

Integrating a New DA Layer with Alt-DA

⚠ The Alt-DA Mode feature is a Beta feature of the MIT licensed OP Stack. While it has received initial review from core contributors, it is still undergoing testing and may have bugs or other issues. [Alt-DA Mode](#) enables seamless integration of any DA Layer, regardless of their commitment type, into the OP Stack. After a DA Server is built for a DA Layer, any chain operator can launch an OP Stack chain using that DA Layer for sustainably low costs.

Build Your DA Server

Our suggestion is for every DA Layer to build and maintain their own DA Server, with support from the OP Labs team along the way. The DA Server will need to be run by every node operator, so we highly recommend making your DA Server open source and MIT licensed.

Design your commitment binary encoding

- It must point to the data on your layer (like block height / hash).
- It must be able to validate the data returned from the data (i.e., include a cryptographic commitment to the data like a hash, merkle proof, or polynomial commitment, this could be done against the block hash with a complex proof).

See the [specs\(opens in a new tab\)](#) for more info on commitment submission.

Claim your da_layer byte

- Claim your [byte\(opens in a new tab\)](#)

Implement the DA Server

- Write a simple HTTP server which supports `get`
- `andput`
- `put`
- is used by the batcher and can return the commitment to the batcher in the body. It should not return until the data is known to be submitted to your DA layer.
- `get`
- should fetch the data. If the data is not available, it should return a 404
- not found. If there are other errors, a different error should be returned.

Run Alt-DA

Follow our guide on [how to operate an Alt-DA Mode chain](#), except instead of using the S3 DA server, use the DA server that you built.

Next Steps

- Want to explore more? See an [example DA server\(opens in a new tab\)](#)
- For more detail on implementing the DA Server [see the specification\(opens in a new tab\)](#)
- .

[Modifying Predeployed Contracts Configure Challenger For Your Chain](#)