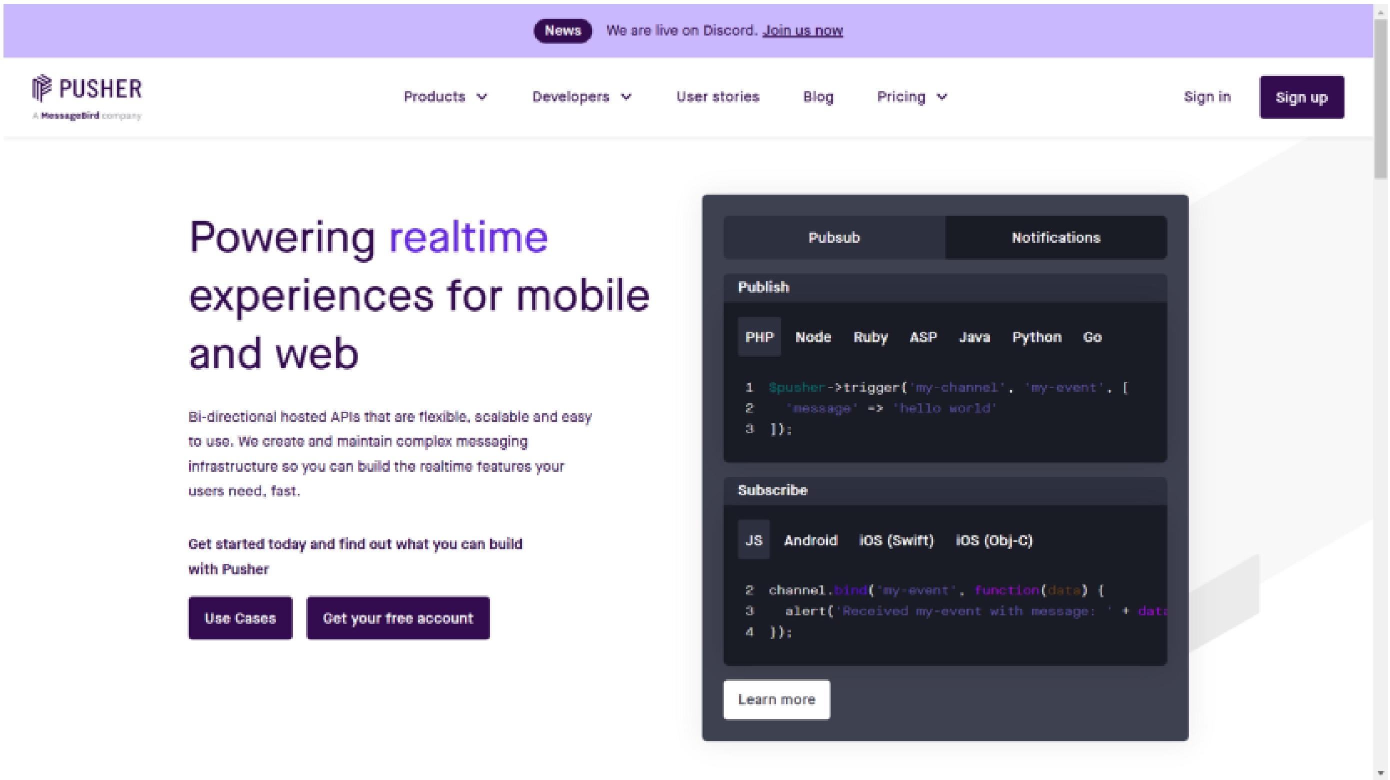
Dasika Sri Bhuvana Vaishnavi

Oct 31 · 3 min read · Listen

Save

Pusher’s Channels

A Comprehensive tutorial on getting started with Pusher’s Channels



Powering realtime experiences for mobile and web

Bi-directional hosted APIs that are flexible, scalable and easy to use. We create and maintain complex messaging infrastructure so you can build the realtime features your users need, fast.

Get started today and find out what you can build with Pusher

```
$pusher->trigger('my-channel', 'my-event', [
  'message' => 'hello world'
]);
```

```
channel.bind('my-event', function(data) {
  alert('Received my-event with message: ' + data
});
```

Channel in a nutshell

Each programme can support any number of channels, and users can pick and choose which ones they want to receive.

Channels provide:

- **Filtering Data:** It’s up to the individual user to decide which channels they want to to receive information on “top-secret-projectX.”

Channel Types:

- Public channels can be subscribed to by anyone who knows their name
- Private channels should have a `private-` prefix. They introduce a mechanism which lets your server control access to the data you are broadcasting
- Private encrypted channels should have a `private-encrypted-` prefix. They extend the authorization mechanism of private channels, adding encryption of the data payloads so that not even Pusher can get access to it without authorization.
- Presence channels should have a `presence-` prefix and are an extension of private channels. They let you ‘register’ user information on your subscription, and let other members of the channel know who’s online
- Cache channels remember the last published message and deliver it to clients when they subscribe. The cache channel is available in public, private, and private-encrypted modes.

Channel Naming :

Channel names should only have letters (both small and capital), numbers, and the following punctuation: `_ - = @ , . ;`

```
foo-bar_1234@=,.;
```

`#` is a reserved character for internal use by Pusher Channels. Applications can’t create channel names containing `#`.

Accessing channels

If a channel is already part of a subscription, it can be found via its name in the channelNameString

The name of the channel to retrieve

Implementing the SDK’s Quickstarts

The pusher protocol lets you use software development kits (SDKs), which makes it easier to use in many different settings. Following is a list of software development kits (SDKs) and the environments in which they can be used to implement the pusher protocol.

> *All these links point to the official documentation to the pusher protocol*

JavaScript quick start

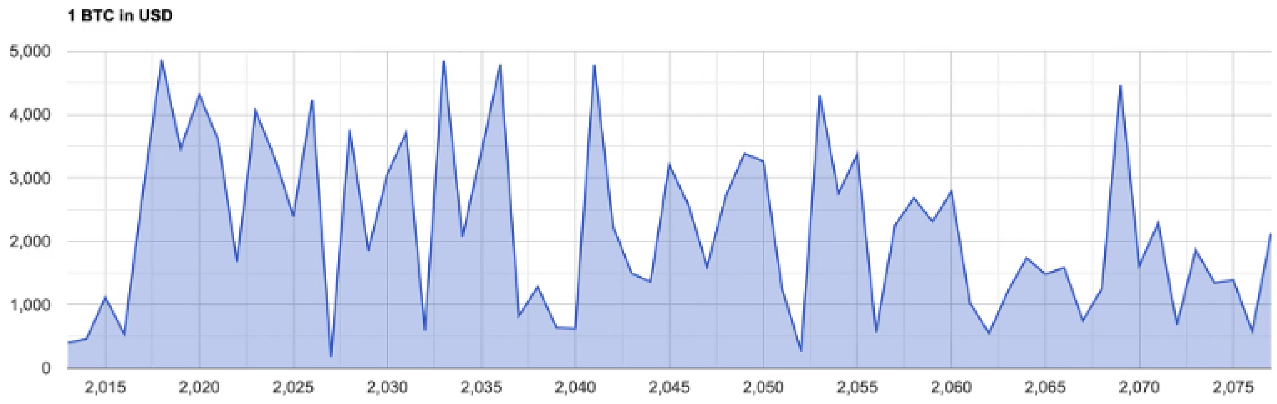
iOS quick start

Android quick start

Flutter quick start

React Native quick start

Usecase Quick starts



JavaScript realtime chart quick start

With JavaScript realtime chart quick start you can do the following

> *Get your free API keys*
>
> *Create your webpage*
>
> *Trigger events from your server*

JavaScript realtime user list quick start

With JavaScript realtime user list art you can do the following

Create your webpage

Create your authorization endpoint server