consent widgets like Newsletter or Email subscription in addition to keeping and managing consent records in the Portal.

With the help of this application, you can configure a customized banner for subscription consent, implement it on your website, and the website users' consent will be recorded in the Adzapier portal.

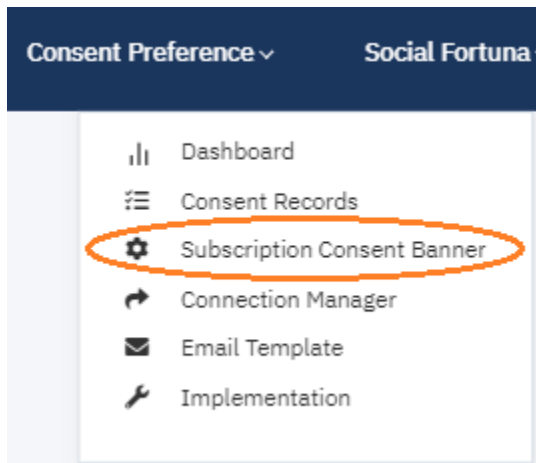
Configure Subscription Consent Banner

In the following way, you can configure a custom banner for subscription consent.

- In the property selection menu, select the desired property.



- In the Consent Preference menu, click Subscription Consent Banner.



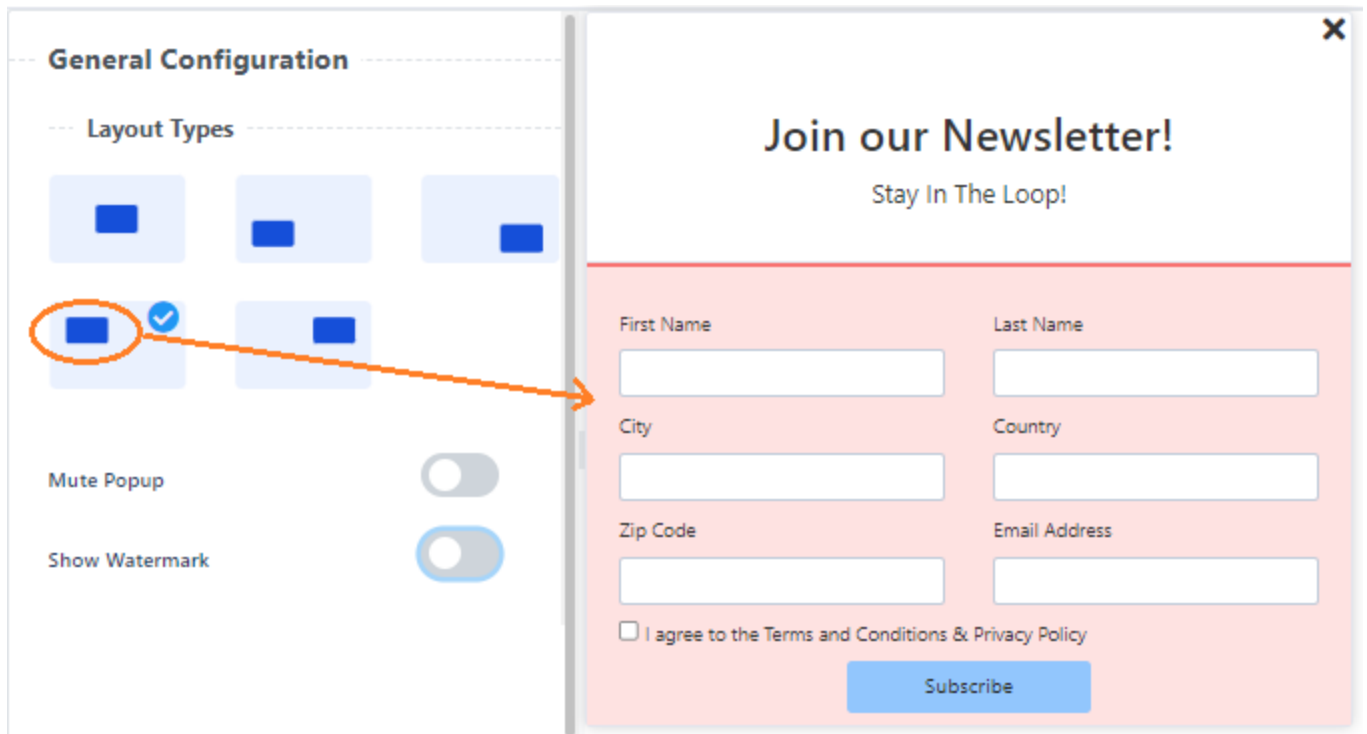
- The Subscription Consent Banner page appears.

Multiple Device Preview

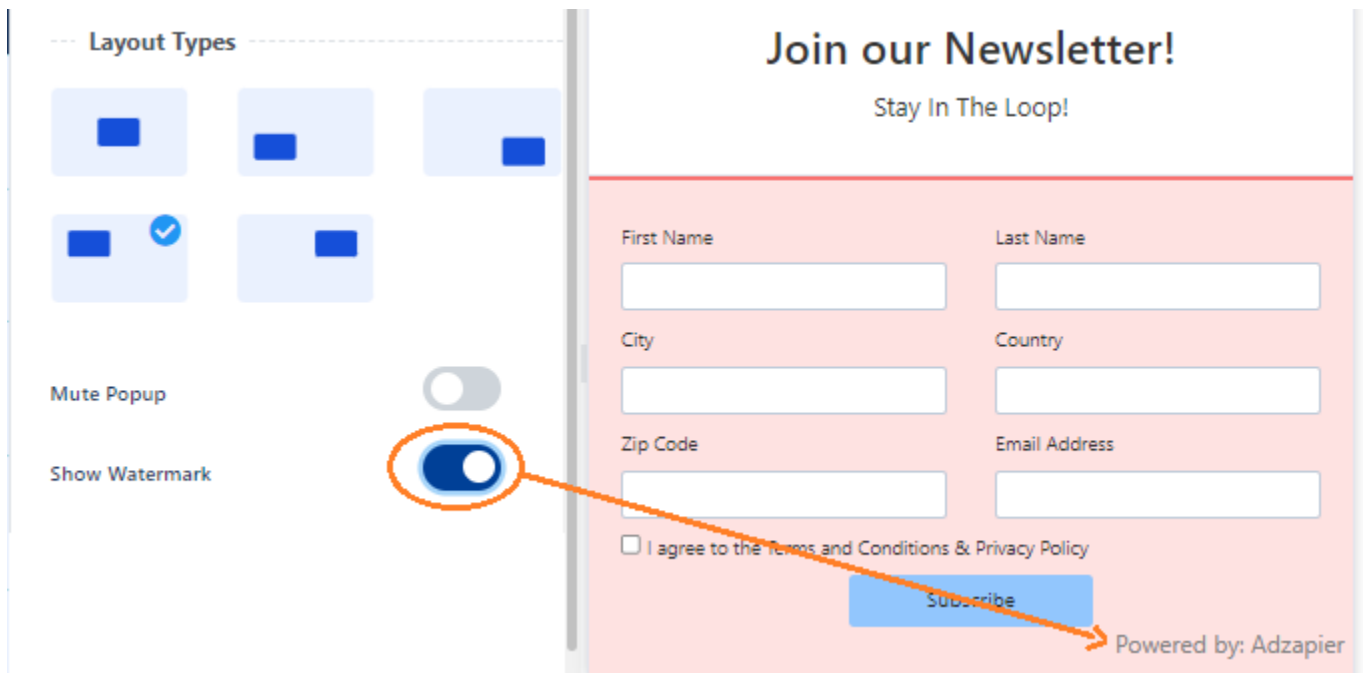
- The default banner preview appears in the window area of the page. You can customize the configuration using tabs in the left pane.

General Configuration

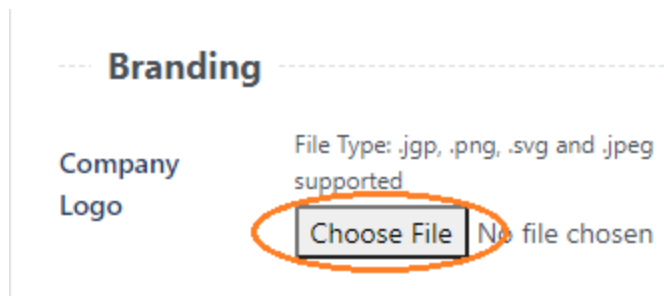
1. Under Layout Types, select any layout type and see the preview.



2. If you want to mute the subscription consent banner for a while, toggle on Mute Popup. And you can publish it again by toggling off this button whenever you want.
3. If you want to remove the branding watermark, toggle off Show Watermark.



4. To upload your logo in the header of the banner, click Choose File, then upload the logo from your computer. The image of a logo in JPG, JPEG, PNG, or SVG can be supported.



The logo preview appears in the banner preview. If you want to remove the logo, click Remove below the image preview.

5. In the Custom CSS area, you can add CSS code to your banner to improve its aesthetic appeal. An example is given below.



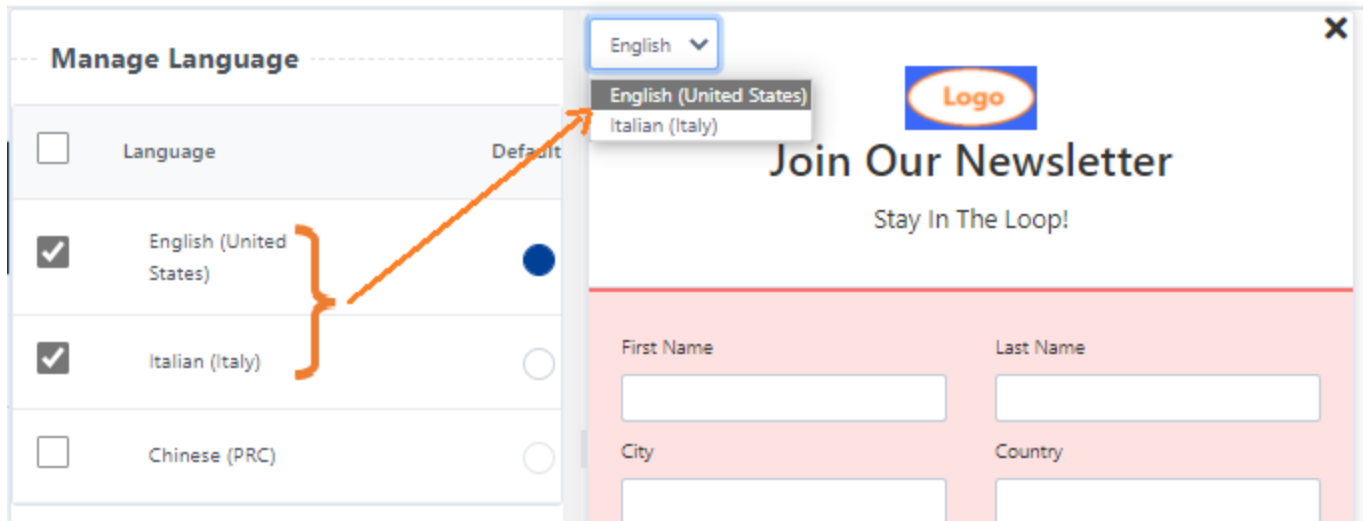
6. This code will turn the text "Terms and Conditions" and "Privacy Policy" into red color.

Manage Languages

You can display the banner in various languages.

The default language of the cookie banner is English. But you can add more language options using this setting.

1. In the left tab, click the Language tab.
2. To add any languages from the list, click the check box right to the language. The selected language options will be added to the banner.
3. To make your preferred language as default, select the language, then click the button in the Default column right to the language.

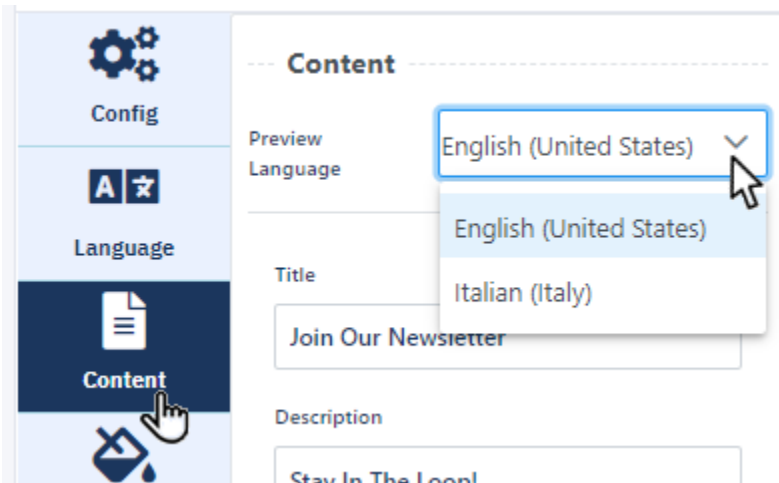


The banner is displayed on your website in the default language. The user can change the banner language using the language dropdown.

Edit Content

You can edit the content of the cookie banner as well as the privacy policy link.

1. In the left pane click the Content tab, then select the Preview Language that you want to edit the content in.



2. Enter the following mandatory details.
 - Terms and Conditions Link
 - Terms and Conditions Version (Numeric value only)
 - Privacy Policy Link
 - Privacy Policy Version (Numeric value only)

The diagram illustrates the mapping between form fields and their HTML placeholders. The placeholders are listed on the left, and the form fields are shown on the right. Arrows indicate the connections:

- Title:** Placeholder: `{Join our Newsletter }` → Field: **Join our Newsletter**
- Description:** Placeholder: `{Stay In The Loop! }` → Field: **Stay In The Loop!**
- Terms and Conditions Link:** Placeholder: `{https://adzapier.com/terms-of-service }` → Field: **I agree to the Terms and Conditions**
- Terms and Conditions Version:** Placeholder: `1` → Field: **1**
- Privacy Policy Link:** Placeholder: `{https://adzapier.com/privacy-policy }` → Field: **Privacy Policy**
- Privacy Policy Version:** Placeholder: `2` → Field: **2**
- Subscribe Button:** Placeholder: `{Subscribe }` → Field: **Subscribe**

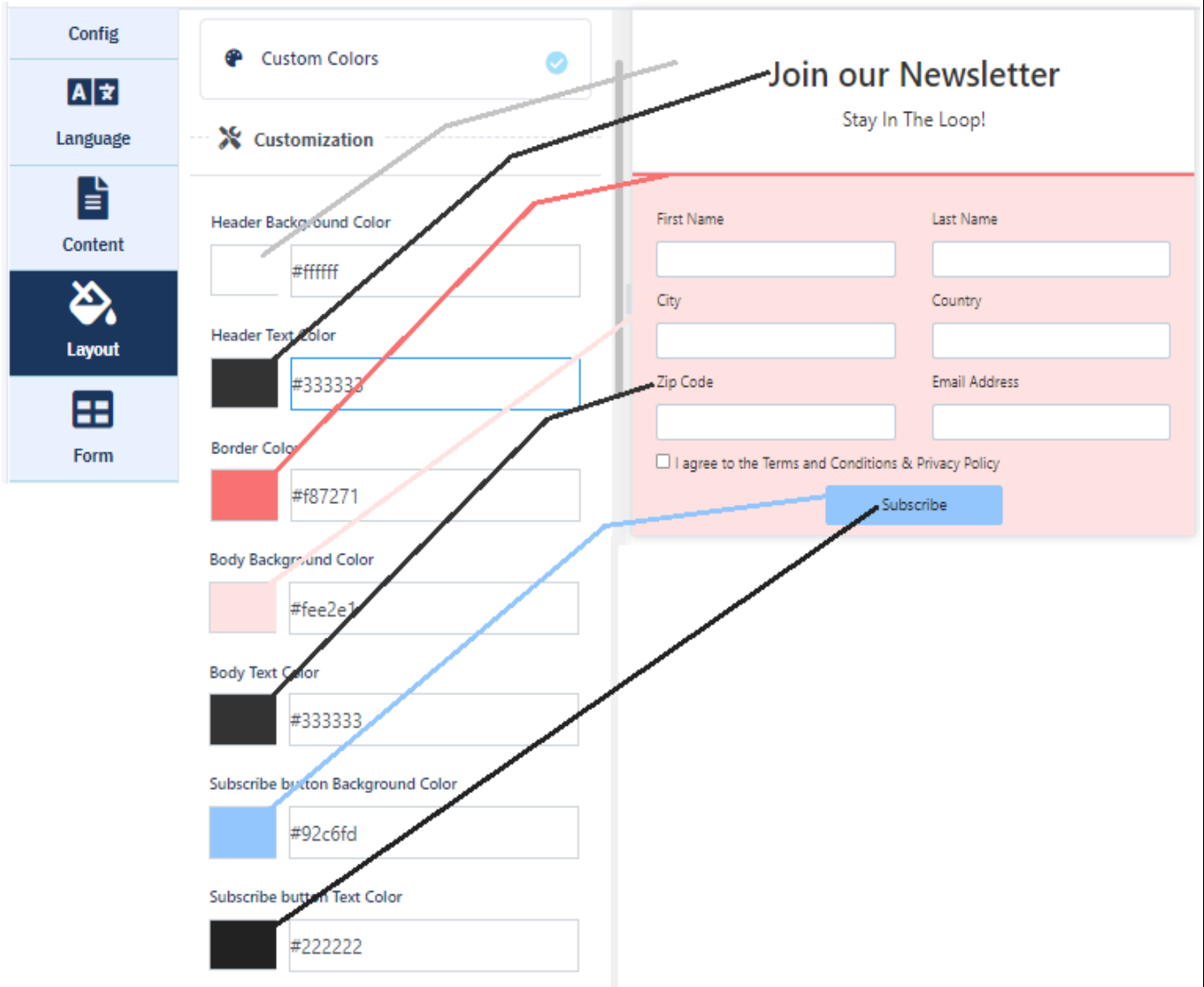
The form fields on the right include: First Name, Last Name, City, Country, Zip Code, Email Address, and a checkbox for agreement to terms and conditions.

3. You can also customize the Title, Description, and Subscribe button title.

Layout

You can choose the predefined light theme and dark theme, or you can customize your theme.

- Under the Layout tab, click Custom Theme to customize the colors.



To change the color, click the color box and select a color or enter a color code.



Form Fields

1. There are eight fields in the banner by default, and if you want to remove any of the fields clear the checkbox right to the field.

Subscription Consent Banner

Config

Language

Content

Layout

Form

Display Frequency

Behavior

Form

First Name	<input checked="" type="checkbox"/>	≡
Last Name	<input checked="" type="checkbox"/>	≡
City	<input checked="" type="checkbox"/>	≡
Country	<input checked="" type="checkbox"/>	≡
Zip Code	<input checked="" type="checkbox"/>	≡
Email	<input checked="" type="checkbox"/>	≡
Phone No	<input checked="" type="checkbox"/>	≡
Landline No	<input checked="" type="checkbox"/>	≡

Englis ▾

Join Our Newsletter

Stay In The Loop!

First name

Last name

City

Country

Zip Code


Email address

Phone Number

Landline Number

☐ I agree to the [Terms and Conditions](#) & [Privacy Policy](#)

Subscribe

2. Each field has predefined settings and if you want to configure any field click on  right to the field.

First Name

☒

≡

Configure Validation

Required

☒

Min Length

3

Max Length

14

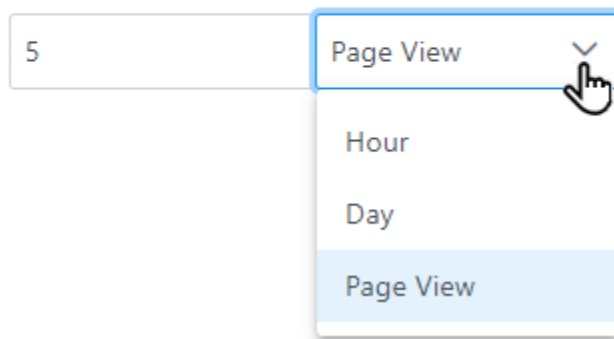
3. You can define the Min Length and Max Length of the text of each field, and you can set the field as either Required or Optional.

Display Frequency

- If you want to define the frequency of displaying the banner, which means the repetition of banner display when the website user closes.
- Enable Display Frequency.

Display Frequency

For Close Subscription Consent Banner Popup - Show
Popup Every



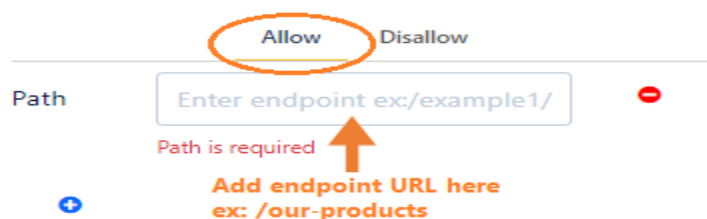
- Enter a frequency value and select any span.



Behavior

You can control how the Subscription Consent Banner behaves by specifying the page URLs where the banner should appear and where it shouldn't.

- By default, the subscription consent banner appears on all the pages of your website.
- If you want to show the banner on some specific pages,
 - Click the Path tab, and under Allow, click Add Path.

Allow/Disallow



- In Path, add the endpoint URL of the page.
- Click  to add more pages in the same way.
 - Else, if you want to show the banner on most of the pages excluding some specific pages,
 - click Disallow, then click Add Path.
 - In Path, add the endpoint URL of the page.
- Click  to add more pages in the same way.

Save and Publish

1. To save the progress of your configuration, click Save.



2. Once you have completed the configurations, click Publish to implement the banner on your website.
3. A Production CDN script pop-up appears.

Please ignore if you have already saved the script, otherwise copy and paste the code into header of your website.

Production CDN



//Please copy and paste this code into the header for each page on your website(s). It is important that this code be the first script in your document.

```
<script type="application/javascript" src="https://staging-cdn.adzapsrv.com/consent-preference/newsletter/v1/web/017663ac-0d00-4e8d-b40f-c199468d25ca/6268aa29-cc6d-43ae-812a-2b505ad92b98-bundle.js"></script>
```

How to integrate Adzapier scripts to:



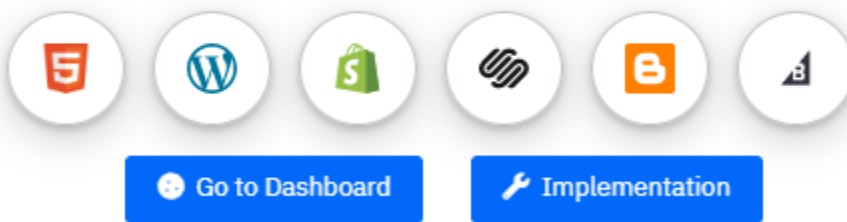
 

4. On the other hand, you can proceed to the Implementation page by clicking the Implementation button or the Implementation option in the Consent Preference menu.
5. Click Copy Script. It copies the script to the clipboard. Now you can add this JS code to your website code to publish the banner.

Note: On the webpage of your domain where you want to implement the Subscription Consent Banner, you must insert the CDN script before the `</head>` closing tag.

6. If you want to learn how to integrate the Adzapier script into your website, click on the respective site and follow the steps.

How to integrate Adzapier scripts to:



Note: If you want to reset the configuration to the default, click Reset to Default and click I Agree.

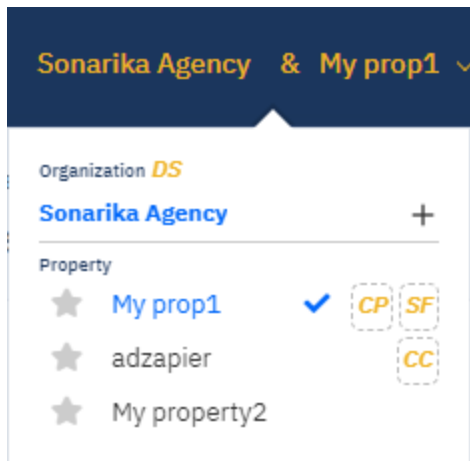
Note: After publishing the Subscription Consent Banner on your website, if you modify any configurations, you must publish to reflect the changes.

Manage Consent Records

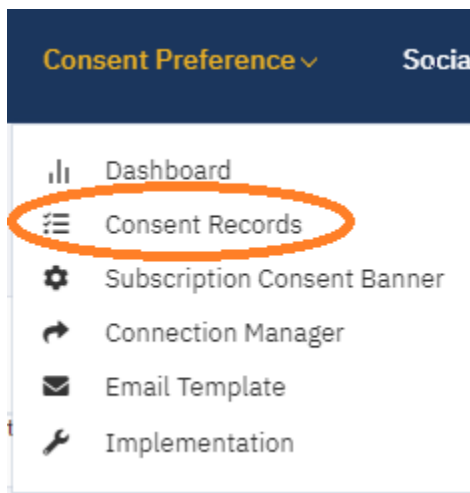
Once you successfully set up the Consent Preference for your property, consents from your domain users are recorded automatically. You can view the details of the individual consents as well as edit the consent details, and manually adding a consent feature is also available in our portal.

View Consent Records

1. In the selection menu, select the property.



2. In the Consent Preference menu, click Consent Records.



3. A list of consent records appears.

Consent Records										<div>+ Add Consent</div> <div>12/14/2022 - 01/13/2023 x</div> <div>Refresh</div> <div>CSV</div> <div>Filter</div>
ID	Owner ID	First Name ↑↓	Last Name ↑↓	Email ↑↓	Country ↑↓	Data Source ↑↓	IP Address	Verified	Created Date	
13fdd340-3eed-4...		satyabrata	das	satya@yopmail.com	India	public	2401:4900:1c**	✓	Dec 29, 2022	
3aad0867-7d14-4...		Niharika	Singh	sassd@yopmail.com			1124.56**	✗	Dec 29, 2022	
e7d42f3f-c002-45...		Sonia	Sahani	sonia@yopmail.com				✗	Jan 13, 2023	
8a9a761b-8a57-4...		Sanu	Singh	sanu@gmail.com		private		✗	Jan 13, 2023	

10 ▾

<<

<

1

>

>>


Showing 1 to 4 of 4 entries

4. Select 10, 20, or 30 from the pagination dropdown and view records as per your requirement.

Apply Filters

1. Select a date range to view the consent records of a specific period on the Consent Records page.

Consent Records									
				+ Add Consent		12/14/2022 - 01/13/2023			
ID	Owner ID	First Name ↑↓	Last Name ↑↓	Email ↑↓	Country ↑↓	Data Source ↑↓	IP Address	Verified	Created Date
13fdd340-3eed-4...		satyabrata	das	satya@yopmail.com	India	public	2401:4900:1c**	✓	Dec 29, 2022
3aad0867-7d14-4...		Niharika	Singh	sassd@yopmail.com			1124.56**	✗	Dec 29, 2022
e7d42f3f-c002-45...		Sonia	Sahani	sonia@yopmail.com				✗	Jan 13, 2023
8a9a761b-8a57-4...		Sanu	Singh	sanu@gmail.com		private		✗	Jan 13, 2023
10 ▾		« < 1 > »			Showing 1 to 4 of 4 entries				

2. To apply more search filters, click .
3. The filtering parameters appear.

Filter the results

First name

First Name

Last name

Last Name

Email

Email

IP Address

IP Address

Source

☐ Both ☐ Private ☐ Public

Cancel

Apply

4. You can sort out the records by User's
 - Email ID
 - First Name
 - Last Name
 - IP Address.
5. After entering the filtering parameters, click Apply.

View Individual Consent Details

In each consent received from the user and manually added consent, you can view the details such as proofs, legal notices, and preferences opt-ins and opt-outs.

1. Go to the Consent Records page, in the list of consent records, click the Consent ID of the consent for which you want to view the details.

Consent Records									
<div> + Add Consent 12/14/2022 - 01/13/2023 × ↺ ↻ 📄 CSV ⌵ </div>									
ID	Owner ID	First Name ↑↓	Last Name ↑↓	Email ↑↓	Country ↑↓	Data Source ↑↓	IP Address	Verified	Created Date
13fdd340-3eed-4...		satyabrata	das	satya@yopmail.com	India	public	2401:4900:1c**	✓	Dec 29, 2022
3aad0867-7d14-4...		Niharika	Singh	sassd@yopmail.com			1124.56**	✗	Dec 29, 2022
e7d42f3f-c002-45...		Sonia	Sahani	sonia@yopmail.com				✗	Jan 13, 2023
8a9a761b-8a57-4...		Sanu	Singh	sanu@gmail.com		private		✗	Jan 13, 2023
<div> 10 ▾ ⏪ < 1 > ⏩ Showing 1 to 4 of 4 entries </div>									

2. The Consent Details page appears.

Consent Details

Current Status of Consent

Edit Consent

Consent ID	First Name	Last Name	Email	Data Source	Country
9a6c4227-4f97-4f66-8752-d0ae8a0af3e0	Satya	sahu	satya@yopmail.com	public	India
Owner ID	IP Address	Created At	Auth Id	Updated At	Additional data
	2401:4900**	Jan 10, 2023		Jan 10, 2023	
Phone Number	Landline Number	Zip Code		City	
9988888888	6676767667677	565656		patna	

Proofs

Legal Notices

Preferences

Recording

Consent Certificate

Proof	Form	Created Date
{"first_name":"Satya","last_name":...}		Jan 10, 2023

- On the Consent Details page, the details of the user such as Email, Name, and IP Address are shown.

How to View the Proof?

- To view the details of the proofs, click the Proofs tab.

Consent Details

Current Status of Consent

Edit Consent

Consent ID	First Name	Last Name	Email	Data Source	Country
9a6c4227-4f97-4f66-8752-d0ae8a0af3e0	Satya	sahu	satya@yopmail.com	public	India
Owner ID	IP Address	Created At	Auth Id	Updated At	Additional data
	2401:4900**	Jan 10, 2023		Jan 10, 2023	
Phone Number	Landline Number	Zip Code		City	
9988888888	6676767667677	565656		patna	

Proofs

Legal Notices


Preferences

Recording

Consent Certificate



Proof	Form	Created Date
{"first_name":"Satya","last_name":...}		Jan 10, 2023

The list of proof documents appears.

2. To view a particular Form of proof, click  right on the proof.
You can read the Form document.

How to View the Legal Notices?

1. To view the details of the legal notices, click the Legal Notices tab.

Proofs Legal Notices Preferences Recording Consent Certificate				
Version	Identifier	Content	Created Date	
1	privacy and policy	adzapiet/terms	Jan 10, 2023	
1	term and condition	adzapiet/terms	Jan 10, 2023	

2. To view the document of any version of the legal notice, click  .

How to View the Preferences?

1. To view the preference opt-ins and opt-outs, click the Preferences tab.

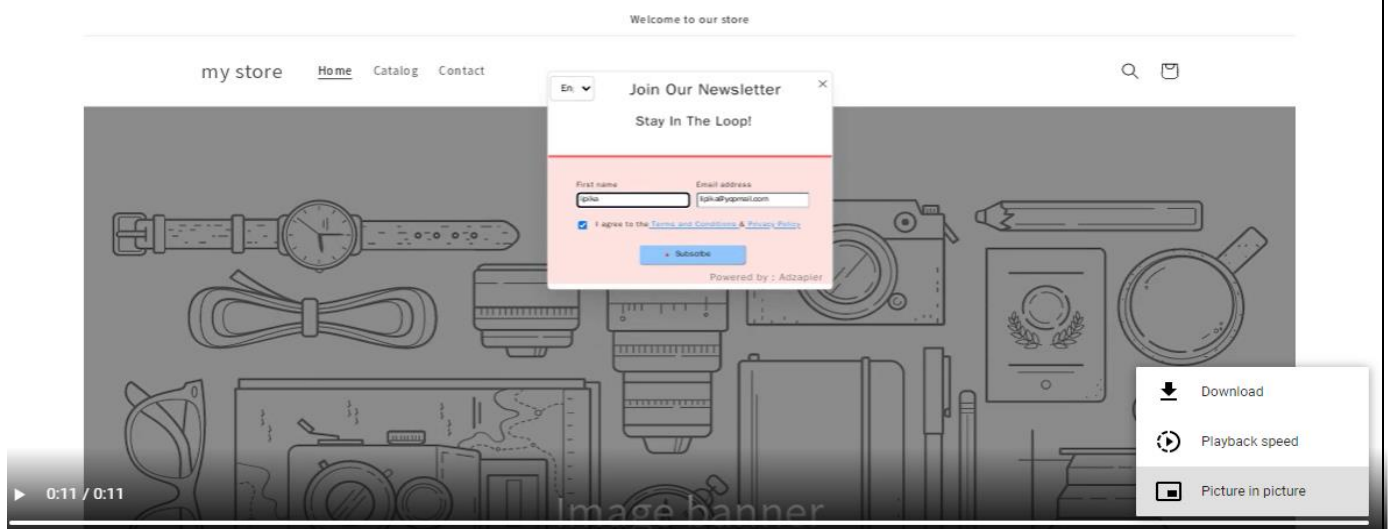
Proofs Legal Notices Preferences Recording Consent Certificate	
Preference	Allowed
privacy_policy	✓
newsletter	✓
terms_and_conditions	✓

How to Watch the Session Recording?

1. The session recording is saved after a user has given consent. Click on the Recording tab, then the following section will appear. Click on the ID to watch the session recording.

Proofs Legal Notices Preferences Recording Consent Certificate		
ID	Status	Recording Date
eab78460-26f2-4fcb-84c6-cd6d28ad23d3	completed	Jan 10, 2023

Note: If the status is pending you must wait a while. when the status is changed to completed, you can watch the recording.



2. You can see the recording in full-screen mode. You can also download the session recording by selecting the three dots located at the bottom. After that click Download.

How to Download Consent Certificate?

1. You can download the Consent Certificate after a user has given consent. Click on the Consent Certificate tab. The following section will appear. Click Generate new certificate.

Proofs
Legal Notices
Preferences
Recording
Consent Certificate

Generate new certificate

Certificate id	Created at	Updated at	Action
No records found			

- A new certificate will generate. Click Show Certificate.

Proofs
Legal Notices
Preferences
Recording
Consent Certificate

Generate new certificate

Certificate id	Created at	Updated at	Action
ee058f43-f6d0-4773-92dc-e0d1dc0454d1	2023-01-09T16:16:50.990723Z	2023-01-09T16:16:50.990723Z	Show Certificate

- The Consent Certificate will appear with all the details of the visitor. Click Print to print the certificate.

Consent Certificate

Trusted Consent Certificate
Certificate ID : d3032a54-2115-46a5-843b-a4f459aa061e
Adzapier

When did they visit ?
Visit date2023-01-11 UTC
Visit Time1:50PM UTC

Where did they visit ?
Page URLhttps://my-cpanel-1141.myschoolify.com/?_ab=0&_fd=0&_so=1

Who visited ?
Remote IP Address2401:4900:1001:1a80:a506:c942::afdd:****
Geographic locationIN-wb-kolkata
BrowserChrome
Operating SystemWindows

What did they see ?
Right click to open
Consent certificate captures a screenshot of page viewed by the visitor
(To view this screenshot right click and open in new tab)
or copy link and paste in new tab)

Print

4. Right-click on the link to view the screenshot in a new tab.

What did they see ?

Consent certificate captures a screenshot of page viewed by the visitor

Right click to open

(To view this screenshot right click and open in new tab or copy link and paste in new tab)

5. The screenshot will look like the following.

English (U)⌵×

Join Our Newsletter Edited

Stay In The Loop! Edited

First Name	Last Name
<input type="text" value="Jagdish"/>	<input type="text" value="More"/>
City	Country
<input type="text" value="Pune"/>	<input type="text" value="India"/>
Zip Code	Email Address
<input type="text" value="411001"/>	<input type="text" value="j.m@yopmail.com"/>
Phone Number	Landline Number
<input type="text" value="1234567890"/>	<input type="text" value="1234567890"/>

☒ I agree to the [Terms and Conditions](#) & [Privacy Policy](#)

Powered by : Adzapier

Adding a Consent (Manually)

You can add consent manually to your property within the portal.

1. Go to Consent Records and click Add Consent.

Add Consent Details appears.

Add Consent Details

First Name

Enter first name

Last Name

Enter last name

Email

Enter Email ID

Authentication ID

Enter Authentication ID

Data Source

☐ Private ☐ Public

Country

Enter Country

Owner ID

Enter Owner ID

IP Address

Enter IP Address

Verified

☐

Phone number

Enter Phone number

Landline number

Enter landline number

City

Enter your city


Zip Code


Enter Zip code

2. Enter all the details of the consent provider and if the details are verified, select the Verified check box.


3. Enter the Preference Name if any, and if the preference is allowed by the user, select the Allowed check box.


Preferences

Preference Name Allowed ☐ 





Proofs

Content Form 





Legal Notice

Identifier Version Content 





Additional Data

Key Value 



[Cancel](#) [Add](#)

4. If you want to add more preferences, click  and enter the preference details. To remove any preference, click .

5. To enter proof details, enter the details in Content and Form. Also, add and remove more proofs same as above.
6. In the same way, you can add the details of Legal Notices.
7. You can also add any additional data by entering the details in the Key and Value field.
Ex: if you want to record that the person who gave consent is 30 years old, enter Age in the Key field and 30 in the value field. You can also add and remove fields the same as above.
8. Once entered all the details, click Add.
The consent will be added to the list of Consent Records.

Editing Consent Details

You can edit the details of the consent provider of any consent available in the consent records.

1. Go to Consent Records, in the list of consent records, click the Consent ID of a consent that you want to edit.
The Consent Details appear.

Consent Details
 Current Status of Consent

Edit Consent

CONSENT ID	FIRST NAME	LAST NAME	EMAIL	DATA SOURCE	COUNTRY
e8ed5ee9-e53c-48f9-acb0-5b9ffc3a1331				public	US

OWNER ID	IP ADDRESS	CREATED AT	AUTH ID	UPDATED AT
	192.186.3456	Jul 6, 2021	12321321	Jul 6, 2021

2. Click Edit.
Update Consent Details pop-up appears.


First Name	Last Name
<input type="text" value="siya"/>	<input type="text" value="pradhan"/>
Email	
<input type="text" value="siya@yopmail.com"/>	
Data Source	Auth ID
<input type="radio"/> Private <input checked="" type="radio"/> Public	<input type="text" value="Enter Authentication ID"/>
Country	Owner Id
<input type="text" value="India"/>	<input type="text" value="Enter Owner ID"/>
IP Address	
<input type="text" value="64.103.1.220 12.34567890123456789012345678901234567890"/>	
Phone number	Landline number
<input type="text" value="8237788999"/>	<input type="text" value="06745657777"/>
City	Zip Code
<input type="text" value="barelih"/>	<input type="text" value="123456"/>
Cancel Update	

3. Make the changes in the details and click Update.

Note: The Auth ID and Owner ID can't be changed.

Export Consent Details

You can export the consent records as a CSV file.

- To export the consent records, click  .

- Then the following pop-up will appear. Select your desired Date Range. You can download Consent Details, Legal Notices, Proofs, and Preferences as a CSV file by clicking on their respective fields.

Export your consent records as a CSV file

Date Range - 09/28/2022 - 10/28/2022

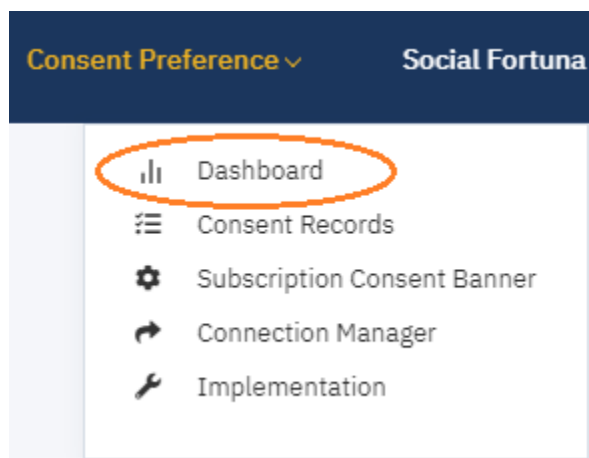
Export Consent Details Export Legal Notices Export Proofs Export Preferences

* Maximum 6 months data can be downloaded

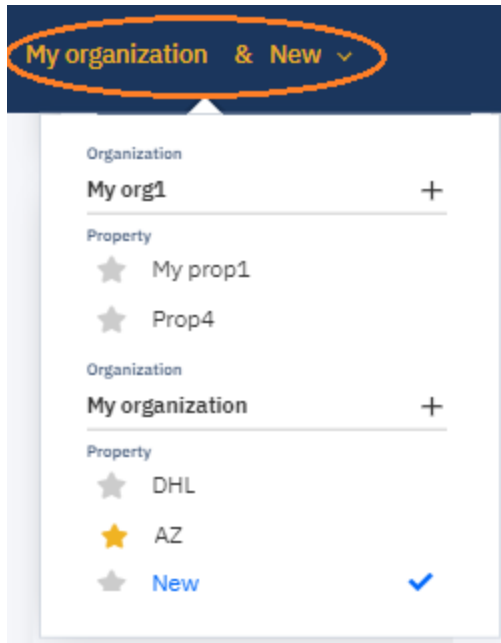
Consent Preference Dashboard

The Consent Preference Dashboard shows the information and analytics on the visitor's consent records of your property.

- To view the Consent Preference Dashboard, click the Consent Preference menu and click Dashboard.



- The Consent Preference Dashboard page appears.
- To change the property, click the property selection menu and select any property.



- The dashboard shows the personalized information of consent records for each property.
- You can change the date range to view the analytics for a particular period.



- The Consent Preference Dashboard shows the following information.

Consents - Count

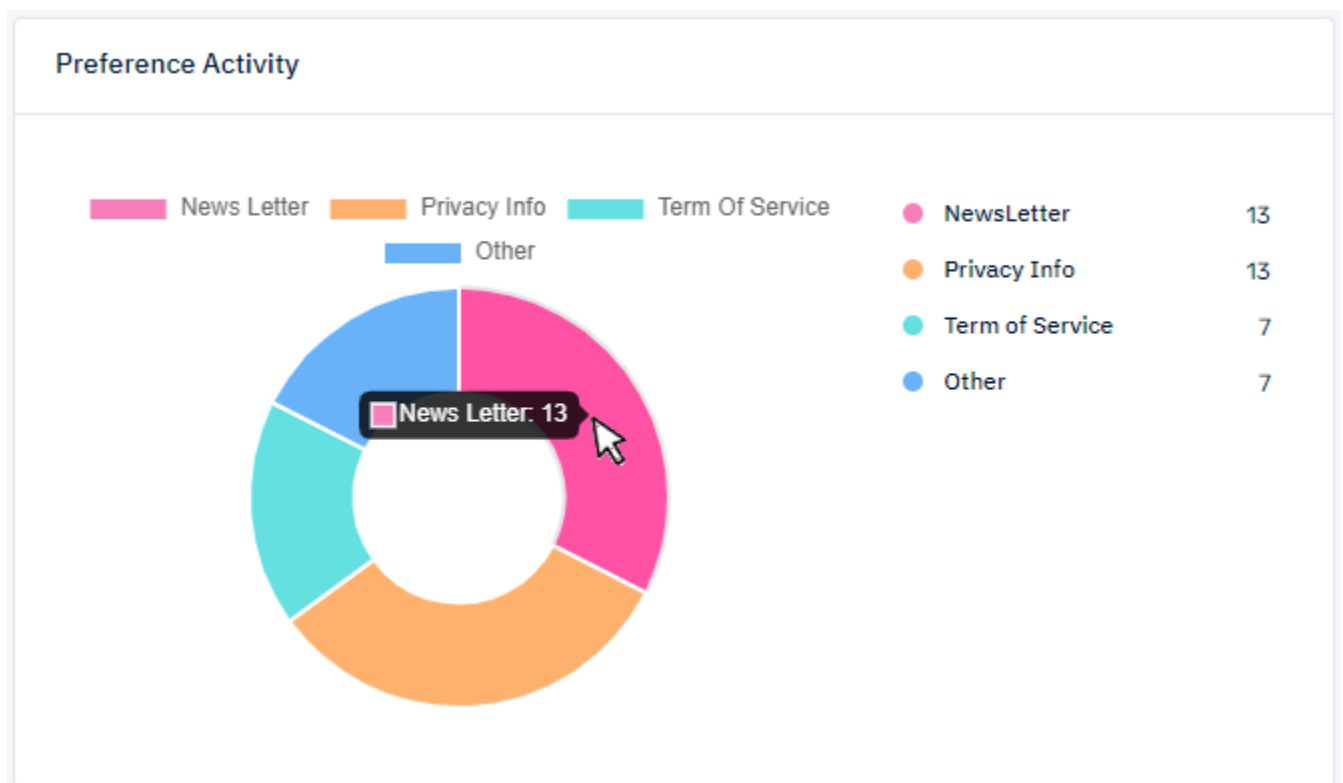
At the top of the consent preference dashboard, it shows the total count of consents as well as the individual count of every opt-in done by the domain users.



For example, the above image shows that 10 users consented through your website, 14 users accepted the legal notices, 13 users subscribed to the newsletter, and so on.

Pie Chart - Preference Activity

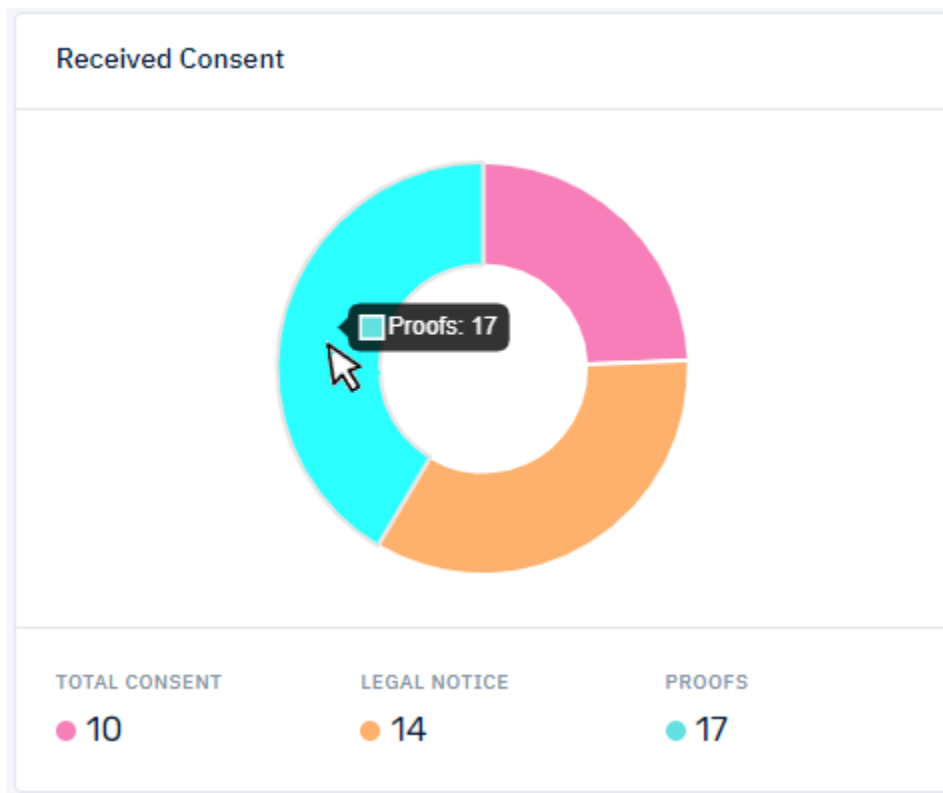
The individual count of preference op-ins by the user on the website is shown as pie chart analytics.



When you hover over each slice of the pie chart, it shows the count of the preference options.

Pie Chart - Received Consent

It also shows the individual count of consents received from the website users as a pie chart analytics.



Consent Preference - JavaScript Documentation

The Consent Preference JavaScript library enables recording the users' consent actions, which are referred to as "subjects".

In this chapter, we are providing a JS library for the above feature.

Setup Script

To install the Consent Preference JS widget, insert your setup code on your website, before closing the `</head>` tag as mentioned in the code below.

```
<script src="https://cdn.primeconsent.com/cmp/consent-preference/production/adzapier-cp-sdk.js" ></script>
```

init Configuration

The `init` function given below defines how the library is initiated instantly and this function is required to be implemented on your web page that has the Consent Preference module.

It is possible to add a callback function as a second parameter of the `init` function. The callback function will be then called once the library has been loaded.

The following are the parameters that you can use inside the `init` configuration object (that is the first parameter on the `init` function which contains the AppID) to customize, for example, the logger or the behavior of the library.

Key	Required	Type	Note
PropID	Required	String	An ID generated for your property within Adzapier Portal
ShowLogs	Optional	Boolean	By default, false
LogType	Optional	String	Use error, log, warn
OwnerID	Optional	String	A unique ID of alpha/numeric value to differentiate a user among multiple users of your account
UseMethodType	Optional	String	GET and POST only (default method POST)

An example code concerning the above parameters is given below.

```
const adzapier = window.CP_SDK_ADZAPIER.init({  
  PropID: 'yourPropID',  
})
```

Connecting the Consent Preference to a Submission Form

To record automatically the consents provided using a submission form, you can either use the `load` function or the `submit` function.

Load Function

The `load` function enables integrating the consent object's fields to input fields of your `form` and automatically recording the consent at submit time.

```
adzapier.load({  
  submitBtnID: 'submit-btn',  
  form: {  
    formID: "form-id",  
    map: {  
      consent: {  
        email: 'your_email',  
        first_name: 'first_name',  
        last_name: 'last_name',  
        data_source: 'data_source',  
        verified: 'verified',  
        auth_id: 'auth_id',  
        country: 'country',  
        ip_address: 'ip_address'  
      },  
      exclude: ['password', 'confirm_password']  
    }  
  },  
},
```

```

consent: {
    legalNotices: [
        {
            identifier: "terms_and_conditions",
            version: 1,
            content: 'https://example.com/term-and-condition'
        },
        {
            identifier: "privacy_policy",
            version: 2,
            content: 'https://example.com/privacy-policy'
        }
    ],

    preferences: [
        { preference: 'terms_and_conditions', allow: true },
        { preference: 'privacy_policy', allow: true },
        { preference: 'other_criterion', allow: true },
    ],

    },

    autodetectIpAddress: true,
}).then(res => {
if (res.success) {
    // success response

```

```

    } else {

        // Error Handle

    }

})

```

The parameters used in the above codes are described below.

Key	Required	Type	Note
submitBtnID	Optional	String or DOM element	If not provided call user using CP_SDK_ADZAPIER.send()
form	No/Yes if consent is not defined	Object	See Form Object
consent	No/Yes if form is not defined	Object	See Consent Object
autodetectIpAddress	Optional	Boolean	A parameter that enables or disables the IP autodetect. 'True' by default

Form Object

Since the form object is optional, an example of the structure of the form object under the `formID` parameter is given below which is passed to `load` and `submit` functions.

Key	Required	Type	Note
formID	Optional	String	The string (ID) or the DOM element of the form from which you want to detect the data
map	Yes, if formID is defined	Object	Map consent attributes directly to the corresponding input's name attribute

consent	No/Yes if <code>formID</code> is not defined	Object	
email	Yes	String	
first_name			
last_name			
data_source	Optional	String	Getting value by name attribute
verified			
auth_id			
country			
ip_address			
exclude	Optional	Array	A list of fields name that we would like to exclude from proofs

Consent Object

An example of the structure of the form object under the `formID` parameter is given below.

Key	Required	Type	Note
consent	Optional	Object	
leagalNotices	Optional	Array	Array containing multiple <code>leagalNotices</code> data
identifier		String	
version		String	Existing value remains until you change

content		String	
proofs	Optional	Array	Array containing multiple proofs data
content		String	
form		String	
preferences	Optional	Array	Set of key-values with user preferences for the consent action
preference		Boolean	<ul style="list-style-type: none"> • 'True' if the user selects the check box. ex: Terms and Conditions. • You can include more preferences with constant values

For more precious understanding, see an example of load function HTML code below.

```
<html>

  <head>

    <title>Consent Preference</title>

  </head>

  <body>

    <form id="form-id">

      <!-- subject -->

      <input type="hidden" value="123ad234" name="ip_address"/>

      <input type="hidden" value="public" name="data_source"/>

      <input type="hidden" value="true" name="verified"/>

      <input type="hidden" value="9045244484" name="auth_id"/>

      <p>

        First name:<br/>
```



```
        <input type="text" name="first_name"/>

    </p>

    <p>

        Last name:<br/>

        <input type="text" name="last_name"/>

    </p>

    <p>

        Country :<br/>

        <input type="text" name="country"/>

    </p>

    <p>

        Email<br/>

        <input type="text" name="your_email"/>

    </p>

    <p>

        Password<br/>

        <input type="password" name="password"/>

    </p>

    <p>

        Confirm Password<br/>

        <input type="password" name="confirm_password"/>

    </p>

    <input type="button" value="Submit" id="submit-btn"/>

</form>
```

```
<script src="http://localhost:5555/adzapier-cp-sdk.js"></script>

<script>

    const adzapier = window.CP_SDK_ADZAPIER.init({

        PropID: '083f0a2a-b70f-4132-ac8f-e326e6885b5d', // Your Current Property

        ShowLogs: true, // Show Console Logs

        OwnerID: '12345'

    })

    adzapier.load({

        submitBtnID: 'submit-btn',

        form: {

            formID: "form-id",

            map: {

                consent: {

                    email: 'your_email',

                    first_name: 'first_name',

                    last_name: 'last_name',

                    data_source: 'data_source',

                    verified: 'verified',

                    auth_id: 'auth_id',

                    country: 'country',

                    ip_address: 'ip_address'

                },

                exclude: ['password', 'confirm_password']

            }

        }

    })

ID
```

```

    },
    consent: {
      legalNotices: [
        {
          identifier: "Terms and Conditions",
          version: 1,
          content: 'https://example.com/terms-and-conditions'
        },
        {
          identifier: "Privacy Policy",
          version: 2,
          content: 'https://example.com/privacy-policy'
        }
      ],
      preferences: [
        { preference: 'terms_and_conditions', allow: true },
        { preference: 'privacy_policy', allow: true },
        { preference: 'other_criterion', allow: true },
      ],
    },
    autodetectIpAddress: true,
  }).then(res => {
    console.log('res', res);
  });

```

```

    })

</script>

</body>

</html>

```

Send Function

The `sendData` function must be used in conjunction with the `load` function and triggers the consent object's recording with the values of the input fields that have been bound by a prior call to the `load` function.

```

function validateForm() {

    if (isValidForm()) {

        // the form has been validated

        adzapier.CP_SDK_ADZAPIER.send();

    }

}

```

Submit Function

The `submit` function enables sending the consent data to Adzapier APIs, that is to record the consent. This submit function can be of two types. 1) Submit with value and 2) Submit by event.

Submit Function with Value

The user can submit the values when the `sendData()` function gets manually called. An example code is given below.

```

adzapier.submit({

    consentData: {

```

```
consent: {  
    email: 'myemail@adzapier.com',  
    first_name: 'Jason',  
    last_name: 'David',  
    data_source: 'private',  
    verified: true,  
    auth_id: 'abc123',  
    country: 'USA',  
    ip_address: '192.168.1.4',  
    additional_data: { // applicable for submit function with value  
        username: 'jason123',  
        status: true,  
        result: 'pass'  
    }  
},  
preferences: [  
    { preference: 'terms_and_conditions', allow: true },  
    { preference: 'privacy_policy', allow: true },  
    { preference: 'other_criterion' allow: true },  
],  
},  
legalNotices: [  
    {  
        identifier: 'term1',
```

```

        version: 1,

        content: 'term1_Content'

    },

    {

        identifier: 'term2',

        version: 2,

        content: 'term2_Content'

    }

],

proofs: [

    {

        content: '{"first_name": "Jason", "last_name": "David", "email": "myemail@adzapier.com" }',

        form: '<form action="/action" method="POST"><p><label>First Name</label><input type="text" name="first_name" /></p><p><label>Last name</label><input type="text" name="last_name" /></p><p><label>E-mail</label><input type="email" name="email" /></p><input type="submit" /></form>'

    }

],

autodetectIpAddress: true,

)

```

Unlike the `load` function, by default the `submit` function sends `consent` object to our server instead of local storage.

Parameters

Key	Required	Type	Note
-----	----------	------	------

Consent	Optional	Object	Fetches the form values directly to the object
Email	Required	String	
first_name	Optional	Object	
last_name			
Country			
data_source			
Verified			
auth_id			
ip_address			
additional_data	Optional	Object	Any key-value pair
leagalNotices	Optional	Array	Array containing multiple <code>leagalNotices</code> data
Identifier		String	
Version		String	Existing value remains until you change
Content		String	
Proofs	Optional	Array	Array containing multiple <code>proofs</code> data
Content		String	
Form		String	
Preferences	Optional	Array	Set of key-values with user preferences for the consent action
Preference		Boolean	<ul style="list-style-type: none"> • 'True' if the user selects the check box. ex: Terms and Conditions. • You can include more preferences with constant values

See an example HTML code of Submit Function with Value below.

```
<html>
```

```
<head>

  <title>Consent Preference</title>

</head>

<body>

<form id="form-id">

  <!-- subject -->

  <input type="hidden" value="123ad234" name="ip_address"/>

  <input type="hidden" value="public" name="data_source"/>

  <input type="hidden" value="true" name="verified"/>

  <input type="hidden" value="9045244484" name="auth_id"/>

  <p>

    First name:<br/>

    <input type="text" name="first_name"/>

  </p>

  <p>

    Last name:<br/>

    <input type="text" name="last_name"/>

  </p>

  <p>

    Country :<br/>

    <input type="text" name="country"/>

  </p>

  <p>

    Email<br/>
```



```

        <input type="text" name="your_email"/>

    </p>

    <p>

        Password<br/>

        <input type="password" name="password"/>

    </p>

    <p>

        Confirm Password<br/>

        <input type="password" name="confirm_password"/>

    </p>

    <input type="button" onclick="submitData()" value="Submit" id="submit-btn"/>

</form>

<script src="http://localhost:5555/adzapier-cp-sdk.js"></script>

<script>

    const adzapier = window.CP_SDK_ADZAPIER.init({

        PropID: '083f0a2a-b70f-4132-ac8f-e326e6885b5d', // Your Current Property

        ShowLogs: true, // Show Console Logs

        OwnerID: '12345'

    })

    function submitData() {

        adzapier.submit({

            consentData: {

```

ID

```

consent: {
    // Provide your form value directly to the object
    email: 'example@adzapier.com', // required Type String
    first_name: 'Rahul', // Optional ;Type Sting
    last_name: 'Sharma', // Optional ;Type Sting
    fullName: 'Rahul Sharma', // Optional ;Type Sting
    data_source: 'private', // Private or Public ; // Optional ;Type
    verified: true, // Email Varified or not // Optional ;Type
    auth_id: '', // auth_id // Optional ;Type Sting
    country: 'india',
    ip_address: '1.02.12',
    additional_data: { // applicable for submit function with value
        username: 'abcd',
        status: true
    }
},
exclude: ['password', 'last_name', 'country'],
preferences: [ // Optional
    {preference: 'terms_and_conditions', allow: true},
    {preference: 'newsletter', allow: true},
    {preference: 'privacy_policy', allow: true},
    { preference: 'your-custom-preference', allow: true }
],

```

Sting

boolean

```

    },
    legalNotices: [ // Optional
        {
            identifier: 'term',
            version: 1,
            content: 'test Content'
        },
        {
            identifier: 'fake',
            version: 123,
            content: 'Fake Content'
        }
    ],
    proofs: [ // Optional
        {
            content: '{ "first_name": "John", "last_name": "Doe", "email": "john@example.com" }',
            form: document.getElementById('form-id').outerHTML
        }
    ],
    autodetectIpAddress: true, // A parameter that enable or disable the ip autodetect. Default to: true
  }, (result) => {
  });

```

```

    }

</script>

</body>

</html>

```

Submit Function by Event

The User can submit the form by an on-click event. An example code is given below.

```

function submitData() {

    adzapier.submit({

        form: {

            formID: "form-id", // Form Element Id Type String; // The string
// (ID) or the DOM element of the form from which you'd like to detect the data

            map: { // optional: map consent attributes directly to the
corresponding

                // input's "name" attribute

                consent: { // Getting Value By Name attribute // required

                    email: 'your_email', // Required; Type String // Name
attribute

                    first_name: 'first_name', // Getting Value By Name
attribute// Optional ; Type Sting

                    last_name: 'last_name', // Optional ;Type Sting Name
attribute

                    data_source: 'data_source', // Optional ; Type Sting
Name attribute

                    verified: 'verified', // Optional ; Type Sting Name
attribute

                    auth_id : 'auth_id', // Optional ; Type Sting Name
attribute

                    country: 'country', // Name attribute

```

```

        ip_address: 'ip_address' // Name attribute
    },
    exclude: ['password', 'confirm_password'] // it will ignore
feild value in proof
    }
},
consent: {
    legalNotices: [
        {
            identifier: "terms_and_conditions",
            version: 1,
            content: 'https://example.com/term-and-condition'
        },
        {
            identifier: "privacy_policy",
            version: 2,
            content: 'https://example.com/privacy-policy'
        }
    ],

    preferences: [
        { preference: 'terms_and_conditions', allow: true },
        { preference: 'news_letter: false },
        { preference: 'other_preference', allow: false },
    ],

```

```

    },
    autodetectIpAddress: true,
  }).then(res => {
    console.log('Func RES:::::::::', res)
  })
}

```

See [Form Object](#) and [Consent Object](#) for the descriptions of the parameters.

Also, see the example below.

```

<html>

  <head>

    <title>Consent Preference</title>

  </head>

  <body>

    <form id="form-id">

      <!-- subject -->

      <input type="hidden" value="123ad234" name="ip_address"/>

      <input type="hidden" value="public" name="data_source"/>

      <input type="hidden" value="true" name="verified"/>

      <input type="hidden" value="9045244484" name="auth_id"/>

      <p>

        First name:<br/>

        <input type="text" name="first_name"/>

      </p>

      <p>

```

```
    Last name:<br/>

    <input type="text" name="last_name"/>

</p>

<p>

    Country :<br/>

    <input type="text" name="country"/>

</p>

<p>

    Email<br/>

    <input type="text" name="your_email"/>

</p>

<p>

    Password<br/>

    <input type="password" name="password"/>

</p>

<p>

    Confirm Password<br/>

    <input type="password" name="confirm_password"/>

</p>

    <input type="button" onclick="submitData()" value="Submit" id="submit-btn"/>

</form>

<script src="http://localhost:5555/adzapier-cp-sdk.js"></script>
```

```

<script>

    const adzapier = window.CP_SDK_ADZAPIER.init({

        PropID: '083f0a2a-b70f-4132-ac8f-e326e6885b5d', // Your Current Property
ID
        ShowLogs: true, // Show Console Logs

        OwnerID: '12345'

    })

    function submitData() {

        adzapier.submit({

            form: {

                formID: "form-id", // Form Element Id Type String; // The string
(ID) or the DOM element of the form from which you'd like to detect the data

                map: { // optional: map consent attributes directly to the
corresponding

                    // input's "name" attribute

                    consent: { // Getting Value By Name attribute // required

                        email: 'your_email', // Required; Type String

                        first_name: 'first_name', // Getting Value By Name
attribute// Optional ; Type Sting

                        last_name: 'last_name', // Optional ;Type Sting

                        data_source: 'data_source', // Optional ; Type Sting

                        verified: 'verified', // Optional ; Type Sting

                        auth_id : 'auth_id', // Optional ; Type Sting

                        country: 'country',

                        ip_address: 'ip_address'

                    },

```



```

    exclude: ['password', 'confirm_password'] // it will ignore
feild value in proof

    }

  },

  consent: {

    legalNotices: [

      {

        identifier: "Terms and Conditions",

        version: 1,

        content: 'https://example.com/terms-and-conditions'

      },

      {

        identifier: "Privacy Policy",

        version: 2,

        content: 'https://example.com/privacy-policy'

      }

    ],

    preferences: [

      { preference: 'terms_and_conditions', allow: true },

      { preference: 'privacy_policy', allow: true },

      { preference: 'other_criterion', allow: true },

    ],

  },

  autodetectIpAddress: true,

```

```

        // Type Boolean

        // Optional ,

        // A parameter that enable or disable the ip autodetect. Default to:
true

    }).then(res => {

        console.log('Func RES::::::::', res)

    })

}

</script>

</body>

</html>

```

Start and Stop the Session Recording manually

To start and stop the session recording manually, follow the code example given below.

```

// Start the session  recording

// Note: Session Recording will auto stop after submitting the consent

adzapier.startRecording();

// Stop the session  recording manually

adzapier.stopRecording();

```

Consent Preference - API documentation

The Consent Preference API enables the functionality of the Consent Record, which is recording the consents provided by the domain users, who are referred to as 'subjects' within the functionality.

Base URL = <https://api.primeconsent.com/api/v1>

Headers

Authentication

To implement the Consent Preference, we provide the following key for the API.

Appid: an ID generated for your account of Adzapier Portal.

See [Generate your App ID](#) to how to generate your App ID.

Content-Type

To send data to our HTTP API, the Content-Type header must be set as mentioned below.

```
Header['Appid'] = 'app_id' // your App ID
```

```
Header['Content-Type'] = 'application/json'
```

API for Consent Preference

To implement the Consent Preference module on your website the following three APIs are provided to you.

Get Consents Records (List) – Method: GET

The following API is to view the list of consent records for a specified period.

```
/consent-preference/get-consent/:pid
```

The following is the accepted query parameters tabulation for each API, the parameters are customizable according to your website.

Parameters

Parameters	Required	Type	Note
pid	Yes	String	Property ID of your selected property

Filter Query-params	Required	Type	Note
limit	Optional	String	Indicates the number of results returned. 10, 20, 30 - Default: 10
page		String	The page to be viewed. Default: 1 (1 st page)
email		String	Filter by email id of the user. It tries to match the partial value with the email id. Default: null
first_name		String	Filter by user's first name. It is case-sensitive. Default: null
last_name		String	Filter by user's last name. It is case-sensitive. Default: null
ip_address		String	Filter by IP address. ex: 59.96.170.213. Default: null.
data_source		String	Filter by the data source. Possible values: Public, Private. Default: Both
start_date		String	Filter by the date of consent created. Returns all consent records between the start_date and end_date. Valid format: 04/17/2021
end_date		String	

The following is an example of the API response for the Consent Records list.

Response

```
{
  "count": 1,
  "response": [
    {
      "id": "57f1713d-b425-4021-b271-262a60c8e0ef",
      "pid": "67a0d68e-94e0-492a-8221-1dabaacd375d",
      "optout": false,
      "country": "USA",
      "owner_id": "01",
      "email": "myemail@adzapier.com",
      "first_name": "Jason",
      "last_name": "David",
      "verified": true,
      "ip_address": "59.96.171.179",
      "data_source": "private", //either public or private
      "additional_data":
        {
          "city": "Princeton",
          "ZIP_Code": "400001"
        },
      "auth_id": "110",
      "legal_notices": [
        {
```

```

      "id": "72f9d235-eb4f-4ee9-96ae-b6c9b2469824",
      "pid": "67a0d68e-94e0-492a-8221-1dabaacd375d",
      "owner_id": "01",
      "consent_id": "57f1713d-b425-4021-b271-262a60c8e0ef",
      "version": 1,
      "identifier": "privacy_policy",
      "content": "<script>",
      "created_at": "2021-07-13T14:08:48.858115Z",
      "updated_at": "2021-07-13T14:08:48.858115Z"
    },
    {
      "id": "7302f00a-f174-469d-92e7-d92a50b8a83b",
      "pid": "67a0d68e-94e0-492a-8221-1dabaacd375d",
      "owner_id": "01",
      "consent_id": "57f1713d-b425-4021-b271-262a60c8e0ef",
      "version": 1,
      "identifier": "terms_and_conditions",
      "content": "<script>",
      "created_at": "2021-07-13T14:08:48.752745Z",
      "updated_at": "2021-07-13T14:08:48.752745Z"
    }
  ],
  "proofs": [
    {

```

```

        "id": "3d18495f-fd1b-4ad0-8594-edfa10fe4025",

        "pid": "67a0d68e-94e0-492a-8221-1dabaacd375d",

        "consent_id": "57f1713d-b425-4021-b271-262a60c8e0ef",

        "content":
        "{\\"firstname\\":\\"Aditya\\",\\"lastname\\":\\"Mishra\\",\\"email\\":\\"gswfsadfsafe42athi@adz
        apier.com\\"}",

        "form": "<form _ngcontent-wfc-c187=\\\"\\\" novalidate=\\\"\\\" id=\\\"registerForm\\\"
        ng-reflect-form=\\\"[object Object]\\\" class=\\\"ng-dirty ng-touched ng-valid</form>",

        "created_at": "2021-07-13T14:08:48.872844Z",

        "updated_at": "2021-07-13T14:08:48.872844Z"

    }

],

"preferences": [

    {

        "id": "f11d5c65-ab1a-4949-8ba7-c1692aa8dc30",

        "pid": "67a0d68e-94e0-492a-8221-1dabaacd375d",

        "consent_id": "57f1713d-b425-4021-b271-262a60c8e0ef",

        "preference": "terms_and_conditions",

        "allow": true,

        "created_at": "2021-07-13T14:08:48.919956Z",

        "updated_at": "2021-07-13T14:08:48.919956Z"

    },

    {

        "id": "ff16fa2a-bab0-4c81-abe5-dca9d8a65e0a",

        "pid": "67a0d68e-94e0-492a-8221-1dabaacd375d",

        "consent_id": "57f1713d-b425-4021-b271-262a60c8e0ef",

```

```

        "preference": "privacy_policy",

        "allow": true,

        "created_at": "2021-07-13T14:08:48.950719Z",

        "updated_at": "2021-07-13T14:08:48.950719Z"

    }

],

    "created_at": "2021-07-13T14:08:48.705921Z",

    "updated_at": "2021-07-13T14:08:48.705921Z"

}

],

"status": 200

}

```

Create Consent – Method: POST

The following is the API when you create a manual consent record.

`/consent-preference/create/:pid`

User-info Parameters

Parameters	Required	Type	Note
owner_id	Optional	String	Get values from the user
email	Required	String	
first_name	Optional	String	
last_name	Optional	String	

country	Optional	String	
ip_address	Optional	String	
authID	Optional	String	
data_source	Optional	Boolean	Select either Public or Private
verified	Optional	Boolean	Either true or false
optout	Optional	Boolean	Either true or false
additional_data	Optional	Object	Any key-value pair

Proof Parameters

Parameters	Required	Type	Note
proofs	Optional	Array	Array of objects contains Proofs data. You can include more Proofs
content		String	
form		String	

Legal Notice Parameters

Parameters	Required	Type	Note
legal_notices	Optional	Array	Array of objects contains Legal Notices data. You can include more Legal Notices
identifier		String	
version		String	

content

String

Preference Parameters

Parameters	Required	Type	Note
preferences	Optional	Array	Array of objects contains Preferences data. You can include more Proofs
preference		String	You can include more preferences with constant values

API Request Payload

```
{  
  "owner_id": "-owner-id",  
  "email": "myemail@adzapier.com",  
  "first_name": "Jason",  
  "last_name": "David",  
  "verified": true,  
  "optout": false,  
  "country": "USA",  
  "auth_id": "user-auth-id",  
  "data_source": "private", //'private' or 'public'  
  "additional_data":  
    {  
      "city": "mumbai",  
      "ZIP_Code": "400001"  
    },  
}
```

```
"proofs": [  
  {  
    "content": "proof_1a",  
    "form": "form_1a"  
  },  
  ],  
"legal_notices": [  
  {  
    identifier: "terms_and_conditions",  
    version: 1,  
    content: 'https://example.com/terms-and-conditions'  
  },  
  {  
    identifier: "privacy_policy",  
    version: 2,  
    content: 'https://example.com/privacy-policy'  
  },  
  ],  
"preferences": [  
  { preference: 'terms_and_conditions', allow: true },  
  { preference: 'privacy_policy', allow: true },  
  { preference: 'others', allow: true },  
  ],  
"ip_address": "127.0.0.1"
```

```
}
```

Response

```
{  
  "consent_id": "bad07d89-4a15-4088-837b-40e4d3b2d6ee",  
  "message": "Record Created",  
  "status": 201,  
  "timestamp": "2021-07-27T16:38:54.873834124Z"  
}
```

Get Consent by ID – Method: GET

The Get Consent by ID function returns the details of a single consent that you select by the Consent ID.

```
/consent-preference/get-consent-by-id/:pid/:id
```

Parameters

Parameters	Required	Type	Note
pid	Yes	String	Property ID of your selected property
id	Yes	String	ID of the consent: generated when hits the API

Response

```
{  
  "id": "bad07d89-4a15-4088-837b-40e4d3b2d6ee" //Consent ID  
  "owner_id": "011",  
  "email": "myemail@adzapier.com",  
}
```

```
"first_name": "James",

"last_name": "David",

"verified": true,

"optout": false,

"country": "UAS",

"data_source": "Private",

"additional_data":

    {

        "city": "Princeton",

        "ZIP_Code": "400001"

    },

"proofs": [

    {

        "content": "content_1a",

        "form": "form_1a"

    },

],

"legal_notices": [

    {

        identifier: "terms_and_condition",

        version: 1,

        content: 'https://example.com/term-and-condition'

    },

    {
```

```

        identifier: "privacy_policy",

        version: 2,

        content: 'https://example.com/privacy-policy'

    }

],

"preferences": [

    { preference: 'terms_and_conditions', allow: true },

    { preference: 'privacy_policy', allow: true },

    { preference: 'other_criteria', allow: false },

],

"ip_address": "127.0.0.1"

}

```

API for Legal Notices

The following are APIs to implement the Legal Notices functionality. A few examples of API request payload and the responses are provided.

Get Legal Notice Records (List) - Method: GET

The following is the API to get the list of Legal Notices shown on the Consent Details page. So, the values of `pid` and `consent_ID` are taken from that consent.

```
/consent-preference/legal-notices/:pid/:consentID
```

```
Header['Appid'] = // your appID
```

Since the Legal Notice list is shown within the Consent Details page, some values are taken from the respective consent.

Parameters

Parameters	Required	Type	Note
pid	Yes	String	Property ID of your selected property
id	Yes	String	ID of the consent given: generated when hits the API

Filter Query-params	Required	Type	Note
limit	Optional	String	Indicates the number of results returned. 10, 20, 30 - Default: 10
page		String	The page to be viewed. Default: 1 (1 st page)

Response

```
{
  "count": 2,
  "response": [
    {
      "id": "0551623e-cb2d-4891-adcb-381963bb7a96",
      "pid": "b41656d4-24a2-416e-9d58-1a06a0746ae7",
      "owner_id": "0011",
      "consent_id": "bf953fae-47d3-4e1e-865c-617908db241a",
      "version": 1,
      "identifier": "terms_and_conditions",
      "content": "https://example.com/terms-and-conditions",
      "created_at": "2021-06-21T12:58:50.256872Z",
      "updated_at": "2021-06-21T22:08:11.256872Z"
    }
  ]
}
```

```

    },
    {
      "id": "05f4da35-24f1-4b54-b1b7-5e4518bc90b2",
      "pid": "184176c8-2519-43ea-a019-8b3ba400fe17",
      "owner_id": "0011",
      "consent_id": "bf953fae-47d3-4e1e-865c-617908db241a",
      "version": 2,
      "identifier": "privacy_policy",
      "content": "https://example.com/privacy-policy",
      "created_at": "2021-06-24T09:48:17.135937Z",
      "updated_at": "2021-06-24T09:48:17.135937Z"
    }
  ],
  "status": 200
}

```

Create Legal Notice - Method: POST

The legal notices opt-in or opt-out by the user is called create preference.

`/consent-preference/legal-notices/:pid/:consentID`

`Header['Appid'] = // your AppID`

Query Parameters

Parameters	Required	Type	Note
------------	----------	------	------

legal_notices	Required	Array	Array of objects containing legal notices data. You can include more Legal Notices
identifier	Optional	String	
version	Optional	String	
content	Optional	String	

Request Payload

```
[
  {
    identifier: "terms_and_conditions",
    version: 1,
    content: 'https://example.com/terms-and-conditions'
  },
  {
    identifier: "privacy_policy",
    version: 1,
    content: 'https://example.com/privacy-policy'
  }
]
```

Response

```
{
  "message": "Record Created",
  "status": 201,
}
```

Get Legal Notice by ID - Method: GET

The opt-in and opt-out details of the legal notices under each consent detail are shown using this API.

```
/consent-preference/legal-notice-by-id/:pid/:consentID/:id
```

```
Header['Appid'] = //appID
```

Parameters

Parameters	Required	Type	Note
pid	Yes	String	Property ID of your selected property
consent_id	Yes	String	ID of the Consent that contains the selected Legal Notice
id	Yes	String	ID of the Legal Notice selected to be viewed

Response

```
{  
  "response": {  
    "id": "07688650-a2a1-4e25-948a-59dd34a2c398",  
    "pid": "63208a7d-771b-4cfd-8859-b254ca5aae23",  
    "owner_id": "9900",  
    "consent_id": "a8acb22b-c548-45eb-948b-3cf23b9413d0",  
    "version": 1,  
    "identifier": "terms_and_conditions",  
    "content": "<content>",  
    "created_at": "2021-05-21T14:58:09.652965Z",  
  }  
}
```

```

    "updated_at": "2021-05-21T14:58:09.652965Z"
  },
  "status": 200
}

```

API for Proofs

The APIs for Proofs are given below, and a few examples of API request payload and the responses are provided.

Get Proofs Records (List) - Method: GET

The following is the API to get the list of Proofs shown on the Consent Details page. So, the values of `pid` and `consent_id` are taken from the particular consent.

```
/consent-preference/proofs-list/:pid/:consentID
```

```
Header['Appid'] = //appID
```

Parameters

Parameters	Required	Type	Note
<code>pid</code>	Yes	String	Property ID of your selected property
<code>id</code>	Yes	String	ID of the consent: generated when hits the API

Filter Query-params	Required	Type	Note
<code>limit</code>	Optional	String	Indicates the number of results returned. 10, 20, 30 - Default: 10

page

String The page to be viewed. Default: 1 (1st page)

Response

```
{
  "count": 2,
  "response": [
    {
      "id": "4bb1b901-7fa5-4d24-9ccb-ea3c75a3cfbe",
      "pid": "083f0a2a-b70f-4132-ac8f-e326e6885b5d",
      "consent_id": "ff827a25-756b-422d-90bf-620933e13e49",
      "content": "sample_content1",
      "form": "form_sample1",
      "created_at": "2021-06-24T10:29:32.212923Z",
      "updated_at": "2021-06-24T10:29:32.212923Z"
    },
    {
      "id": "622f727b-93c1-4a97-8fed-5c84ff7995e7",
      "pid": "083f0a2a-b70f-4132-ac8f-e326e6885b5d",
      "consent_id": "ff827a25-756b-422d-90bf-620933e13e49",
      "content": "sample_content2",
      "form": "form_sample2",
      "created_at": "2021-06-24T10:34:01.608246Z",
      "updated_at": "2021-06-24T10:34:01.608246Z"
    }
  ]
}
```

```
],  
  "status": 200  
}
```

Create Proof - Method: POST

Submitting the proof details by the user is called create preference.

`/consent-preference/proofs/:pid/:consentID`

`Header['Appid'] = // your appID`

Query Parameters

Parameters	Required	Type	Note
proofs	Required	Array	Array of objects containing proofs data. You can include more Proofs
content	Optional	String	
form	Optional	String	

Request Payload

```
[  
  {  
    "content": "sample_content",  
    "form": "sample_form"  
  },  
  {  
    "content": "sample_content2",
```

```
"form": "sample_form2"
```

```
}
```

```
]
```

Response

```
{
```

```
  "message": "Record Created",
```

```
  "status": 201,
```

```
}
```

Get Proof by ID - Method: GET

The proof details submitted by the user under each consent are shown using this API.

```
/consent-preference/proof-by-id/:pid/:consentID/:id
```

```
Header['Appid'] = //your appID
```

Parameters

Parameters	Required	Type	Note
pid	Yes	String	Property ID of your selected property
consent_id	Yes	String	ID of the Consent that contains the selected Proof
id	Yes	String	ID of the Proof selected to be viewed

Response

```
{
```

```
  "response": {
```

```

      "id": "ef4b83d2-ccd0-4f20-8d15-0efd2a6699e0",
      "pid": "083f0a2a-b70f-4132-ac8f-e326e6885b5d",
      "consent_id": "07b3757e-60d2-48d3-bb73-47dd8de8db99",
      "content": "test",
      "form": "test",
      "created_at": "2021-06-24T10:11:59.190368Z",
      "updated_at": "2021-06-24T10:11:59.190368Z"
    },
    "status": 200
  }
}

```

API for Preference

The APIs for Preference are given below, and a few examples of API request payload and the responses are provided.

Get Preference Records (List) - Method: GET

The following is the API to get the list of Preferences shown on the Consent Details page. So, the values of `pid` and `consent_id` are taken from the particular consent.

```
/consent-preference/preference/:pid/:consentID
```

```
Header['Appid'] = //appID
```

Parameters

Parameters	Required	Type	Note
pid	Yes	String	Property ID of your selected property

id	Yes	String	ID of the consent given: generated when hits the API
----	-----	--------	--

Filter Query-params	Required	Type	Note
limit	Optional	String	Indicates the number of results returned. 10, 20, 30 - Default: 10
page		String	The page to be viewed. Default: 1 (1 st page)

Response

```
{
  "count": 2,
  "response": [
    {
      "id": "f4067ece-4978-414b-9ba3-60ac6d602260",
      "pid": "67a0d68e-94e0-492a-8221-1dabaacd375d",
      "consent_id": "0e8a7daf-fb9b-4927-b74e-1875676b3aae",
      "preference": "terms_and_conditions",
      "allow": true,
      "created_at": "2021-07-21T07:45:52.940978Z",
      "updated_at": "2021-07-21T07:45:52.940978Z"
    },
    {
      "id": "16167rfg-4978-546b-9rt3-23dg6d602260",
      "pid": "67a0d68e-94e0-492a-8221-1dabaacd375d",
      "consent_id": "0e8a7daf-fb9b-4927-b74e-1875676b3aae",
```



```

        "preference": "privacy_policy",

        "allow": true,

        "created_at": "2021-07-21T07:45:52.940978Z",

        "updated_at": "2021-07-21T07:45:52.940978Z"

    }

],

"status": 200

}

```

Create Preference - Method: POST

The preference opt-in or opt-out by the user is called create preference.

`/consent-preference/preference/:pid/:consentID`

Header['Appid'] = //appID

Query Parameters

Parameters	Required	Type	Note
preferences	Required	Array	Array of objects containing proofs data. You can include more Proofs
preference	Optional	String	You can include more preferences with constant values
preference	Optional	String	

Request Payload

```

[

    {

```

```

    "allow": true,

    "preference": "privacy_policy"

  },

  {

    "allow": true,

    "preference": "terms_and_conditions"

  }

]

```

Response

```

{

  "message": "Record Created",

  "status": 201,

}

```

Get Preference by ID - Method: GET

The opt-in and opt-out details of the preference under each consent detail are shown using this API.

`/consent-preference/preference-by-id/:pid/:consentID/:id`

`Header['Appid'] = //appID`

Parameters

Parameters	Required	Type	Note
pid	Yes	String	Property ID of your selected property

consent_id	Yes	String	ID of the Consent that contains the selected Preference
id	Yes	String	ID of the Preference selected to be viewed

Response

```
{
  "response": {
    "id": "f4067ece-4978-414b-9ba3-60ac6d602260",
    "pid": "67a0d68e-94e0-492a-8221-1dabaacd375d",
    "consent_id": "0e8a7daf-fb9b-4927-b74e-1875676b3aae",
    "preference": "terms_and_conditions",
    "allow": true,
    "created_at": "2021-07-21T07:45:52.940978Z",
    "updated_at": "2021-07-21T07:45:52.940978Z"
  },
  "status": 200
}
```

TCPA (Telephone Consumer Protection Act)

INTRODUCTION

A brief introduction to Adzapier and how it assists its clients with TCPA (Telephone Consumer Protection Act) compliance.

With GDPR (General Data Protection Regulation) and CCPA (California Consumer Privacy Act) compliance, Adzapier has created a secure and user-friendly application to gather and manage the consent and preferences of visitors to your website.

For you folks, we have some happy news. We now allow our customers to follow TCPA (Telephone Consumer Protection Act) regulations and use their visitor's data with prior consent. By doing this, we assist our clients in avoiding legal trouble and penalties.

The Reason TCPA has Evolved.

Why does TCPA enter the picture and what the justification is?

Early in the 1980s, there were hardly any phones available. There were landline phones in some homes. When we first got the calls, we were so excited that we rushed to the phone to pick them up. But as time went on, mobile phones became increasingly commonplace. Additionally, a lot of businesses obtain our contact information from various sources and call us at random for marketing purposes, constantly bothering us. Additionally, they use SMS to sell their items. These businesses occasionally annoy people at any moment, which hinders their work and irritates the client.

Describe TCPA

Complete details on TCPA.

To control telemarketing calls, auto-dialed calls, prerecorded calls, artificial voices, text messages, and faxes, President George H.W. Bush signed the Telephone Consumer Protection Act (TCPA) into law in 1991 by United State Congress. TCPA aids in defending consumers against opportunistic marketing calls. The Telephone Consumer Protection Act forbids calling any customers whose numbers are listed on the Do-Not-Call registry. The FTC (Federal Trade Commission) supervises the Do Not Call register on behalf of consumers. The FCC (Federal Communications Commission) has the authority to impose TCPA rules and regulations. By TCPA legislation, businesses dealing with customer data, such as customer phone numbers, are required to obtain the prior consent of the customer they intend to contact for marketing purposes.

TCPA guidelines and Regulations

List all TCPA rules and regulations.

1. Prior consent from customers should be obtained by businesses before contacting them.
2. Without the customer's permission, the TCPA forbids the use of ATDS (Automatic Telephone Dialing System).
3. Without the customer's consent, the TCPA forbids the use of pre-recorded telemarketing messages.
4. Marketing calls to numbers on the Federal DNC (Do Not Call) registry are not permitted under TCPA.
5. Callers must identify themselves, their business, and their contact information, according to TCPA requirements.
6. The best time range is between 8 am and 9 pm in the customer's time zone, according to the TCPA guidelines. Companies should contact customers within this ideal period.
7. According to TCPA, companies should help customers who want to opt-out of being solicited.
8. Companies are required by TCPA to keep a record of customers' choice to opt-in and opt-out using the DNC (Do Not Call) registry.
9. Companies are permitted to transmit communications in some exceptional circumstances without obtaining consent first. These notifications resemble alarms about potential fraud or warnings to not share your OTP with others, or they may serve as a reminder for taking urgent medications or getting vaccinated.

Key Advantages of TCPA

What benefits will a business experience if it complies with TCPA?

Everybody has a rigid schedule these days. Nobody has enough free time to talk on the phone and answer these marketing calls whenever they come in. TCPA regulations benefit these clients for this.

1. Defend the consumer against invasive calls
2. Safeguarding users against unsolicited SMS, Faxes.
3. It can shield you from legal action for personal data breaches.
4. You might avoid millions of fines, by adhering to TCPA.
5. TCPA aids in boosting security.
6. TCPA facilitates better communication.

How Adzapier assists its clients in maintaining TCPA

Detailed information on how Adzapier assists its clients in adhering to TCPA and using their visitor's data.

Any corporation that violates the TCPA will incur excessive costs as a result. TCPA infractions can result in fines of up to 500\$, and if the violation is committed deliberately, they can result in fines of up to 1500\$.

Oh my God, I hope I didn't terrify you. You are safe here, so don't be alarmed. The Adzapier Family will assist you in maintaining compliance with TCPA.

We will record and retain the session when your website visitors give their consent. You can say, "Hey, here is the proof with your information," If any of your visitors claim that you violated their privacy by calling or texting them for any commercial or marketing purposes. With your visitors' information, we are providing you a Consent Certificate (which you can

download from the portal [Click here to know how to download the Consent Certificate](#)) and a Screenshot of the experience so that you have adequate evidence to defend against legal action.