

# Using LavaSDK

## The Backend

☹️ The Problem : Making calls to API providers is computationally expensive. Creating a backend application requires generating a private key locally and maintaining a sufficient balance to make calls on multiple chains.

✓ The Solution : Lava simplifies multi-chain backend development with free [subscriptions](#) on all projects through our gateway. Lava allows any user to sign up to the gateway. A default subscription is provided courtesy of Lava and requires no additional configurations. This subscription is rate-limited but should be substantial enough for many use cases.

Get Started : You can get started right from the [Lava Gateway](#) ! We recognize that a hosted solution for project/subscription management is not ideal for all users and some may want more granular control of subscriptions or limits. Users who are interested in self-custodial project management should learn more about [subscriptions](#) .

## ↩ Recommended Flow

Although you can subscribe your private key from the commandline, the easiest way to get started is through Lava's Gateway.

1. Sign up to the Gateway. [Register now](#)
2. if you haven't already!
3. Create a Project and Select your APIs!
4. Go to Settings -> Project Keys
5. Click Add New Key
6. and decide whether you want to Generate a new Key
7. or Add a public Key
8. Give a moment for your key to sync!
9. Install the SDK into your project -> `npm install @lavanet-lava-sdk`
10. Select any of your API accesses (LavaSDK -> Use in BackEnd
11. ) & copy & paste your code snippet into your code.

## Manual Key Creation

A private-key, public-key pair can also be generated using our [lavad](#) binary. The [installation guide](#) can be very helpful for users looking to use `lavad` . For more information on how to create a lava address consult the [existing documentation](#) .

If you already have `lavad` installed - you can easily generate a lava address by entering the following command: `lavad keys add`

If you have an address that exists and no longer remember its private key, you can always use `lavad keys export-unarmored-hex --unsafe` to see your private key printed as a string. Do not share the output with anyone!

Your private-key, public-key pair can be used for backend development provided you have a subscription! [Edit this page](#) [Previous](#) [Getting Started](#) [Next](#) [Frontend Use](#)