Using LavaSDK

The Frontend

☠The Problem : Using private keys on the frontend without special provisions is inherently unsafe. In ordinary development scenarios, private keys from a user can be leaked through the browser.

✅The Solution : Lava uses a unique solution calledbadges to solve these limitations. Abadge consists of several parts and is used in lieu of a private key. It must, however, be signed by an external server that holds the relevant authorizing party's private key. Thedefault Badge Server is hosted by Lava and requires no additional configurations.

Get Started : You can get started right from theLava Gateway ! We recognize that a hosted solution is not ideal for every use case. Users who are interested in accomplishing the highest levels of decentralization may run their ownbadge server .

↪ Recommended Flow

Although you can host your own badge server, 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. Open an API and click LavaSDK.
5. Install the SDK into your project ->npm install @lavanet-lava-sdk
6. Copy & paste your code snippet into your code.

Badges

Usage ⚙

Badges are objects passed to the SDK instance which allow a user to forgo the usage of private keys. A badge has the following format:

const myBadge =

{ badgeServerAddress :

"https://badges.lavanet.xyz" ,

// Or your own Badge-Server URL projectId :

""

//input your project ID from the Gateway or custom setup } ; A user can initialize the SDK using a badge instead of a privatekey

const lavaSDK =

await

LavaSDK . create ({ badge : myBadge , chainID :

"LAV1" , rpcInterface :

"rest" , }) ; And make calls all the same - no privatekeys exposed!

const info =

await lavaSDK . sendRelay ({ method :

"GET" , url :

"/node_info" , }) ; That's it. For most users, there is no additional information required to successfully use LavaSDK on the front end! Edit this page Previous Backend Use_ Next Examples_