Get Started with Avail

Start Operating

To run an Avail light client, simply install the Avail CLInpm package:

npm

i

-g

@availproject/cli Then, run:

avail

lc

up That's it!

If you're seeking more control over the configuration or are interested in running a full node, check out the deployment guides below.

Action Tokens Needed Staking Technical Skill Required Client BinaryRun a Light Client No No Basic v1.7.6 (opens in a new tab) Run a Full Node / Validator No No Moderate v1.10.0.0 (opens in a new tab)

Start Building

The Avail API offers a robust set of features that enable application developers to interact seamlessly with the Avail network. This guide provides a curated list of example code snippets in JavaScript/TypeScript, Rust, and Go. These examples cover essential tasks such as establishing a connection to a local node, querying data from the network, and executing transactions on the blockchain.

Before diving into the examples, ensure you have a local Avail node running or configure your settings to align with an existing network.

Detailed instructions on how to execute these examples are available in the Avail GitHub repository: *TypeScript Instructions (opens in a new tab) * Go Instructions (opens in a new tab) * Rust Instructions (opens in a new tab)

Sample Workflow

Follow these steps to understand a sample end-to-end flow of data availability in Avail. Examples are provided for JavaScript, Go, and Rust.

Step 1: Establishing Connection

Before you can interact with the Avail network, you need to establish a connection to a node.

Language Example Link JavaScript Establishing Connection (opens in a new tab) Go Establishing Connection (opens in a new tab) Rust Headers (opens in a new tab)

Step 2: Submitting Data

Once connected, the next step is to submit data to the network. This could be any blob of data that you want to make available.

Language Example Link JavaScript Submitting Blob Data (opens in a new tab) Go Submitting Blob Data (opens in a new tab) Rust Submitting Blob Data (opens in a new tab)

Step 3: Dispatching Data Root (Optional)

If you need to dispatch the data root, you can do so at this stage. This is optional and depends on your use case.

Language Example Link JavaScript Dispatching Data Root (opens in a new tab) Rust Submit Data and Dispatch Data Root (opens in a new tab)

Step 4: Querying Data and Proofs

After submitting your data, you might want to query it to ensure it's been properly stored and is retrievable.

Language Example Link JavaScript Querying Data Proof (opens in a new tab) Go Querying Data Proof (opens in a new tab) Rust Democracy External (opens in a new tab)

Step 5: Monitoring and Metrics

Finally, you can monitor the network and your data by listening for new blocks and other metrics.

Language Example Link JavaScript Listening for New Blocks (opens in a new tab) Go Listening for New Blocks (opens in a new tab)

API Examples

The following table lists various code examples for interacting with the Avail network. These examples are organized by functionality and the supported programming languages.

Functionality Supported Languages Data Submission TypeScript (opens in a new tab) ,Go (opens in a new tab) ,Rust (opens in a new tab) Dispatching Data Root TypeScript (opens in a new tab) Creating Application Key TypeScript (opens in a new tab) ,Rust (opens in a new tab) Asset Transfer TypeScript (opens in a new tab) ,Go (opens in a new tab) Querying Proof/Data Proof TypeScript (opens in a new tab) ,Go (opens in a new tab) Querying Application Data TypeScript (opens in a new tab) Event Monitoring (Blocks/Data Submission) TypeScript (opens in a new tab) ,Go (opens in a new tab) ,Go (opens in a new tab) Network Connection & Info TypeScript (opens in a new tab) ,Go (opens in a new tab) Internal Operations Go (opens in a new tab) Rust + & Substrate Examples VariousRust (opens in a new tab) examples like Data Availability Bridge Actor, Democracy External, etc.