2

Set up the SDK

After you have installed the SDK, you first need to set it up. To get started, you have to instantiate and configure the LI.FI SDK:

```

Copy import{ LiFi }from'@lifi/sdk' constlifi=newLiFi({ integrator:'Your dApp/company name' })

// or

const{LiFi}=require("@lifi/sdk") constlifi=newLiFi({ integrator:'Your dApp/company name' })

```

The optionalconfig parameter can be used to pass custom configuration to the SDK:

```

Copy typeConfigUpdate={ apiUrl?:string apiKey?:string rpcs?:Record multicallAddresses?:Record defaultExecutionSettings?:ExecutionSettings defaultRouteOptions?:RouteOptions }

```

TheapiUrl defines which backend should be called. This only has to be edited when accessing a dedicated API endpoint.

If you are interested in a dedicated API endpoint for your service, reach out via our Discord or file an issue in our repositoryhere . TheapiKey allows you to authenticate on the API. This will give you access to custom rate limits. Check ourAPI documentation for information on how to apply for an API key.

rpcs ormulticallAddresses can be passed if custom endpoints/addresses should be used, for example to prevent rate limiting.

TheRouteOptions are explained in more detail here . An explanation ofExecutionSettings is available here .

Node.js Compatibility

SDK uses native Fetch API and in Node.js 16 and lower versionsfetch is not natively included, so it needs to be added manually. To provide access to Fetch API you need to patchglobal object usingnode-fetch orcross-fetch .

Here is anexample using cross-fetch:

```

Copy importfetch,{ Headers,Request,Response }from'cross-fetch'

if(!globalThis.fetch) { constglobalThisAny:any=globalThis globalThisAny.fetch=fetch globalThisAny.Headers=Headers globalThisAny.Request=Request globalThisAny.Response=Response }

```

Last updated5 months ago On this page Was this helpful?Export as PDF