AppIds on Avail DA?

DON'T CARE ABOUT THEORY?

If you are just looking for the easiest way to create your own AppID on Avail to continue building, you can skip to this page .
LOOKING FOR PROGRAMMATIC INSTRUCTIONS?

You can check out ourAPI reference for the same.

Introduction

As a general-purpose base layer, Avail is designed to support many modular chains at the same time, providing consensus and data availability to all of them simultaneously.

How does this work? Avail headers contain an index that allows a given modular chain (or "application" in Avail terminology) to determine and downloadonly the sections of a block that have data for that particular application.

This has significant benefits, including:

- Modular applications are mainly unaffected by other uses of the
- base layer at the same time.
- Block sizes can increase without requiring applications to fetch
- more data because they don't need to fetch the whole block, only what's
- relevant to them.
- This filtering is done using the "application id" (AppId).

Data availability sampling is still done on the entire block, however--this is the process where clients sample tiny parts of the block at random to verify availability.

Consider a random block on Avail DA

It might contain data blobs from a variety of different rollups pertaining to their different execution environments. Think of the EVM, the SVM, Stackr, ZKsync chains, OP stack, and many more. If all of this data is randomnly strewn about in a block, it would be tedious for an a rollup to parse through all of it just to fetch the data it needs.

Now think of the same data neatly arranged into its own sections

But what if all of that same data in the same block was organised into different sections, stored alongside it's peers. Each of these'sections' would be identified by anAppID . A developer now does not need to parse through the entire block to find the data they need, they can simply query data from theAppID they are interested in.

These'sections'

are flexible

All of the rollups running on Avail DA probably won't submit data in all of the blocks. Thus, it is likely someAppIDs will be empty in some blocks. On the flipside, some rollups might need to submit more data than usual in a particular block. None of this is an issue on Avail DA, the individual block builds as needed in the moment.

Tell me more

Let's learn more about AppIds by going through a real-life example.

1. We recommend you go through the whole page before trying to go through the same steps.
2. Although there are multiple ways to retrieve existing AppIds and generate new ones, using theAvail DA explorer(opens in a new tab)
3. to do so
4. is a good way to start.
5. TheAvail DA explorer(opens in a new tab)
6. is very powerful and can be used in a variety of ways.
7. For now though, let's stick toAppIds
8. .
9. Make sure you're on thechain state
10. section of the explorer. You can access it bysimply clicking this link(opens in a new tab)
11. ,
12. or by navigating to it through thedeveloper
13. tab near the top right.

14. Make sure you've selected thedataAvailability
15. pallet and theappKeys
16. method.
17. Uncheck theinclude option
18. toggle, and click on the+
19. button next to the method name.
20. You will fetch a list of all registeredAppIds
21. on Avail DA.

EachappID consists of 3 fields:

- key
- : This is a string that is the name of theappID
- . EachappID
- should have a unique name.
- owner
- : This is the address of the account that created theappID
- . A single address can create multipleAppIds
- .
- id
- : This is the unique integer index of theappID
- . It is incremented by 1 everytime a newappID
- is created. Whenever a newappID
- is created, it is automatically assigned the next availableid
- .
- Next, check theinclude option
- toggle, and enterbased avail
- as thebytes
- input. Call the function. What do you see?
- You will be returned a pair ofowner
- andid
- , which together with thekey
- you entered, form a uniqueappID
- .
- TheappKeys
- method is essentially a mapping that returns theowner
- andid
- of anappID
- given itskey
- (key
- => (owner
- ,id
-)).
- By checking theinclude option
- toggle, you are essentially filtering the output.

How to check the next availableappID

?

Anyone can create their ownappID on Avail DA. The process is entirely democratic, and it's rather simple too.

Let us first check out the next availableappID on the newtork.

1. Within thedataAvailability
2. pallet, select thenextAppId
3. method.
4. No need to pass any params, just click the+
5. button next to the method name.
6. You will be returned the next availableindex
7. /id

8. for a newappID
9. .

How to register my ownappID

?

1. Make sure you have one or more Avail DA wallets connected to the explorer. If you don't know how to do so,
2. you can follow our docs onsetting up a new wallet
3. .
4. Simplyclick this link(opens in a new tab)
5. OR navigate to theextrinsics
6. section of the explorer through thedeveloper
7. tab.

Please note that theDeveloper tab does not show theextrinsics section at all if you don't have a wallet set up on the explorer or an extension wallet connected to it. So make sure you have anAvail DA wallet set up before moving forward. 1. Select thedataAvailability 2. pallet, and thecreateApplicationKey 3. method.

1. Enter akey
2. for yourappID
3. . It can be anything you like, really.

4. This is how it should look like in the end:

5. Click onSubmit Transaction

6. , and then click onSign and Submit
7. in the box that pops up.

DO NOT CHANGE THEappID FOR THIS TRANSACTION

1. Each and every single transaction on Avail DA has anappID
2. associated with it, which isgreater than or equal to
3. 0
4. .
5. A transaction or data submission with theappID
6. of0
7. is used forchain-level
8. operations.
9. This is what we need to use for creating a newappID
10. , since the act of creating a newappID
11. has nothing to do with a specific 'app
12. ' on Avail DA.
13. This field would instead have been a positive integer if we, for example, were submitting data to a specific application on Avail DA.
14. Authorize the transaction through your wallet, and you're done! You've successfully created your ownappID
15. on Avail DA.
16. You can verify7
17. by using the steps covered earlier to query theappKeys
18. method :)

How to submit data to myappID

?

You can submit data to your, or any otherappID on Avail DA using the explorer by calling thesubmitData extrinsic from within thedataAvailability pallet.

1. Make sure you have one or more Avail DA wallets connected to the explorer. If you don't know how to do so,
2. you can follow our docs onsetting up a new wallet
3. .
4. Simplyclick this link(opens in a new tab)
5. OR navigate to theextrinsics
6. section of the explorer through thedeveloper
7. tab.
8. Select thedataAvailability
9. pallet, and thesubmitData

10. method

10. method.
11. Enter a randomdata
12. string that you want to submit to yourappID
13. , and then click onSubmit Transaction
14. .
15. Fill in theAppID
16. that you want to submit the data to. Click onSign and Submit
17. to authorize the transaction through your wallet.
18. Wait for the transaction to be included in a block and then open the detailed view of the block.
19. Click on your specific transaction to see it's details. This is what it should look like:

A few more things of note

1. As stated earlier, thekey
2. andid
3. fields of everyappID
4. are unique. This means if you try to create anappID
5. with the samekey
6. as an existing one,
7. the operation will fail. This is why it makes sense to use theappKeys
8. method to check if your desired name is already taken.
9. If you're a developer and are looking for more programmatic instructions, you can check out ourAPI reference
10. .
11. Anyone can submit any sort of data to anyappID
12. regardless of whether or not they created it.
13. But what does this mean? And is this an attack vector?
14. This is where it is important to understand that Avail DA is a DA layer, not an execution environment.
15. We are not concerned with the validity of the data being submitted, only with its availability, which means we can support a wide variety of applications across multiple tech stacks.
16. This does not constitute an attack vector since that any app or execution layer building on top of Avail DA can always set up
17. certain rules to filter out unwanted data submissions.
18. They could for example make it so that only data submitted with a particular signature, i.e. from a particular address, is accepted.
19. All other data submitted to the particularappID
20. is treated as spam, and ignored.

Learn more about Avail Get started with the Avail Apps Explorer