<repository>

LUKSO-Hardhat-template

overview

repository: https://github.com/CJ42/LUKSO-Hardhat-template/
userName: CJ42
repository: LUKSO-Hardhat-template
branch: main
date: 2023-11-27T10:16:38+01:00 (1701076598)
description:
hello everyone and welcome to this uh in
this Workshop tutorial where today we
are going to learn how to create an nft
collection using the luus standard lp7
or LS
sp8 so in in today's video we're going
to see how we can create different type
of nfts and we will see different
example uh for instance um for instance
like the example that you have listed
here to create different type of
collections so things like um nfts where
you need to create large
quantities like digital tickets for
events or large quantities of nft with
the same design that are um that act as
collectible or using um or doing things
like creating a collection like a basic
nft collections or dynamic nfts that can
evolve or a more advanced use case like
collections of sub
collections so to get started uh we are
going to use uh template that I've
created that use the latest version at
the current time of hard heart and eers
and uh to you so you can run the
following instruction to um to clone the
repository and install all the
dependencies and you need to make sure
that you have uh this version of
nodejs and if you don't you can just
install it using you can just update
your node version using
NVM okay so let's get started so first
of all we need to there is a first
concept what we need to learn and
understand which is uh the the type of

token that we are going to create so the
type of digital
asset and when we create a digital asset
smart contract on loo we need to specify
the token type under the following data
key in thec 75 y storage so the there
are three type that we can Define we can
Define either a token or we can Define
um an nft or we can define a collection
a token will mean that each item uh have
the same value as another and nft mean
will mean that each item are unique with
our own design and the collection will
mean that uh it's a group of
nfts okay so let's get
started so we're going to first start by
creating a digital ticket for an
event and to do that we can use the
standard LSP 7 and Set uh the decimals
to zero using the non-divisible flag
when we deploy the token and so the
token type will be zero which is a token
and this is useful when we want to to
create large quantities of nfts where
each nft is the
same um and we can use that especially
in context where we need to Mint large
quantities of nfts all at once and
thanks to lp7 set non divisible we can
do that because we will be able to
create uh to Mint to call the Min
function once with the recipient amount
recipient and the amount of token to to
me to create and uh then we
um and then we will just call the and
and yeah this will just create the whole
batch all at once so for creating this
is useful uh because we can just do that
in a single transaction instead of
calling M and the token ID a thousand
time so a good example of use case for
that are Cinema tiet uh things like
event tickets for events or uh where we
want to
um Supply uh the total amount that we
want to create and then we can
afterwards uh transfer each of them
individually or each attendant like each
attendant to the event can basically
claim the
nft so this is an example of how we can
represent an an evate as an nft using

lp7 so as you can see we will just
inherit the sb7 minable um uh contract
and uh we just Define the name of the
event the symbol of the token and who on
the contract we mark this contract as
indivisible meaning that uh it's a when
you get a digital ticket for the event
it's a full ticket you cannot split the
token in the ticket in half uh which
mean which is the parameter non
divisible
here and then we can set the token type
uh as we said the4 token type to a token
meaning that what we are creating
represent an actual uh token so one
ticket has the same same value as
another tickets there are all both
interchangeable and we can mint a batch
of thousand ticket of 100 tickets here
all at once um as follow and we are
using here the traditional uh
distribution uh the pattern that is
often used where uh basically tokens are
just minted
to one address that will then distribute
this uh these ticket
okay now we're going to see the second
example where uh we can use still the
same standard sp7 to create 100
collectible with the same
design and so it will be the same we
just use LS sp7 set to non divisible and
the token type is one and this is useful
when we actually want to
differentiate when we want to create
large quantities of nfts efficiently but
we want to differentiate it the asset we
are creating is either uh a token or an
NFD uh so in this case here all the LD
has have the same design is still useful
to large quantities and a comparison
with a real world here is Pokemon is
like basically trading cards if you're
familiar with trading cards uh a lot of
people collect them things like Pokemon
or magic recovering or baseball card um
and there are a large there are multiple
uh quantity of uh such card for instance
a Charizard or one of baseball C like
that that have been created in multiple
quantities but people uh collect have a
collectible aspect so this is how we can

use the standard to map to an example
from The Real
World and this is how that would look
like in uh solidity we will still use LS
sp7 so we can create a large quantity
but only if we set the non divisible
flag to True uh because it's the same
you a trading card you cannot rip it in
half and give me half you just own the
full card or you you don't own any C or
you don't own the card and same you can
see here in the Constructor we're going
to set the the lp4 token type to be an
NFD okay now we're going to see a third
example where we are going to create
collection um where each collectible is
unique and this is a bit similar to what
already exist um and to do that we are
going to use LS sp8 and we will set the
token type uh to a collection and this
is the most simple way to create nfts on
so really uh that mimics the current nft
collection that
exist uh so if you want to recreate
popular collections similar to what's
already there like out there like crypto
punks or bard Apes collection like that
you can use this um um this mixing of Ls
sp8 and the token type to
three and um yeah this is useful
especially when you want to create an
nft with the metadata is going to be set
in stone it's not going to change um and
so like you create vft you set its
metadata and the metadata is is setup
and very
relatively speaking example with a weird
word again is uh it's basically painting
or digital art you you basically create
U when when a painting is created it's
uh the the art pieces is there it's not
going to change it's just painting and
in oil painting and uh we have the same
now nowadays with digital art so for
instance this collection of digital art
the uh the art pie is there it's not
going to it's not going to change so um
so yeah we can use the this combination
to reproduce something uh
similar okay so we going to do bit of uh
uh coding in solid and we're going to
see how we can create uh uh such a basic

example and we will look we will do that
by inheriting the LSP contract so I will
go over the uh the repository if you
have downloaded it and you have
installed it you we'll see that there is
a playground file that we can use to
we're going to use to create our
collection and you just need to install
the package with Yar if you haven't done
it yet and what we're going to do is we
are going to create our contract or nft
collection
contract so let's call it digital art
collection and in order to make it as an
nft we are going to
import the LSP create
identifiable asset
contract and we are going to import it
from the loo SM contract
package which is uh
here so you have to specify
contract you have to
specify
SP
8 and we are going to
take this
one okay so we're going to take this one
and we are going to add it into the
inheritance and here we will uh write
the logic
code
here okay so as you can see here if you
try try to compile the contract uh and
if you run npm run build which is the
command for compiling you will get a a
first error from the sotic compiler
saying that we have no argument that
have been passed to the
Constructor um and the reason is that
ls8 expect uh to take uh arguments on
deployment so we're going to see in a
minute now these arguments so when you
deploy an lp8 an nft contract on you you
uh have to specify the following you
have to specify the name of the token so
in this sense the in this in the context
of this tutorial the name of a
collection the symbol um who owns the
nft uh contract who control this
contract uh so this address is basically
going to be the one that can uh set all
the metadata on the contract and in the

future if his Contra is to be own by uh someone else uh for instance uh you the whole uh whole man let's say in the real world Contex context the whole management of the of the collection is uh is moved to like an organization or another person then this is the the address specified as the owner that will be able to do
that and finally we will uh have to define the type of uh nfts and token IDs and we will see that in a second so it's technically two ways in Sol here to Define that we can either use predefined deployment values uh so we hard code that in the sity code or we can specify the values in the Gs and TS JavaScript or typescript so the first method is useful when you know already what's going to be the name of a collection um so you don't have to write the code in JavaScript and deploy like that you just can hardcode everything into the contract uh uh and it it's just going to be easier it's going to be a bit cheaper to deploy because the things are not the arguments not do not need to be passed and extracted from loal data uh and the second method is good when you are creating something more like a a template temp a contract template that can be reused to uh recreate nfts um uh collection okay
so the to give an example in form of code the method one is like that you just specify an argument as literal values inside the inheritance uh all the second method is you define a Constructor and you will write the uh the arguments uh as parameter and then you will specify the arguments in your JavaScript code like this so this is an example with eers and type chain for the deploying uh the nft
collection so I'm going to just carry on on uh the code here so in my case what I'm going to do is that I going to do method one I'm just going to um create the uh I'm just going to write it down because I kind of have an idea of what kind of digital art collection I'm going to create I'm going to create um a

collection of uh
3D um let's call it 3D artistic
Cube I'm just going to create basically
let's imagine we are creating different
cubes uh as digital art painting with
different shapes and colors and this
would be our nft collection so this is
the uh collection
name the symbol we are going to say
3D
cube so this is what will appear in the
block
Explorer uh under the name under the
address the owner is going to be the I'm
going to specify the address that will
deploy the
contract and then um and then we'll have
to specify as well the the token ID
type so for the owner you can uh
basically if you look at the contract
here if you
if you look at the
contract we can basically go
into let's look at factual contract
code so for owner here is basically the
address that you create the contract and
the fourth will be the token ID type
we're going to specify that in a minute
if you want what you can do is that you
can also if you know an address that
will deploy
the the the if you know the address that
will own if that you will own the N
collection so for instance you want to
use one address to deploy the contract
and one another address to specify as
who will have control over it um you can
just specify uh like this as a constant
uh both you can just say LS sp8
collection owner and here as an
example and I will just
say if
this and you can just specify
the um you can just basically specify
the address here yeah it's just going to
ask
me to check some the
address so you could have like the
put the address here to
specify so you just can basically do
that this will be the contract on this
okay now we need to look at uh this

parameter the token ID type so what is
it so basically when you specify when
you create an LP collection um you have
a parameter that you need to specify
which is the lsba token ID
type and the lsba token ID type is
basically what you're going to use to
define how the ID
identifier um of each nft will be
represented so with lp4 token type you
specify how the how you have to
interpret the assets inside the the
contract we have lspa token ID type this
is really for nfts this is for
specifying how you what you are what
which method you're going to use to uh
create the identifiers for each nfts
and so in the LSP standard we have five
way five ways to Define to to to do that
we can uh basic we I'm going to go
through the main uh 3 one here and uh
and then we will have another future
example for the the last
one so let's give first example we can
first use the token ID type zero which
is basically your unique number and you
can use that for things like you start
at token id1 and you increment as you
meant new nfts and a popular example
that exist outfir is basically the
crypto Punk collection the token IDs
started at one and then they got
incremented uh so and there is a there
has been 10,000 crypto Punk that got me
that got created and each crypto Punk
has the identifier 1 2 345 and that
increments as it
goes a second example example for token
type string is we can use the example
that we had uh and this is useful for
when you actually want to say the
identifier of the token ID is going to
be a whole name String um and to use a
example of digital art this is useful
this token ID type is useful for this
purpose because if you create a let's
say a digital art and you call one
painting um here exmo file
9853 um this is easier to identify the
the token ID like that and yeah and or
if you um let's say
um let's say you like if you create the

the digital jond you will you will
basically let's say you recreate the
digital version of popular uh painting
you can say uh you can name the token ID
the digital Jon like the famous painting
of DaVinci it's easier to identify the
digital piece of art by its name rather
by an identifier that doesn't properly
match so depending on the use case the
token ID type is easier is easier for uh
identify
NFD and so here in this case each
digital piece of art is uh represented
by its name
another use of the token type for
instance is the number two um which can
be used for like things like a really
unique and specific uh ID uh and let's
say for instance you have a physical
good in the real world that you want to
map on the blockchain with something
like an NFC chip or a QR code um search
Goode can have a serial number for
instance you can have a collection of
Rolex watches and they have all their
unique serial number and if you want to
map that onto the smart contract
collection for the Rolex collection uh
you can use this token ID type where
token ID is the serial number of the
item and we can import all these token
in solidity as follow uh they have they
are defined in the LSP smart
contract and you and in our case we are
going to to use the tokened type number
so let's carry on into our playground we
just going to um so I had put a zero
just for uh Sting the compiler warning
um so we just going
to
specify we're just going to import that
so we're going to saycore
lp8 token ID
type number I'm going to use the more
simple
one and I'm going to
do LSP smart
contract
contract in
here I'm going to go here
again and I'm going to specify LP
constant. so and I will pass this

argument
here or in the case of a painting we
could say string because it's more
appropriate with what we were using
before okay so now that we have uh we
have um created our first basis for nft
contract uh there is we have a still one
problem is that uh we have no public
method for
minting uh new
nfts so we need uh and the reason is
that the lspa contract by itself doesn't
have this because uh because
every uh every contract Creator will
Define its own way to Mint and create
new nfts so we have two method it's
either we create our own function uh to
create our own nft Supply or we
basically
um how we um uh use our
preset so the first method will be to
create your own function in this case uh
we will call it uh let's say create new
Cube
and and so we call it let's say create
new Cube uh create new Cube and we say
Cube name address
recipient and we will have our public
method we make it only the owner so that
only the contract owner can create a
cube and then we need to
define the token
ID and we basically have to
pass so we need
to pass the name of the cube as B so the
token ID are B 32 so we need to pass the
token ID as B B and then we need
to pass the the byes as a B
for so we have our token ID and then we
can pull the internal mean function and
as you can see it's going to take the
recipient of the ID of the token ID the
forse parameter and the the data so the
recipient we have it
in
in the in the function parameter then
we're going to specify true and then the
data and then the data sorry we going
just specify no data this is basically
any extra data that we want to send uh
when we notify the recipient that we
have created a new n that here hased a

new uh nftd
and yeah so here I will not put anything
and the true the force here the true
parameter is basically uh parameter
called Force this is for enforcing this
is for U basically ensuring if forcing
if we are transferring an nft to an
address uh that cannot handle the nft or
not so in in the Lu
ecosystem
basically the recipient have to be smart
contract uh of tokens have to be smart
contract that implement the lsp1
standard the lsp1 universal receiver
standard and this is to ensure that they
can handle and register uh and and
register these uh
tokens so if you specify by default it's
uh if you specify false if you are
sending the minting or transferring the
nft to an externally own account or a
smart contract that doesn't implement
lsp1 this will fail in this case we're
just going to put two um to uh just uh
mean for anyone uh a universal profile
uh contract and implement lp1 or uh any
smart contract or any EA so one syntax
that I like to write in general it's
it's basically um it's basically writing
the name parameter so that it's very
it's very clear what uh everything uh
mean mean and you can do that in
solidity uh um as follow you can
just you can just
basically
take each of this
parameter yeah and you B yeah this is
BAS basically the name parameter way to
uh to call an a function so this is
clear what each parameter uh means
here but in our case we're not going to
even we are not going to uh we are not
really going to uh do that we just going
to um we can just create just going to
create a public function called we in
okay yeah we so this is the first me
method and the second method is you can
use a preset contract that we have which
is called LSB
minable and you can narrate it
here and when you have uh so I would
just comment out this method and when

you use this pH inable what happen and
is
basic is basically the following you
have a public uh method added into the
API that you can call to uh to create
your uh NF collection so if you if we
look here if we look for mint we'll see
we have a public Min function here in
the AI uh that take all these parameters
um I take all of these parameters it
takes the token
ID Yeah so basically we created here two
ways
to create
nfts so now that we have created the nft
we can what we can do is that we can
specify uh the metadata R for each nft
and the most simple way um is to um
basically have all the metadata of each
nft the links and that result in the
contract and you can do this by setting
the base
URI uh
the Bas Ur and this data key and then to
get the metadata each nft you then
append you take extract this Bas and you
append the token
ID and uh what this parameter takes is
basically the it's the hash function uh
it's for
verifiability and this is the base um
and this is the constant uh for
varability and this is the base U uh
that you will
use an example of how we can use it for
this is for both apes basically both
apes for metadata they all have the same
base U and the metadata never change so
you just they just all have base URI and
there is the token ID appended at the
end yeah so if we look carefully here
basically we uh this is the example in
JavaScript where we do that and yeah as
I say the under the LSP token metad
database U we have a tole that we had
defined here and the first value is the
first for bite of a hash of a word K 256
parenthesis with utf8 in
parenthesis is the has of uh that and
this is used for uh verifiability
between ensure that when we create the
Json the URL to the metadata it's

verifiable and um and then we have the
base URI appended at the
end okay the fourth example is for
dynamic nfts nfts that can change their
traits and attributes over
time um so in this case the combination
that we will use is s sp8 uh plus the
number three for the token type meaning
we are going to create a
collection and so when we do that each
token ID here is uh an nft F can be its
own smart contract uh and for example
contract an ERC 75y smart contract and
it can have an infinite amount of
entries as data keys to specify specific
attributes this is very useful if you
want to uh create an energ collection
where um the attribute of nfg can change
so you can create basically
digital items or entities as nft that
can evolve um and this is useful for
things like uh nft that you want to make
customizable um so
the the basically the user owns an nft
that is specific to him but it can also
even customize it or things like some
use case are like characters in a video
game but that can evolve as the
character by equipment or weapons um so
for instance um an example from this is
an example from famous video game
Assassin's Creed where the character can
uh upgrade and downgrade and change his
equipment like his helmet his weapon his
uniform and you can technically store
all of these under the e75 wi storage of
a contract of that represent the
nftd so to do that what we do is that we
uh still we we inherit LSP 8 and we uh
and we use the token ID type address and
we will set the token type data key to
uh
collection and the so this is the whole
collection that contain all the the each
nft identifier and this is the contract
that will be for uh the actual uh nft
itself with the metadata that it
contains and this contract basically uh
here specify the owner of nft and the
collection that it belongs to and this
collection can be specified under the
LSP reference contract data key so this

way you can uh Point backwards to uh
when you quer this nft as a smart
contract you can know from which
collection it it it it comes from you
can go from both ways you can basically
um um check uh under this
contract who owns Which nft and the
token ID will be at the address of this
contract and in this contract the value
stored under the lsph reference contract
will be the address uh that uh this
contract belongs
to okay and the final example which is
the most complex of all of them is that
you can use LS sp8 and lp7 uh in
multiple combination to create a
collection of
subcollections so this is a very
Advanced use case so this the standard
will be still LS sp8 and the token type
uh will be still free as a collection
and in this case the LSP contract is
actually a whole collection uh it's
really uh it can it it it's it's it list
all the uh all the sub collection that
it contains and each token ID is
actually another smart contract that
it's an lp7 contract that is still non
divisible and this represent uh one part
of a sub
collection and um very
um um a very uh a good example which is
quite uh common to anyone in the crystal
space is to create the famous sticker
that we all collect when we go to uh
crypto events but in a digital form so
if you go to a crypto event you will see
that that any Booth everyone uh has a
whole set of stickers and there is
different design for each stickers but
there is multiple of them and at luo the
sticker where they very popular are the
loo Goods sticker the loo super good and
the one the in whole silver so basically
each what happens here is that
this is the whole lp8 contract and LSP
and each Contra each of the stickers is
actually an LSP 7 contract with a
certain number of digital sticker that
exist so this is will be one lp7
contract this will be another one and
this will be another one and there will

be a limited number of each sticker so
I've represented with a different
example here where you have the nft
contract lp8 that contained all the list
of stickers digital stickers created so
this is these all these digital stickers
are basically nfts so this is the whole
collection and the token ID type is
addressed because each sticker is a
smart contract it's an LSP 7 so the
token ID uh 416 a4f it's basically the
address of np7 contract that has the
silver dog uh sticker and and you have
that for each uh each basically sticker
and in this way you can create things
like a collection of subcollections
where each subcollection has a limited
number of item available um so for
instance in this case the the the
stickered silver shiny dog
has there is a lot of quantity available
and the the the the second one here is a
bit less common while the last one the
six ice cat it's very rare there's only
5,000 digital stickers ever created and
still each of these uh tokens are
non-divisible so you can use this uh
approach also to create a collection
with a sub collection inside them that
have with where each item have is more
scar than
others and yeah this is basically how
you can uh compose the different LSP
standards to create different type of
nftd collection as you want and the
example here will be that uh and the
form of code snippet uh that will be the
following you have um it's more it's
better
viewed in the in vs code where you
have
here you have the whole LF nft
collection so this is the LSP
contract when you deploy it you have you
specify the lp4 token type and you can
technically uh visit this is a sort
example where you create a new sticker
you you uh specify the name and the
sybol of a stinker of a of a digital
sticker and you say how many sticker you
are going to create so reusing for
example if you are going to create

100,000 or 20,000 or
5,000 um and you define a price for
people who want to buy the the the
sticker and you just create a new
contract and once the contract has been
created you uh get back his uh uh
address here and you um you basically uh
then uh create the sticker ID the token
ID and under the uh under this contract
you are just going to um basically set
the token ID and then you will uh you
will min the the token ID for the
stickers so this sticker will then
belong to this other collector ction and
when you are uh basically
here creating the sticker you are what's
happening is that you are just
specifying the the type of uh
sticker um and yeah and and here uh it's
actually not collection it's
ft and here in this case you just
minting the total amount um that uh that
exist and yeah that's about it that's
how you can use the different LSPs in
combination to uh create uh different
type of nft collection I hope you enjoy
the video and thank you

<file>

path: /.env.example

url: https://github.com/CJ42/LUKSO-Hardhat-template/blob/main/.env.example

```
PRIVATE_KEY=
```

</file>

<file>

path: /.nvmrc

url: https://github.com/CJ42/LUKSO-Hardhat-template/blob/main/.nvmrc

```
16.20
```

</file>

<file>

path: /.prettierrc

url: https://github.com/CJ42/LUKSO-Hardhat-template/blob/main/.prettierrc

```
{
  "plugins": ["prettier-plugin-solidity"],
  "overrides": [
    {
      "files": ["*.js", "*.ts"],
      "options": {
        "tabWidth": 2,
        "printWidth": 100,
        "trailingComma": "all",
        "singleQuote": true,
        "semi": true
      }
    },
    {
      "files": "*.sol",
      "options": {
        "tabWidth": 4,
        "printWidth": 80,
        "compiler": "0.8.15"
      }
    }
  ]
}
```

</file>

<file>

path: /.tool-versions

url: https://github.com/CJ42/LUKSO-Hardhat-template/blob/main/.tool-versions

```
nodejs 16.20.2
```

</file>

<file>

path: /README.md

url: https://github.com/CJ42/LUKSO-Hardhat-template/blob/main/README.md

LUKSO Hardhat Template

This project demonstrates how to use LSP7 or LSP8 to create a basic NFT collection on LUKSO using the @lukso/lsp-smart-contracts (https://github.com/lukso-network/universalprofile-smart-contracts);

It uses:

- Hardhat v2.18.2
- Ethers v6
- Artifacts generated by Typechain

Setup

Make sure you are using node js v16.20. You can set this up with nvm (https://github.com/nvm-sh/nvm).

```
nvm install 16.20
nvm use 16.20
```

Installation

```
git clone https://github.com/CJ42/LUKSO-Hardhat-template.git
cd LUKSO-Hardhat-Template
yarn
```

</file>

<file>

path: /contracts/Example 1/EventTicketsNFT.sol

url: https://github.com/CJ42/LUKSO-Hardhat-template/blob/main/contracts/Example 1/EventTicketsNFT.sol

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.4;

// modules
import {
    LSP7Mintable
} from "@lukso/lsp-smart-contracts/contracts/LSP7DigitalAsset/presets/LSP7Mintable.sol";

import {_LSP4_TOKEN_TYPE_DATA_KEY, TokenType} from "../TokenTypes.sol";

contract EventTicketsNFT is LSP7Mintable {
    constructor(
        string memory eventName,
        string memory tokenSymbol,
        address contractOwner
    )
        LSP7Mintable(
            eventName,
            tokenSymbol,
            contractOwner,
            true // make the token non divisible
        )
    {
        // set the token type
        _setData(_LSP4_TOKEN_TYPE_DATA_KEY, abi.encode(TokenType.TOKEN));
    }
}
```

</file>

<file>

path: /contracts/Example 2/DigitalTradingCards.sol

url: https://github.com/CJ42/LUKSO-Hardhat-template/blob/main/contracts/Example 2/DigitalTradingCards.sol

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.4;

// modules
import {
   LSP7Mintable
} from "@lukso/lsp-smart-contracts/contracts/LSP7DigitalAsset/presets/LSP7Mintable.sol";

import {_LSP4_TOKEN_TYPE_DATA_KEY, TokenType} from "../TokenTypes.sol";

contract DigitalTradingCards is
   LSP7Mintable(
      "Digital Trading Card",
      "DTC",
      msg.sender,
      true // cards are non divisible (you cannot tear a card in 2 and give me half)
   )
{
   constructor() {
      // set the token type
      _setData(_LSP4_TOKEN_TYPE_DATA_KEY, abi.encode(TokenType.NFT));
   }
}
```

</file>

<file>

path: /contracts/Example 3/BasicNFTCollection.sol

url: https://github.com/CJ42/LUKSO-Hardhat-template/blob/main/contracts/Example 3/BasicNFTCollection.sol

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.4;

// modules
import {
    LSP8Mintable
} from "@lukso/lsp-smart-contracts/contracts/LSP8IdentifiableDigitalAsset/presets/LSP8Mintable.sol";

// constants
import {
    _LSP8_TOKENID_TYPE_NUMBER
} from "@lukso/lsp-smart-contracts/contracts/LSP8IdentifiableDigitalAsset/LSP8Constants.sol";
import {_LSP4_TOKEN_TYPE_DATA_KEY, TokenType} from "../TokenTypes.sol";

contract BasicNFTCollection is LSP8Mintable {
    constructor(
        string memory nftCollectionName,
        string memory nftCollectionSymbol,
        address contractOwner
    )
        LSP8Mintable(
            nftCollectionName,
            nftCollectionSymbol,
            contractOwner,
            _LSP8_TOKENID_TYPE_NUMBER
        )
    {
        // set token type
        _setData(_LSP4_TOKEN_TYPE_DATA_KEY, abi.encode(TokenType.COLLECTION));
    }
}
```

</file>

<file>

path: /contracts/Example 4/DynamicNFTCollection.sol

url: https://github.com/CJ42/LUKSO-Hardhat-template/blob/main/contracts/Example 4/DynamicNFTCollection.sol

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.4;

// modules
import {
    LSP8Mintable
} from "@lukso/lsp-smart-contracts/contracts/LSP8IdentifiableDigitalAsset/presets/LSP8Mintable.sol";

// constants
import {
    _LSP8_TOKENID_TYPE_ADDRESS
} from "@lukso/lsp-smart-contracts/contracts/LSP8IdentifiableDigitalAsset/LSP8Constants.sol";

import {_LSP4_TOKEN_TYPE_DATA_KEY, TokenType} from "../TokenTypes.sol";

contract DynamicNFTCollection is LSP8Mintable {
    constructor(
        string memory dynamicNFTCollectionName,
        string memory dynamicNFTCollectionSymbol,
        address contractOwner
    )
        LSP8Mintable(
            dynamicNFTCollectionName,
            dynamicNFTCollectionSymbol,
            contractOwner,
            _LSP8_TOKENID_TYPE_ADDRESS
        )
    {
        // Set the type of the token
        _setData(_LSP4_TOKEN_TYPE_DATA_KEY, abi.encode(TokenType.COLLECTION));
    }
}
```

</file>

<file>

path: /contracts/Example 4/DynamicNFTMetadataContract.sol

url: https://github.com/CJ42/LUKSO-Hardhat-template/blob/main/contracts/Example 4/DynamicNFTMetadataContract.sol

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.4;

// modules
import {ERC725Y} from "@erc725/smart-contracts/contracts/ERC725Y.sol";

contract DynamicNFT is ERC725Y {

    bytes32 constant _LSP8_REFERENCE_CONTRACT =
0x708e7b881795f2e6b6c2752108c177ec89248458de3bf69d0d43480b3e5034e6;

    constructor(address nftOwner, address nftCollection) ERC725Y(nftOwner) {
        /**
         * @dev set the reference to the NFT collection that this metadata contract belongs to
         *
         * {
         *     "name": "LSP8ReferenceContract",
         *     "key": "0x708e7b881795f2e6b6c2752108c177ec89248458de3bf69d0d43480b3e5034e6",
         *     "keyType": "Singleton",
         *     "valueType": "(address,bytes32)",
         *     "valueContent": "(Address,bytes32)"
         * }
         */
        _setData(_LSP8_REFERENCE_CONTRACT, abi.encodePacked(nftCollection,
bytes32(bytes20(address(this)))));
    }
}
```

</file>

<file>

path: /contracts/Example 5/LSP7DigitalStickers.sol

url: https://github.com/CJ42/LUKSO-Hardhat-template/blob/main/contracts/Example 5/LSP7DigitalStickers.sol

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.4;

import {
    LSP7DigitalAsset
} from "@lukso/lsp-smart-contracts/contracts/LSP7DigitalAsset/LSP7DigitalAsset.sol";
import {
    LSP7Burnable
} from "@lukso/lsp-smart-contracts/contracts/LSP7DigitalAsset/extensions/LSP7Burnable.sol";

import {_LSP4_TOKEN_TYPE_DATA_KEY, TokenType} from "../TokenTypes.sol";

contract LSP7DigitalStickers is LSP7DigitalAsset, LSP7Burnable {
    constructor(
        string memory stickerName,
        string memory stickerSymbol,
        address contractOwner,
        uint256 totalNumberOfStickersInCirculation
    ) LSP7DigitalAsset(stickerName, stickerSymbol, contractOwner, true) {
        // Set the type of the token
        _setData(_LSP4_TOKEN_TYPE_DATA_KEY, abi.encode(TokenType.NFT));

        // mint the total number of stickers that the LSP8 Collection owns
        _mint({
            to: msg.sender,
            amount: totalNumberOfStickersInCirculation,
            force: true,
            data: abi.encodePacked(
                "creating ",
                totalNumberOfStickersInCirculation,
                " new stickers for this collection!"
            )
        });
    }
}
```

</file>

<file>

path: /contracts/Example 5/LSP8DigitalStickersCollection.sol

url: https://github.com/CJ42/LUKSO-Hardhat-template/blob/main/contracts/Example 5/LSP8DigitalStickersCollection.sol

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.4;

// modules
import {
    LSP8IdentifiableDigitalAsset
```

```
} from "@lukso/lsp-smart-contracts/contracts/LSP8IdentifiableDigitalAsset/LSP8IdentifiableDigitalAsset.sol";
import {LSP7DigitalStickers} from "./LSP7DigitalStickers.sol";

// constants
import {
    _LSP8_TOKENID_TYPE_ADDRESS
} from "@lukso/lsp-smart-contracts/contracts/LSP8IdentifiableDigitalAsset/LSP8Constants.sol";

import {_LSP4_TOKEN_TYPE_DATA_KEY, TokenType} from "../TokenTypes.sol";

/**
 * @dev Pseudo code example, LSP8 collection of digital stickers, where each sticker is a LSP7 contract
with a limited supply available.
 * This contract is for testing and tutorial purpose.
 */
contract LSP8DigitalStickersCollection is
    LSP8IdentifiableDigitalAsset(
        "Example Digital Stickers Collection",
        "EDSC",
        msg.sender,
        _LSP8_TOKENID_TYPE_ADDRESS
    )
{
    constructor() {
        // Set the type of the token
        _setData(_LSP4_TOKEN_TYPE_DATA_KEY, abi.encode(TokenType.COLLECTION));
    }

    // Constants ---

    bytes32 internal constant _LSP8_REFERENCE_CONTRACT_DATA_KEY =
        0x708e7b881795f2e6b6c2752108c177ec89248458de3bf69d0d43480b3e5034e6;

    bytes32 internal constant _STICKER_PRICE_DATA_KEY =
        keccak256("Price Per Sticker");

    // Errors ---

    error FailedDeployingLSP7StickerContract(string stickerName);

    function createNewStickerCollection(
        string calldata stickerName,
        string calldata stickerSymbol,
        uint256 numberOfStickers,
        uint256 pricePerSticker // in wei
    ) external onlyOwner {
        try
            new LSP7DigitalStickers(
                stickerName,
                stickerSymbol,
                address(this),
                numberOfStickers
            )
        returns (LSP7DigitalStickers newSticker) {
            // example:
            // newSticker (contract address)= 0x9760e8347D5E8f9042726E3B4DdE9Ae787476101

            // NFT tokenId will be
0x0000000000000000000000009760e8347D5E8f9042726E3B4DdE9Ae787476101
            bytes32 stickerId = bytes32(abi.encode(address(newSticker)));

            bytes32[] memory dataKeysToSet = new bytes32[](2);
            bytes[] memory dataValuesToSet = new bytes[](2);

            // Set Data Key "LSP8ReferenceContract",
            dataKeysToSet[0] = _LSP8_REFERENCE_CONTRACT_DATA_KEY;
            dataValuesToSet[0] = abi.encodePacked(address(this), stickerId); // "valueType": "
(address,bytes32)"

            // Set Data Key specific to the price per sticker
            dataKeysToSet[1] = _STICKER_PRICE_DATA_KEY;
            dataValuesToSet[1] = abi.encode(pricePerSticker);

            newSticker.setDataBatch(dataKeysToSet, dataValuesToSet);

            // need to pass &nbsp;force = true&nbsp; as will attempt to notify &nbsp;universalReceiver&nbsp;
function of this contract
            // which does not implement LSP1
            _mint({
                to: address(this),
                tokenId: stickerId,
                force: true,
                data: ""
            });
        } catch (bytes memory) {
            revert FailedDeployingLSP7StickerContract(stickerName);
        }
    }
}
```

</file>

<file>

path: /contracts/Playground.sol

url: https://github.com/CJ42/LUKSO-Hardhat-template/blob/main/contracts/Playground.sol

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.4;

// modules
import {
    LSP8Mintable
} from "@lukso/lsp-smart-contracts/contracts/LSP8IdentifiableDigitalAsset/presets/LSP8Mintable.sol";

// constants
import {
    _LSP8_TOKENID_TYPE_STRING
} from "@lukso/lsp-smart-contracts/contracts/LSP8IdentifiableDigitalAsset/LSP8Constants.sol";

// put the address to specify the owner of the LSP8 collection and use a different deployer address
address constant _LSP8_COLLECTION_OWNER = 0xCAfEcAfeCAfECaFeCaFecaFecaFECafECafeCaFe;

contract DigitalArtCollection is
    LSP8Mintable(
        "3D Artistic Cubes", // collection name
        "3DCuB", // symbol
        _LSP8_COLLECTION_OWNER, // contract owner
        _LSP8_TOKENID_TYPE_STRING
    )
{
    function createNewCube(
        string memory cubeName,
        address recipient
    ) public onlyOwner {
        bytes32 tokenId = bytes32(bytes(cubeName));
        _mint({to: recipient, tokenId: tokenId, force: true, data: ""});
    }
}
```

</file>

<file>

path: /contracts/TokenTypes.sol

url: https://github.com/CJ42/LUKSO-Hardhat-template/blob/main/contracts/TokenTypes.sol

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.4;

bytes32 constant _LSP4_TOKEN_TYPE_DATA_KEY =
0xe0261fa95db2eb3b5439bd033cda66d56b96f92f243a8228fd87550ed7bdfdb3;

enum TokenType {
    TOKEN,
    NFT,
    COLLECTION
}
```

</file>

<file>

path: /hardhat.config.ts

url: https://github.com/CJ42/LUKSO-Hardhat-template/blob/main/hardhat.config.ts

```
import { HardhatUserConfig } from 'hardhat/config';
import { NetworkUserConfig } from 'hardhat/types';
import '@nomicfoundation/hardhat-toolbox';

import * as dotenv from 'dotenv';
dotenv.config();

function getTestnetChainConfig(): NetworkUserConfig {
  const config: NetworkUserConfig = {
    url: 'https://rpc.testnet.lukso.network',
    chainId: 4201,
  };

  if (process.env.PRIVATE_KEY !== undefined) {
    config['accounts'] = [process.env.PRIVATE_KEY];
  }

  return config;
}

const config: HardhatUserConfig = {
  solidity: {
    version: '0.8.19',
    settings: {
      optimizer: {
        enabled: true,
        runs: 200,
      },
    },
  },
  typechain: {
    outDir: 'typechain-types',
    target: 'ethers-v6',
  },
  networks: {
    luksoTestnet: getTestnetChainConfig(),
  },
};

export default config;
```

</file>

<file>

path: /package.json

url: https://github.com/CJ42/LUKSO-Hardhat-template/blob/main/package.json

```
{
  "name": "LUKSO-Hardhat-Template",
  "description": "Example repo for Hackathon to build a NFT collection using LSP7 or LSP8",
  "scripts": {
    "clean": "hardhat clean",
    "build": "hardhat compile"
  },
  "devDependencies": {
    "@nomicfoundation/hardhat-chai-matchers": "^2.0.0",
    "@nomicfoundation/hardhat-ethers": "^3.0.0",
    "@nomicfoundation/hardhat-network-helpers": "^1.0.0",
    "@nomicfoundation/hardhat-toolbox": "^3.0.0",
    "@nomicfoundation/hardhat-verify": "^1.0.0",
    "@typechain/ethers-v6": "^0.4.0",
    "@typechain/hardhat": "^8.0.0",
    "@types/chai": "^4.2.0",
    "@types/mocha": ">=9.1.0",
    "@types/node": ">=16.0.0",
    "chai": "^4.2.0",
    "dotenv": "^16.3.1",
    "ethers": "^6.4.0",
    "hardhat": "^2.18.2",
    "hardhat-gas-reporter": "^1.0.8",
    "prettier-plugin-solidity": "^1.1.3",
    "solidity-coverage": "^0.8.0",
    "ts-node": ">=8.0.0",
    "typechain": "^8.1.0",
    "typescript": ">=4.5.0"
  },
  "dependencies": {
    "@erc725/smart-contracts": "^6.0.0",
    "@lukso/lsp-factory.js": "^3.1.1",
    "@lukso/lsp-smart-contracts": "^0.12.0-rc.0"
  }
}
```

</file>

<file>

path: /scripts/DeployAndSetLSP8MetadataBaseURI.ts

url: https://github.com/CJ42/LUKSO-Hardhat-template/blob/main/scripts/DeployAndSetLSP8MetadataBaseURI.ts

```
import { ethers } from "hardhat";

import {BasicNFTCollection, BasicNFTCollection__factory} from "../typechain-types";

const LSP8TokenMetadataBaseURI =
  '0x1a7628600c3bac7101f53697f48df381ddc36b9015e7d7c9c5633d1252aa2843';

async function deployAndSetLSP8MetadataBaseURI() {
    const accounts = await ethers.getSigners();
    const deployer = accounts[0];

    const boredApesCollection: BasicNFTCollection = await new BasicNFTCollection__factory(
        deployer,
      ).deploy('BoredApeYachtClub', 'BAYC', deployer.address);

      await boredApesCollection.setData(
        LSP8TokenMetadataBaseURI,
        ethers.concat(
            // &nbsp;0x6f357c6a&nbsp; represents the hash function identifier,
            // the first 4 bytes of keccak256('keccak256(utf8)')
            // to be used to ensure that the metadata of the NFT is set in stone and cannot be changed
            // (for verifiability purpose)
            ["0x6f357c6a", ethers.toUtf8Bytes('ipfs://your-base-uri-on-ipf-goes-here/')]
        )
      );
}

deployAndSetLSP8MetadataBaseURI().catch((error) => {
    console.error(error);
    process.exitCode = 1;
  });
```

</file>

<file>

path: /scripts/deploy.ts

url: https://github.com/CJ42/LUKSO-Hardhat-template/blob/main/scripts/deploy.ts

```
import { ethers } from "hardhat";

import {BasicNFTCollection, BasicNFTCollection__factory} from "../typechain-types";

async function deployLSP8Collection() {
  const accounts = await ethers.getSigners();
  const deployer = accounts[0];

  const nftCollection: BasicNFTCollection = await new BasicNFTCollection__factory(deployer).deploy(
    "NFT Collection Name", // collection name
    "NFT", // collection symbol
    deployer.address
  );

}

deployLSP8Collection().catch((error) => {
  console.error(error);
  process.exitCode = 1;
});
```

</file>

<file>

path: /scripts/mintTickets.ts

url: https://github.com/CJ42/LUKSO-Hardhat-template/blob/main/scripts/mintTickets.ts

```
import { ethers } from "hardhat";

import {
  EventTicketsNFT,
  EventTicketsNFT__factory
} from "../typechain-types";

async function deployAndCreateTickets() {
  const accounts = await ethers.getSigners();
  const deployer = accounts[0];

  const luksoMeetupTickets: EventTicketsNFT = await new EventTicketsNFT__factory(
    deployer
  ).deploy(
    "LUKSO Meetup #2",
    "MUP2",
    deployer.address,
  )

  // create 100 entry tickets.
  // Give them to the deployer initially, who will distribute them afterwards.
  await luksoMeetupTickets.mint(
    deployer.address, // recipient
    100, // amount
    true, // force sending to an EOA
    "minting 100 tickets" // data
  );
}

deployAndCreateTickets().catch((error) => {
  console.error(error);
  process.exitCode = 1;
 });
```

</file>

<file>

path: /scripts/setDigitalStickerMetadata.ts

url: https://github.com/CJ42/LUKSO-Hardhat-template/blob/main/scripts/setDigitalStickerMetadata.ts

```
import { LSPFactory } from '@lukso/lsp-factory.js';
import { ethers } from 'hardhat';

import * as dotenv from "dotenv"
dotenv.config()

const provider = 'https://rpc.testnet.lukso.network';

const lspFactory = new LSPFactory(provider, {
  deployKey: process.env.PRIVATE_KEY, // Private key of the account which will deploy smart contracts
  chainId: 4201,
});

async function setStickerMetadata() {
    const accounts = await ethers.getSigners()
    const signer = accounts[0]

    console.log(process.env.PRIVATE_KEY)
    console.log(signer)
}

setStickerMetadata().catch((error) => {
    console.error(error);
    process.exitCode = 1;
  });
```

</file>

<file>

path: /test/NFTCollection.ts

url: https://github.com/CJ42/LUKSO-Hardhat-template/blob/main/test/NFTCollection.ts

```
import { time, loadFixture } from '@nomicfoundation/hardhat-toolbox/network-helpers';
import { anyValue } from '@nomicfoundation/hardhat-chai-matchers/withArgs';
import { expect } from 'chai';
import { ethers } from 'hardhat';

import { BasicNFTCollection, BasicNFTCollection__factory } from '../typechain-types';

const LSP8MetadataTokenURI = '0x1339e76a390b7b9ec9010000';
const LSP8ReferenceContract =
'0x708e7b881795f2e6b6c2752108c177ec89248458de3bf69d0d43480b3e5034e6';
const LSP8TokenMetadataBaseURI =
  '0x1a7628600c3bac7101f53697f48df381ddc36b9015e7d7c9c5633d1252aa2843';

describe('BoredApes Collection', () => {
  it('test', async () => {
    const accounts = await ethers.getSigners();
    const deployer = accounts[0];

    const boredApesCollection: BasicNFTCollection = await new BasicNFTCollection__factory(
      deployer,
    ).deploy('BoredApeYachtClub', 'BAYC', deployer.address);

    await boredApesCollection.setData(
      LSP8TokenMetadataBaseURI,
      ethers.toUtf8Bytes('ipfs://your-base-uri-on-ipf-goes-here/'),
    );

    expect(await boredApesCollection.getData(LSP8TokenMetadataBaseURI)).to.equal(
      ethers.toUtf8Bytes('ipfs://your-base-uri-on-ipf-goes-here/'),
    );
  });
});
```

</file>

<file>

path: /tsconfig.json

url: https://github.com/CJ42/LUKSO-Hardhat-template/blob/main/tsconfig.json

```json
{
  "compilerOptions": {
    "target": "es2020",
    "module": "commonjs",
    "esModuleInterop": true,
    "forceConsistentCasingInFileNames": true,
    "strict": true,
    "skipLibCheck": true,
    "resolveJsonModule": true
  }
}
```

</file>

</repository>

Demo-AddingLSP17Extensions

overview

repository: https://github.com/YamenMerhi/Demo-AddingLSP17Extensions/
userName: YamenMerhi
repository: Demo-AddingLSP17Extensions
branch: main
date: 2023-11-27T10:16:38+01:00 (1701076598)
description:
hello everyone it's y from the tech team
in this video we're going to learn how
to extend the functionalities of a
universal profile using LSP 17 contract
extension standard so before going into
the LSP 17 let's start with the
universal
profile a universal profile is an lsp0
ERC 725 account and when we go to the
spec we see that it's a nonbi smart
contract account which mean it have
basic functionalities like executing
setting data Val dating signature
ownership management being notified
about different actions and reacting on
it and a unified way to interact with
the
contract so here we can see most of the
essential functionality of a universal
profile and once it's deployed we will
just have these native function so
imagine someone wants to add a specific
function to the universal profile after
it has been deployed should he create a
new Universal profile with this
functionality obviously not because your
Universal profile will hold your
reputation which you cannot migrate to a
new one because whenever you deploy a
new account the address will change and
you'll need to migrate your data and
your tokens which is not an easy thing
to do whenever you want to add a new
functionality to the universal profile
so what's the
solution the solution is LSP 17 contract
extension which is a standard that the

universal profile adopt that will allow
for the contract to receive function
calls of function that does not exist
natively in the contract and forward it
to an extension where it actually exists
passing the relevant information in this
way even after deployment smart
contracts will have the ability to
support new
functionalities so here for the
universal profile we can see that we can
add extension and why is this useful and
needed is because for example this
profile will be deployed at a certain
stage in time but even after deployment
the process of standardization in the
ecosystem will continue so other
standard will get adopted in the future
and the profile might need to support
them similar to what's happening now
with EIP 4337 where in order to be
compatible with the standard you'll need
to have a function called validate user
op which you can see on the screen that
we can add as an extension and also some
token require a specific function that
return abides for Value in order to make
a safe transfer successful like erc721
and
1155 which anyone can decide to support
by adding the relevant extension so LSP
17 not just ensure the ability to extend
the profile with new function and
interface IDs but also emphasize on the
point that a universal profile is a
non-biased account where only essential
feature are within the account and
everything else can be added as an
extension so we're going to see a demo
on how to add the extension to the
universal profile and what you can do
with
them so I created a hard hat project and
you could easily do that by doing npmi
hard hat and then you'll install hard
hat and after this you would need to
initiate the project so you do npx
hardhead in it um I created a hardhead
project using typescript but feel free
to do it with JavaScript if you're more
comfortable with it so you'll get a hard
hat project and then you would have some

templates in contract scripts uh but
this is the main template that we're
going to use and then we're expand from
this uh and at the end of the video you
can go to the this GitHub repo where
you'll find all of the final code so the
package that I'm using are prettier hard
hat. EnV so you just install EnV uh to
put the environment variables uh also
we're using the LSP smart contract
package so you would also need to
download it in this way because we need
to extract some constant and uh we're
going to use some contracts uh so You'
put that into the EnV so I created a
sample file for the EnV so we'll need
the private key of the browser extension
controller there will be other tutorials
that show how to extract that from the
browser extension and we would need the
universal profile address and obviously
you would put that into the EnV I will
not show you this because it contain my
private key so we have this main setup
I'm going to explain it so we import
ethers we load the environment variables
we import this Universal profile ABI
from the LSP smart contract package and
then we connect our code to the loxo
testnet we initiate the signer which is
the browser extension controller and we
create an instance of the universal
profile using the address that we
imported from the environment
variable and the universal profile AI so
we can interact with a function of the
universal profile and the provider which
is where the universal profile actually
exists which is on the test n
uh so I'm going to run this script so
you can see
the so you can see my I can see the
address of my browser extension
controller and the universal profile and
then we're going to start so we're going
to start by calling a function that does
not exist on the universal profile and
I'm going to take this main
setup and I'm going to put it here and
because we're going to call a function
that does not exist we going to create a
new API so let's say the function that

does not exist is called buy me a
Coffey so this is the function and now
we're going
to update the universal profile with the
new AI which is this one and we're going
to try to call it and we're going to see
what's going to happen so we connect to
the signer so the browser extension
controller will be executing this
transaction and then we're going to try
to call the
function and let's see what
happened so it's going to revert so
execution reverted unknown custom error
and if we take
the the signature and we put it into the
bytes for dictionary
so we just do this and then we extract
all of this thing take the first four
bites we're going to see that this
custom error is for no extension F found
for function selector so this means that
this function which is buy me a coffee
does not exist natively on the up and
also there is no extension for it so
what we're going to do is that we're
going to go create a new contract we
call this contract coffee maker and
we're going to put this function here
and then at the end we're going to add
this contract as an extension for the
universal profile so here we inherited
the LSP 17 extension contract because it
provides some utility functions that
we're going to see so we have extendable
message data extendable message. sender
and extendable message Dov
value so I create uh I explained this
before but what's happening is that a
person uh or an address is calling the
universal profile with this function and
then the universal profile is calling
the contract that will be added as an
extension so that mean the message.
sender of the universal profile is the
actual person calling but the message.
sender of the extension will be the
universal profile so if we're going to
do some validation based on like who who
called the extension we cannot do
message. sender because in this case it
will be the universal profile address so

because we're uh appending data to the
extensions we can access that with the
extendable message sender so that mean
this is the person who actually called
the universal profile with the function
selector and we can also get access to
the extendable message.
value so we can see it here extendable
message. Val which is how much this
person have sent while calling this
function
and the universal profile will obviously
communicate this information to the
extension and if you go to the
specification of the LSP Z you will see
how you can um do some configuration in
order to make the value forwarded to the
actual extension but in our case we'll
just uh see how it's communicated so
we're going to proceed with adding the
function that we agreed on its name
which is buy me a
coffee so we're going to create a public
function and we're not going to do
anything uh complex we're just going to
create a public variable called coffee
revenue and this coffee
Revenue will be increased each
time send us value so we're going to say
that the coffein coffee Revenue
will increase each time someone send
value to the universal profile so in
case it's uh deployed on the uh Luxo
Network it will be licks how much licks
you're sending so we're going to deploy
this it's super simple so obviously if
we run the script again it will still
fail because we just coded the solidity
part and we didn't deploy it or add it
as an extension yet so what we're going
to do is that we're going to do npx
hardat compile
and
then we're going to add the function
that does not exist so we're going to
take the main
setup put it here and now we would need
several stuff so first we need to
deploy the coffee
maker so we're going to take the factory
from the type chain and then
we're going to say that this is the

coffee
maker we're going to attach the signer
just going to deploy
it and here we can just
log the
address of the copy
extension Target so there's the
difference between
uh ether's V6 and V5 so if you're using
V5 it will be here do address instead of
do Target but I'm using the latest
version so we can deploy it but at the
same time let's add um let's add it as
an extension so what we're going to do
is that if we go here we can probably
take the function
signature of the function which is this
one buy me a coffee
but we're not going to take it from here
what we're going to do is through the
code we're going to say
that by
me coffee selector is equal to ethers
then we're going to use the ID method
we're going to say we're going to copy
the function
name so we get that
correctly and then this will result in a
32 by hash so we're just going to
take the first 10 characters which mean
it's ox and then four bytes which is the
function
selector um after this we're going to
construct the data key of adding uh the
extension so we're going to
use the ERC 725 data key which are
provided by the package so you can
import that from the package and here
you can use some utility libraries like
R c725 JS to construct the data and
fetch them from the contract but I will
try to do it without them to to also
emphasize that if it's a simple thing
you can do it without the uh library but
if you're if you have many fetching and
constructing it is recommended to use
ERC 725 GS or the utilities of the LSP
smart contract so I'm going to construct
the data key um by
saying constructed data key equal so
first we take the ERC 75y data key which
is the LSP 17 and we take the prefix so

the prefix will be 12 bytes which which
are defined in the standard and now we
need to take this uh 4 byte selector and
Pad it so we're going to use ethers
do0 pad bytes this one and then we're
going to take the selector
and we're going to pad this by 20 bytes
because it's 12 and we want to make it
as 32 by data key and then at the end
we're just going to remove the ZX from
the
front so this is the constructed data
key and now what we're going to do is
that we're going to set this data key
with the address of the extension so
what we're saying is that whenever the
universal profile is called with this
function selector call the extension
which we deployed here so this is what
we're going to do we're going to set
data on the universal profile using the
signer and here it's important to note
that um the signer which is the browser
extension should have all the permission
needed here the permission needed are
ADD extension which if you want to
modify you'll need edit extension but
make sure that uh your Universal profile
browser extension controller have the
ADD extension uh permission there's
other tutorials that show how to detect
this and how to add it in case you don't
have it so we're going to call set data
and then this is the construct data key
and the data value would be the coffee
maker address which we can access by
doing
this and then we would need to wait
uh is mind so now we're adding function
that does not exist let's see if we made
some mistake hopefully we didn't okay we
can take so we can see that it's
deployed and we can take the address so
we can use it later and we can put it
here so this is the address of the
coffee maker extension so everything
went through and now if we try to call
the function that does not exist second
time hopefully it will go
through npx hard but we need to make
sure of the requirement of the function
if you had some requirements here you

need to make sure these are fulfilled so
here we can modify it to say require
that um the extendable message do value
is more than five in this way you're
requiring that anyone who wants to buy
you a coffee that he send you more than
five way or like you can put it whatever
value you want so we're going to call
the function
again so we're going to try to call the
function after adding it as an
extension and we're going to see that
this time it didn't trovert
actually so this time we're calling it
and what we're doing is that we're
saying the extendable message do value
is incremented with how much value we
send so here we didn't send we didn't
send any value so I'm going to try to
say I'm going to
send um so let's let's see the default
okay let's take the main setup and check
the
extension so this is the address of the
extension so here we're
saying the ABI we don't need to use the
universal profile Abi I'm going to say
the ABI
is
function coffee
Revenue then it returns a un to 56 let's
make sure it's correct
yep and then we're going to instantiate
the
extension with this
address
and with this
Ai and with this provider so we don't
need to use these ones and now what
we're going to do is that we're just
going to call the extension with
the this signer and which function we're
going to call is the
coffee Revenue do static call so here
we're going to do a static
call
and so we're going to put the result and
then we're going to
log the result hopefully it will go
through so now what we're doing is that
we're checking the extension if there's
some

changes it's checking the
extension so here it showed zero because
actually when we called the extension
here we didn't forward any value and
let's see now if we forward some value
if it will work or not so I'm going to
send seven way here we didn't put any
restrictions so we can send as much as
we want so I'm going to
send 7 way and I'm going to do calling a
function that does not
exist and we can see it takes some time
so it can get mined on the
blockchain okay let's check now if the
revenue went
up let's do check extension hopefully we
can see yeah you can see that it went
from zero to 7 which is actually the
number that we sent so this can show you
that you can even control in the
extension the value that can be sent to
your Universal profile while calling a
function that does not exist so this
give you a big amount of flexibility so
the last thing we're going to do uh is
that we're going to try to also extend
the universal profile with other
interface ID so the whole point of
extending with function is to add new
functionalities to the universal profile
but what usually websites do is that
they check an interface ID for the
universal profile profile if it's
supported so they can start preparing to
call a function so if you go to the
universal profile now you can see that
it support ERC 75x and ERC 725 Y and
what websites do is before doing any
transaction they check if if it support
this interface ID and if it doesn't
support it it just don't do the call so
now we're going to create a new contract
and let's call this contract
supports
interface and
then just name this we don't need
actually now the extension just for
Simplicity I would do
function support
interface it will take a byes
for
then public and it will return aoan

whether this interface ID is supported
or
not so for the Simplicity we're just
going to hardcode it and say if it's
support ABC ab a a b b c c d and we're
going to
compile it went through now what we're
going to do take the main set up and put
it in calling supports interface what
we're going to do is just simply call
the up because the supports interface
function exist on the up but we just
want to extend it so we do up connect to
the signer and recall
supports
interface with Ox aab BB CC
DD and let's see the result obviously
now we didn't add the extension yet so
it's okay if it
returns false let's
see so actually it returned false
because this interface ID is not
supported by the universal profile and
what we're going to do now is adding
this um we're going to add this
extension so here we can modify this
code to just instead
of adding the extension of buy me a
coffee we're going to do supports
interface byes
4 here instead of the coffee maker we're
going to take the supports interface
factorys interface
extension and we can log log
it
supports
interface and then we just replace it
here and so what we're
doing we're first deploying the uh
extension that extend the uh supports
interface function this extension will
support obviously it's here the this
interface
ID and then we're Computing the selector
of supports interface and we're
constructing it with the with the LSP 17
extension prefix and then we're setting
it on the universal profile so let's run
the script of adding the
extension so it's going to take a little
bit of time so we can see this is the
address not the coffee extension ah I

left it
so we can do support
words actually it's okay no one is
seeing now so we can take this and wait
till get mined so this is actually this
is a typo it's not the address of the
coffee extension it's the address of the
supports interface we're going to take
it
and we're
gonna actually we don't need it because
we're going to directly call uh we're
going to directly check the interface ID
on the universal profile so before we
checked the interface ID this one and it
returned false but now if we do it
again calling supports interface it will
be true so that means not just you can
extend the functionality of a universal
profile with new functions but you can
also make the universal profile support
new interface IDs and this is good for
detection of what feature they Universal
profile will eventually have so this was
the demo as I said if you want to look
at the code this will be all published
to this GitHub repository thank you

<file>

path: /.env.sample

url: https://github.com/YamenMerhi/Demo-AddingLSP17Extensions/blob/main/.env.sample

```
PRIVATE_KEY=
UP_ADDR=
```

</file>

<file>

path: /README.md

url: https://github.com/YamenMerhi/Demo-AddingLSP17Extensions/blob/main/README.md

Demo-AddingLSP17Extensions</file>

<file>

path: /contracts/CoffeeMaker.sol

url: https://github.com/YamenMerhi/Demo-AddingLSP17Extensions/blob/main/contracts/CoffeeMaker.sol

```
// SPDX-License-Identifier: UNLICENSED
pragma solidity ^0.8.17;

import {LSP17Extension} from "@lukso/lsp-smart-contracts/contracts/LSP17ContractExtension/LSP17Extension.sol";

contract CoffeeMaker is LSP17Extension {
    uint256 public coffeeRevenue;

    function buyMeACoffee() public {
        coffeeRevenue += _extendableMsgValue();
    }
}
```

</file>

<file>

path: /contracts/SupportsInterface.sol

url: https://github.com/YamenMerhi/Demo-AddingLSP17Extensions/blob/main/contracts/SupportsInterface.sol

```
// SPDX-License-Identifier: UNLICENSED
pragma solidity ^0.8.17;

contract SupportsInterface {
    function supportsInterface(bytes4 interfaceId) public pure returns (bool) {
        return interfaceId == 0xaabbccdd;
    }
}
```

</file>

<file>

path: /hardhat.config.ts

url: https://github.com/YamenMerhi/Demo-AddingLSP17Extensions/blob/main/hardhat.config.ts

```
import { HardhatUserConfig } from "hardhat/config";
import "@nomicfoundation/hardhat-toolbox";

const config: HardhatUserConfig = {
  solidity: "0.8.19",
};

export default config;
```

</file>

<file>

path: /package.json

url: https://github.com/YamenMerhi/Demo-AddingLSP17Extensions/blob/main/package.json

```
{
  "dependencies": {
    "@lukso/lsp-smart-contracts": "^0.12.0-rc.0",
    "dotenv": "^16.3.1",
    "hardhat": "^2.18.3",
    "prettier": "^3.0.3"
  },
  "devDependencies": {
    "@nomicfoundation/hardhat-toolbox": "^3.0.0"
  }
}
```

</file>

<file>

path: /scripts/AddingAFunctionThatDoesNotExist.ts

url: https://github.com/YamenMerhi/Demo-AddingLSP17Extensions/blob/main/scripts/AddingAFunctionThatDoesNotExist.ts

```
import { ethers } from 'hardhat';
import * as dotenv from 'dotenv';
import { abi as UP_ABI } from '@lukso/lsp-smart-contracts/artifacts/UniversalProfile.json';

import { CoffeeMaker__factory } from '../typechain-types';
import { ERC725YDataKeys } from '@lukso/lsp-smart-contracts';

// load env vars
dotenv.config();

// Update those values in the .env file
const { UP_ADDR, PRIVATE_KEY } = process.env;

async function main() {

    // setup provider
    const provider = new ethers.JsonRpcProvider('https://rpc.testnet.lukso.network');
    // setup signer (the browser extension controller)
    const signer = new ethers.Wallet(PRIVATE_KEY as string, provider);

    let UP = new ethers.Contract(UP_ADDR as string, UP_ABI, provider);

    console.log('      EOA: ', signer.address);
    console.log('      Universal Profile: ', await UP.getAddress());

    const coffeeMakerExtension = await new CoffeeMaker__factory(signer).deploy();
    console.log("The address of the  extension", coffeeMakerExtension.target);

    const buyMeACoffeeSelector = ethers.id("buyMeACoffee()").substring(0, 10);

    const constructedDataKey = ERC725YDataKeys.LSP17.LSP17ExtensionPrefix +
ethers.zeroPadBytes(buyMeACoffeeSelector, 20).substring(2);

    const tx = await UP.connect(signer).setData(constructedDataKey, coffeeMakerExtension.target);
    await tx.wait();
}

main()
    .then(() => process.exit(0))
    .catch((error) => {
        console.error(error);
        process.exit(1);
    });
```

</file>

<file>

path: /scripts/CallingAFunctionThatDoesNotExist.ts

url: https://github.com/YamenMerhi/Demo-AddingLSP17Extensions/blob/main/scripts/CallingAFunctionThatDoesNotExist.ts

```
import { ethers } from 'hardhat';
import * as dotenv from 'dotenv';
import { abi as UP_ABI } from '@lukso/lsp-smart-contracts/artifacts/UniversalProfile.json';

// load env vars
dotenv.config();

// Update those values in the .env file
const { UP_ADDR, PRIVATE_KEY } = process.env;

async function main() {

    // setup provider
    const provider = new ethers.JsonRpcProvider('https://rpc.testnet.lukso.network');
    // setup signer (the browser extension controller)
    const signer = new ethers.Wallet(PRIVATE_KEY as string, provider);

    let UP = new ethers.Contract(UP_ADDR as string, UP_ABI, provider);

    console.log('      EOA: ', signer.address);
    console.log('      Universal Profile: ', await UP.getAddress());

    const abi = ["function buyMeACoffee()"];
    UP = new ethers.Contract(UP_ADDR as string, abi, provider);


    const valueToSend = 7;

    const tx = await UP.connect(signer).buyMeACoffee({ value: valueToSend });
    await tx.wait();

}

main()
    .then(() => process.exit(0))
    .catch((error) => {
        console.error(error);
        process.exit(1);
    });
```

</file>

<file>

path: /scripts/CallingSupportsInterface.ts

url: https://github.com/YamenMerhi/Demo-AddingLSP17Extensions/blob/main/scripts/CallingSupportsInterface.ts

```
import { ethers } from 'hardhat';
import * as dotenv from 'dotenv';
import { abi as UP_ABI } from '@lukso/lsp-smart-contracts/artifacts/UniversalProfile.json';

// load env vars
dotenv.config();

// Update those values in the .env file
const { UP_ADDR, PRIVATE_KEY } = process.env;

async function main() {

    // setup provider
    const provider = new ethers.JsonRpcProvider('https://rpc.testnet.lukso.network');
    // setup signer (the browser extension controller)
    const signer = new ethers.Wallet(PRIVATE_KEY as string, provider);

    let UP = new ethers.Contract(UP_ADDR as string, UP_ABI, provider);

    console.log('    EOA: ', signer.address);
    console.log('    Universal Profile: ', await UP.getAddress());


    const interfaceIdToQuery = "0xaabbccdd";

    const isSupported = await UP.connect(signer).supportsInterface(interfaceIdToQuery);
    console.log(isSupported);
}

main()
    .then(() => process.exit(0))
    .catch((error) => {
        console.error(error);
        process.exit(1);
    });
```

</file>

<file>

path: /scripts/CheckingExtension.ts

url: https://github.com/YamenMerhi/Demo-AddingLSP17Extensions/blob/main/scripts/CheckingExtension.ts

```
// Coffee Maker extension 0x77D4195988Ebd257c0F7Bfc50715A1FCe3eDCA6a

import { ethers } from 'hardhat';
import * as dotenv from 'dotenv';

// load env vars
dotenv.config();

// Update those values in the .env file
const { PRIVATE_KEY } = process.env;

async function main() {

  // setup provider
  const provider = new ethers.JsonRpcProvider('https://rpc.testnet.lukso.network');
  // setup signer (the browser extension controller)
  const signer = new ethers.Wallet(PRIVATE_KEY as string, provider);


  const abi = ["function coffeeRevenue() returns (uint256)"];

  // Insert the extension address here
  const extensionAddress = "0x...."

  let extension = new ethers.Contract(extensionAddress, abi, provider);

  const result = await extension.connect(signer).coffeeRevenue.staticCall();
  console.log(result);
}

main()
  .then(() => process.exit(0))
  .catch((error) => {
    console.error(error);
    process.exit(1);
  });
```

</file>

<file>

path: /scripts/SupportingANewInterfaceId.ts

url: https://github.com/YamenMerhi/Demo-AddingLSP17Extensions/blob/main/scripts/SupportingANewInterfaceId.ts

```
import { ethers } from 'hardhat';
import * as dotenv from 'dotenv';
import { abi as UP_ABI } from '@lukso/lsp-smart-contracts/artifacts/UniversalProfile.json';

import { SupportsInterface__factory } from '../typechain-types';
import { ERC725YDataKeys } from '@lukso/lsp-smart-contracts';

// load env vars
dotenv.config();

// Update those values in the .env file
const { UP_ADDR, PRIVATE_KEY } = process.env;

async function main() {

  // setup provider
  const provider = new ethers.JsonRpcProvider('https://rpc.testnet.lukso.network');
  // setup signer (the browser extension controller)
  const signer = new ethers.Wallet(PRIVATE_KEY as string, provider);

  let UP = new ethers.Contract(UP_ADDR as string, UP_ABI, provider);

  console.log('     EOA: ', signer.address);
  console.log('     Universal Profile: ', await UP.getAddress());

  // The contract to deploy which have the function supportsInterface that supports
  // the new interfaceId that the UniversalProfile needs to support
  const supportsInterfaceExtension = await new SupportsInterface__factory(signer).deploy();
  console.log("The address of the  extension", supportsInterfaceExtension.target);


  const supportsInterfaceSelector = ethers.id("supportsInterface(bytes4)").substring(0, 10);
  const constructedDataKey = ERC725YDataKeys.LSP17.LSP17ExtensionPrefix +
ethers.zeroPadBytes(supportsInterfaceSelector, 20).substring(2);

  const tx = await UP.connect(signer).setData(constructedDataKey, supportsInterfaceExtension.target);
  await tx.wait();
}

main()
  .then(() => process.exit(0))
  .catch((error) => {
    console.error(error);
    process.exit(1);
  });
```

</file>

<file>

path: /scripts/mainSetup.ts

url: https://github.com/YamenMerhi/Demo-AddingLSP17Extensions/blob/main/scripts/mainSetup.ts

```
import { ethers } from 'hardhat';
import * as dotenv from 'dotenv';
import { abi as UP_ABI } from '@lukso/lsp-smart-contracts/artifacts/UniversalProfile.json';

// load env vars
dotenv.config();

// Update those values in the .env file
const { UP_ADDR, PRIVATE_KEY } = process.env;

async function main() {

  // setup provider
  const provider = new ethers.JsonRpcProvider('https://rpc.testnet.lukso.network');
  // setup signer (the browser extension controller)
  const signer = new ethers.Wallet(PRIVATE_KEY as string, provider);

  let UP = new ethers.Contract(UP_ADDR as string, UP_ABI, provider);

  console.log('    EOA: ', signer.address);
  console.log('    Universal Profile: ', await UP.getAddress());
}

main()
  .then(() => process.exit(0))
  .catch((error) => {
    console.error(error);
    process.exit(1);
  });
```

</file>

<file>

path: /tsconfig.json

url: https://github.com/YamenMerhi/Demo-AddingLSP17Extensions/blob/main/tsconfig.json

```
{
 "compilerOptions": {
  "target": "es2020",
  "module": "commonjs",
  "esModuleInterop": true,
  "forceConsistentCasingInFileNames": true,
  "strict": true,
  "skipLibCheck": true,
  "resolveJsonModule": true
 }
}
```

</file>

</repository>

<repository>

Demo-AddingLSP1UniversalReceiverDelegates

overview

repository: https://github.com/YamenMerhi/Demo-AddingLSP1UniversalReceiverDelegates/
userName: YamenMerhi
repository: Demo-AddingLSP1UniversalReceiverDelegates
branch: main
date: 2023-11-27T10:16:38+01:00 (1701076598)
description:
hello everyone it's yam from the teting
in this video we're going to learn how
to add reactivity to the universal
profile using lsp1 universal receiver
standard so let's go through the
universal receiver and know more about
the problem that this standard solve
we're going to do that by taking an
example of an erc20 token since it's the
most used currently in the space during
an erc20 token transfer the sender will
call the transfer function on the erc20
token and then this token contract will
just decrease the balance of the sender
and increase the balance of the
recipient without further interaction
the recipient will be totally excluded
from the interaction regardless if it's
an externally owned account or a smart
contract so this recipient cannot know
which token it owns and even other
standard where a function on the
recipient is defined the notification is
limited and designed in a bad way so for
instance ERC 721 1155 and 223 Define
function functions to be available in
the recipient contract to accomplish a
successful safe transfer so let's say as
a recipient my contract is deployed and
two years down the road a new nft
standard get adopted my contract will
not have the ability to receive these
kind of tokens because it won't be
implementing the mapped ERC XXX received
function other than that these function
are not mandatory to be called these
will be called only on safe transfer and

also the standard does not defy an event
to be emitted on the recipient so
offchain clients cannot easily filter
what tokens you own as a recipient they
would need to query the whole blockchain
and token contract to know what token
you own so this shows that this way of
notifying is broken and we need to find
a better way and this is possible with
the universal receiver function that we
standardized in
lsp1 so lsp1 Define a function called
universal receiver and the whole point
of this function is to emit an event to
the network network with the data passed
to it so let's say you inform a smart
contract about a specific message an
event will be emitted with who's
informing which is the sender how much
value he's sending what type of message
he's sending and the additional data
attached to it and the response of the
universal receiver function not not just
that this function can allow you to
react on call so for instance you can
specify in the function that whenever a
specific entity send you a token half of
these token will be sent to a vault you
own or distributed equally to other
recipient addresses the standard also
take modularity into consideration so
it's not recommended to hard code a
specific logic within the universal
receiver function but rather delegate
the responsibility of reacting to an
external contract called the universal
receiver delegate and this is because
you might need to change how you react
on specific call so now you can decide
to split all the token you receive 50/50
between you and the secret vault or
later you might you might want to disect
allow anyone from sending you a token or
filter the spam token and only allow
whitelisted token so universal receiver
implement the univers universal profile
implement the universal receiver
function and as you can see here you can
have your tokens inform the sender and
inform the recipient about the transfer
in this way not just you'll get notified
but also you can react on the token

transfer as shown on the screen the
universal receiver delegate can contain
any logic from rejecting the transfer
forwarding the asset to a vault or a
staking pool
Etc so in a previous video we explained
why there is a need to have LSP 17
contract extensions within the universal
profile and how you can add them however
it's important to differentiate between
notifying the universal profile using
universal receiver and calling an
extension on the universal profile when
a token will notify you the events will
be emitted on your profile in this way
if I need to know what tokens or
followers or whatever people decide to
notify about you can just listen to the
universal receiver event emitted on the
profile address however with LSP 17
contract extension websites need to
fetch what extension you previously had
from the events and then fetch the
events from these extension gathered so
this will be much slower and unreliable
plus the universal receiver is inclusive
meaning that by default you'll not
reject any call to inform you unless you
decide to however with the extent tion
if someone is informing You by calling
the extension that you did not add yet
the call will fail so it's all about
having more flexibility and knowing
where is the right place to perform
certain actions LSP 17 contract
extension are to extend the
functionality of the universal profile
and universal receiver is to inform you
about incoming transaction and you being
able to react on them and with that
being said let's see a demo on how to
use the universal receiver and how to
add universal receiver delegates
so I created a new hard hat project you
can do that by installing hard
hat so you'll do npm install hard hat
and then you'll do npx hard hat in it
where you initiate a new hard hat
project so I uh created a typescript
project uh feel free to do that with
JavaScript if it's easier for you uh
these are the packages that I'm using so

you need to install the LSP smart
contract package by doing that uh I use
frer
andv uh to have the environment
variables and this package will be
automatically installed with hard hat so
for the samples of the uh environment
variables you'll need to add the private
key of your browser extension controller
so there will be other tutorials that
show how to uh extract the private key
from the browser extension and here you
would put the address so obviously you
need to put that in the EnV not in the
SLE this is just to show you how to do
it so we're going to start by looking at
the main setup that I created um we have
here we're importing ethers from hard
hat we are taking the universal profile
AI from the LSP smart contract package
and we're instantiating uh a new
instance of the universal profile by
passing The Universal profile address
the ABI and the provider which connect
us to the Luxo test so we can run this
script to see
our um addresses so we're going do the
main
setup and we can see here this is the
address of the browser extension
controller and this is the address of
our Universal profile so in this
specific tutorial we're going to uh call
the universal receiver function and
we're going to watch the events emited
and we're going to say why this is
useful so first let's start by adding a
listener to the universal receiver event
so we're going to create this and we're
going to do Universal profile on and
we're going to listen to the universal
receiver so obviously there's different
parameter with that there's the from
who's calling the universal receiver the
value he's sending the type ID which is
a unique identifier about the action
that the person is calling to notify
about there's the received data
additional data that
you want to notify about and then we can
just
say say the

universal receiver event was
emitted can say the universal
receiver was called
by
from
passing
this
amount has
value to
modify about this type
ID with
these data so we can do is run this
script we can run this script here
here and go create
a new
terminal and within our main setup we're
going to try to call the universal
receiver function and see uh if the
event will be emitted so obviously
events take some time to to be detected
but let's see if it's going to be super
so so what we're going to do is that
we're going to call the UN receiver
function on the profile so the universal
receiver function take two parameter the
type ID and data so the type ID
represent a specific action that you
want to notify about so it should be
like a unique identifier for the action
that you want to notify about so we can
say
um
that where we want to notify about the
royalty so we create the hash of the
word royalty and this will be our unique
identifier and then as data we can pass
whatever we want like the address or
like royalty about like which token
contract or about which product you're
selling so but for now we're just going
to use this dummy
data and then we're going to wrap that
in the
transaction let
run this
transaction and obviously even if you
didn't touch your universal receiver
function before the transaction will
pass because we said the universal
receiver function is inclusive meaning
whenever you deploy your Universal
profile you don't need to do any setup

if someone call this universal receiver
function the transaction will pass
because it's just emitting an event and
it's searching for universal receiver
delegates to react on the uh on the
notification but if you didn't set any
universal receiver delegate we should
expect it to pass so if we go to the
other terminal and wait for a bit we can
see that the event was
emitted so so maybe not this one here we
can see that the universal receiver was
called by Ox E4 which is the signer
because we have seen that whenever I log
the signer it's this one passing value
we didn't send any to notify about which
is the hash of the word royalty and with
a b b c d d as data so what we can do is
pass some data here and see if it
changed can see we're sending
tenway and then we run the script
again and we wait a
bit till it's mind on the blockchain
and we go and we can see that now it's
updated with 10 value and even this not
just uh can be done with externally
owned account by calling the universal
receiver so you can have a token so
let's say we're
creating my token which is and first
let's
import let's import
nsp7 digital
asset
from so there would be other tutorial
that show you how to deploy and uh
create an LSP 7 digital asset for now
we're just going to use it as a sample
to see that a token would notify you
about um about the token
transfer so the lp7 digital asset we
need to pass some
arguments like
um we need to
pass the name of the token
token the
symbol we need to pass the owner which
is let's say the digal sender and then
uh whether it's divisible or not for now
it's not very
important and I'm going to create a
function that's called mint and takes a

recipient and it's called by anyone just
for the sake of the
demo so the amount will be 10 the
recipient
this let's set the true and as data I
just want to pass ZX or let's keep it
empty and now I'll just compile the
contract and I would need to deploy so
I'll just go here take
this and deploy the contract so I need
to
import
and I take the
factory and the signer will deploy this
one and what we're going to do is that
directly we're going
to we're going to Mint some
tokens this is
designer minute and we're going to put
the universal profile
address
so this
then so we don't need this line
and let's run this script which is
deploying contractor
so what did I do as a
mistake
probably
yep so we're waiting till the contract
is
deployed and we're minting some tokens
to the universal profile which is
supposed to notify also the universal
profile so hopefully we can see that
reflect on the
events that we're listening
to so let's wait so it's
deployed so now after fixing the code I
just needed to add a guas limit for some
reason um so I added a guess limit and
it went through it didn't revert so we
can see here that it didn't revert and
hopefully whenever I minted this um
token to the universal profile it should
notify the unal profile and we're going
to see if it did notify or not so we're
going to go here and we see that
actually notified so now the universal
receiver was called by this address we
see that it's a different now not the
signer is calling the universal receiver
because actually the signer is calling

the mint function and the token contract
is notifying the universal profile about
the transfer so this is the address of
the token contract and we say we see
that it not with zero value because it's
not sending a native token it's just the
the token that we
created about this specific type ID so
if you go to the LSP 7 spec you see that
this type ID is to represent that you're
receiving an LS sp7 token so now even
you can say that you want to filter the
universal receiver event with this type
ID and this way you'll just get the
notification when you receive an LSP 7
token and and these obviously are the
data associated with the transfer that
you can uh choose from the token
contract so the whole idea of notifying
and listen listening to the event of the
anniversary receiver is to say that now
we don't need to index the whole
blockchain anymore and check where uh is
the transaction where my address is
associated with now we can just listen
to the universary receiver uh event and
know what uh are we being notified about
or what we receed is it like a token
follower a royalty
anything so now let's see how we can use
the univers receiver delegate so we'll
stick to the example of the token so we
have this LSP 7 digital asset token and
when whenever we're minting or
transferring a token to a recipient this
recipient If he if it's a universal
profile then the universal receiver of
function of this Universal profile will
be called with specific parameters that
we're going to going to see now so if we
go to the LSP 7
specification we can see if we go to the
transfer function uh which applies the
same to the mint function we can see
that if the token recipient is a
contract that support the lsp1 interface
which is the case in uh the universal
profile uh it should call the tokens
recipient universal receiver function
with a parameter below so this is the
type ID that the universal profile will
be notified with so we're going to take

it and for for now let's just put it
here so we can create a logic in our
universal receiver delegate that depend
on this type ID so for instance in this
tutorial let's take a simple example
which is um allowing just whitelisted
token to be received so what we're going
to do is that we're going to create the
universal receiver function universal
receiver delegate in our in the case of
a universal receiver delegate it should
take this form the sender will be the
person not notifying the original
universal receiver function value how
much he sent the type ID which is passed
and the data is pass so we can just do
that and then returns
bites so this need to be added with the
times we have
it so now we can say if the type
ID is equal
equal this one which signal that we're
receiving an LSP 7
token we need to requ we need to First
add a mapping that will hold the logic
of white
listing address then Boolean
public is white
listed and we can have a
function
change white listing then the address of
the token all the
status and for Simplicity we're not
going to add any modifier now but this
should be only callable by the universal
profile so he can control uh which token
to allow or not but for now let's just
keep it simple
and just do
that so uh we're gonna say that
require that the sender which is the
person notifying the universal receiver
function so we're going to
say we're going to
require that the sender is white
listed if he's not white listed it's
going to revert so let's compile
obviously we're going
to ignore the rest we don't need the
value or the data so it's not necessary
that we use them it's just we we use
what we need so we're going to do that

and then we're going to
compile and an important thing that I
just forgot is the support of the lsp1
delegate interface if we don't support
this interface ID then this function
will not run so obviously we need to do
support
interface byes
four so this is a critical aspect to the
universal receiver delegate and we
say
since we're
importing can just take the interface ID
from here and now we can compile and
this should be working so now we're just
compiling and compiled successfully now
we need to deploy this universal
receiver delegate and add it to the uh
Universal profile
so now let's take the invers receiver
delegate Factory actually let's keep
both so we can deploy a
new token so we deployed a new
token let's remove the logic of so this
is y to
Target and then we're going to deploy
the universal receiver dedicate
and we're going
to
emit so let's do
that let's also set a gas limit so we
don't fall so this gas limit is a bit
exaggerated
but it will work so let need ex that
from script deploying contracts it might
take a bit of time
because so the first one is the
address of we're going to put it here
somewhere this is the address of the
token and this is the address of the
receiver
dedicate
cool so now the step would be to first
check if we meant some tokens to the
universal profile it will not fail
because we didn't set the universal
receiver delegate yet so what we can do
is
just so we need to
instantiate the Token contract
so let's do
that let's

do we take the address of the token from
here we just put it here and the API so
I created a function called mint with
address
so it's very simple don't need to make
it
complicated and now we would just call
the mint function to the universal
profile and hopefully it should go
through so we're going to wait a
bit
and it should be
working because it didn't
revert so it didn't trovert obviously
that mean it's working and it Meed some
token but now what we're going to do is
that we're going to add this universal
receiver delegate which will revert
whenever we send the token because it's
not wh listed yet so what we're going to
do
is first we need to construct the data
key of the uh universal receiver
delegate So currently uh the universal
profiles deployed by luo will have a
universal receiver delegate that will be
responsible of uh writing the address of
the assets you receive and the vaults
that you receive and this is a common
behavior that everyone agreed to but
anyone can change this universal
receiver delegate but now we're not
going to mess this
uh setup and we're just going to add
this as an extra universal receiver
delegate just related to the type ID of
receiving LSP 7 token so we're going to
do that we're going to start by
constructing the data key so we're going
to say we need to
import the ERC 725 Y data
key the luo
package we're going to construct
so the type ID we put it right there I
guess so this is the type ID we're going
to need
it type ID of
lp7
recipient
constructed data key
equal and then uh we're going to use the
lsp1 universal receiver prefix which is

which is 12 bytes and then we're going
to concatenate that with the type ID so
we'll need to
take we need to remove the first two
bytes and then take
um uh 20 bytes so it will be 2 to 42 and
then this will be a bytes 32 data key
and then the address will obviously be
the address of the universal receiver
delegate so what we're going to do is
we're going to create a
transaction
that call set data on the universal
profile so if you're doing that with the
universal profile deployed by Lo so you
need to make sure that the address of
the universal profile browser extension
uh the the browser extension controller
has the permission of uh add universal
receiver delegate or edit un receiver
delegate if you're going to remove one
and replace it with this one so there
will be other tutorials that show how to
give uh the universal receiver delegate
permission or like how to edit it from
the extension or programmatically so I'm
not going to do it in this one so I just
do constructed data key and then the
address of the universal receiver
dedicates and then we're going to wait
so we're adding aners receiver
delegate so let's see whether it goes
through or
not so it went through so we assume that
we added our universal receiver delegate
and we're going to make sure uh if it's
working or not so now what we're going
to do is that we're going to try to Min
some tokens to the universal profile and
see if it's reverting or
not we do so we just put them in the
deploying contract it's not
accurate but let just put the code
to
so after waiting for a bit the
transaction went through and it actually
reverted so we can see transaction
execution reverted and here's me trying
again to run the script and it actually
reverted and this is because our
universal receiver delegate is now
reverting on nonwhite listed token so

what we're going to do is that we're
going to go to um
yeah what we're going to do is now we're
going to
whitelist the the token that we have in
order to do that we need to interact
with the universal receiver delegate
contract so also we're going to do that
from the same
contract
so we just uh first load the API so we
need to see what we did rename the
function take this and then instant API
okay and
then we just
instantiate my univers delegate contract
we have the address of the universary
receiv delegate
here then this is the AI and the
provider and we just
do proceed
with connecting with the
ler and then we're going to call the
function so we're going to say change
white listing so first we need the
address of the token so we can Whit list
it we already have it saved
here and then we're going to say true so
it means that it's white listed and then
we're GNA wait till the transaction is
mined so just in case I'm GNA add the
gas
limit or maybe not let's see if it go
through and we wait a bit till the
transaction is
mined and now the trans ction is mined
and we know that we whitelisted the
token so now we try again and this time
the transaction need to go through and
for that to make sure that it went
through I will be listening here to the
universal receiver event so I'm going to
run this
script I'm going to run it here and now
I'm going to proceed with minting a
token to see if it's going to notify and
if it's going to pass through or
not let's
see so we wait a bit and we see that it
didn't revert so this is the first sign
that our universal receiver delegate
worked because earlier it reverted when

we did a mint and now when we go to the
or receiver event we see that this is
the address actual address of the uh of
the token so we can compare it you can
see that this is the address of the
token which mean it minted some token
and notify about this one which is the
type ID that we already set a universal
receiver Delegate for it to react on it
and we have some other information that
is passed by the token
contract so that shows that we can add
any logic to the universal receiver
delegate and make it run on whatever
action we decide to so that needs
standardization in LSP 7 we did it so we
can see here that we
standardize uh the notification for
receiving token and for sending them and
in this way you can react on them so uh
this uh example was a bit simple if you
need to go through more advanced uh
example you can go to our uh docs you
can see here in the guides section
create a universal receiver delegate and
this one is a bit Advanced so you can
see how you can split the LSP 7 token
like transfer a percentage to another
recipient which is a bit more advanced
than just wh
listing so that's the guide uh for today
and if you need to check the code this
will be all organized under the uh this
GitHub repository thank you

<file>

path: /.env.sample

url: https://github.com/YamenMerhi/Demo-AddingLSP1UniversalReceiverDelegates/blob/main/.env.sample

```
PRIVATE_KEY=
UP_ADDR=
```

</file>

<file>

path: /README.md

url: https://github.com/YamenMerhi/Demo-AddingLSP1UniversalReceiverDelegates/blob/main/README.md

Demo-AddingLSP1UniversalReceiverDelegates</file>

<file>

path: /contracts/MyToken.sol

url: https://github.com/YamenMerhi/Demo-AddingLSP1UniversalReceiverDelegates/blob/main/contracts/MyToken.sol

```solidity
// SPDX-License-Identifier: Apache-2.0
pragma solidity ^0.8.4;

import {LSP7DigitalAsset} from "@lukso/lsp-smart-contracts/contracts/LSP7DigitalAsset/LSP7DigitalAsset.sol";

contract MyToken is LSP7DigitalAsset {
    constructor() LSP7DigitalAsset("Token", "TKN", msg.sender, false) {}

    function mint(address recipient) public {
        _mint(recipient, 10, true, "");
    }
}
```

</file>

<file>

path: /contracts/UniversalReceiverDelegate.sol

url: https://github.com/YamenMerhi/Demo-AddingLSP1UniversalReceiverDelegates/blob/main/contracts/UniversalReceiverDelegate.sol

```
// SPDX-License-Identifier: Apache-2.0
pragma solidity ^0.8.4;

// 0x20804611b3e2ea21c480dc465142210acf4a2485947541770ec1fb87dee4a55c

import {ILSP1UniversalReceiverDelegate} from "@lukso/lsp-smart-
contracts/contracts/LSP1UniversalReceiver/ILSP1UniversalReceiverDelegate.sol";

contract UniversalReceiverDelegate is ILSP1UniversalReceiverDelegate {
    mapping(address => bool) public isWhitelisted;

    function changeWhitelisting(address token, bool status) public {
        isWhitelisted[token] = status;
    }

    function universalReceiverDelegate(
        address sender,
        uint256 /*value*/,
        bytes32 typeId,
        bytes memory /*data*/
    ) public view returns (bytes memory) {
        if (
            typeId ==
            0x20804611b3e2ea21c480dc465142210acf4a2485947541770ec1fb87dee4a55c
        ) {
            require(isWhitelisted[sender]);
        }

        return "";
    }

    function supportsInterface(bytes4 interfaceId) public pure returns (bool) {
        return interfaceId == type(ILSP1UniversalReceiverDelegate).interfaceId;
    }
}
```

</file>

<file>

path: /hardhat.config.ts

url: https://github.com/YamenMerhi/Demo-AddingLSP1UniversalReceiverDelegates/blob/main/hardhat.config.ts

```
import { HardhatUserConfig } from "hardhat/config";
import "@nomicfoundation/hardhat-toolbox";

const config: HardhatUserConfig = {
  solidity: "0.8.17",
};

export default config;
```

</file>

<file>

path: /package.json

url: https://github.com/YamenMerhi/Demo-AddingLSP1UniversalReceiverDelegates/blob/main/package.json

```
{
  "dependencies": {
    "@lukso/lsp-smart-contracts": "^0.12.0-rc.0",
    "dotenv": "^16.3.1",
    "hardhat": "^2.18.3",
    "prettier": "^3.0.3"
  },
  "devDependencies": {
    "@nomicfoundation/hardhat-toolbox": "^3.0.0"
  }
}
```

</file>

<file>

path: /scripts/AddingAUniversalReceiverDelegate.ts

url: https://github.com/YamenMerhi/Demo-AddingLSP1UniversalReceiverDelegates/blob/main/scripts/AddingAUniversalReceiverDelegate.ts

```
import { ethers } from 'hardhat';
import * as dotenv from 'dotenv';
import { abi as UP_ABI } from '@lukso/lsp-smart-contracts/artifacts/UniversalProfile.json';
import { ERC725YDataKeys, LSP1_TYPE_IDS } from '@lukso/lsp-smart-contracts';

// load env vars
dotenv.config();

// Update those values in the .env file
const { UP_ADDR, PRIVATE_KEY } = process.env;

async function main() {

  // setup provider
  const provider = new ethers.JsonRpcProvider('https://rpc.testnet.lukso.gateway.fm');
  // setup signer (the browser extension controller)
  const signer = new ethers.Wallet(PRIVATE_KEY as string, provider);

  let UP = new ethers.Contract(UP_ADDR as string, UP_ABI, provider);

  console.log('     EOA: ', signer.address);
  console.log('     Universal Profile: ', await UP.getAddress());

  const typeIdLSP7Recipient = LSP1_TYPE_IDS.LSP7Tokens_RecipientNotification;
  const constructedDataKey = ERC725YDataKeys.LSP1.LSP1UniversalReceiverDelegatePrefix +
typeIdLSP7Recipient.substring(2, 42);

  // The address of the deployed URD
  const UniversalReceiverDelegateAddress = "0x..."

  const tx = await UP.connect(signer).setData(constructedDataKey, UniversalReceiverDelegateAddress);
  await tx.wait();

}

main()
  .catch((error) => {
    console.error(error);
    process.exit(1);
  });
```

</file>

<file>

path: /scripts/CallingFunctionOnDeployedContract.ts

url: https://github.com/YamenMerhi/Demo-AddingLSP1UniversalReceiverDelegates/blob/main/scripts/CallingFunctionOnDeployedContract.ts

```
import { ethers } from 'hardhat';
import * as dotenv from 'dotenv';
import { abi as UP_ABI } from '@lukso/lsp-smart-contracts/artifacts/UniversalProfile.json';
import { MyToken__factory, UniversalReceiverDelegate__factory } from '../typechain-types';

// load env vars
dotenv.config();

// Update those values in the .env file
const { UP_ADDR, PRIVATE_KEY } = process.env;

async function main() {

    // setup provider
    const provider = new ethers.JsonRpcProvider('https://rpc.testnet.lukso.network');
    // setup signer (the browser extension controller)
    const signer = new ethers.Wallet(PRIVATE_KEY as string, provider);

    const deployedContractAddress = "0x...";

    // let abi = ["function mint(address)"];
    // let myToken = new ethers.Contract(deployedContractAddress, abi, provider);


    // const recipientAddress = "0x.."
    // const tx = await myToken.connect(signer).mint(recipientAddress, { gasLimit: 400_000 });
    // await tx.wait();

    // const abi = ["function changeWhitelisting(address token, bool status)"]
    // let myUniversalReceiverDelegate = new ethers.Contract(deployedContractAddress, abi, provider);

    // const addressToWhitelist = "0x..."
    // const tx = await myUniversalReceiverDelegate.connect(signer).changeWhitelisting(addressToWhitelist,
true);
    // await tx.wait();




}



main()
    .catch((error) => {
        console.error(error);
        process.exit(1);
    });
```

</file>

<file>

path: /scripts/CallingUniversalReceiver.ts

url: https://github.com/YamenMerhi/Demo-AddingLSP1UniversalReceiverDelegates/blob/main/scripts/CallingUniversalReceiver.ts

```
import { ethers } from 'hardhat';
import * as dotenv from 'dotenv';
import { abi as UP_ABI } from '@lukso/lsp-smart-contracts/artifacts/UniversalProfile.json';

// load env vars
dotenv.config();

// Update those values in the .env file
const { UP_ADDR, PRIVATE_KEY } = process.env;

async function main() {

    // setup provider
    const provider = new ethers.JsonRpcProvider('https://rpc.testnet.lukso.gateway.fm');
    // setup signer (the browser extension controller)
    const signer = new ethers.Wallet(PRIVATE_KEY as string, provider);

    let UP = new ethers.Contract(UP_ADDR as string, UP_ABI, provider);

    console.log('     EOA: ', signer.address);
    console.log('     Universal Profile: ', await UP.getAddress());

    const typeId = ethers.id("Royalty");
    const data = "0xaabbccdd";
    const valueToSend = 10;

    const tx = await UP.connect(signer).universalReceiver(typeId, data, { value: valueToSend });
    await tx.wait();
}

main()
    .catch((error) => {
        console.error(error);
        process.exit(1);
    });
```

</file>

<file>

path: /scripts/DeployingContract.ts

url: https://github.com/YamenMerhi/Demo-AddingLSP1UniversalReceiverDelegates/blob/main/scripts/DeployingContract.ts

```
import { ethers } from 'hardhat';
import * as dotenv from 'dotenv';
import { abi as UP_ABI } from '@lukso/lsp-smart-contracts/artifacts/UniversalProfile.json';
import { MyToken__factory, UniversalReceiverDelegate__factory } from '../typechain-types';

// load env vars
dotenv.config();

// Update those values in the .env file
const { UP_ADDR, PRIVATE_KEY } = process.env;

async function main() {

  // setup provider
  const provider = new ethers.JsonRpcProvider('https://rpc.testnet.lukso.network');
  // setup signer (the browser extension controller)
  const signer = new ethers.Wallet(PRIVATE_KEY as string, provider);


  const myToken = await new MyToken__factory(signer).deploy({ gasLimit: 20_000_000 });
  console.log(myToken.target);

  const urd = await new UniversalReceiverDelegate__factory(signer).deploy({ gasLimit: 20_000_000 });
  console.log((urd.target));
}


main()
  .catch((error) => {
    console.error(error);
    process.exit(1);
  });
```

</file>

<file>

path: /scripts/ListeningUniversalReceiver.ts

url: https://github.com/YamenMerhi/Demo-AddingLSP1UniversalReceiverDelegates/blob/main/scripts/ListeningUniversalReceiver.ts

```
import { ethers } from 'hardhat';
import * as dotenv from 'dotenv';
import { abi as UP_ABI } from '@lukso/lsp-smart-contracts/artifacts/UniversalProfile.json';

// load env vars
dotenv.config();

// Update those values in the .env file
const { UP_ADDR, PRIVATE_KEY } = process.env;

async function main() {

    // setup provider
    const provider = new ethers.JsonRpcProvider('https://rpc.testnet.lukso.gateway.fm');
    // setup signer (the browser extension controller)
    const signer = new ethers.Wallet(PRIVATE_KEY as string, provider);

    let UP = new ethers.Contract(UP_ADDR as string, UP_ABI, provider);

    console.log('     EOA: ', signer.address);
    console.log('     Universal Profile: ', await UP.getAddress());


    UP.on("UniversalReceiver", (from, value, typeId, receivedData, returnedData) => {
        console.log(&nbsp;The UniversalReceiver was called by ${from} passing ${value} value, to notify about ${typeId} with  these ${receivedData} data&nbsp;);
    })


}

main()
    .catch((error) => {
        console.error(error);
        process.exit(1);
    });
```

</file>

<file>

path: /scripts/mainSetup.ts

url: https://github.com/YamenMerhi/Demo-AddingLSP1UniversalReceiverDelegates/blob/main/scripts/mainSetup.ts

```
import { ethers } from 'hardhat';
import * as dotenv from 'dotenv';
import { abi as UP_ABI } from '@lukso/lsp-smart-contracts/artifacts/UniversalProfile.json';

// load env vars
dotenv.config();

// Update those values in the .env file
const { UP_ADDR, PRIVATE_KEY } = process.env;

async function main() {

  // setup provider
  const provider = new ethers.JsonRpcProvider('https://rpc.testnet.lukso.gateway.fm');
  // setup signer (the browser extension controller)
  const signer = new ethers.Wallet(PRIVATE_KEY as string, provider);

  let UP = new ethers.Contract(UP_ADDR as string, UP_ABI, provider);

  console.log('     EOA: ', signer.address);
  console.log('     Universal Profile: ', await UP.getAddress());

  const typeId = ethers.id("Royalty");
  const data = "0xaabbccdd"
  const tx = await UP.connect(signer).universalReceiver(typeId, data, { value: 10 });
  await tx.wait();
}

main()
  .catch((error) => {
    console.error(error);
    process.exit(1);
  });
```

</file>

<file>

path: /tsconfig.json

url: https://github.com/YamenMerhi/Demo-AddingLSP1UniversalReceiverDelegates/blob/main/tsconfig.json

```
{
  "compilerOptions": {
    "target": "es2020",
    "module": "commonjs",
    "esModuleInterop": true,
    "forceConsistentCasingInFileNames": true,
    "strict": true,
    "skipLibCheck": true,
    "resolveJsonModule": true
  }
}
```

</file>

</repository>

<repository>

createUP

overview

repository: https://github.com/biancabuzea200/createUP/
userName: biancabuzea200
repository: createUP
branch: main
date: 2023-11-27T10:16:38+01:00 (1701076598)
description:
hi there it's Bianca from the Deval team
in this video we're going to create a
universal profile programmatically and
after we have our Universal profile
we're going to read Universal profile
data first the contents of uh this video
so in the first part we're going to
create an account programmatically in
part number two we're going to deploy a
universal profile programmatically and
in part number three after we have our
Universal profile we're going to read
the data from our
profile here you can have by scanning
this code uh you can access the GitHub
repository and I strongly encourage you
to actually follow uh with the GitHub
repository in front because you will
find very useful comments in the code
and also you will find the different
libraries and dependencies in the rme
file one note here uh when you are
deployed loing a universal profile uh
you have two options so the first option
is you deploy the universal profile
using the universal profile browser
extension and when you're doing this way
your transaction gas fees will be
subsidized by the Lux of relay service
the second version that we're going to
cover today is when you deploy your
Universal profile
programmatically in this case you'll
have to cover the gas fees uh for your
transaction by yourself however there
are still cases when you will need to
create a universal profile

programmatically so this is uh very
useful to
know now brief recap on what is a
universal profile a universal profile is
a smart contract based account that
represents your profile on chain so to
this profile you can attach different
metadata such as links descriptions
profile pictures cover photos uh tags
Etc Now demo time show me the code I
would like to invite you to access uh
this link and follow along with the
video to create our Universal profile
first we'll need an account to do that
we're going to use the web FS Library so
first we're going to import it now we're
going to create uh an instance we're
going to pass uh in here here
the LX RPC endpoint for the test net
next we're going to create our account
uh we're going to use the create
method and we are going to display to
the console the private key and the
public key next we want to display our
balance we're going to do so by creating
an Asing function we can call it uh get
balance of course uh at the beginning
our balance will be zero but we are
going to fund our key using the Luxo
testet faucet because we're going to
need uh some testet leaks to deploy our
Universal profile last but not least uh
we are going
to call our get balance function and
let's see this in
action
as you can see here it has generated our
private key our public key and we have
our balance of zero so now it's time
that we go ahead and we fund our uh
public I'm right now on the Luxo docs
I'm going to go to networks I'm going to
go to test net Network parameters I'm
looking for faucet and and here you can
find the testet authenticated
faucet in order to get funds uh you need
to make a tweet and paste the URL from
the Tweet you have the whole explanation
uh here in the bottom then you can press
uh give me l t testet loo that's it next
we are going to deploy our Universal
profile okay so now in a separate file

I'll call this uh deploy up
I'm going to work on creating and
deploying uh my Universal profile
similar to before I'm going to import a
web free library and create a web free
object where I'm passing the locko
testnet RPC endpoint however if you
prefer using eers instead of uh web free
you will find under the eers folder
corresponding to uh the GitHub reple for
these tutorials all the files using
eers after this I'm going to use uh LSP
uh Factory I'm going to import uh LSP uh
Factory LSP Factory it's a library that
does the heavy lifting for deploying
Universal profiles and other digital
assets such as LSP 7 or LSP
8 and then I'm going to uh
import my uh I'm going to import uhv and
I'm doing this because I I want to save
my private key uh Secret in
TMV so what I'm going to do
now I am uh going to uh load my private
key from the EMV file and store in a
private in a constant called private
key when it comes to the UA I'm going to
define the variable my UA and and here
I'm going to use the private key to
account uh function this function create
an account object uh from a private key
so this is exactly uh What uh we are
looking
for next is time to to use our LSP uh
Factory Library so we're going to create
a new instance of LSP factory and we're
going to pass the testet RPC uh endpoint
uh and uh the EA's uh private key uh
this will deploy basically the universal
profile smart contract so we have the
RPC endpoint and we have the deploy key
which is uh the private key and the
chain ID uh we have uh 4201 for The Lox
test
net in The Next Step uh I'm going to
Define an async function called create
uh Universal uh profile let's say uh
this uh will allow uh need to return the
deployed contracts so I'm going to use
deploy function from the LSP Factory and
I'm going to pass uh two uh parameters
the first parameter is the controller
key this is essentially the EA that will

be uh controlling the universal
profile and the second
parameter is the LSP free profile so the
LSP uh free profile it contains the
profile
metadata and basically it allows to
define specific data key uh to Define uh
properties for the universal profile so
you can see here that I have a name I
have a description I have tags I have
links but you can add
other uh data keys for example they can
be one for a profile
image now that uh we have discovered uh
what I'm going to do is I want to store
the address of the universal profile in
a variable called uh my up address and
then I will uh output this to the
console so now all I have to do uh is to
return um the deployed
contracts and then of course uh to call
uh
the create Universal profile
function and this should be it now all I
have to do is uh to run our
file it will take a while for the smart
contracts uh to be uh
deployed but um this should work
and here we have the address of my uh
Universal
profile okay so now for the last part we
want to read profile data from our
freshly created Universal profile to do
that I created a new file uh you can
find it under the name of read profile
data.js similarly to
before we're going to import a web free
in order to connect to an RPC endpoint
we're also going to import this time RC
uh 725 this allows us to easily uh
interact with u ERC 725
schemas and also we're going to
import our LSP free profile
schema now in order to uh read our
profile data we're going to use uh the
fetch profile
function
so as you can see here we are attempting
to create a new ERC 725 uh object and to
do that uh we need to pass in four uh
parameters youc LSP free profile schema
the address which is the universal

profile
address the
provider and we need also a config which
is the the ipfs uh
Gateway so so uh we attempt to create a
new ERC 725 object and we attempt to
fetch data from this object once we
created the object and if we get an
error most likely means that this is not
ANC 725 uh contract we're going to
display this uh error
message so now all we need to do uh is
to basically Define uh this uh such that
uh we have them as parameters
so first we're going
to have our sample profile data this is
the universal profile address that we
have from earlier the RPC endpoint uh we
are going to pass the loxo RPC endpoint
and the ipfs
Gateway that we're going to uh use and
then we also need
to Define uh the
parameters for the ERC 725 instance and
these are the provider and the
config
finally we want to call the fetch uh
data uh fetch profile uh
function so uh by doing that we want to
display uh to the console uh the profile
data as a as a Json so so now we're
going to do that we're going to save our
file and
run uh profile read profile
data and here we have our schema that we
just defined
earlier beautiful

<file>

path: /.env.example

url: https://github.com/biancabuzea200/createUP/blob/main/.env.example

```
MY_PRIVATE_KEY=
MY_PUPLIC_KEY=
MY_UP_ADDRESS=
```

</file>

<file>

path: /README.md

url: https://github.com/biancabuzea200/createUP/blob/main/README.md

<!-- Improved compatibility of back to top link: See: https://github.com/othneildrew/Best-README-Template/pull/73 -->

<a name="readme-top"></a>

<!--
*** Thanks for checking out the Best-README-Template. If you have a suggestion
*** that would make this better, please fork the repo and create a pull request
*** or simply open an issue with the tag "enhancement".
*** Don't forget to give the project a star!
*** Thanks again! Now go create something AMAZING! :D
-->

<!-- PROJECT SHIELDS -->
<!--
*** I'm using markdown "reference style" links for readability.
*** Reference links are enclosed in brackets [ ] instead of parentheses ( ).
*** See the bottom of this document for the declaration of the reference variables
*** for contributors-url, forks-url, etc. This is an optional, concise syntax you may use.
*** https://www.markdownguide.org/basic-syntax/#reference-style-links
-->

<!-- ABOUT THE PROJECT -->

Create a Universal Profile Programmatically and Read Profile Data

About The Project

This repo aims to get you up to speed with creating a Universal Profile on the LUKSO blockchain programmatically and reading data from it.

<!-- TABLE OF CONTENTS -->
<details>
<summary>Table of Contents</summary>
<ol>
<li>
<a href="#about-the-project">About The Project</a>
<ul>
<li><a href="#built-with">Built With</a></li>
</ul>
</li>
<li>
<a href="#getting-started">Getting Started</a>
<ul>
<li><a href="#prerequisites">Prerequisites</a></li>
<li><a href="#installation">Installation</a></li>

</ul>
</li>
<li><a href="#usage">Usage</a></li>
<li><a href="#contributing">Contributing</a></li>
<li><a href="#contact">Contact</a></li>
</ol>
</details>

Built With

- Node.js (https://nodejs.org/en)

<p align="right">(<a href="#readme-top">back to top</a>)</p>

<!-- GETTING STARTED -->

Getting Started

These examples are very straightforward and all can be run simply with node. We do need to do some minimal setup first though.

Prerequisites

This is an example of how to list things you need to use the software and how to install them.

- nvm

  We will be changing our node version for some examples so we will be using nvm. If you do not have it set up take a look here (https://github.com/nvm-sh/nvm#installing-and-updating). This will also ensure you have node.js and npm available.

Installation

1. Clone the repo

   ```
   git clone https://github.com/biancabuzea200/createUP
   ```

2. Install NPM packages

   ```
   npm install
   ```

3. If you have an UP address or Wallet for LUKSO testnet already you can enter them in the .env

   ```
   MY_PRIVATE_KEY=
   MY_PUPLIC_KEY=
   MY_UP_ADDRESS=
   ```

4. Install the libraries

   ```
   npm install @lukso/lsp-factory.js
   npm install web3
   ```

<p align="right">(<a href="#readme-top">back to top</a>)</p>

<!-- USAGE EXAMPLES -->

Usage

This repo goes over some simple examples that you can build on.

For more examples, please refer to the Documentation (https://docs.lukso.tech/)

Basic Usage

1. Navigate to the directory you are interested in, both the ethers and web3.js examples are equivalent.

2. Follow any additional instructions left as comments in each file. The normal order to follow is generateKeys.js -> deployUP.js -> readProfileData.js since this order generates all values needed for subsequent examples.

3. Each example can be run simply with

   ```
   node [filepath]/filename.js
   ```

4. Make sure to add any newly generated variables to the .env before proceeding to the next example.

   ```
   MY_PRIVATE_KEY=
   MY_PUPLIC_KEY=
   MY_UP_ADDRESS=
   ```

<p align="right">(<a href="#readme-top">back to top</a>)</p>

<!-- CONTRIBUTING -->

Contributing

Contributions are what make the open source community such an amazing place to learn, inspire, and create. Any contributions you make are greatly appreciated.

If you have a suggestion that would make this better, please fork the repo and create a pull request. You can also simply open an issue with the tag "enhancement".
Don't forget to give the project a star! Thanks again!

1. Fork the Project
2. Create your Feature Branch (git checkout -b feature/AmazingFeature)
3. Commit your Changes (git commit -m 'Add some AmazingFeature')
4. Push to the Branch (git push origin feature/AmazingFeature)
5. Open a Pull Request

<p align="right">(<a href="#readme-top">back to top</a>)</p>

<!-- CONTACT -->

Contact

Bianca Buzea - @buzea200 (https://twitter.com/buzea200)

Project Link: https://github.com/biancabuzea200/createUP (https://github.com/biancabuzea200/createUP)

<p align="right">(<a href="#readme-top">back to top</a>)</p>

<!-- MARKDOWN LINKS & IMAGES -->
<!-- https://www.markdownguide.org/basic-syntax/#reference-style-links -->
</file>

<file>

path: /ethers/deployUP.js

url: https://github.com/biancabuzea200/createUP/blob/main/ethers/deployUP.js

```
// You might need to use node v16.0.0 (nvm use 16.0.0) when running this
// This is because node made a change and a flag is required in the fetch command, and the ipfs-http-client hasn't made an update
// So only older versions of node will work.
// You can do this with &nbsp;nvm use 16.0.0&nbsp;

import { LSPFactory } from '@lukso/lsp-factory.js';
import dotenv from 'dotenv/config';
import { ethers } from "ethers";


const PRIVATE_KEY = process.env.MY_PRIVATE_KEY;
const myEOA = new ethers.Wallet(PRIVATE_KEY);

const lspFactory = new LSPFactory('https://rpc.testnet.lukso.network/', {
    deployKey: PRIVATE_KEY,
    chainId: 4201, // LUKSO Testnet
});

async function createUniversalProfile() {
    const deployedContracts = await lspFactory.UniversalProfile.deploy({
        controllerAddresses: [myEOA.address],
        lsp3Profile: {
            name: 'My Universal Profile',
            description: 'My Cool Universal Profile',
            tags: ['Public Profile'],
            links: [
                {
                    title: 'My Website',
                    url: 'https://my-website.com/',
                },
            ],
        },
    });

    const myUPAddress = deployedContracts.LSP0ERC725Account.address;
    console.log('my Universal Profile address: ', myUPAddress);

    // Now we can add this UP address to our .env file
    return deployedContracts;
}
createUniversalProfile();
```

</file>

<file>

path: /ethers/generateKeys.js

url: https://github.com/biancabuzea200/createUP/blob/main/ethers/generateKeys.js

```
import { ethers } from "ethers";

const provider = new ethers.JsonRpcProvider("https://rpc.testnet.lukso.network/");

let account = ethers.Wallet.createRandom(provider);
console.log("My private key is: " + account.privateKey);
console.log("My address is: " + account.address);

// Now we can add this key and address to our .env file

async function getBalance() {
    let balance = await provider.getBalance(account.address);
    console.log("My balance: " + ethers.formatEther(balance) + "ether LYXt");
}

getBalance();
```

</file>

<file>

path: /package.json

url: https://github.com/biancabuzea200/createUP/blob/main/package.json

```
{
  "name": "createup",
  "version": "1.0.0",
  "description": "",
  "type": "module",
  "main": "createUP.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC",
  "dependencies": {
    "@lukso/lsp-factory.js": "^3.1.1",
    "dotenv": "^16.3.1",
    "isomorphic-fetch": "^3.0.0",
    "web3": "^1.10.2"
  }
}
```

</file>

<file>

path: /src/deployUP.js

url: https://github.com/biancabuzea200/createUP/blob/main/src/deployUP.js

```
// You might need to use node v16.0.0 (nvm use 16.0.0) when running this
// This is because node made a change and a flag is required in the fetch command, and the ipfs-http-client hasn't made an update
// So only older versions of node will work.
// You can do this with &nbsp;nvm use 16.0.0&nbsp;


import Web3 from 'web3';
const web3 = new Web3("https://rpc.testnet.lukso.network");
import { LSPFactory } from '@lukso/lsp-factory.js';
import dotenv from 'dotenv/config';

const PRIVATE_KEY = process.env.MY_PRIVATE_KEY;
const myEOA = web3.eth.accounts.privateKeyToAccount(PRIVATE_KEY);

const lspFactory = new LSPFactory('https://rpc.testnet.lukso.network/', {
  deployKey: PRIVATE_KEY,
  chainId: 4201, // LUKSO Testnet
});

async function createUniversalProfile() {
  const deployedContracts = await lspFactory.UniversalProfile.deploy({
    controllerAddresses: [myEOA.address],
    lsp3Profile: {
      name: 'My Universal Profile',
      description: 'My Cool Universal Profile',
      tags: ['Public Profile'],
      links: [
        {
          title: 'My Website',
          url: 'https://my-website.com/',
        },
      ],
    },
  });

  const myUPAddress = deployedContracts.LSP0ERC725Account.address;
  console.log('my Universal Profile address: ', myUPAddress);

  // Now we can add this UP address to our .env file

  return deployedContracts;
}
createUniversalProfile();
```

</file>

<file>

path: /src/generateKeys.js

url: https://github.com/biancabuzea200/createUP/blob/main/src/generateKeys.js

```
import Web3 from "web3";
const web3 = new Web3("https://rpc.testnet.lukso.network/");

let account = web3.eth.accounts.create();
console.log("My private key is: " + account.privateKey);
console.log("My address is: " + account.address);

// Now we can add this key and address to our .env file

async function getBalance() {
    let balance = await web3.eth.getBalance(account.address);
    console.log("My balance: " + web3.utils.fromWei(balance, "ether") + " LYXt");
}

getBalance();
```

</file>

<file>

path: /src/readProfileData.js

url: https://github.com/biancabuzea200/createUP/blob/main/src/readProfileData.js

```
// Import and Network Setup

import Web3 from 'web3';
import { ERC725 } from '@erc725/erc725.js';
import lsp3ProfileSchema from '@erc725/erc725.js/schemas/LSP3ProfileMetadata.json' assert { type: 'json'
};

// Our static variables
//const SAMPLE_PROFILE_ADDRESS2 = '0x6979474Ecb890a8EFE37daB2b9b66b32127237f7';
const SAMPLE_PROFILE_ADDRESS = "0xB031363560403179Aac100d51864e27fFF4D7807";
const RPC_ENDPOINT = 'https://rpc.testnet.lukso.network';
const IPFS_GATEWAY = 'https://api.universalprofile.cloud/ipfs';

// Parameters for ERC725 Instance
const provider = new Web3.providers.HttpProvider(RPC_ENDPOINT);
const config = { ipfsGateway: IPFS_GATEWAY };

async function fetchProfile(address) {
    try {
        const profile = new ERC725(lsp3ProfileSchema, address, provider, config);
        return await profile.fetchData();
    } catch (error) {
        console.log(error)
        return console.log('This is not an ERC725 Contract');
    }
}

fetchProfile(SAMPLE_PROFILE_ADDRESS).then((profileData) =>
    console.log(JSON.stringify(profileData, undefined, 2)),
);
```

</file>

</repository>

<repository>

relayer-example

overview

repository: https://github.com/magalimorin18/relayer-example/
userName: magalimorin18
repository: relayer-example
branch: main
date: 2023-11-27T10:16:38+01:00 (1701076598)
description:
welcome to all of you to the looko
buildup hackathon number two I'm very
happy today to be presenting a workshop
on the service relayer for a little
introduction on who I am so my name is
magali Morin I am a software engineer at
luo and the projects I've been working
on since uh I've joined luo are the
service relayer on which I will be
talking today and also the indexer that
you can find at search. Universal
profile. Cloud which indexes the
universal profiles on the blockchain and
also the tokens and
nft so the objective of this Workshop is
that you get an idea of how to use a
service relayer how to send execute
relay call transaction how to deploy
Universal profiles and finally I will do
a demo with uh the GitHub repository
that I have prepared for this
Workshop all along the the workshop you
can always refer at docs. loco. Tech
where you will find very detailed
information on everything that I will be
talking about so don't hesitate to visit
d.o. tech for any question that you
have so today a big barrier of Entry to
the blockchain is the fact that users
needs to buy token of a network on an
exchange platform do a kyc etc etc along
and a very a discouraging process before
they are able to send transaction to the
blockchain the idea of the service
relayer is to replace this two time step
for the user with just a onetime step
for the user the user just needs to

subscribe to a monthly subscription to
the service relayer and then the service
relayer sends and pay for the
transaction fee of the users every time
a user wants to make an action on the
blockchain
so basically the service relayer removes
a barrier of Entry of the user to use
the
blockchain let's get into more details
on how this flow Works a user signs a
transaction with a key this key doesn't
need any fund on it it just needs to
have permission to execute a transaction
on the universal
profile this transaction is sent to the
service relayer the service relayer then
sends this transaction and sends it to
the
blockchain the user needs to send the
transaction to the service relayer this
transaction is an object with different
parameters the first parameter is the
address this address is the universal
profile address of the user on which the
transaction will be
executed then we have a transaction
object the transaction object consist of
the ABI the ABI of the transaction to be
executed a signature this is the signed
transaction with the key of the user
which doesn't have any
fund the nons the nons is the nons of
the signing key of the
transaction and finally the validity
time stamp the validity time stamp is
the duration for which the transaction
will be considered valid in this case
the validity time stamps is set to zero
this means that the transaction is valid
for an indefinite period of
time then the service relayer will
encode the execute relay call
transaction with the parameters given by
the
user and finally it will signs and send
the transaction to the
blockchain this transaction will be
executed on the blockchain and the
service relayer will return to the user
the transaction
hash now let's get into more detail on

how to deploy a universal profile
a universal profile deployment required
to deploy several smart contracts and
these smart contracts need to interact
with each
other in fact a universal profile is a
combination of different contracts on
one side an ERC 725 account contract and
also an LS sp6 key manager
contract this LS sp6 key manager
contract is linked to the ERC 725
account
contract
then we can also set additional settings
like a universal receiver delegate or
lsb3 metadata on the ERC 725 account
contract in order to orchestrate the
deployment of these very different
contracts and to set additional data on
top of this deployment we use the LSP 23
standard now let's get into the demo
code so first I would like to present
some tools that will be useful to you
when you will be be tacking with the
repository first the luo faet this faet
allows you to get some fund on your keys
from the Luxo test net the second tool
that will be very useful to you is ERC
725 inspect it allows you very easily to
know what is inside your Universal
profile now let's get into the repo and
into the demo
code let's clone the Repository
and do the installation
setup once you have followed the
installation process you should have an
environment file with a private key
which have funds on it so this key
relayer private key is the key that was
in Pink So this key is the key that will
be used to send execute relay code to
the blockchain you can use the faet in
order to fund this
key
let's run the project by running yarn
Dev this
will generate the server
and here we see that the server was
successfully started at Locos
3000 so now we can um use the script in
order to execute a relay call or deploy
a universal profile first let's start by

executing a relay
call
in the script execute relay a call we
see that we need
to give two parameters in input the
first parameter is the universal profile
address on which we want to execute this
relay and then it's the user private key
this private key is the key that was in
gold in the slide before this key
doesn't need any fund it only needs to
have permission on the universal profile
in order to execute the ABI
payload in this script as we don't fill
in the input parameters any API a
default set data AI will be executed in
our on our Universal
profile you can find this ABI uh in the
jate body execute. TS
file it's this one so this is the AI
that will be executed on the universal
Pro profile it's a set data transaction
and it will set this default lsb3
profile
data as we want to execute a set data
transaction on the universal profile we
need to make sure that the user private
key has set data permission on the
universal
profile so now let's execute the script
and see if the lsb3 profile that are
correctly set set on our Universal
profile to execute the
script we will
run yarn run
execute here we see that the execute
relay call request is being sent to the
blockchain by the service
relayer this execute relay call request
is signed by this wallet
address
and here we have that the transaction
was successfully sent and we see that
our transaction was
successful then we can see on ERC 725
inspect if the LSP profile metadata were
correctly set on our Universal
profile so let's go on ERC 725
inspect and here paste the contract
address here is the universal profile
address address on which we executed the
relay

call and here we can see everything that
is on this Universal
profile we see in lsb3 profile that here
we have the lsb3 profile metadata that
we wanted to set on it so the execute
relay C transaction was successfully
executed
now let's deploy a universal profile
with the script so the input parameters
are a list of controllers addresses with
permissions and lsb3 profile metadata to
set on the deployed Universal
profile so let's get into more detail on
what does these list of controller means
so once the universal profile will be
deployed we want to set controller
addresses on it with very specific
permission in order to to encode and
encode this permission we can also use
ERC 725 inspect I will do this here to
show you how to do it on your c725
inspect there is this key manager
permission page where you can paste the
permissions and then it will decode for
you the
permission that it corresponds to here
we can see that this corresponds to a
set data
permission so once the universal profile
will be deployed it will have a
controller with this address that will
have the s data permission on
it it will also have this address which
will have other permissions on it and
also this address with these permissions
and finally it will be deployed with
this lsb3 profile
metadata so let's run the script deploy
up in order to deploy a universal
profile and check if it was deployed
with a COR correct
inputs for this we run the script yarn
run deploy
up and let's see so now the relayer
received a request to deploy a universal
profile and here we see that the
universal profile was successfully
deployed and here is the transaction
hash on the Block Explorer it was
successful and we can inspect the
universal
profile on RC 725 inspect so if we go to

the LSP 6 address permission here we see
that this Universal profile was deployed
with three addresses as controllers this
is what we decided and let's check that
each of these addresses have the correct
permissions so for this I will use the
data fetcher here I will paste the
universal profile
address and here I want to inspect the
address permission for a specific
address here I will paste the address of
the
controller omitting the two first
bytes and I want to get the data and
here we see that the permissions that
this address have on the universal
profile is only set data and this is
what we were
expecting you can find more information
on how each parameters is generated in
the read me please go through there in
detail as it will give you better
Insight on how each endpoint works and
how a transaction execute relay call or
a deploy Universal profile is executed
through the relayer finally I thought
about some ideas for the hackathon that
could be nice to uh discuss about the
nons management system so in this
implementation of the rayer there is
just one key that that signs and send
the transaction to the blockchain
however if there was just one key the
relayer will be a very slow system so
ideally we want several keys that are
signing and sending transaction uh to
the blockchain in parallel but for this
we need to have a good non management of
each
Keys then another idea was the batch of
transaction in order to reduce the cost
of gas fee the relayer could send users
transaction in batch this has the
advantages of reducing the gas fee but
also on the other side disadvantages so
for example uh we could not send um the
transaction hash back to the user for
each transaction that they want to make
and also if one of the transaction would
happen to fail then the whole batch of
transaction would
fail and finally another idea would be

link to the quota
management how does user would uh pay or
retribute to the service relater for
paying their transaction fees uh would
they pay on a monthly basis or would um
it be another system that pay for the
user I think uh in this subject uh we
can be very creative on how to manage
quota so this is something definitely
worth uh thinking about so thank you
very much for having followed this work
work shop I it was really a pleasure for
me to present my work on the relayer
don't hesitate if you have any question
to come up to me on Discord I will do my
best to answer all your
questions I wish you the best of luck
for the hackathon and I'm very very
excited to see all the project that you
will come up with and hopefully in the
future we can collaborate together and
create and build very cool stuff on the
looks of
blockchain

<file>

path: /.env.example

url: https://github.com/magalimorin18/relayer-example/blob/main/.env.example

```
HOST=0.0.0.0
PORT=3000
RPC_ENDPOINT=https://rpc.testnet.lukso.network
CHAIN_ID=4201
RELAYER_BASE_URL=http://0.0.0.0:3000

RELAYER_PRIVATE_KEY=
```

</file>

<file>

path: /README.md

url: https://github.com/magalimorin18/relayer-example/blob/main/README.md

Relayer Example

[Slides (https://drive.google.com/file/d/1HY87a4yj06YxAg7FyTPnfXq0PxOkng93/view?usp=sharing)](https://drive.google.com/file/d/1HY87a4yj06YxAg7FyTPnfXq0PxOkng93/view?usp=sharing) of the workshop presentation.

Description

A Service Relayer allows to send transactions on behalf of users to cover their gas costs.

Installation

To begin sending transaction with the Relayer create a .env file to hold configuration parameters:

```
cp .env.example .env
```

- RELAYER_PRIVATE_KEY : Used by the relayer to send the transaction to the blockchain on behalf of the user. Need to be funded with sufficient LYX balance to execute transactions on the blockchain.
  To fund this key use the faucet (https://faucet.testnet.lukso.network)

Install dependencies

```
yarn
```

Run the project

```
yarn dev
```

This command will generate the types of the smart contracts and start the server at default address http://0.0.0.0:3000

Scripts

Execute relay call

Parameters to set:

```
{
  "universalProfileAddress": "0xda09F6F4acC07647C275319A7176A2D96E7cbA5c", //Universal Profile on which you want to execute the transaction.

  "userPrivateKey": "0x04a3042380a01c4600df3cd06a86755f51426493d8a0ba12118f47f64d84471e", //Private key to sign the executeRelayCall transaction. It needs permission on the Universal Profile to execute the given transaction. It doesnt need to have LYX on it.

  "abi": "0x.." // optional. The abi-encoded transaction data (e.g: a function call on the Universal Default value: SET_DATA abi-encoded transaction.
}
```

⚠ Update from the oral prensentation: The parameters are loaded from the json file directly. <file-name.json> needs to be located in the same folder as the execute.ts file.

Execute the script :

```
yarn run execute <file-name.json>
```

Find more information on this script : scripts/execute-relay-call/execute.ts

Deploy a Universal Profile

Parameters to set :

```
{
  "controllers": [
    {
      "address": "0x479a0BD32ba791E92d8bF9BF263eF63780182645",
      "permissions": "0x0000000000000000000000000000000000000000000000000000000000040000"
    }
  ], // A list of controller addresses with permissions to be set on the deployed Universal Profile.

  "lsp3ProfileMetadata": "0x.." //optional. LSP3 Metadata to set on the deployed Universal Profile.
}
```

Encode and decode permissions: [ERC725-Inspect (https://erc725-inspect.lukso.tech/key-manager)](https://erc725-inspect.lukso.tech/key-manager)

Execute the script:

⚠ Update from the oral prensentation: The parameters are loaded from the json file directly. <file-name.json> needs to be located in the same folder as the execute.ts file.

```
yarn run deploy-up <file-name.json>
```

Find more information on this script : scripts/deploy-up/deploy-up.ts

Endpoints

POST /execute

Executes a transaction on behalf of a user using executeRelayCall function of the Key Manager owning the user's Universal Profile.

Request body

```
{
  "address": "0xBB645D97B0c7D101ca0d73131e521fe89B463BFD",
  "transaction": {
    "abi":
"0x7f23690c5ef83ad9559033e6e941db7d7c495acdce616347d28e90c7ce47cbfcfcad3bc50000000000000
0000000000000000000000000000000000000000000000040000000000000000000000000000000000000
0000000000000000000000000596f357c6aa5a21984a83b7eef4cb0720ac1fcf5a45e9d84c653d97b71bbe8
9b7a728c386a697066733a2f2f516d624b43744b4d7573376741524470617744687a32506a4e36616f64346
b69794e436851726d3451437858454b0000000000000000",
    "signature":
"0x43c958b17295867491695 99d7e776f18afc6223c7da21107161477d291d497973b4fc50a724b1b2ab98f3f
8cf1d5cdbbbdf3512e4fbfbdc39732229a15beb14a1b",
    "nonce": 1,
    "validityTimestamps": "0x0000000000000000000000006420f3f0000000000000000000000000065ec82d0"
  }
}
```

- address - The address of the Universal Profile.

- transaction - An object containing the transaction parameters which will be executed with executeRelayCall.
    - abi - The abi-encoded transaction data (*e.g: a function call on the Universal Profile smart contract*) which will be passed as the payload parameter to the executeRelayCall function.
    - signature - The signed message. See section below for how to construct the signature.
    - nonce - The nonce of the user key signing the transaction. Fetched by calling getNonce(address address, uint128 channelId) on the LSP6 KeyManager contract of the Universal Profile.
    - validityTimestamps (optional) - Two concatenated uint128 timestamps which indicate a time duration for which the transaction will be considered valid. If no validityTimestamps parameter is passed the relayer should assume that validityTimestamps is 0 and the transaction will be valid indefinitely until it is executed.

Response

The transaction hash of the executeRelayCall transaction.

```
{
  "transactionHash": "0xBB645D97B0c7D101ca0d73131e521fe89B463BFD"
}
```

Specification to construct the executeRelayCall signature

Construct the message

```
const message = ethers.utils.solidityPack(
  ["uint256", "uint256", "uint256", "uint256", "uint256", "bytes"],
  [
    25, //LSP25 Version
    "4201", //Chain Id
    { _hex: "0x0c", _isBigNumber: true }, //Nonce of the user signing key
    0, //Validity Timestamps
    0, //Amount of native tokens to transfer (in Wei)

0x7f23690ccafecafecafecafecafecafecafecafecafecafecafecafecafecafecafecafe000000000000000000000
000000000000000000000000000000000000000004000000000000000000000000000000000000000000000
00000000000000000004cafecafe00000000000000000000000000000000000000000000000000000000,
//Abi payload
  ]
);
```

Sign the message

Important : The message must be signed by a key belonging to the user that has permissions on the Universal Profile in order to execute the abi payload. This key doesn't need LYX.

EIP191 signer (https://www.npmjs.com/package/@lukso/eip191-signer.js) library is used to generate the signature using the function signDataWithIntendedValidator.

```
const eip191Signer = new EIP191Signer();

const { signature } = eip191Signer.signDataWithIntendedValidator(
  keyManagerAddress, // Key Manager of the Universal Profile
  message, // Constructed message as shown in the section above
  privateKey // User key with permissions on the Universal Profile
);
```

Transaction Gate

This implementation of the relayer can only process one transaction at a time. To prevent nonce reuse errors, a transaction gate is implemented which will block incoming transactions if there is already a transaction pending.

POST /universal-profile

Deploy a Universal Profile with given controller addresses and LSP3 Metadata. Universal Profiles will by default be deployed with a Universal Receiver Delegate.

```
{
  "controllers": [
    {
      "address": "0x479a0BD32ba791E92d8bF9BF263eF63780182645",
      "permissions": "0x0000000000000000000000000000000000000000000000000000000000040000"
    },
    {
      "address": "0xDD06fC1f047103ea047654c57C6f6F754E91CD24",
      "permissions": "0x00000000000000000000000000000000000000000000000000000000003f3f7f"
    }
  ],
  "lsp3ProfileMetadata":
"0x6f357c6ae8b1db2db7ce8a555019ab31c557e9934145c0baa7fdae3f1a131e736d3d4e8e697066733a2f2f516d583543365939614a35646879697533546446704b625879765343434345386a435477514436374743784d705a31"
}
```

- controllers - The addresses and permissions to set on the deployed Universal Profile. In order to encode and decode permissions you can use [ERC725 Inspect (https://erc725-inspect.lukso.tech/key-manager)](https://erc725-inspect.lukso.tech/key-manager)
- lsp3ProfileMetadata - optional. LSP3 Metadata to set on the deployed Universal Profile.

Response

```
{
  "universalProfileAddress": "0x8303Aa4dAD1825e524868E4224020f04D2446db5",
  "transactionHash": "0x58b473fef1c52da3ff32d77d11da15079ca12d1688db3b431d0a12cb3a600f56"
}
```

How the deployement of a Universal Profile works.

Deploying a Universal Profile requires to deploy smart contracts that need to interact with each others. The steps are the following :

1. Deploying a ERC725Account contract
2. Deploying a LSP6 Key Manager and linking it to the previously deployed ERC725Account contract
3. Optional : Setting additional data on the deployed ERC725Account contract like a Universal Receiver Delegate or LSP3 Metadata.

LSP23 standard is used in oder to deploy these contracts interdependently but also linked together as some contracts require other contracts addresses at the time of their deployment. LSP23 manages this by deploying a primary contract (in our case an ERC725Account contract) and a secondary contracts (in our case a LSP6 Key Manager contract), linking them together.

To use the LSP23 standard, one can interact with the LSP23LinkedContractsFactory deployed on LUKSO Testnet at address: 0x2300000A84D25dF63081feAa37ba6b62C4c89a30.

In this repository, I used LSP23 to deploy ERC1167 minimal proxies with the following base contract addresses:

```
universalProfileInitAddress = "0x0000000000e6300463CDbbF7ECF223a63397C489";
keyManagerInitAddress = "0x000000000A6cAE9b1bB3d9DA92BFf3569b77707E";
postDeploymentModuleAddress = "0x000000000066093407b6704B89793beFfD0D8F00";
```

In this relayer implementation, the Post Deployment Module sets the following data on the deployed Universal Profile:

- A Universal Receiver Delegate deployed on LUKSO Testnet at address 0xA20454137b47440C71fE4DD203D25D69F0b34535
- Optional : LSP3 Profile Metadata if passed by the user as parameter.

More detailed information on LSP23-LinkedContractsFactory (https://github.com/lukso-network/LIPs/blob/main/LSPs/LSP-23-LinkedContractsFactory.md#deployerc1167proxies)

Notes

This relayer doesn't implement the /quota endpoint standardized in the LSP15-TransactionRelayServiceAPI (https://github.com/lukso-network/LIPs/blob/main/LSPs/LSP-15-TransactionRelayServiceAPI.md). You can find an example of this endpoint in this repository (https://github.com/CallumGrindle/dummy-transaction-relayer) made by Callum Grindle which shows another example of a relayer service.

In version 0.12 of @lukso/lsp-smart-contracts library, userPrivateKey will need to have permission EXECUTE_RELAY_CALL on the Universal Profile.

Ideas for hackaton

Nonce management

Nonce management systems to allow the relayer to process concurrent transaction. Multiple keys can sign transactions in parallel.

Batch transaction

Send transaction to the blockchain in batch using executeRelayCallBatch function.

Advantages:

- Reduce cost of gas fee

Disadvantages:

- If one transaction fails from the batch then the whole batch fails.
- Users can receive the transaction hash of their transaction.

Quota management

Manage user's quota.

- How users retribute the relayer.
- How / when quota is refilled.
  </file>

<file>

path: /package.json

url: https://github.com/magalimorin18/relayer-example/blob/main/package.json

```
{
  "name": "relayer-example",
  "version": "0.0.1",
  "description": "Example relayer for LUKSO workshop.",
  "main": "app.ts",
  "author": "Magali Morin",
  "scripts": {
    "generate-types": "typechain --target ethers-v5 --out-dir 'types/ethers-v5' './node_modules/@lukso/lsp-smart-contracts/artifacts/*.json'",
    "dev": "yarn run generate-types && npx nodemon -r dotenv/config src/app.ts",
    "build": "tsc -p tsconfig.json",
    "execute": "npx ts-node scripts/execute-relay-call/execute.ts",
    "deploy-up": "npx ts-node scripts/deploy-up/deploy-up.ts"
  },
  "dependencies": {
    "@lukso/eip191-signer.js": "^0.2.2",
    "@lukso/lsp-smart-contracts": "^0.11.1",
    "@types/express": "^4.17.16",
    "@types/node": "^18.11.18",
    "axios": "^1.4.0",
    "dotenv": "^16.0.3",
    "ethers": "5.6.2",
    "express": "^4.18.2",
    "http-status": "^1.6.2",
    "typescript": "^4.9.5",
    "winston": "^3.8.2"
  },
  "devDependencies": {
    "@typechain/ethers-v5": "^10.2.0",
    "@typescript-eslint/eslint-plugin": "^5.50.0",
    "@typescript-eslint/parser": "^5.50.0",
    "eslint": "^8.33.0",
    "eslint-config-prettier": "^8.6.0",
    "eslint-import-resolver-typescript": "^3.5.3",
    "eslint-plugin-import": "^2.27.5",
    "eslint-plugin-prettier": "^4.2.1",
    "eslint-plugin-unicorn": "^45.0.2",
    "nodemon": "^2.0.20",
    "ts-node": "^10.9.1",
    "typechain": "^8.1.1"
  }
}
```

</file>

<file>

path: /scripts/deploy-up/deploy-up.ts

url: https://github.com/magalimorin18/relayer-example/blob/main/scripts/deploy-up/deploy-up.ts

```
/**
 * This script deploys a Universal Profiles based on the inputs variable.
 * An example of the structure of the inputs can be found in scripts/deploy-up/example-up-inputs.json.
 * It needs to be of type DeployUpRequest.
 *
 * Parameters :
 *&nbsp;controllers&nbsp; : A list of controller addresses with permissions to be set on the deployed Universal Profile.
 *&nbsp;lsp3ProfileMetadata&nbsp;: optional. LSP3 Metadata to set on the deployed Universal Profile.
 *
 * ⚠ Update from the oral prensentation: The parameters are loaded from the json file directly.
 * To execute the script:
 * &nbsp;yarn run deploy-up example-up-inputs.json&nbsp;
 *
 * The file example-up-inputs.json needs to be in the same folder as deploy-up.ts file.
 *
 * - The script will deploy a Universal Profile with controller addresses from the example-up-inputs.json file
 * and assign then the permissions you set.
 * e.g. If in the file example-up-inputs.json you set in the controllers array :
 *    {
 *     "address": "0x479a0BD32ba791E92d8bF9BF263eF63780182645",
 *    "permissions": "0x0000000000000000000000000000000000000000000000000000000000040000"
 *    }
 * Then, the address 0x479a0BD32ba791E92d8bF9BF263eF63780182645 will have SET_DATA permission on the UP.
 * In order to encode and decode permission you can use : &nbsp;https://erc725-inspect.lukso.tech/key-manager&nbsp;
 *
 * - The script will set a Universal Receiver Delegate on the deployed Universal Profile.
 *
 * - If you provide lsp3ProfileMetadata in the example-inputs file, the script will set LSP3 metadata on the deployed Universal Profile.
 *
 */

import axios from "axios";
import {
  CHAIN_ID,
  RELAYER_BASE_URL,
  RELAYER_PRIVATE_KEY,
} from "../../src/globals";
import { DeployUpRequest, DeployUpResponse } from "../../src/interface";

const fileName = process.argv[2];

const checkInputVariables = async () => {
  if (!fileName) {
    throw new Error(
      "Please specify the file name containing the deploy-up parameters."
    );
  }
  const filePath = "./" + fileName;
  console.log("      Loading deploy-up parameters from path:", filePath);
  const inputs: DeployUpRequest = await import(filePath);

  if (!inputs?.controllers) {
    throw new Error("No controllers provided in the inputs variable.");
  }

  inputs.controllers.forEach((controller) => {
    if (!controller?.address || !controller?.permissions) {
      throw new Error("Invalid input parameters for controllers.");
    }
  });

  return inputs;
};

const checkEnvVariables = () => {
  if (!RELAYER_PRIVATE_KEY) {
    throw new Error("No RELAYER_PRIVATE_KEY provided.");
  }
  if (!CHAIN_ID) {
    throw new Error("No CHAIN_ID provided.");
  }
};

const main = async () => {
  const inputs = await checkInputVariables();
  checkEnvVariables();

  let response;
  try {
    console.log("      Sending Universal Profile deployment request ...");
    response = await axios.post(
      RELAYER_BASE_URL + "/universal-profile",
      inputs
    );

    const { universalProfileAddress, transactionHash, keyManagerAddress } =
      response?.data as DeployUpResponse;

    console.log(
      &nbsp;      Successfully sent transaction:
https://explorer.execution.testnet.lukso.network/tx/${transactionHash}&nbsp;
    );
    console.log(
      &nbsp;      Universal Profile deployed at address ${universalProfileAddress} with       Key Manager
${keyManagerAddress}&nbsp;
    );
  } catch (error: any) {
    console.log(
      &nbsp;      Error executing /universal-profile endpoint: ${error?.response?.data?.message}&nbsp;
    );
    return;
  }
};

main()
  .then(() => {
    console.log("     Done.");
  })
  .catch((error: any) => {
    console.log("     ", error);
  });
```

</file>

<file>

path: /scripts/deploy-up/example-up-inputs.json

url: https://github.com/magalimorin18/relayer-example/blob/main/scripts/deploy-up/example-up-inputs.json

```
{
  "controllers": [
    {
      "address": "0x479a0BD32ba791E92d8bF9BF263eF63780182645",
      "permissions": "0x0000000000000000000000000000000000000000000000000000000000040000"
    },
    {
      "address": "0xDD06fC1f047103ea047654c57C6f6F754E91CD24",
      "permissions": "0x00000000000000000000000000000000000000000000000000000000003f3f7f"
    },
    {
      "address": "0x55535425c795C972b2723492e76f0c8D61dBFFB4",
      "permissions": "0x00000000000000000000000000000000000000000000000000000000003f3f7f"
    }
  ],
  "lsp3ProfileMetadata":
"0x6f357c6ae8b1db2db7ce8a555019ab31c557e9934145c0baa7fdae3f1a131e736d3d4e8e697066733a2f2f516d583543365939614a35646879697533546446704b6258797653434345386a435477514436374743784d705a31"
}
```

</file>

<file>

path: /scripts/execute-relay-call/constants.ts

url: https://github.com/magalimorin18/relayer-example/blob/main/scripts/execute-relay-call/constants.ts

```
export const DEFAULT_LSP3_PROFILE_DATA =

"0x6f357c6a3e2e3b435dd1ee4b8a2435722ee5533ea3f6cf6cb44c7fc278ac57ea1480295e697066733a2f2f516d5861714d67646971664b79313843735747768534a4c62626136316f6676666857387175506e6e6a6e76625966";

```

</file>

<file>

path: /scripts/execute-relay-call/example-execute-inputs.json

url: https://github.com/magalimorin18/relayer-example/blob/main/scripts/execute-relay-call/example-execute-inputs.json

```
{
  "universalProfileAddress": "0xda09F6F4acC07647C275319A7176A2D96E7cbA5c",
  "userPrivateKey": "0x04a3042380a01c4600df3cd06a86755f51426493d8a0ba12118f47f64d84471e"
}

```

</file>

<file>

path: /scripts/execute-relay-call/execute.ts

url: https://github.com/magalimorin18/relayer-example/blob/main/scripts/execute-relay-call/execute.ts

```
/**
 * This script executes a relay call on a Universal Profile based on the inputs variable.
 * An example of the structure of the inputs can be found in scripts/execute-relay-call/example-execute-inputs.json. It needs to be of type ExecuteRequest.
 *
 * Parameters :
 *&nbsp;universalProfileAddress&nbsp; : Universal Profile on which you want to execute an abi.
 *&nbsp;userPrivateKey&nbsp; : private key to sign the executeRelayCall message. This key needs to have permission on the universalProfileAddress to execute the payload. This key doesnt need to have LYX on it.
 *&nbsp;abi&nbsp;: optional. An abi to execute on the universal profile. By default, a SET_DATA will be executed.
 *
 * IMPORTANT : If you don't specify an abi parameters in the inputs variable, make sure the userPrivateKey has SET_DATA
 * permission on the Universal Profile as by default a SET_DATA transaction will be executed as the execute relay call.
 *
 * ⚠ Update from the oral prensentation: The parameters are loaded from the json file directly.
 * To execute the script:
 * &nbsp;yarn run execute example-execute-inputs.json&nbsp;
 *
 * The file example-execute-inputs.json needs to be in the same folder as execute.ts file.
 *
 * It is important that the userPrivateKey in the inputs has permissions to execute the payload you want on the Universal Profile.
 * e.g. if you choose to execute a payload that TRANSFER_VALUE on the Universal Profile then the user private key needs TRANSFER_VALUE permission.
 *
 */

import axios from "axios";
import {
  CHAIN_ID,
  RELAYER_BASE_URL,
  RELAYER_PRIVATE_KEY,
} from "../../src/globals";
import { generateExecuteParameters } from "./generate-body-execute";
import { ExecuteRequest } from "../../src/interface";

const fileName = process.argv[2];

const checkInputVariables = async () => {
  if (!fileName) {
    throw new Error(
      "Please specify the file name containing the execute parameters."
    );
  }
  const filePath = "./" + fileName;
  console.log("      Loading execute parameters from path:", filePath);
  const inputs: ExecuteRequest = await import(filePath);

  if (!inputs?.universalProfileAddress) {
    throw new Error(
      "No universalProfileAddress provided in the inputs variable."
    );
  }

  if (!inputs?.userPrivateKey) {
    throw new Error("No userPrivateKey provided in the inputs variable.");
  }

  return inputs;
};

const checkEnvVariables = () => {
  if (!RELAYER_PRIVATE_KEY) {
    throw new Error("No RELAYER_PRIVATE_KEY provided.");
  }

  if (!CHAIN_ID) {
    throw new Error("No CHAIN_ID provided.");
  }
};
```

```
const main = async () => {
  const inputs = await checkInputVariables();
  checkEnvVariables();

  const body = await generateExecuteParameters(
    inputs.universalProfileAddress,
    inputs.userPrivateKey,
    inputs?.abi
  );

  let response;
  try {
    console.log("     Sending Execute Relay Call request ...");
    response = await axios.post(RELAYER_BASE_URL + "/execute", body);
  } catch (error: any) {
    console.log(&nbsp;     Error executing /execute endpoint&nbsp;);
    error?.response?.data?.message && console.log(error.response.data.message);
    return;
  }

  console.log(
    &nbsp;     Successfully sent transaction:
https://explorer.execution.testnet.lukso.network/tx/${response.data.transactionHash}&nbsp;
  );
};

main()
  .then(() => {
    console.log("    Done.");
  })
  .catch((error: any) => {
    console.log("    ", error);
  });
```

</file>

<file>

path: /scripts/execute-relay-call/generate-body-execute.ts

url: https://github.com/magalimorin18/relayer-example/blob/main/scripts/execute-relay-call/generate-body-execute.ts

```
import { BigNumber, Wallet, ethers } from "ethers";
import { getProvider } from "../../src/libs/ethers.service";
import {
  LSP6KeyManager__factory,
  UniversalProfile__factory,
} from "../../types/ethers-v5";
import { EIP191Signer } from "@lukso/eip191-signer.js";
import { CHAIN_ID } from "../../src/globals";
import { ERC725YDataKeys } from "@lukso/lsp-smart-contracts";
import { DEFAULT_LSP3_PROFILE_DATA } from "./constants";

export const generateExecuteParameters = async (
  universalProfileAddress: string,
  userPrivateKey: string,
  abi?: string
) => {
  const provider = getProvider();

  const universalProfile = UniversalProfile__factory.connect(
    universalProfileAddress,
    provider
  );
  console.log(&nbsp;     Universal Profile address : ${universalProfileAddress}&nbsp;);

  const keyManagerAddress = await universalProfile.owner();
  console.log(&nbsp;     Key Manager address : ${keyManagerAddress}&nbsp;);

  const keyManager = LSP6KeyManager__factory.connect(
    keyManagerAddress,
    provider
  );

  const wallet = new Wallet(userPrivateKey, provider);
  const walletAddress = wallet.address;
  console.log(&nbsp;     Wallet address signing transaction : ${walletAddress}&nbsp;);

  let nonce: BigNumber;
  try {
    nonce = await keyManager.getNonce(walletAddress, 0); //Use of channel 0
  } catch (error) {
    throw new Error(
      &nbsp;     Error while fetching nonce of the Key Manager. ${error}&nbsp;
    );
  }

  if (!abi) {
    abi = universalProfile.interface.encodeFunctionData("setData", [
      ERC725YDataKeys.LSP3.LSP3Profile,
      DEFAULT_LSP3_PROFILE_DATA,
    ]);
  }

  const validityTimestamps = 0;

  const message = ethers.utils.solidityPack(
    ["uint256", "uint256", "uint256", "uint256", "uint256", "bytes"],
    [
      25, // LSP25 Version
      CHAIN_ID,
      nonce,
      validityTimestamps,
      0, // the amount of native tokens to transfer (in Wei)
      abi,
    ]
  );

  const eip191Signer = new EIP191Signer();
  const { signature } = eip191Signer.signDataWithIntendedValidator(
    keyManagerAddress,
    message,
    userPrivateKey // This address needs permissions on the universalProfileAddress
  );

  const transactionObject = {
    abi,
    signature: signature,
    nonce, // Nonce has to be a Big number
    validityTimestamps,
  };

  return {
    address: universalProfileAddress,
    transaction: transactionObject,
  };
};
```

</file>

<file>

path: /src/app.ts

url: https://github.com/magalimorin18/relayer-example/blob/main/src/app.ts

```
import express, { Express, Request, Response } from "express";
import http from "http";
import { AddressInfo } from "net";
import executeController from "./execute-relay-call/execute.controller";
import deployUpController from "./deploy-up/deploy-up.controller";

const host = process.env.HOST || "0.0.0.0";
const port = process.env.PORT || "3000";

const createApp = () => {
  const app: Express = express();

  // Configure the app to be able to receive different types of payload
  app.use(express.urlencoded({ extended: true }));
  app.use(express.json());

  // Security : do not display the technology used to power the server
  app.disable("x-powered-by");

  app.get("/health", (_req: Request, res: Response) => {
    res.send("Backend is running.");
  });

  return app;
};

const startServer = () => {
  const app = createApp();

  app.use("/", executeController);
  app.use("/", deployUpController);

  const server = http.createServer(app).listen({ host, port }, () => {
    const serverInfo = server.address() as AddressInfo;

    console.log(
      &nbsp;      Server successfully started at http://${serverInfo.address}:${serverInfo.port}&nbsp;
    );
  });

  const signalTraps: NodeJS.Signals[] = ["SIGTERM", "SIGINT", "SIGUSR2"];
  signalTraps.forEach((type) => {
    process.once(type, async () => {
      console.log(&nbsp;process.once ${type}&nbsp;);
      server.close(() => {
        console.log("Closing server.");
      });
    });
  });
};

startServer();
```

</file>

<file>

path: /src/deploy-up/deploy-up.controller.ts

url: https://github.com/magalimorin18/relayer-example/blob/main/src/deploy-up/deploy-up.controller.ts

```
import express, { Request, Response } from "express";
import { DeployUpRequest, ExecutePayload } from "../interface";
import httpStatus from "http-status";
import { deployUpWithKm } from "./deploy-up.service";

const deployUpController = express.Router();

const deployUp = async (req: Request, res: Response) => {
  const { controllers, lsp3ProfileMetadata } = req.body as DeployUpRequest;

  try {
    const response = await deployUpWithKm(controllers, lsp3ProfileMetadata);
    res.send({
      universalProfileAddress: response?.universalProfileAddress,
      transactionHash: response?.transactionHash,
      keyManagerAddress: response?.keyManagerAddress,
    });
  } catch (error: any) {
    console.log("      An error occured when executing transaction.", error);

    res.status(httpStatus.INTERNAL_SERVER_ERROR).send("Internal Server Error");
  }
};

deployUpController.post("/universal-profile", deployUp);

export default deployUpController;
```

</file>

<file>

path: /src/deploy-up/deploy-up.service.ts

url: https://github.com/magalimorin18/relayer-example/blob/main/src/deploy-up/deploy-up.service.ts

```
import { ERC725YDataKeys } from "@lukso/lsp-smart-contracts";
import { BigNumber, Wallet } from "ethers";
import {
  BytesLike,
  defaultAbiCoder,
  hexZeroPad,
  randomBytes,
} from "ethers/lib/utils";

// types
import {
  LSP23LinkedContractsFactory__factory,
  LSP6KeyManagerInit__factory,
  UniversalProfileInit__factory,
} from "../../types/ethers-v5";
```

```
import { RELAYER_PRIVATE_KEY } from "../globals";
import { getProvider } from "../libs/ethers.service";
import { Controllers } from "../interface";

/// ------ DO NOT CHNAGE ------
const lsp23Address = "0x2300000A84D25dF63081feAa37ba6b62C4c89a30";
const upPostDeploymentModuleAddress =
  "0x000000000066093407b6704B89793beFfD0D8F00";

const universalProfileInitAddress =
  "0x0000000000e6300463CDbbF7ECF223a63397C489";
const keyManagerInitAddress = "0x000000000A6cAE9b1bB3d9DA92BFf3569b77707E";
const universalReceiverDelegateAddress =
  "0xA20454137b47440C71fE4DD203D25D69F0b34535";
/// --------------------------

export const deployUpWithKm = async (
  controllers: Controllers[],
  lsp3ProfileMetadata?: string
) => {
  /// ------ Contract Instance for the &nbsp;LSP23LinkedContractsFactory&nbsp; ------
  const LSP23LinkedContractsFactory =
    new LSP23LinkedContractsFactory__factory().attach(lsp23Address);
  /// ------------------------------------------------------------------

  /// ------ Contract Instance for the &nbsp;UniversalProfileInit&nbsp; ------
  const universalProfileInit = new UniversalProfileInit__factory().attach(
    universalProfileInitAddress
  );
  /// ------------------------------------------------------------------

  /// ------ Contract Instance for the &nbsp;LSP6KeyManagerInit&nbsp; ------
  const keyManagerInit = new LSP6KeyManagerInit__factory().attach(
    keyManagerInitAddress
  );
  /// ------------------------------------------------------------------

  /// ------ Generate Random Salt for Universal Profile deployment (we can also use custom salt) ------
  const salt = randomBytes(32);
  /// ----------------------------------------------------------------------------------------------

  /// ------ Data for Universal Profile deployment ------
  const primaryContractDeploymentInit = {
    salt,
    fundingAmount: 0,
    implementationContract: universalProfileInit.address,
    initializationCalldata: universalProfileInit.interface.encodeFunctionData(
      "initialize",
      [upPostDeploymentModuleAddress]
    ),
  };
  /// --------------------------------------------------
  /// ------ Data for Key Manager deployment ------
  const secondaryContractDeploymentInit = {
    fundingAmount: 0,
    implementationContract: keyManagerInit.address,
    addPrimaryContractAddress: true,
    initializationCalldata:
      universalProfileInit.interface.getSighash("initialize"),
    extraInitializationParams: "0x",
  };
  /// ---------------------------------------------------

  /// ------ Encode Data Keys & Values for updating permissions & LSP3Metadata ------

  const controllerKeys = controllers.flatMap((controller, index) => [
    ERC725YDataKeys.LSP6["AddressPermissions[]"].index +
      hexZeroPad(BigNumber.from(index).toHexString(), 16).slice(2), //Key of the controller address index

    ERC725YDataKeys.LSP6["AddressPermissions:Permissions"] +
      controller.address.slice(2), // Key of the controller address permission
  ]);

  const controllerValues = controllers.flatMap((controller) => [
    controller.address,
    controller.permissions,
  ]);

  const data = [
    {
      key: ERC725YDataKeys.LSP1.LSP1UniversalReceiverDelegate,
      value: universalReceiverDelegateAddress,
    },
  ];

  if (lsp3ProfileMetadata) {
    data.push({
      key: ERC725YDataKeys.LSP3.LSP3Profile,
      value: lsp3ProfileMetadata,
    });
  }

  const dataKeys = data ? data.map(({ key }) => key) : [];
  const dataValues = data ? data.map(({ value }) => value) : [];

  const encodedSetData = defaultAbiCoder.encode(
    ["bytes32[]", "bytes[]"],
    [
      [
        ERC725YDataKeys.LSP6["AddressPermissions[]"].length,
        ...controllerKeys,
        ...dataKeys,
      ],
      [
        hexZeroPad(BigNumber.from(controllers.length).toHexString(), 16),
        ...controllerValues,
        ...dataValues,
      ],
    ]
  );
  /// ------------------------------------------------------------------------------

  const provider = getProvider();
  const signer = new Wallet(RELAYER_PRIVATE_KEY, provider);

  /// ------ Pre-calculate the addresses for the Universal Profile & Key Manager ------
  const [universalProfileAddress, keyManagerAddress] =
    await LSP23LinkedContractsFactory.connect(
      signer
    ).callStatic.deployERC1167Proxies(
      primaryContractDeploymentInit,
      secondaryContractDeploymentInit,
      upPostDeploymentModuleAddress,
      encodedSetData
    );

  console.log(
    &nbsp;     Deploying Universal Profile at address ${universalProfileAddress} with      Key Manager
  ${keyManagerAddress}...&nbsp;
  );
  /// --------------------------------------------------------------------------------

  /// ------ Deploy the Universal Profile & Key Manager ------
  const deploymentTransaction = await LSP23LinkedContractsFactory.connect(
    signer
  ).deployERC1167Proxies(
    primaryContractDeploymentInit,
    secondaryContractDeploymentInit,
    upPostDeploymentModuleAddress,
    encodedSetData
  );
  /// -------------------------------------------------------

  console.log(
    &nbsp;     Transaction sent
  https://explorer.execution.testnet.lukso.network/tx/${deploymentTransaction.hash}&nbsp;
  );

  await deploymentTransaction.wait(1);

  return {
    universalProfileAddress,
    keyManagerAddress,
    transactionHash: deploymentTransaction.hash,
  };
```

```
};
```

</file>

<file>

path: /src/execute-relay-call/execute.controller.ts

url: https://github.com/magalimorin18/relayer-example/blob/main/src/execute-relay-call/execute.controller.ts

```
import express, { Request, Response } from "express";
import { ExecutePayload } from "../interface";
import httpStatus from "http-status";
import { executeTransaction } from "./execute.service";

const executeController = express.Router();

const execute = async (req: Request, res: Response) => {
  const { address, transaction } = req.body as ExecutePayload;

  try {
    const transactionHash = await executeTransaction(address, transaction);
    res.send({ transactionHash });
  } catch (error: any) {
    console.log(
      "      An error occured when executing transaction.",
      error?.message
    );
    if (error?.message.includes("Transaction in progress")) {
      res
        .status(httpStatus.TOO_MANY_REQUESTS)
        .send({ message: error?.message });
      return;
    }

    res.status(httpStatus.INTERNAL_SERVER_ERROR).send("Internal Server Error");
  }
};

executeController.post("/execute", execute);

export default executeController;
```

</file>

<file>

path: /src/execute-relay-call/execute.service.ts

url: https://github.com/magalimorin18/relayer-example/blob/main/src/execute-relay-call/execute.service.ts

```
import { BigNumber, ethers } from "ethers";
import {
  LSP6KeyManager__factory,
  UniversalProfile__factory,
  LSP6KeyManagerInit__factory,
} from "../../types/ethers-v5";

import { getProvider } from "../libs/ethers.service";
import { TransactionParameters } from "../interface";
import { transactionGate, waitForTransaction } from "../libs/listener.service";
import { signTransaction } from "../libs/signer.service";

export const executeTransaction = async (
  address: string,
  transaction: TransactionParameters
) => {
  console.log(&nbsp;      Received execute request for Universal Profile ${address}&nbsp;);

  transactionGate();

  const { signature, nonce, abi, validityTimestamps } = transaction;

  const provider = getProvider();
  const lsp6Interface = LSP6KeyManagerInit__factory.createInterface();

  const universalProfile = UniversalProfile__factory.connect(address, provider);
  const keyManagerAddress = await universalProfile.owner();
  const keyManager = LSP6KeyManager__factory.connect(
    keyManagerAddress,
    provider
  );

  let gasLimit: BigNumber;
  try {
    gasLimit = await keyManager.estimateGas.executeRelayCall(
      transaction.signature,
      transaction.nonce,
      validityTimestamps,
      transaction.abi
    );
  } catch (error) {
    gasLimit = BigNumber.from(3000000);
    console.log(
      "      Unable to estimate gas. Setting default value to gas Limit"
    );
  }

  const transactionData = lsp6Interface.encodeFunctionData("executeRelayCall", [
    signature,
    nonce,
    validityTimestamps,
    abi,
  ]);

  const signedTransaction = await signTransaction(
    {
      to: keyManagerAddress,
      transactionData,
      gasLimit,
    },
    provider
  );

  let transactionResponse: ethers.providers.TransactionResponse;
  try {
    transactionResponse = await provider.sendTransaction(
      signedTransaction.signerSignature
    );
  } catch (error) {
    throw Error(&nbsp;     Error sending transaction to the blockchain. ${error}&nbsp;);
  }

  console.log("     Waiting for transaction to be mined...");

  await waitForTransaction(transactionResponse);

  const transactionHash = ethers.utils.keccak256(
    signedTransaction.signerSignature
  );

  console.log(
    &nbsp;     Successfully sent transaction:
https://explorer.execution.testnet.lukso.network/tx/${transactionHash}&nbsp;
  );

  return transactionHash;
};
```

</file>

<file>

path: /src/globals.ts

url: https://github.com/magalimorin18/relayer-example/blob/main/src/globals.ts

```
export const RPC_ENDPOINT =
  process.env.RPC_ENDPOINT || "https://rpc.testnet.lukso.network";

export const CHAIN_ID = process.env.CHAIN_ID || "4201";

export const RELAYER_PRIVATE_KEY =
  process.env.RELAYER_PRIVATE_KEY ||
  "0x04a3042380a01c4600df3cd06a86755f51426493d8a0ba12118f47f64d84471e";

export const RELAYER_BASE_URL = process.env.RELAYER_BASE_URL
  ? process.env.RELAYER_BASE_URL
  : "http://0.0.0.0:3000";
```

</file>

<file>

path: /src/interface.ts

url: https://github.com/magalimorin18/relayer-example/blob/main/src/interface.ts

```
import { BigNumber, BytesLike } from "ethers";

export interface TransactionParameters {
  abi: string;
  signature: string;
  nonce: BigNumber;
  validityTimestamps: number;
}

export interface ExecutePayload {
  address: string;
  transaction: TransactionParameters;
}

export interface SigningRequest {
  transactionData: string;
  to: string;
  gasLimit: BigNumber;
}

export interface SigningResponse {
  signerSignature: string;
  signerAddress: string;
  nonce: number;
}
export interface DeployUpRequest {
  controllers: Controllers[];
  lsp3ProfileMetadata?: string;
}

export interface DeployUpResponse {
  universalProfileAddress: string;
  keyManagerAddress: string;
  transactionHash: string;
}

export interface ExecuteRequest {
  universalProfileAddress: string;
  userPrivateKey: string;
  abi?: string;
}

export interface Controllers {
  address: string;
  permissions: BytesLike;
}
```

</file>

<file>

path: /src/libs/ethers.service.ts

url: https://github.com/magalimorin18/relayer-example/blob/main/src/libs/ethers.service.ts

```
import { ethers } from "ethers";
import { RPC_ENDPOINT } from "../globals";

let provider: ethers.providers.JsonRpcProvider;

export function getProvider() {
  if (!provider) {
    provider = new ethers.providers.JsonRpcProvider(RPC_ENDPOINT);
  }

  return provider;
}
```

</file>

<file>

path: /src/libs/listener.service.ts

url: https://github.com/magalimorin18/relayer-example/blob/main/src/libs/listener.service.ts

```
import { ethers } from "ethers";

let acceptNextTransaction = true;
let currentTransactionHash: string;

export async function waitForTransaction(
  transaction: ethers.providers.TransactionResponse
) {
  acceptNextTransaction = false;
  currentTransactionHash = transaction.hash;

  await transaction.wait();

  console.log(&nbsp;      Validated transaction ${transaction.hash}&nbsp;);

  acceptNextTransaction = true;
}

export function transactionGate() {
  if (!acceptNextTransaction) {
    throw new Error(
      &nbsp;Transaction in progress. Waiting until transaction ${currentTransactionHash} has been validated&nbsp;
    );
  }
}
```

</file>

<file>

path: /src/libs/signer.service.ts

url: https://github.com/magalimorin18/relayer-example/blob/main/src/libs/signer.service.ts

```
import { BigNumber, ethers } from "ethers";

import { getProvider } from "./ethers.service";
import { CHAIN_ID, RELAYER_PRIVATE_KEY } from "../globals";
import { SigningRequest, SigningResponse } from "../interface";

let signer: ethers.Wallet;

async function getSigner() {
  const provider = getProvider();

  if (!RELAYER_PRIVATE_KEY) {
    throw new Error("No signing key set");
  }

  if (!signer) {
    signer = new ethers.Wallet(RELAYER_PRIVATE_KEY, provider);
  }

  return signer;
}

export async function signTransaction(
  signingRequest: SigningRequest,
  provider: ethers.providers.JsonRpcProvider
): Promise<SigningResponse> {
  const signer = await getSigner();

  let signerBalanceInWei: ethers.BigNumber;
  try {
    signerBalanceInWei = await signer.getBalance();
  } catch (error) {
    throw new Error(&nbsp;    Unable to get signing key balance. ${error}&nbsp;);
  }

  console.log(
    &nbsp;    Signer address ${
      signer?.address
    } has a balance of ${ethers.utils.formatEther(signerBalanceInWei)} LYX&nbsp;
  );

  const { to, gasLimit, transactionData } = signingRequest;

  if (Number(gasLimit) > Number(signerBalanceInWei)) {
    const errorMessage = &nbsp;    Insufficiant balance. Gas Limit : ${gasLimit} SignerKeyBalance :
${signerBalanceInWei}&nbsp;;
    throw new Error(errorMessage);
  }

  const signerAddress = signer.address;
  console.log(&nbsp;    Signing transaction with signing key ${signerAddress}&nbsp;);

  let signerNonce: number;
  try {
    signerNonce = await provider.getTransactionCount(signerAddress);
  } catch (error) {
    throw Error(&nbsp;    Unable to get signing key nonce. ${error}&nbsp;);
  }

  const transactionParameters = {
    to,
    from: signerAddress,
    nonce: signerNonce,
    gasLimit,
    value: 0,
    type: 2,
    chainId: Number.parseInt(CHAIN_ID),
    data: transactionData,
  };

  let populatedTransaction;
  try {
    populatedTransaction = await signer.populateTransaction(
      transactionParameters
    );
  } catch (error) {
    throw Error(
      &nbsp;      Unable to populate transaction ${transactionParameters}. ${error}&nbsp;
    );
  }

  let signerSignature: string;
  try {
    signerSignature = await signer.signTransaction(populatedTransaction);
  } catch (error) {
    throw Error(
      &nbsp;      Error signing transaction ${populatedTransaction}. ${error}&nbsp;
    );
  }

  return {
    signerSignature,
    signerAddress: signerAddress,
    nonce: signerNonce,
  };
}
```

</file>

<file>

path: /tsconfig.json

url: https://github.com/magalimorin18/relayer-example/blob/main/tsconfig.json

```
{
  "compilerOptions": {
    "target": "es2016" /* Set the JavaScript language version for emitted JavaScript and include compatible library declarations. */,
    "module": "commonjs" /* Specify what module code is generated. */,
    "outDir": "./build" /* Specify an output folder for all emitted files. */,
    "esModuleInterop": true /* Emit additional JavaScript to ease support for importing CommonJS modules. This enables 'allowSyntheticDefaultImports' for type compatibility. */,
    "forceConsistentCasingInFileNames": true /* Ensure that casing is correct in imports. */,
    "strict": true /* Enable all strict type-checking options. */,
    "skipLibCheck": true /* Skip type checking all .d.ts files. */
  }
}
```

</file>

</repository>

<repository>

lukso-siwe-demo

overview

repository: https://github.com/richtera/lukso-siwe-demo/
userName: richtera
repository: lukso-siwe-demo
branch: main
date: 2023-11-27T10:16:38+01:00 (1701076598)
description:
hi I'm Andreas RoR principal engineer at
Luxo and I would like to talk about
authentication using Universal profile
and sign in with
ethereum I'd like to compare it with oo
and show what kind of Primitives you
would have to implement in order to
allow it to identify
authorize and log in a user into our
system first I'd like to discuss you
know what what is login you know why do
we need
it so essentially it's to identify the
user to give them permissions for what
they're allowed to do and to keep track
of them in terms of you know when they
make more requests to recognize them
again so they don't have to sign in
every single time they make a
request comparing it
to using normal transactions on the
blockchain you're essentially signing
one of those every single time you make
an interaction so for a
normal D you may not need authentication
but if you have some kind of backend
environment that needs to recognize the
user then authentication is a a good
thing and sign in with ethereum is
better a better way of dealing with it
than having an O provider so to define
the terms what login composes so
basically identification of the user is
to convert the user into a
unique user identity uh that may be just
a random number of bytes or it may be in
like in this case in Universal profiles

it may be the address of the
account but it can be something
Anonymous or something
personal um usually it doesn't have P ID
data in it but it's something that can
make the user known to the system some
systems even use emails
there
um then for
authorization that's basically a set of
permissions that are associated with a
user la l in so on an email system a
user might be able to read certain
folders and not read other ones and
those are permissions and they're called
authorization and session tracking
just it
explains a way to the server so that the
server can uh give the user a token and
when they come back the user sends that
token along so that the user can
reauthenticate themselves and have
access to the same permissions that they
have as they had before without this
they would have to log in every single
time they make a
request so let's talk about the identify
part and compare it to how it used to
work in oo or how it still working in
oath and kind of
compare when you compare oath to sign in
with ethereum inside of oo you have to
have a database that keeps track of
users and their
credentials and when a user logs in the
credentials are being sent to the
server username password it could be a
pass key but it's it's some kind of
information that represents the user
that the o server can look up in the
database
validate and then it can reply with an
authentication token or a code that can
then be exchanged for a token but
essentially it identifies the user
verifies them and then sends the reply
on the sign in with ethereum
side the user already
certifies the message with the private
key so there's no need to expose any
credentials on the server side the
server can validate the message

without having a database table that
keeps track of the
user now let's compare
authorization between o and uh sign in
with ethereum so those pieces could be
identical where where you would look up
the user in the database and look up the
permissions and then reply or you could
have some kind of onchain storage like a
a contract or a a key value store that
keeps track of which user has which
permissions
um since sign in with ethereum can
already prove that the user exists and
that it's a legitimate user all you need
to do is look up the permissions and and
return um a token for the user that
contains
that um alternatively you can even have
the user write those permissions in the
initial package and just validate that
they indeed should get those
permissions in terms of session tracking
things are pretty much very
similar um if you're using sign in with
ethereum uh it makes certain things a
little easier but let's see what the
difference is so in terms of session
tokens like JWT
tokens they're pretty much similar uh
when the tokens expire although when you
want to talk about refresh tokens that
might last for like a month or something
like that usually they need to be
registered
somewhere um you could register them
either in a
database or you could register them on
on chain in a key Value Store
um so that you can roke them if you ever
need
to on the sign in ethereum signing it
itself is really easy so you may prefer
to just have the user sign in
again
now I'm going to be talking about a
little demo that I put together that
kind of shows how the different pieces
go
together so first I want to quickly show
the some of the code point out certain
things so essentially the code is

centered around creation and validation
of this JWT tokens so in this
case I'm creating a JWT on the clients
side so um in order to do that you first
have to create the sign in with ethereum
message there's certain parameters that
are
required uh in this case you pass in the
statement which the user in this case
provides but normally that's hardcoded
uh for
you um you specify when the token was
signed or issued and when it expires in
this case it's only 10 seconds since the
client token you usually don't do much
with it you first send it to the server
and exchange it for server side
token in this case I'm I'm putting the
digital identity of the client in
here and the digital identity of the
website you prepare a message sign the
message and then here I'm doing a little
um prepending a header taking the
message the signature and then all base
64 URL encoding it this will just make
it act and look like a regular JWT token
that people use on the net to identify
users and to send as Sparrow tokens on
requests um exchange
token is a pretty simple function you
basically send the barer token to the
back end and the back end then responds
with the original token and a new token
a new signature and so on and a new
JWT if you want to just verify the token
that's basically sending it to any API
request so you would just send it as a
barer token and get the
response so there's nothing special
going on in very ification if you wanted
to verify the token on the client that's
a little more difficult because on the
client you have to check whether the
token was a server side token or a
client side token if it is a client side
token the address it's going to match
with your un Universal profile
address so in that case you have to call
the contract and call is valid signature
give it the hash and the signature and
then it will tell you if it
succeeded by returning this magic value

here 1626 ba
7e if it if it doesn't match
then the signature didn't
match for the other case whether whether
it's a seright token you got to make
sure it's your token so in this case I'm
checking for my dead in the
resources if that H if that succeeds
then I ask the server for its public
address in this case the public key
which is just a hex hex number of the
wallet then I recover that address from
the message by sending it the message in
the signature and then you can compare
whether the server public key is equal
to the
address and if it is then good otherwise
it's not
good uh but what's kind of cool is that
this allows the client to actually
validate messages from the server so if
the server
included it's
JWT inside of the response header you
could validate that that response
actually came from that
server
the server implementation is basically
two routines it's decode token so again
you split it up you Val you create the
sign in with ethereum message uh you can
verify the
message if the wallet address was you if
it's the server
address um otherwise you can uh verify
it using the LSP Zer ERC 725
account
so again those two cases if it's the
wallet address of the server then you
can just call the plane verify uh
message if it's the client signed token
then you ask is valid signature you get
the magic
number if it worked and if it's not the
magic number that you for an exception
for exchanging of token it's basically
the same you decode the token and then
you just create a new one but this time
you created on the
server so in this case I just take the
original sign in with ethereum message
that the user did and modify the

resources to include the web as a
digital identity the user as a digital
identity make sure that you don't
include include the server wallet as the
digital identity so I just
always remove it just in case it got in
there in this case you can make the
token last
longer um but otherwise you know it's
it's the same kind of code that's on the
client so you sign it with your wallet
you preent the header and again
serialize it base 64 URL encoded
and you can return it to the client so I
set up a little web
applic web tab that that kind of shows
this
so let's put in like a message by
logging
in
you agree to the
terms and
conditions
you click the login
button you
basically approve that
signature that will give you a client
side token here's the here's how the
token is going to look like it includes
this
message and has this um statement and
has the
signature you can verify it on the
client the client says yeah okay I
I knew what this was I recognized the
user you can verify it on the server by
sending it to the server so the server
says okay yep I know I know who this
is um you can now exchange it for a
server
token so the server came back it
returned the original message the new
message um you notice that the server
put itself into the resources
uh again it knew the account so it
Returns the account Returns the JWT
token which you can then verify on the
client so the client knew what this was
it was able to verify
it since the server accepts both the
client side token and the server side
token you can just send resend it and

refresh and get a new server side token
so in this case you can see that the two
messages are basically the same except
that the dates and times are
different then uh you can verify that on
the server so you can verify the new
newly generated one on the server and
that works
also so that's basically what I wanted
to show I put the the demo in GitHub so
you can check it out
see if you'd like to uh play with
it here's a little summary of what the
pieces are inside of this little demo um
I put the source code in the GitHub repo
so you can you can check it
out

<file>

path: /.dockerignore

url: https://github.com/richtera/lukso-siwe-demo/blob/main/.dockerignore

```
*
!/src
!/index.html
!/tsconfig.*
!/pnpm-lock.yaml
!/package.json
!/public
!/tailwind.config.js
!/vite.config.js
!/postcss.config.js
!/polyfills.ts
!/server
```

</file>

<file>

path: /.env.example

url: https://github.com/richtera/lukso-siwe-demo/blob/main/.env.example

```
VITE_WALLET_CONNECT_PROJECT_ID=
VITE_APP_DOMAIN=example.com
VITE_MNEMONIC=
```

</file>

<file>

path: /.eslintrc.json

url: https://github.com/richtera/lukso-siwe-demo/blob/main/.eslintrc.json

```
{
  "extends": ["next/core-web-vitals", "eslint:recommended", "prettier", "plugin:prettier/recommended",
"plugin:@typescript-eslint/recommended", "plugin:unicorn/recommended"],
  "plugins": ["prettier", "@typescript-eslint"],
  "rules": {
    "prettier/prettier": "error",
    //TYPESCRIPT
    "@typescript-eslint/no-unused-vars": ["error", { "vars": "all", "args": "all", "argsIgnorePattern": "^_" }],
    "unicorn/no-useless-undefined": "off",
    "unicorn/no-nested-ternary": "off",
    // NEXT
    "@next/next/no-img-element": "off",
    // REACT
    "react/no-unescaped-entities": "off",
    // Unicorn
    "unicorn/prevent-abbreviations": [
      "error",
      {
        "replacements": {
          "e": false,
          "i": false,
          "props": false
        }
      }
    ],
    "unicorn/filename-case": ["off"],
    "unicorn/catch-error-name": ["off"],
    "@typescript-eslint/no-explicit-any": "off",
    // OTHER
    "no-console": ["error", { "allow": ["warn", "error"] }],
    "import/order": [
      "error",
      {
        "groups": [["builtin", "external", "index"], ["sibling", "parent", "internal"], "object", "type"],
        "newlines-between": "always",
        "alphabetize": { "order": "asc" }
      }
    ],
    "@typescript-eslint/ban-ts-comment": "off"
  }
}
```

</file>

<file>

path: /.npmrc

url: https://github.com/richtera/lukso-siwe-demo/blob/main/.npmrc

```
strict-peer-dependencies = false
```

</file>

<file>

path: /.nvmrc

url: https://github.com/richtera/lukso-siwe-demo/blob/main/.nvmrc

```
v18.12.1

```

</file>

<file>

path: /.prettierrc

url: https://github.com/richtera/lukso-siwe-demo/blob/main/.prettierrc

```
{
  "trailingComma": "all",
  "tabWidth": 2,
  "semi": true,
  "singleQuote": true,
  "printWidth": 80
}

```

</file>

<file>

path: /.tool-versions

url: https://github.com/richtera/lukso-siwe-demo/blob/main/.tool-versions

```
nodejs 18.12.1

```

</file>

<file>

path: /.vscode/launch.json

url: https://github.com/richtera/lukso-siwe-demo/blob/main/.vscode/launch.json

```
{
  // Use IntelliSense to learn about possible attributes.
  // Hover to view descriptions of existing attributes.
  // For more information, visit: https://go.microsoft.com/fwlink/?linkid=830387
  "version": "0.2.0",
  "configurations": [
    {
      "name": "Launch via NPM",
      "request": "launch",
      "runtimeArgs": ["run-script", "dev"],
      "runtimeExecutable": "npm",
      "cwd": "${workspaceFolder}",
      "skipFiles": ["<node_internals>/**"],
      "type": "node",
      "outputCapture": "std",
      "console": "integratedTerminal"
    }
  ]
}
```

</file>

<file>

path: /.vscode/settings.json

url: https://github.com/richtera/lukso-siwe-demo/blob/main/.vscode/settings.json

```
{
  "editor.formatOnSave": true,
  "editor.formatOnPaste": true
}
```

</file>

<file>

path: /Dockerfile

url: https://github.com/richtera/lukso-siwe-demo/blob/main/Dockerfile

```
FROM node:18.17.1-slim as builder
RUN apt-get update \
  && apt-get install python3 make gcc g++ git -y \
  && npm i -g pnpm
WORKDIR /app
COPY package.json pnpm-lock.yaml /app/
RUN pnpm install --frozen-lockfile
WORKDIR /run
COPY package.json pnpm-lock.yaml /run/
RUN pnpm install --frozen-lockfile --prod --shamefully-hoist
COPY . /app/
WORKDIR /app
ARG VITE_WALLET_CONNECT_PROJECT_ID
ARG VITE_APP_DOMAIN=example.com
ENV VITE_WALLET_CONNECT_PROJECT_ID=$VITE_WALLET_CONNECT_PROJECT_ID
ENV VITE_APP_DOMAIN=$VITE_APP_DOMAIN
RUN pnpm run typechain && pnpm run build

FROM node:18.17.1-slim
ENV PORT=8080
ENV NODE_ENV=production
COPY --from=builder /app/dist.server /app
COPY --from=builder /run/node_modules /app/node_modules
COPY package.json pnpm-lock.yaml /app/
COPY --from=builder /app/dist/ /app/web/
ENV BASE_DIR=/app/web
ENV VITE_MNEMONIC=""
CMD ["node", "/app/server/index.js"]
```

</file>

<file>

path: /README.md

url: https://github.com/richtera/lukso-siwe-demo/blob/main/README.md

This demo uses amongst other thing wagmi (https://wagmi.sh) + Web3Modal (https://web3modal.com/) + Vite (https://vitejs.dev/) project bootstrapped with create-wagmi (https://github.com/wagmi-dev/wagmi/tree/main/packages/create-wagmi)

Demo

A demo implementation of this dapp is deployed at.
https://siwe-demo.richtera.app (https://siwe-demo.richtera.app)

Getting Started

Get a wallet connect projectId and put it into .env (copy .env.example and fill it in)
The app will create a mnemonic for you the first time you launch if it's not defined.
You can copy the random one it generates into the .env so your server tokens will be valid during a restart.

Install pnpm and nodejs (use asdf or nvm to get the right)
Run pnpm install
Run pnpm run typegen (one time to bootstrap the typechain wrappers for lsp-smart-contracts)
Run pnpm run dev in your terminal, and then open https://localhost:5173 (https://localhost:5173) in your browser.

Once the webpage has loaded, changes made to files inside the src/ and server/ directories (e.g. src/App.tsx) will automatically update the webpage and server.
The server is a small fastify server which allows token exchange and verify of JWT tokens.

Requests on server

GET /exchange

Present a SIWE bearer token to exchange it for a server side SIWE token
The public address of the server for the server token
is available at GET /.well-known/public-key or as a jwks at GET /.well-known/jwks

GET /verify

Present SIWE client or server bearer token to get verified.

Learn more

To learn more about Vite (https://vitejs.dev/) or wagmi (https://wagmi.sh), check out the following resources:

- wagmi Documentation (https://wagmi.sh) – learn about wagmi Hooks and API.
- wagmi Examples (https://wagmi.sh/examples/connect-wallet) – a suite of simple examples using wagmi.
- Web3Modal Documentation (https://web3modal.com) – learn more about Web3Modal (configuration, theming, advanced usage, etc).
- Vite Documentation (https://vitejs.dev/) – learn about Vite features and API.)
  </file>

<file>

path: /cloudbuild.yaml

url: https://github.com/richtera/lukso-siwe-demo/blob/main/cloudbuild.yaml

```
steps:
  - name: gcr.io/cloud-builders/docker
    args:
      - build
      - '--build-arg'
      - 'VITE_WALLET_CONNECT_PROJECT_ID=$_VITE_WALLET_CONNECT_PROJECT_ID'
      - '--build-arg'
      - 'VITE_APP_DOMAIN=$_VITE_APP_DOMAIN'
      - '-t'
      - >-
        $_AR_HOSTNAME/$PROJECT_ID/cloud-run-source-deploy/$REPO_NAME/$_SERVICE_NAME:$COMMIT_SHA
      - .
      - '-f'
      - Dockerfile
    id: Build
  - name: gcr.io/cloud-builders/docker
    args:
      - push
      - >-
        $_AR_HOSTNAME/$PROJECT_ID/cloud-run-source-deploy/$REPO_NAME/$_SERVICE_NAME:$COMMIT_SHA
    id: Push
  - name: 'gcr.io/google.com/cloudsdktool/cloud-sdk:slim'
    args:
      - run
      - services
      - update
      - $_SERVICE_NAME
      - '--platform=managed'
      - >-
        --image=$_AR_HOSTNAME/$PROJECT_ID/cloud-run-source-deploy/$REPO_NAME/$_SERVICE_NAME:$COMMIT_SHA
      - >-
        --labels=managed-by=gcp-cloud-build-deploy-cloud-run,commit-sha=$COMMIT_SHA,gcb-build-id=$BUILD_ID,gcb-trigger-id=$_TRIGGER_ID
      - '--region=$_DEPLOY_REGION'
      - '--quiet'
    id: Deploy
    entrypoint: gcloud
images:
  - >-
    $_AR_HOSTNAME/$PROJECT_ID/cloud-run-source-deploy/$REPO_NAME/$_SERVICE_NAME:$COMMIT_SHA
options:
  substitutionOption: ALLOW_LOOSE
substitutions:
  _TRIGGER_ID: ''
  _DEPLOY_REGION: us-central1
  _AR_HOSTNAME: us-central1-docker.pkg.dev
  _PLATFORM: managed
  _SERVICE_NAME: lukso-siwe-demo
  _VITE_WALLET_CONNECT_PROJECT_ID: ''
  _VITE_APP_DOMAIN: siwe-demo.richtera.app

tags:
  - gcp-cloud-build-deploy-cloud-run
  - gcp-cloud-build-deploy-cloud-run-managed
```

</file>

<file>

path: /index.html

url: https://github.com/richtera/lukso-siwe-demo/blob/main/index.html

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Web3 Auth: Using Universal Profile and SIWE for Login</title>
    <link rel="apple-touch-icon" sizes="180x180" href="/apple-touch-icon.png" />
    <link rel="icon" type="image/png" sizes="32x32" href="/favicon-32x32.png" />
    <link rel="icon" type="image/png" sizes="16x16" href="/favicon-16x16.png" />
    <link rel="manifest" href="/site.webmanifest" />
    <link rel="mask-icon" href="/safari-pinned-tab.svg" color="#5bbad5" />
    <meta name="msapplication-TileColor" content="#da532c" />
    <meta name="theme-color" content="#ffffff" />
    <script type="module" src="./polyfills.ts"></script>
  </head>
  <body>
    <div id="root"></div>
    <script type="module" src="/src/main.tsx"></script>
  </body>
</html>
```

</file>

<file>

path: /package.json

url: https://github.com/richtera/lukso-siwe-demo/blob/main/package.json

```
{
  "name": "wagmi-template-vite-with-web3modal",
  "version": "0.0.1",
  "type": "module",
  "private": true,
  "scripts": {
    "dev": "concurrently -k 'nodemon --watch ./server/index.ts --exec \"ts-node-esm --project=./tsconfig.server.json ./server/index.ts\"' 'vite dev'",
    "build": "vite build && tsc --project ./tsconfig.server.json",
    "typechain": "typechain --target ethers-v6 --out-dir libs/contracts './node_modules/@lukso/lsp-smart-contracts/artifacts/*.json'"
  },
  "dependencies": {
    "@fastify/cors": "^8.4.0",
    "@fastify/formbody": "^7.4.0",
    "@fastify/static": "^6.11.2",
    "@lukso/lsp-smart-contracts": "0.12.0-rc.0",
    "@noble/secp256k1": "^2.0.0",
    "@tailwindcss/forms": "^0.5.6",
    "@typechain/ethers-v6": "^0.5.1",
    "@web3modal/ethereum": "2.4.1",
    "@web3modal/react": "2.4.1",
    "autoprefixer": "^10.4.16",
    "base64url": "^3.0.1",
    "buffer": "^6.0.3",
    "dotenv": "^16.3.1",
    "ethers": "^6.8.0",
    "ethers-v6\n": "link:@typechain/ethers-v6\n",
    "fastify": "^4.24.3",
    "jose": "^5.0.1",
    "key-encoder": "^2.0.3",
    "postcss-import": "^15.1.0",
    "postcss-nesting": "^12.0.1",
    "process": "^0.11.10",
    "react": "^18.2.0",
    "react-dom": "^18.2.0",
    "react-json-view": "^1.21.3",
    "siwe": "^2.1.4",
    "tailwindcss": "^3.3.5",
    "util": "^0.12.4",
    "viem": "~0.3.36",
    "wagmi": "^1.0.6"
  },
  "devDependencies": {
    "@types/node": "^20.8.9",
    "@types/react": "^18.0.9",
    "@types/react-dom": "^18.0.3",
    "@typescript-eslint/eslint-plugin": "^6.9.0",
    "@vitejs/plugin-react": "^4.0.0",
    "concurrently": "^8.2.2",
    "eslint": "^8.52.0",
    "eslint-config-next": "^14.0.0",
    "eslint-config-prettier": "^9.0.0",
    "eslint-plugin-prettier": "^5.0.1",
    "eslint-plugin-unicorn": "^48.0.1",
    "nodemon": "^3.0.1",
    "prettier": "^3.0.3",
    "ts-node": "^10.9.1",
    "typechain": "^8.3.2",
    "typescript": "^5.0.4",
    "vite": "^4.3.5",
    "vite-plugin-mkcert": "^1.16.0"
  }
```

```
}
```

</file>

<file>

path: /pnpm-lock.yaml

url: https://github.com/richtera/lukso-siwe-demo/blob/main/pnpm-lock.yaml

```
lockfileVersion: '6.0'

settings:
  autoInstallPeers: true
  excludeLinksFromLockfile: false

dependencies:
  '@fastify/cors':
    specifier: ^8.4.0
    version: 8.4.0
  '@fastify/formbody':
    specifier: ^7.4.0
    version: 7.4.0
  '@fastify/static':
    specifier: ^6.11.2
    version: 6.11.2
  '@lukso/lsp-smart-contracts':
    specifier: 0.12.0-rc.0
    version: 0.12.0-rc.0(@babel/core@7.23.2)
  '@noble/secp256k1':
    specifier: ^2.0.0
    version: 2.0.0
  '@tailwindcss/forms':
    specifier: ^0.5.6
    version: 0.5.6(tailwindcss@3.3.5)
  '@typechain/ethers-v6':
    specifier: ^0.5.1
    version: 0.5.1(ethers@6.8.0)(typechain@8.3.2)(typescript@5.2.2)
  '@web3modal/ethereum':
    specifier: 2.4.1
    version: 2.4.1(@wagmi/core@1.4.5)(viem@0.3.50)
  '@web3modal/react':
    specifier: 2.4.1
    version: 2.4.1(react-dom@18.2.0)(react@18.2.0)
  autoprefixer:
    specifier: ^10.4.16
    version: 10.4.16(postcss@8.4.31)
  base64url:
    specifier: ^3.0.1
    version: 3.0.1
  buffer:
```

```
  specifier: ^6.0.3
  version: 6.0.3
dotenv:
  specifier: ^16.3.1
  version: 16.3.1
ethers:
  specifier: ^6.8.0
  version: 6.8.0
"ethers-v6\n":
  specifier: |
    link:@typechain/ethers-v6
  version: |
    link:@typechain/ethers-v6
fastify:
  specifier: ^4.24.3
  version: 4.24.3
jose:
  specifier: ^5.0.1
  version: 5.0.1
key-encoder:
  specifier: ^2.0.3
  version: 2.0.3
postcss-import:
  specifier: ^15.1.0
  version: 15.1.0(postcss@8.4.31)
postcss-nesting:
  specifier: ^12.0.1
  version: 12.0.1(postcss@8.4.31)
process:
  specifier: ^0.11.10
  version: 0.11.10
react:
  specifier: ^18.2.0
  version: 18.2.0
react-dom:
  specifier: ^18.2.0
  version: 18.2.0(react@18.2.0)
react-json-view:
  specifier: ^1.21.3
  version: 1.21.3(@types/react@18.0.25)(react-dom@18.2.0)(react@18.2.0)
siwe:
  specifier: ^2.1.4
  version: 2.1.4(ethers@6.8.0)
tailwindcss:
  specifier: ^3.3.5
  version: 3.3.5(ts-node@10.9.1)
util:
  specifier: ^0.12.4
  version: 0.12.5
viem:
  specifier: ~0.3.36
  version: 0.3.50(typescript@5.2.2)
  wagmi:
    specifier: ^1.0.6
    version: 1.4.5(@types/react@18.0.25)(react-dom@18.2.0)(react@18.2.0)(typescript@5.2.2)(viem@0.3.50)

devDependencies:
  '@types/node':
    specifier: ^20.8.9
    version: 20.8.9
  '@types/react':
    specifier: ^18.0.9
    version: 18.0.25
  '@types/react-dom':
    specifier: ^18.0.3
    version: 18.0.9
  '@typescript-eslint/eslint-plugin':
    specifier: ^6.9.0
    version: 6.9.0(@typescript-eslint/parser@6.9.0)(eslint@8.52.0)(typescript@5.2.2)
  '@vitejs/plugin-react':
    specifier: ^4.0.0
    version: 4.1.0(vite@4.5.0)
  concurrently:
    specifier: ^8.2.2
    version: 8.2.2
  eslint:
    specifier: ^8.52.0
    version: 8.52.0
  eslint-config-next:
    specifier: ^14.0.0
    version: 14.0.0(eslint@8.52.0)(typescript@5.2.2)
  eslint-config-prettier:
    specifier: ^9.0.0
    version: 9.0.0(eslint@8.52.0)
  eslint-plugin-prettier:
    specifier: ^5.0.1
    version: 5.0.1(eslint-config-prettier@9.0.0)(eslint@8.52.0)(prettier@3.0.3)
  eslint-plugin-unicorn:
    specifier: ^48.0.1
    version: 48.0.1(eslint@8.52.0)
  nodemon:
    specifier: ^3.0.1
    version: 3.0.1
  prettier:
    specifier: ^3.0.3
    version: 3.0.3
  ts-node:
    specifier: ^10.9.1
    version: 10.9.1(@types/node@20.8.9)(typescript@5.2.2)
  typechain:
    specifier: ^8.3.2
    version: 8.3.2(typescript@5.2.2)
  typescript:
      specifier: ^5.0.4
      version: 5.2.2
    vite:
      specifier: ^4.3.5
      version: 4.5.0(@types/node@20.8.9)
    vite-plugin-mkcert:
      specifier: ^1.16.0
      version: 1.16.0(vite@4.5.0)

packages:

  /@aashutoshrathi/word-wrap@1.2.6:
    resolution: {integrity: sha512-1Yjs2SvM8TflER/OD3cOjhWWOZb58A2t7wpE2S9XfBYTiIl+XFhQG2bjy4Pu1I+EAlCNUzRDYDdFwFYUKvXcIA==}
    engines: {node: '>=0.10.0'}
    dev: true

  /@account-abstraction/contracts@0.6.0:
    resolution: {integrity: sha512-8ooRJuR7XzohMDM4MV34I12Ci2bmxfE9+cixakRL7lA4BAwJKQ3ahvd8FbJa9kiwkUPCUNtj+/zxDQWYYalLMQ==}
    dev: false

  /@adraffy/ens-normalize@1.10.0:
    resolution: {integrity: sha512-nA9XHtlAkYfJxY7bce8DcN7eKxWWCWkU+1GR9d+U6MbNpfwQp8TI7vqOsBsMcHoT4mBu2kypKoSKnghEzOOq5Q==}
    dev: false

  /@adraffy/ens-normalize@1.9.0:
    resolution: {integrity: sha512-iowxq3U30sghZotgl4s/oJRci6WPBfNO5YYgk2cIOMCHr3LeGPcsZjCEr+33Q4N+oV3OABDAtA+pyvWjbvBifQ==}
    dev: false

  /@adraffy/ens-normalize@1.9.4:
    resolution: {integrity: sha512-UK0bHA7hh9cR39V+4gl2/NnBBjoXIxkuWAPCaY4X7fbH4L/azIi7ilWOCjMUYfpJgraLUAqkRi2BqrjME8Rynw==}
    dev: false

  /@alloc/quick-lru@5.2.0:
    resolution: {integrity: sha512-UrcABB+4bUrFABwbluTIBErXwvbsU/V7TZWfmbgJfbkwiBuziS9gxdODUyuiecfdGQ85jglMW6juS3+z5TsKLw==}
    engines: {node: '>=10'}
    dev: false

  /@ampproject/remapping@2.2.0:
    resolution: {integrity: sha512-qRmjj8nj9qmLTQXXmaR1cck3UXSRMPrbsLJAasZpF+t3riI71BXed5ebIOYwQntykeZuhjsdweEc9BxH5Jc26
w==}
    engines: {node: '>=6.0.0'}
    dependencies:
      '@jridgewell/gen-mapping': 0.1.1
      '@jridgewell/trace-mapping': 0.3.17

  /@babel/code-frame@7.22.13:
    resolution: {integrity: sha512-
XktuhWlJ5g+3TJXc5upd9Ks1HutSArik6jf2eAjYFyIOf4ej3RN+184cZbzDvbPnuTJIUhPKKJE3cIsYTiAT3w==}
    engines: {node: '>=6.9.0'}
    dependencies:
      '@babel/highlight': 7.22.20
      chalk: 2.4.2

  /@babel/compat-data@7.23.2:
    resolution: {integrity: sha512-
0S9TQMmDHlqAZ2ITT95irXKfxN9bncq8ZCoJhun3nHL/lLUxd2NKBJYoNGWH7S0hz6fRQwWlAWn/ILM0C7
0KZQ==}
    engines: {node: '>=6.9.0'}

  /@babel/core@7.23.2:
    resolution: {integrity: sha512-
n7s51eWdaWZ3vGT2tD4T7J6eJs3QoBXydv7vkUM06Bf1cbVD2Kc2UrkzhiQwobfV7NwOnQXYL7UBJ5VPU
+RGoQ==}
    engines: {node: '>=6.9.0'}
    dependencies:
      '@ampproject/remapping': 2.2.0
      '@babel/code-frame': 7.22.13
      '@babel/generator': 7.23.0
      '@babel/helper-compilation-targets': 7.22.15
      '@babel/helper-module-transforms': 7.23.0(@babel/core@7.23.2)
      '@babel/helpers': 7.23.2
      '@babel/parser': 7.23.0
      '@babel/template': 7.22.15
      '@babel/traverse': 7.23.2
      '@babel/types': 7.23.0
      convert-source-map: 2.0.0
      debug: 4.3.4
      gensync: 1.0.0-beta.2
      json5: 2.2.3
      semver: 6.3.1
    transitivePeerDependencies:
      - supports-color

  /@babel/generator@7.23.0:
    resolution: {integrity: sha512-
lN85QRR+5IbYrMWM6Y4pE/noaQtg4pNiqeNGX60eqOfo6gtEj6uw/JagelB8vVztSd7R6M5n1+PQkDbHbBR
U4g==}
    engines: {node: '>=6.9.0'}
    dependencies:
      '@babel/types': 7.23.0
      '@jridgewell/gen-mapping': 0.3.2
      '@jridgewell/trace-mapping': 0.3.17
      jsesc: 2.5.2

  /@babel/helper-compilation-targets@7.22.15:
    resolution: {integrity: sha512-
y6EEzULok0Qvz8yyLkCvVX+02ic+By2UdOhylwUOvOn9dvYc9mKICJuuU1n1XBI02YWsNsnrY1kc6DVbjcX
btw==}
    engines: {node: '>=6.9.0'}
    dependencies:
      '@babel/compat-data': 7.23.2
      '@babel/helper-validator-option': 7.22.15
      browserslist: 4.22.1
      lru-cache: 5.1.1
      semver: 6.3.1

  /@babel/helper-define-polyfill-provider@0.4.3(@babel/core@7.23.2):
    resolution: {integrity: sha512-
WBrLmuPP47n7PNwsZ57pqam6G/RGo1vw/87b0Blc53tZNGZ4x7YvZ6HgQe2vo1W/FR20OgjeZuGXzudPi
XHFug==}
    peerDependencies:
      '@babel/core': ^7.4.0 || ^8.0.0-0 <8.0.0
    dependencies:
      '@babel/core': 7.23.2
      '@babel/helper-compilation-targets': 7.22.15
      '@babel/helper-plugin-utils': 7.22.5
      debug: 4.3.4
      lodash.debounce: 4.0.8
      resolve: 1.22.8
    transitivePeerDependencies:
      - supports-color
    dev: false

  /@babel/helper-environment-visitor@7.22.20:
    resolution: {integrity: sha512-
zfedSIzFhat/gFhWfHtgWvlec0nqB9YEIVrpuwjruLlXfUSnA8cJB0miHKwqDnQ7d32aKo2xt88/xZptwxbfhA==}
    engines: {node: '>=6.9.0'}

  /@babel/helper-function-name@7.23.0:
    resolution: {integrity: sha512-
OErEqsrxjZTJciZ4Oo+eoZqeW9UIiOcuYKRJA4ZAgV9myA+pOXhhmpfNCKjEH/auVfEYVFJ6y1Tc4r0eIApqi
w==}
    engines: {node: '>=6.9.0'}
    dependencies:
      '@babel/template': 7.22.15
      '@babel/types': 7.23.0

  /@babel/helper-hoist-variables@7.22.5:
    resolution: {integrity: sha512-
wGjk9QZVzvknA6yKIUURb8zY3grXCcOZt+/7Wcy8O2uctxhplmUPkOdlgoNhmdVee2c92JXbf1xpMtVNbfox
Rw==}
    engines: {node: '>=6.9.0'}
    dependencies:
```

```
    '@babel/types': 7.23.0

  /@babel/helper-module-imports@7.22.15:
    resolution: {integrity: sha512-
0pYVBnDKZO2fnSPCrgM/6WMc7eS20Fbok+0r88fp+YtWVLZrp4CkafFGIp+W0VKw4a22sgebPT99y+FDN
MdP4w==}
    engines: {node: '>=6.9.0'}
    dependencies:
      '@babel/types': 7.23.0

  /@babel/helper-module-transforms@7.23.0(@babel/core@7.23.2):
    resolution: {integrity: sha512-
WhDWw1tdrlT0gMgUJSlX0IQvoO1eN279zrAUbVB+KpV2c3Tylz8+GnKOLllCS6Z/iZQEyVYxhZVUdPTqs2Y
YPw==}
    engines: {node: '>=6.9.0'}
    peerDependencies:
      '@babel/core': ^7.0.0
    dependencies:
      '@babel/core': 7.23.2
      '@babel/helper-environment-visitor': 7.22.20
      '@babel/helper-module-imports': 7.22.15
      '@babel/helper-simple-access': 7.22.5
      '@babel/helper-split-export-declaration': 7.22.6
      '@babel/helper-validator-identifier': 7.22.20

  /@babel/helper-plugin-utils@7.22.5:
    resolution: {integrity: sha512-
uLls06UVKgFG9QD4OeFYLEGteMIAa5kpTPcFL28yuCIIzsf6ZyKZMllKVOCZFhiZ5ptnwX4mtKdWCBE/uT4
amg==}
    engines: {node: '>=6.9.0'}

  /@babel/helper-simple-access@7.22.5:
    resolution: {integrity: sha512-
n0H99E/K+Bika3++WNL17POvo4rKWZ7lZEp1Q+fStVbUi8nxPQEBOlTmCOxW/0JsS56SKKQ+ojAe2pHKJ
HN35w==}
    engines: {node: '>=6.9.0'}
    dependencies:
      '@babel/types': 7.23.0

  /@babel/helper-split-export-declaration@7.22.6:
    resolution: {integrity: sha512-
AsUnxuLhRYsisFiaJwvp1QF+I3KjD5FOxut14q/GzovUe6orHLesW2C7d754kRm53h5gqrz6sFl6sxc4BVtE/g==
}
    engines: {node: '>=6.9.0'}
    dependencies:
      '@babel/types': 7.23.0

  /@babel/helper-string-parser@7.22.5:
    resolution: {integrity: sha512-
mM4COjgZox8U+JcXQwPijIZLElkgEpO5rsERVDJTc2qfCDfERyob6k5WegS14SX18IIjv+XD+GrqNumY5JR
CDw==}
    engines: {node: '>=6.9.0'}
  /@babel/helper-validator-identifier@7.22.20:
    resolution: {integrity: sha512-Y4OZ+ytlatR8AI+8KZfKuL5urKp7qey08ha31L8b3BwewJAoJamTzyvxPR/5D+KkdJCGPq/+8TukHBlY10FX9A==}
    engines: {node: '>=6.9.0'}

  /@babel/helper-validator-option@7.22.15:
    resolution: {integrity: sha512-bMn7RmyFjY/mdECUbgn9eoSY4vqvacUnS9i9vGAGttgFWesO6B4CYWA7XlpbWgBt71iv/hfbPlynohStqnu5hA==}
    engines: {node: '>=6.9.0'}

  /@babel/helpers@7.23.2:
    resolution: {integrity: sha512-Izchcp8SjTSVe/fPmLwtWVBFC7+Tbn8LGHDVfDp9JGxpAY5opSaEFgt8UQvrnECWOTdji2mOWMz1rOhkHscmGQ==}
    engines: {node: '>=6.9.0'}
    dependencies:
      '@babel/template': 7.22.15
      '@babel/traverse': 7.23.2
      '@babel/types': 7.23.0
    transitivePeerDependencies:
      - supports-color

  /@babel/highlight@7.22.20:
    resolution: {integrity: sha512-dkdMCN3py0+ksCgYmGG8jKeGA/8Tk+gJwSYYlFGxG5lmhfKNoAy004YpLxpS1W2J8m/EK2Ew+yOs9pVRwO89mg==}
    engines: {node: '>=6.9.0'}
    dependencies:
      '@babel/helper-validator-identifier': 7.22.20
      chalk: 2.4.2
      js-tokens: 4.0.0

  /@babel/parser@7.23.0:
    resolution: {integrity: sha512-vvPKKdMemU85V9WE/l5wZEmImpCtLqbnTvqDS2U1fJ96KrxoW7KrXhNsNCblQlg8Ck4b85yxdTyelsMUgFUXiw==}
    engines: {node: '>=6.0.0'}
    hasBin: true
    dependencies:
      '@babel/types': 7.23.0

  /@babel/plugin-transform-react-jsx-self@7.22.5(@babel/core@7.23.2):
    resolution: {integrity: sha512-nTh2ogNUtxbiSbxaT4Ds6aXnXEipHweN9YRgOX/oNXdf0cCrGn/+2LozFa3lnPV5D90MkjhgckCPBrsoSc1a7g==}
    engines: {node: '>=6.9.0'}
    peerDependencies:
      '@babel/core': ^7.0.0-0
    dependencies:
      '@babel/core': 7.23.2
      '@babel/helper-plugin-utils': 7.22.5
    dev: true

  /@babel/plugin-transform-react-jsx-source@7.22.5(@babel/core@7.23.2):
    resolution: {integrity: sha512-
yIiRO6yobeEIaI0RTbIr8iAK9FcBHLtZq0S89ZPjDLQXBA4xvghaKqI0etp/tF3htTM0sazJKKLz9oEiGRtu7w==}
    engines: {node: '>=6.9.0'}
    peerDependencies:
      '@babel/core': ^7.0.0-0
    dependencies:
      '@babel/core': 7.23.2
      '@babel/helper-plugin-utils': 7.22.5
    dev: true

  /@babel/plugin-transform-runtime@7.23.2(@babel/core@7.23.2):
    resolution: {integrity: sha512-
XOntj6icgzMS58jPVtQpiuF6ZFWxQiJavISGx5KGjRj+3gqZr8+N6Kx+N9BApWzgS+DOjIZfXXj0ZesenOWDy
A==}
    engines: {node: '>=6.9.0'}
    peerDependencies:
      '@babel/core': ^7.0.0-0
    dependencies:
      '@babel/core': 7.23.2
      '@babel/helper-module-imports': 7.22.15
      '@babel/helper-plugin-utils': 7.22.5
      babel-plugin-polyfill-corejs2: 0.4.6(@babel/core@7.23.2)
      babel-plugin-polyfill-corejs3: 0.8.6(@babel/core@7.23.2)
      babel-plugin-polyfill-regenerator: 0.5.3(@babel/core@7.23.2)
      semver: 6.3.1
    transitivePeerDependencies:
      - supports-color
    dev: false

  /@babel/runtime@7.23.2:
    resolution: {integrity: sha512-
mM8eg4yl5D6i3lu2QKPuPH4FArvJ8KhTofbE7jwMUv9KX5mBvwPAqnV3MlyBNqdp9RyRKP6Yck8TrfYrPvX
3bg==}
    engines: {node: '>=6.9.0'}
    dependencies:
      regenerator-runtime: 0.14.0

  /@babel/template@7.22.15:
    resolution: {integrity: sha512-
QPErUVm4uyJa60rkI73qneDacvdvzxshT3kksGqlGWYdOTIUOwJ7RDUL8sGqslY1uXWSL6xMFKEXDS3ox
2uF0w==}
    engines: {node: '>=6.9.0'}
    dependencies:
      '@babel/code-frame': 7.22.13
      '@babel/parser': 7.23.0
      '@babel/types': 7.23.0
```

```
 /@babel/traverse@7.23.2:
   resolution: {integrity: sha512-
azpe59SQ48qG6nu2CzcMLbxUudtN+dOM9kDbUqGq3HXUJRlo7i8fvPoxQUzYgLZ4cMVmuZgm8vvBpNeR
hd6XSw==}
   engines: {node: '>=6.9.0'}
   dependencies:
     '@babel/code-frame': 7.22.13
     '@babel/generator': 7.23.0
     '@babel/helper-environment-visitor': 7.22.20
     '@babel/helper-function-name': 7.23.0
     '@babel/helper-hoist-variables': 7.22.5
     '@babel/helper-split-export-declaration': 7.22.6
     '@babel/parser': 7.23.0
     '@babel/types': 7.23.0
     debug: 4.3.4
     globals: 11.12.0
   transitivePeerDependencies:
     - supports-color

 /@babel/types@7.23.0:
   resolution: {integrity: sha512-
0oIyUfKoI3mSqMvsxBdclDwxXKXAUA8v/apZbc+iSyARYou1o8ZGDxbUYyLFoW2arqS2jDGqJuZvv1d/io1ax
g==}
   engines: {node: '>=6.9.0'}
   dependencies:
     '@babel/helper-string-parser': 7.22.5
     '@babel/helper-validator-identifier': 7.22.20
     to-fast-properties: 2.0.0

 /@coinbase/wallet-sdk@3.7.2:
   resolution: {integrity: sha512-
lIGvXMsgpsQWci/XOMQIJ2nIZ8JUy/L+bvC0wkRaYarr0YylwpXrJ2gRM3hCXPS477pkyO7N/kSiAoRgEXUd
JQ==}
   engines: {node: '>= 10.0.0'}
   dependencies:
     '@metamask/safe-event-emitter': 2.0.0
     '@solana/web3.js': 1.87.3
     bind-decorator: 1.0.11
     bn.js: 5.2.1
     buffer: 6.0.3
     clsx: 1.2.1
     eth-block-tracker: 6.1.0
     eth-json-rpc-filters: 5.1.0
     eth-rpc-errors: 4.0.2
     json-rpc-engine: 6.1.0
     keccak: 3.0.2
     preact: 10.11.3
     qs: 6.11.0
     rxjs: 6.6.7
     sha.js: 2.4.11
     stream-browserify: 3.0.0
     util: 0.12.5
```

```
    transitivePeerDependencies:
      - bufferutil
      - encoding
      - supports-color
      - utf-8-validate
    dev: false

  /@cspotcode/source-map-support@0.8.1:
    resolution: {integrity: sha512-
IchNf6dN4tHoMFIn/7OE8LWZ19Y6q/67Bmf6vnGREv8RSbBVb9LPJxEcnwrcwX6ixSvaiGoomAUvu4YSxXr
Vgw==}
    engines: {node: '>=12'}
    dependencies:
      '@jridgewell/trace-mapping': 0.3.9

  /@csstools/selector-specificity@3.0.0(postcss-selector-parser@6.0.13):
    resolution: {integrity: sha512-
hBI9tfBtuPIi885ZsZ32IMEU/5nlZH/KOVYJCOh7gyMxaVLGmLedYqFN6Ui1LXkI8JlC8IsuC0rF0btcRZKd5g==
}
    engines: {node: ^14 || ^16 || >=18}
    peerDependencies:
      postcss-selector-parser: ^6.0.13
    dependencies:
      postcss-selector-parser: 6.0.13
    dev: false

  /@erc725/smart-contracts@6.0.0(@babel/core@7.23.2):
    resolution: {integrity: sha512-
6okutGGL9xbg/MSgAof2FU1UcSNE/z3p9TORlROVGaM3gi1A6FQQ7fDqtBYkPtvHureX8yS9gP7xPt3PRbP
43Q==}
    dependencies:
      '@openzeppelin/contracts': 4.9.3
      '@openzeppelin/contracts-upgradeable': 4.9.3
      solidity-bytes-utils: 0.8.0(@babel/core@7.23.2)
    transitivePeerDependencies:
      - '@babel/core'
      - bufferutil
      - encoding
      - supports-color
      - utf-8-validate
    dev: false

  /@esbuild/android-arm64@0.18.20:
    resolution: {integrity: sha512-
Nz4rJcchGDtENV0eMKUNa6L12zz2zBDXuhj/Vjh18zGqB44Bi7MBMSXjgunJgjRhCmKOjnPuZp4Mb6OKqt
MHLQ==}
    engines: {node: '>=12'}
    cpu: [arm64]
    os: [android]
    requiresBuild: true
    dev: true
    optional: true
  /@esbuild/android-arm@0.18.20:
    resolution: {integrity: sha512-fyi7TDI/ijKKNZTUJAQqiG5T7YjJXgnzkURqmGj13C6dCqckZBLdl4h7bkhHt/t0WP+zO9/zwroDvANaOqO5Sw==}
    engines: {node: '>=12'}
    cpu: [arm]
    os: [android]
    requiresBuild: true
    dev: true
    optional: true

  /@esbuild/android-x64@0.18.20:
    resolution: {integrity: sha512-8GDdlePJA8D6zlZYJV/jnrRAi6rOiNaCC/JclcXpB+KIuvfBN4owLtgzY2bsxnx666XjJx2kDPUmnTtR8qKQUg==}
    engines: {node: '>=12'}
    cpu: [x64]
    os: [android]
    requiresBuild: true
    dev: true
    optional: true

  /@esbuild/darwin-arm64@0.18.20:
    resolution: {integrity: sha512-bxRHW5kHU38zS2lPTPOyuyTm+S+eobPUnTNkdJEfAddYgEcll4xkT8DB9d2008DtTbl7uJag2HuE5NZAZgnNEA==}
    engines: {node: '>=12'}
    cpu: [arm64]
    os: [darwin]
    requiresBuild: true
    dev: true
    optional: true

  /@esbuild/darwin-x64@0.18.20:
    resolution: {integrity: sha512-pc5gxlMDxzm513qPGbCbDukOdsGtKhfxD1zJKXjCCcU7ju50O7MeAZ8c4krSJcOIJGFR+qx21yMMVYwiQvyTyQ==}
    engines: {node: '>=12'}
    cpu: [x64]
    os: [darwin]
    requiresBuild: true
    dev: true
    optional: true

  /@esbuild/freebsd-arm64@0.18.20:
    resolution: {integrity: sha512-yqDQHy4QHevpMAaxhhIwYPMv1NECwOvIpGCZkECn8w2WFHXjEwrBn3CeNIYsibZ/iZEUemj++M26W3cNR5h+Tw==}
    engines: {node: '>=12'}
    cpu: [arm64]
    os: [freebsd]
    requiresBuild: true
    dev: true
    optional: true

  /@esbuild/freebsd-x64@0.18.20:
    resolution: {integrity: sha512-
tgWRPPuQsd3RmBZwarGVHZQvtzfEBOreNuxEMKFcd5DaDn2PbBxfwLcj4+aenoh7ctXcbXmOQIn8HI6mC
Sw5MQ==}
    engines: {node: '>=12'}
    cpu: [x64]
    os: [freebsd]
    requiresBuild: true
    dev: true
    optional: true

  /@esbuild/linux-arm64@0.18.20:
    resolution: {integrity: sha512-
2YbscF+UL7SQAVIpnWvYwM+3LskyDmPhe31pE7/aoTMFKKzIc9lLbyGUpmmb8a8AixOL61sQ/mFh3jEjHY
FvdA==}
    engines: {node: '>=12'}
    cpu: [arm64]
    os: [linux]
    requiresBuild: true
    dev: true
    optional: true

  /@esbuild/linux-arm@0.18.20:
    resolution: {integrity: sha512-
/5bHkMWnq1EgKr1V+Ybz3s1hWXok7mDFUMQ4cG10AfW3wL02PSZi5kFpYKrptDsgb2WAJIvRcDm+qIvXf
/apvg==}
    engines: {node: '>=12'}
    cpu: [arm]
    os: [linux]
    requiresBuild: true
    dev: true
    optional: true

  /@esbuild/linux-ia32@0.18.20:
    resolution: {integrity: sha512-
P4etWwq6IsReT0E1KHU40bOnzMHoH73aXp96Fs8TIT6z9Hu8G6+0SHSw9i2isWrD2nbx2qo5yUqACgdfV
Gx7TA==}
    engines: {node: '>=12'}
    cpu: [ia32]
    os: [linux]
    requiresBuild: true
    dev: true
    optional: true

  /@esbuild/linux-loong64@0.18.20:
    resolution: {integrity: sha512-
nXW8nqBTrOpDLPgPY9uV+/1DjxoQ7DoB2N8eocyq8I9XuqJ7BiAMDMf9n1xZM9TgW0J8zrquIb/A7s3BJv7r
jg==}
```

```
    engines: {node: '>=12'}
    cpu: [loong64]
    os: [linux]
    requiresBuild: true
    dev: true
    optional: true

  /@esbuild/linux-mips64el@0.18.20:
    resolution: {integrity: sha512-d5NeaXZcHp8PzYy5VnXV3VSd2D328Zb+9dEq5HE6bw6+N86JVPExrA6O68OPwobntbNJ0pzCpUFZTo3w0GyetQ==}
    engines: {node: '>=12'}
    cpu: [mips64el]
    os: [linux]
    requiresBuild: true
    dev: true
    optional: true

  /@esbuild/linux-ppc64@0.18.20:
    resolution: {integrity: sha512-WHPyeScRNcmANnLQkq6AfyXRFr5D6N2sKgkFo2FqguP44Nw2eyDlbTdZwd9GYk98DZG9QItIiTlFLHJHjxP3FA==}
    engines: {node: '>=12'}
    cpu: [ppc64]
    os: [linux]
    requiresBuild: true
    dev: true
    optional: true

  /@esbuild/linux-riscv64@0.18.20:
    resolution: {integrity: sha512-WSxo6h5ecI5XH34KC7w5veNnKkju3zBRLEQNY7mv5mtBmrP/MjNBCAlsM2u5hDBlS3NGcTQpoBvRzqBcRtpq1A==}
    engines: {node: '>=12'}
    cpu: [riscv64]
    os: [linux]
    requiresBuild: true
    dev: true
    optional: true

  /@esbuild/linux-s390x@0.18.20:
    resolution: {integrity: sha512-+8231GMs3mAEth6Ja1iK0a1sQ3ohfcpzpRLH8uuc5/KVDFneH6jtAJLFGafpzpMRO6DzJ6AvXKze9LfFMrIHVQ==}
    engines: {node: '>=12'}
    cpu: [s390x]
    os: [linux]
    requiresBuild: true
    dev: true
    optional: true

  /@esbuild/linux-x64@0.18.20:
    resolution: {integrity: sha512-
UYqiqemphJcNsFEskc73jQ7B9jgwjWrSayxawS6UVFZGWrAAtkzjxSqnoclCXxWtfwLdzU+vTpcNYhpn43uP
1w==}
    engines: {node: '>=12'}
    cpu: [x64]
    os: [linux]
    requiresBuild: true
    dev: true
    optional: true

  /@esbuild/netbsd-x64@0.18.20:
    resolution: {integrity: sha512-
iO1c++VP6xUBUmltHZoMtCUdPlnPGdBom6IrO4gyKPFFVBKioIlmVooR5I83nTew5UOYrk3gIJhbZh8X44y0
6A==}
    engines: {node: '>=12'}
    cpu: [x64]
    os: [netbsd]
    requiresBuild: true
    dev: true
    optional: true

  /@esbuild/openbsd-x64@0.18.20:
    resolution: {integrity: sha512-
e5e4YSsuQfX4cxcygw/UCPIEP6wbIL+se3sxPdCiMbFLBWu0eiZOJ7WoD+ptCLrmjZBK1Wk7I6D/I3NglUGO
xg==}
    engines: {node: '>=12'}
    cpu: [x64]
    os: [openbsd]
    requiresBuild: true
    dev: true
    optional: true

  /@esbuild/sunos-x64@0.18.20:
    resolution: {integrity: sha512-
kDbFRFp0YpTQVVrqUd5FTYmWo45zGaXe0X8E1G/LKFC0v8x0vWrhOWSLITcCn63lmZIxfOMXtCfti/RxN/
0wnQ==}
    engines: {node: '>=12'}
    cpu: [x64]
    os: [sunos]
    requiresBuild: true
    dev: true
    optional: true

  /@esbuild/win32-arm64@0.18.20:
    resolution: {integrity: sha512-
ddYFR6ItYgoaq4v4JmQQaAI5s7npztfV4Ag6NrhiaW0RrnOXqBkgwZLofVTlq1daVTQNhtI5oieTvkRPfZrePg==
}
    engines: {node: '>=12'}
    cpu: [arm64]
    os: [win32]
    requiresBuild: true
    dev: true
    optional: true

  /@esbuild/win32-ia32@0.18.20:
    resolution: {integrity: sha512-Wv7QBi3ID/rROT08SABTS7eV4hX26sVduqDOTe1MvGMjNd3EjOz4b7zeexIR62GTIEKrfJXKL9LFxTYgkyeu7g==}
    engines: {node: '>=12'}
    cpu: [ia32]
    os: [win32]
    requiresBuild: true
    dev: true
    optional: true

  /@esbuild/win32-x64@0.18.20:
    resolution: {integrity: sha512-kTdfRcSiDfQca/y9QIkng02avJ+NCaQvrMejlsB3RRv5sE9rRoeBPISaZpKxHELzRxZyLvNts1P27W3wV+8geQ==}
    engines: {node: '>=12'}
    cpu: [x64]
    os: [win32]
    requiresBuild: true
    dev: true
    optional: true

  /@eslint-community/eslint-utils@4.4.0(eslint@8.52.0):
    resolution: {integrity: sha512-1/sA4dwrzBAyeUoQ6oxahHKmrZvsnLCg4RfxW3ZFGGmQkSNQPFNLV9CUEFQP1x9EYXHTo5p6xdhZM1Ne9p/AfA==}
    engines: {node: ^12.22.0 || ^14.17.0 || >=16.0.0}
    peerDependencies:
      eslint: ^6.0.0 || ^7.0.0 || >=8.0.0
    dependencies:
      eslint: 8.52.0
      eslint-visitor-keys: 3.4.3
    dev: true

  /@eslint-community/regexpp@4.10.0:
    resolution: {integrity: sha512-Cu96Sd2By9mCNTx2iyKOmq10v22jUVQv0lQnlGNy16oE9589yE+QADPbrMGCkA51cKZSg3Pu/aTJVTGfL/qjUA==}
    engines: {node: ^12.0.0 || ^14.0.0 || >=16.0.0}
    dev: true

  /@eslint/eslintrc@2.1.2:
    resolution: {integrity: sha512-+wvgpDsrB1YqAMdEUCcnTlpfVBH7Vqn6A/NT3D8WVXFIaKMlErPIZT3oCIAVCOtarRpMtelZLqJeU3t7WY6X6g==}
    engines: {node: ^12.22.0 || ^14.17.0 || >=16.0.0}
    dependencies:
      ajv: 6.12.6
      debug: 4.3.4
      espree: 9.6.1
      globals: 13.23.0
      ignore: 5.2.4
      import-fresh: 3.3.0
      js-yaml: 4.1.0
      minimatch: 3.1.2
      strip-json-comments: 3.1.1
    transitivePeerDependencies:
      - supports-color
    dev: true

  /@eslint/js@8.52.0:
    resolution: {integrity: sha512-
mjZVbpaeMZludF2fsWLD0Z9gCref1Tk4i9+wddjRvpUNqqcndPkBD09N/Mapey0b3jaXbLm2kICwFv2E64Qin
A==}
    engines: {node: ^12.22.0 || ^14.17.0 || >=16.0.0}
    dev: true

  /@ethereumjs/common@2.5.0:
    resolution: {integrity: sha512-
DEHjW6e38o+JmB/NO3GZBpW4lpaiBpkFgXF6jLcJ6gETBYpEyaA5nTimsWBUJR3Vmtm/didUEbNjajskugZ
ORg==}
    dependencies:
      crc-32: 1.2.2
      ethereumjs-util: 7.1.5
    dev: false

  /@ethereumjs/common@2.6.5:
    resolution: {integrity: sha512-
lRyVQOeCDaIVtgfbowla32pzeDv2Obr8oR8Put5RdUBNRGr1VGPGQNGP6elWIpgK3YdpzqTOh4GyUGOur
eVeeA==}
    dependencies:
      crc-32: 1.2.2
      ethereumjs-util: 7.1.5
    dev: false

  /@ethereumjs/tx@3.3.2:
    resolution: {integrity: sha512-
6AaJhwg4ucmwTvw/1qLaZUX5miWrwZ4nLOUsKyb/HtzS3BMw/CasKhdi1ims9mBKeK9sOJCH4qGKOBGy
JCeeog==}
    dependencies:
      '@ethereumjs/common': 2.6.5
      ethereumjs-util: 7.1.5
    dev: false

  /@ethereumjs/tx@3.5.2:
    resolution: {integrity: sha512-
gQDNJWKrSDGu2w7w0PzVXVBNMzb7wwdDOmOqczmhNjqFxFuIbhVJDwiGEnxFNC2/b8ifcZzY7MLcluizo
hRzNw==}
    dependencies:
      '@ethereumjs/common': 2.6.5
      ethereumjs-util: 7.1.5
    dev: false
  /@ethersproject/abi@5.7.0:
    resolution: {integrity: sha512-
351ktp42TiRcYB3H1OP8yajPeAQstMW/yCFokj/AthP9bLHzQFPlOrxOcwYEDkUAICmOHljvN4K39OMTMU
a9RA==}
    dependencies:
      '@ethersproject/address': 5.7.0
      '@ethersproject/bignumber': 5.7.0
      '@ethersproject/bytes': 5.7.0
      '@ethersproject/constants': 5.7.0
      '@ethersproject/hash': 5.7.0
      '@ethersproject/keccak256': 5.7.0
      '@ethersproject/logger': 5.7.0
      '@ethersproject/properties': 5.7.0
      '@ethersproject/strings': 5.7.0
    dev: false

  /@ethersproject/abstract-provider@5.7.0:
    resolution: {integrity: sha512-
R41c9UkchKCpAqStMYUpdunjo3pkEvZC3FAwZn5S5MGbXoMQOHIdHItezTETxAO5bevtMApSyEhn9+CH
cDsWBw==}
    dependencies:
      '@ethersproject/bignumber': 5.7.0
      '@ethersproject/bytes': 5.7.0
      '@ethersproject/logger': 5.7.0
      '@ethersproject/networks': 5.7.1
      '@ethersproject/properties': 5.7.0
      '@ethersproject/transactions': 5.7.0
      '@ethersproject/web': 5.7.1
    dev: false

  /@ethersproject/abstract-signer@5.7.0:
    resolution: {integrity: sha512-
a16V8bq1/Cz+TGCkE2OPMTOUDLS3grCpdjoJCYNnVBbdYEMSgKrU0+B90s8b6H+ByYTBZN7a3g76jdIJi
7UfKQ==}
    dependencies:
      '@ethersproject/abstract-provider': 5.7.0
      '@ethersproject/bignumber': 5.7.0
      '@ethersproject/bytes': 5.7.0
      '@ethersproject/logger': 5.7.0
      '@ethersproject/properties': 5.7.0
    dev: false

  /@ethersproject/address@5.7.0:
    resolution: {integrity: sha512-
9wYhYt7aghVGo758POM5nqcOMaE168Q6aRLJZwUmiqSrAungkG74gSSeKEIR7ukixesdRZGPgVqme6vm
xs1fkA==}
    dependencies:
      '@ethersproject/bignumber': 5.7.0
      '@ethersproject/bytes': 5.7.0
      '@ethersproject/keccak256': 5.7.0
      '@ethersproject/logger': 5.7.0
      '@ethersproject/rlp': 5.7.0
    dev: false

  /@ethersproject/base64@5.7.0:
    resolution: {integrity: sha512-Dr8tcHt2mEbsZr/mwTPIQAf3Ai0Bks/7gTw9dSqk1mQvhW3XvRlmDJr/4n+wg1JmCl16NZue17CDh8xb/vZ0sQ==}
    dependencies:
      '@ethersproject/bytes': 5.7.0
    dev: false

  /@ethersproject/bignumber@5.7.0:
    resolution: {integrity: sha512-n1CAdIHRWjSucQO3MC1zPSVgV/6dy/fjL9pMrPP9peL+QxEg9wOsVqwD4+818B6LUEtaXzVHQiuivzRoxPxUGw==}
    dependencies:
      '@ethersproject/bytes': 5.7.0
      '@ethersproject/logger': 5.7.0
      bn.js: 5.2.1
    dev: false

  /@ethersproject/bytes@5.7.0:
    resolution: {integrity: sha512-nsbxwgFXWh9NyYWo+U8atvmMsSdKJprTcICAkvbBffT75qDocbuggBU0SJiVK2MuTrp0q+xvLkTnGMPK1+uA9A==}
    dependencies:
      '@ethersproject/logger': 5.7.0
    dev: false

  /@ethersproject/constants@5.7.0:
    resolution: {integrity: sha512-DHI+y5dBNvkpYUMiRQyxRBYBefZkJfo70VUkUAsRjcPs47muV9evftfZ0PJVCXYbAiCght0DtcF9srFQmIgWA==}
    dependencies:
      '@ethersproject/bignumber': 5.7.0
    dev: false

  /@ethersproject/hash@5.7.0:
    resolution: {integrity: sha512-qX5WrQfnah1EFnO5zJv1v46a8HW0+E5xuBBDTwMFZLuVTx0tbU2kkx15NqdjxecrLGatQN9FGQKpb1FKdHCt+g==}
    dependencies:
      '@ethersproject/abstract-signer': 5.7.0
      '@ethersproject/address': 5.7.0
      '@ethersproject/base64': 5.7.0
      '@ethersproject/bignumber': 5.7.0
      '@ethersproject/bytes': 5.7.0
      '@ethersproject/keccak256': 5.7.0
      '@ethersproject/logger': 5.7.0
      '@ethersproject/properties': 5.7.0
      '@ethersproject/strings': 5.7.0
    dev: false
  /@ethersproject/keccak256@5.7.0:
    resolution: {integrity: sha512-
2UcPboeL/iW+pSg6vZ6ydF8tCnv3Iu/8tUmLLzWWGzxWKFFqOBQFLo6uLUv6BDrLgCDfN28RJ/wtByx+jZ4
KBg==}
    dependencies:
      '@ethersproject/bytes': 5.7.0
      js-sha3: 0.8.0
    dev: false

  /@ethersproject/logger@5.7.0:
    resolution: {integrity: sha512-
0odtFdXu/XHtjQXJYA3u9G0G8btm0ND5Cu8M7i5vhEcE8/HmF4LbdqanwyvE4uQTr2tx6b7fQRmgLrsnpQlmni
g==}
    dev: false

  /@ethersproject/networks@5.7.1:
    resolution: {integrity: sha512-
n/MufjFYv3yFcUyfhnXotyDlNdFb7onmkSy8aQERi2PjNcnWQ66xXxa3XlS8nCcA8aJKJjlIMNJTC7tu80Gwp
Q==}
    dependencies:
      '@ethersproject/logger': 5.7.0
    dev: false

  /@ethersproject/properties@5.7.0:
    resolution: {integrity: sha512-
J87jy8suntrAkIZtecpxEPxY//szqr1mlBaYlQ0r4RCaiD2hjheqF9s1LVE8vVuJCXisjIP+JgtK/Do54ej4Sw==}
    dependencies:
      '@ethersproject/logger': 5.7.0
    dev: false

  /@ethersproject/rlp@5.7.0:
    resolution: {integrity: sha512-
rBxzX2vK8mVF7b0Tol44t5Tb8gomOHkj5guL+HhzQ1yBh/ydjGnpw6at+X6Iw0Kp3OzzzkcKp8N9r0W4kYSs9
w==}
    dependencies:
      '@ethersproject/bytes': 5.7.0
      '@ethersproject/logger': 5.7.0
    dev: false

  /@ethersproject/signing-key@5.7.0:
    resolution: {integrity: sha512-
MZdy2nL3wO0u7gkB4nA/pEf8lu1TlFswPNmy8AiYkfKTdO6eXBJyUdmHO/ehm/htHw9K/qF8ujnTyUAD+Ry5
4Q==}
    dependencies:
      '@ethersproject/bytes': 5.7.0
      '@ethersproject/logger': 5.7.0
      '@ethersproject/properties': 5.7.0
      bn.js: 5.2.1
      elliptic: 6.5.4
      hash.js: 1.1.7
    dev: false
  /@ethersproject/strings@5.7.0:
    resolution: {integrity: sha512-
/9nu+lj0YswRNSH0NXYqrh8775XNyEdUQAuf3f+SmOrnVewcJ5SBNAjF7lpgehKi4abvNNXyf+HX86czCdJ8
Mg==}
    dependencies:
      '@ethersproject/bytes': 5.7.0
      '@ethersproject/constants': 5.7.0
      '@ethersproject/logger': 5.7.0
    dev: false

  /@ethersproject/transactions@5.7.0:
    resolution: {integrity: sha512-
kmcNicCp1lp8qanMTC3RIikGgoJ80ztTyvtsFvCYpSCfkjhD0jZ2LOrnbcuxuToLIUYYf+4XwD1rP+B/erDIhQ==
}
    dependencies:
      '@ethersproject/address': 5.7.0
      '@ethersproject/bignumber': 5.7.0
      '@ethersproject/bytes': 5.7.0
      '@ethersproject/constants': 5.7.0
      '@ethersproject/keccak256': 5.7.0
      '@ethersproject/logger': 5.7.0
      '@ethersproject/properties': 5.7.0
      '@ethersproject/rlp': 5.7.0
      '@ethersproject/signing-key': 5.7.0
    dev: false

  /@ethersproject/web@5.7.1:
    resolution: {integrity: sha512-
Gueu8lSvyjBWL4cYsWsjh6MtMwM0+H4HvqFPZfB6dV8ctbP9zFAO73VG1cMWae0FLPCtz0peKPpZY8/ugJ
JX2w==}
    dependencies:
      '@ethersproject/base64': 5.7.0
      '@ethersproject/bytes': 5.7.0
      '@ethersproject/logger': 5.7.0
      '@ethersproject/properties': 5.7.0
      '@ethersproject/strings': 5.7.0
    dev: false

  /@fastify/accept-negotiator@1.1.0:
    resolution: {integrity: sha512-
OIHZrb2ImZ7XG85HXOONLcJWGosv7sIvM2ifAPQVhg9Lv7qdmMBNVaai4QTdyuaqbKM5eO6sLSQOYI7w
EQeCJQ==}
    engines: {node: '>=14'}
    dev: false

  /@fastify/ajv-compiler@3.5.0:
    resolution: {integrity: sha512-
ebbEtlI7dxXF5ziNdr05mOY8NnDiPB1XvAlLHctRt/Rc+C3LCOVW5imUVX+mhvUhnNzmPBHewUkOFgGlCx
gdAA==}
    dependencies:
      ajv: 8.12.0
      ajv-formats: 2.1.1(ajv@8.12.0)
      fast-uri: 2.2.0
    dev: false

  /@fastify/cors@8.4.0:
    resolution: {integrity: sha512-
MIVvMTenltToByTpLwlWtO+7dQ3l2J+1OpmGrx9JpSNWo1d+dhfNCOi23zHhxdFhtpDzfwGwCsKu9DTeG7k
7nQ==}
    dependencies:
      fastify-plugin: 4.5.1
      mnemonist: 0.39.5
    dev: false

  /@fastify/deepmerge@1.3.0:
    resolution: {integrity: sha512-
J8TOSBq3SoZbDhM9+R/u77hP93gz/rajSA+K2kGyijPpORPWUXHUpTaleoj+92As0S9uPRP7Oi8IqMf0u+ro
6A==}
    dev: false

  /@fastify/error@3.4.0:
    resolution: {integrity: sha512-
e/mafFwbK3MNqxUcFBLgHhgxsF8UT1m8aj0dAlqEa2nJEgPsRtpHTZ3ObgrgkZ2M1eJHPTwgyUl/tXkvabsZ
dQ==}
    dev: false

  /@fastify/fast-json-stringify-compiler@4.3.0:
    resolution: {integrity: sha512-
aZAXGYo6m22Fk1zZzEUKBvut/CIIQe/BapEORnxiD5Qr0kPHqqI69NtEMCme74h+at72sPhbkb4ZrLd1W3K
RLA==}
    dependencies:
      fast-json-stringify: 5.8.0
    dev: false

  /@fastify/formbody@7.4.0:
    resolution: {integrity: sha512-
H3C6h1GN56/SMrZS8N2vCT2cZr7mIHzBHzOBa5OPpjfB/D6FzP9mMpE02ZzrFX0ANeh0BAJdoXKOF2e7I
bV+Og==}
    dependencies:
      fast-querystring: 1.1.2
      fastify-plugin: 4.5.1
    dev: false

  /@fastify/send@2.1.0:
    resolution: {integrity: sha512-
yNYiY6sDkexoJR0D8IDy3aRP3+L4wdqCpvx5WP+VtEU58sn7USmKynBzDQex5X42Zzvw2gNzzYgP90Uf
WShLFA==}
    dependencies:
      '@lukeed/ms': 2.0.1
      escape-html: 1.0.3
      fast-decode-uri-component: 1.0.1
      http-errors: 2.0.0
      mime: 3.0.0
    dev: false

  /@fastify/static@6.11.2:
    resolution: {integrity: sha512-EH7mh7q4MfNdT7N07ZVlwsX/ObngMvQ7KBP0FXAuPov99Fjn80KSJMdxQhhYKAKWW1jXiFdrk8X7d6uGWdZFxg==}
    dependencies:
      '@fastify/accept-negotiator': 1.1.0
      '@fastify/send': 2.1.0
      content-disposition: 0.5.4
      fastify-plugin: 4.5.1
      glob: 8.1.0
      p-limit: 3.1.0
    dev: false

  /@humanwhocodes/config-array@0.11.13:
    resolution: {integrity: sha512-JSBDMiDKSzQVngfRjOdFXgFfklaXI4K9nLF49Auh21lmBWRLIK3+xTErTWD4KU54pb6coM6ESE7Awz/FNU3zgQ==}
    engines: {node: '>=10.10.0'}
    dependencies:
      '@humanwhocodes/object-schema': 2.0.1
      debug: 4.3.4
      minimatch: 3.1.2
    transitivePeerDependencies:
      - supports-color
    dev: true

  /@humanwhocodes/module-importer@1.0.1:
    resolution: {integrity: sha512-bxveV4V8v5Yb4ncFTT3rPSgZBOpCkjfK0y4oVVVJwIuDVBRMDXrPyXRL988i5ap9m9bnyEEjWfm5WkBmtffLfA==}
    engines: {node: '>=12.22'}
    dev: true

  /@humanwhocodes/object-schema@2.0.1:
    resolution: {integrity: sha512-dvuCeX5fC9dXgJn9t+X5atfmgQAzUOWqS1254Gh0m6i8wKd10ebXkfNKiRK+1GWi/yTvvLDHpoxLr0xxxeslWw==}
    dev: true

  /@jridgewell/gen-mapping@0.1.1:
    resolution: {integrity: sha512-sQXCasFk+U8lWYEe66WxRDOE9PjVz4vSM51fTu3Hw+ClTpUSQb718772vH3pyS5pShp6lvQM7SxgIDXXXmOX7w==}
    engines: {node: '>=6.0.0'}
    dependencies:
      '@jridgewell/set-array': 1.1.2
      '@jridgewell/sourcemap-codec': 1.4.14

  /@jridgewell/gen-mapping@0.3.2:
    resolution: {integrity: sha512-
```

```
mh65xKQAzI6iBcFzwv28KVWSmCkdRBWoOh+bYQGW3+6OZvbbN3TqMGo5hqYxQniRcH9F2VZIoJCm4
pa3BPDK/A==}
    engines: {node: '>=6.0.0'}
    dependencies:
      '@jridgewell/set-array': 1.1.2
      '@jridgewell/sourcemap-codec': 1.4.14
      '@jridgewell/trace-mapping': 0.3.17

  /@jridgewell/resolve-uri@3.1.0:
    resolution: {integrity: sha512-
F2msla3tad+Mfht5cJq7LSXcdudKTWCVYUgw6pLFOOHSTtZlj6SWNYAp+AhuqLmWdBO2X5hPrLcu8cVP8
fy28w==}
    engines: {node: '>=6.0.0'}

  /@jridgewell/set-array@1.1.2:
    resolution: {integrity: sha512-
xnkseuNADM0gt2bs+BvhO0p78Mk762YnZdsuzFV018NoG1Sj1SCQvpSqa7XUaTam5vAGasABV9qXASM
KnFMwMw==}
    engines: {node: '>=6.0.0'}

  /@jridgewell/sourcemap-codec@1.4.14:
    resolution: {integrity: sha512-
XPSJHWmi394fuUuzDnGz1wiKqWfo1yXecHQMRf2l6hztTO+nPru658AyDngaBe7isIxEkRsPR3FZh+s7iVa4
Uw==}

  /@jridgewell/trace-mapping@0.3.17:
    resolution: {integrity: sha512-
MCNzAp77qzKca9+W/+I0+sEpaUnZoeasnghNeVc41VZCEKaCH73Vq3BZZ/SzWIgrqE4H4ceI+p+b6C0mHf
9T4g==}
    dependencies:
      '@jridgewell/resolve-uri': 3.1.0
      '@jridgewell/sourcemap-codec': 1.4.14

  /@jridgewell/trace-mapping@0.3.9:
    resolution: {integrity: sha512-
3Belt6tdc8bPgAtbcmdtNJlirVoTmEb5e2gC94PnkwEW9jI6CAHUeoG85tjWP5WquqfavoMtMwiG4P926ZKKu
Q==}
    dependencies:
      '@jridgewell/resolve-uri': 3.1.0
      '@jridgewell/sourcemap-codec': 1.4.14

  /@ledgerhq/connect-kit-loader@1.1.2:
    resolution: {integrity: sha512-
mscwGroSJQrCTjtNGBu+18FQbZYA4+q6Tyx6K7CXHl6AwgZKbWfZYdgP2F+fyZcRUdGRsMX8QtvU61Vc
GGtO1A==}
    dev: false

  /@lit-labs/ssr-dom-shim@1.1.2:
    resolution: {integrity: sha512-
jnOD+/+dSrfTWYfSXBXlo5l5f0q1UuJo3tkbMDCYA2lKUYq79jaxqtGEvnRoh049nt1vdo1+45RinipU6FGY2g==
}
    dev: false
```

```

 /@lit/reactive-element@1.6.3:
   resolution: {integrity: sha512-
QuTgnG52Poic7uM1AN5yJ09QMe0O28e10XzSvWDz02TJiiKee4stsiownEIadWm8nYzyDAyT+gKzUoZmiW
QtsQ==}
   dependencies:
     '@lit-labs/ssr-dom-shim': 1.1.2
   dev: false

 /@lukeed/ms@2.0.1:
   resolution: {integrity: sha512-
Xs/4RZltsAL7pkvaNStUQt7netTkyxrS0K+RILcVr3TRMS/ToOg4I6uNfhB9SlGsnWBym4U+EaXq0f0cEMNkH
A==}
   engines: {node: '>=8'}
   dev: false

 /@lukso/lsp-smart-contracts@0.12.0-rc.0(@babel/core@7.23.2):
   resolution: {integrity: sha512-
yyX5AbsPCfOwH/tVUQp4Arf3ysKviqoPI9UBbAkImWjxQCocI6K1PBKCH77kCeuOFzpPBGcHo4LqlSkhbqv
YNw==}
   dependencies:
     '@account-abstraction/contracts': 0.6.0
     '@erc725/smart-contracts': 6.0.0(@babel/core@7.23.2)
     '@openzeppelin/contracts': 4.9.3
     '@openzeppelin/contracts-upgradeable': 4.9.3
     solidity-bytes-utils: 0.8.0(@babel/core@7.23.2)
   transitivePeerDependencies:
     - '@babel/core'
     - bufferutil
     - encoding
     - supports-color
     - utf-8-validate
   dev: false

 /@metamask/eth-sig-util@4.0.1:
   resolution: {integrity: sha512-
tghyZKLHZjcdlDqCA3gNZmLeR0XvOE9U1qoQO9ohyAZT6Pya+H9vkBPcsyXytmYLNgVoin7CKCmweo/R4
3V+tQ==}
   engines: {node: '>=12.0.0'}
   dependencies:
     ethereumjs-abi: 0.6.8
     ethereumjs-util: 6.2.1
     ethjs-util: 0.1.6
     tweetnacl: 1.0.3
     tweetnacl-util: 0.15.1
   dev: false

 /@metamask/safe-event-emitter@2.0.0:
   resolution: {integrity: sha512-
/kSXhY692qiV1MXu6EeOZvg5nECLclxNXcKCxJ3cXQgYuRymRHpdx/t7JXfsK+JLjwA1e1c1/SBrlQYpusC2
9Q==}
   dev: false
```

```
  /@metamask/utils@3.6.0:
    resolution: {integrity: sha512-9cIRrfkWvHblSiNDVXsjivqa9Ak0RYo/1H6tqTqTbAx+oBK2Sva0lWDHxGchOqA7bySGUJKAWSNJvH6gdHZ0gQ==}
    engines: {node: '>=14.0.0'}
    dependencies:
      '@types/debug': 4.1.10
      debug: 4.3.4
      semver: 7.5.4
      superstruct: 1.0.3
    transitivePeerDependencies:
      - supports-color
    dev: false

  /@motionone/animation@10.16.3:
    resolution: {integrity: sha512-QUGWpLbMFLhyqKlngjZhjtxM8IqiJQjLK0DF+XOF6od9nhSvlaeEpOY/UMCRVcZn/9Tr2rZO22EkuCIjYdI74g==}
    dependencies:
      '@motionone/easing': 10.16.3
      '@motionone/types': 10.16.3
      '@motionone/utils': 10.16.3
      tslib: 2.4.1
    dev: false

  /@motionone/dom@10.16.4:
    resolution: {integrity: sha512-HPHlVo/030qpRj9R8fgY50KTN4Ko30moWRTA3L3imrsRBmob93cTYmodln49HYFbQm01lFF7X523OkKY0DX6UA==}
    dependencies:
      '@motionone/animation': 10.16.3
      '@motionone/generators': 10.16.4
      '@motionone/types': 10.16.3
      '@motionone/utils': 10.16.3
      hey-listen: 1.0.8
      tslib: 2.4.1
    dev: false

  /@motionone/easing@10.16.3:
    resolution: {integrity: sha512-HWTMZbTmZojzwEuKT/xCdvoMPXjYSyQvuVM6jmM0yoGU6BWzsmYMeB4bn38UFf618fJCNtP9XeC/zxtKWfbr0w==}
    dependencies:
      '@motionone/utils': 10.16.3
      tslib: 2.4.1
    dev: false

  /@motionone/generators@10.16.4:
    resolution: {integrity: sha512-geFZ3w0Rm0ZXXpctWsSf3REGywmLLujEjxPYpBR0j+ymYwof0xbV6S5kGqqsDKgyWKVWpUlnqQYvQfL6fRbXeg==}
    dependencies:
      '@motionone/types': 10.16.3
      '@motionone/utils': 10.16.3
      tslib: 2.4.1
    dev: false

  /@motionone/svelte@10.16.4:
    resolution: {integrity: sha512-zRVqk20lD1xqe+yEDZhMYgftsuHc25+9JSo+r0a0OWUJFocjSV9D/+UGhX4xgJsuwB9acPzXLr20w40VnY2PQA==}
    dependencies:
      '@motionone/dom': 10.16.4
      tslib: 2.4.1
    dev: false

  /@motionone/types@10.16.3:
    resolution: {integrity: sha512-W4jkEGFifDq73DlaZs3HUfamV2t1wM35zN/zX7Q79LfZ2sc6C0R1baUHZmqc/K5F3vSw3PavgQ6HyHLd/MXcWg==}
    dev: false

  /@motionone/utils@10.16.3:
    resolution: {integrity: sha512-WNWDksJIxQkaI9p9Z9z0+K27xdqISGNFy1SsWVGaiedTHq0iaT6iZujby8fT/ZnZxj1EOaxJtSfUPCFNU5CRoA==}
    dependencies:
      '@motionone/types': 10.16.3
      hey-listen: 1.0.8
      tslib: 2.4.1
    dev: false

  /@motionone/vue@10.16.4:
    resolution: {integrity: sha512-z10PF9JV6SbjFq+/rYabM+8CVlMokgl8RFGvieSGNTmrkQanfHn+15XBrhG3BgUfvmTeSeyShfOHpG0i9zEdcg==}
    dependencies:
      '@motionone/dom': 10.16.4
      tslib: 2.4.1
    dev: false

  /@next/eslint-plugin-next@14.0.0:
    resolution: {integrity: sha512-Ye37nNI09V3yt7pzuzSQtwlvuJ2CGzFszHXkcTHHZgNr7EhTMFLipn3VSJChy+e5+ahTdNApPphc3qCPUsn10A==}
    dependencies:
      glob: 7.1.7
    dev: true

  /@noble/curves@1.0.0:
    resolution: {integrity: sha512-2upgEu0iLiDVDZkNLeFV2+ht0BAVgQnEmCk6JsOch9Rp8xfkMCbvbAZlA2pBHQc73dbl+vFOXfqkf4uemdn0bw==}
    dependencies:
      '@noble/hashes': 1.3.0
    dev: false

  /@noble/curves@1.2.0:
    resolution: {integrity: sha512-
oYclrNgRaM9SsBUBVbb8M6DTV7ZHRTKugureoYEncY5c65HOmRzvSiTE3y5CYaPYJA/GVkrhXEoF0M3Y
a9PMnw==}
    dependencies:
      '@noble/hashes': 1.3.2
    dev: false

  /@noble/hashes@1.1.2:
    resolution: {integrity: sha512-
KYRCASVTv6aeUi1tsF8/vpyR7zpfs3FUzy2Jqm+MU+LmUKhQ0y2FpfwqkCcxSg2ua4GALJd8k2R76WxwZ
GbQpA==}
    dev: false

  /@noble/hashes@1.3.0:
    resolution: {integrity: sha512-
ilHEACi9DwqJB0pw7kv+Apvh50jiiSyR/cQ3y4W7lOR5mhvn/50FLUfsnfJz0BDZtl/RR16kXvptiv6q1msYZg==}
    dev: false

  /@noble/hashes@1.3.2:
    resolution: {integrity: sha512-
MVC8EAQp7MvEcm30KWENFjgR+Mkmf+D189XJTkFIlwohU5hcBbn1ZkKq7KVTi2Hme3PMGF390DaL52b
eVrIihQ==}
    engines: {node: '>= 16'}
    dev: false

  /@noble/secp256k1@1.6.3:
    resolution: {integrity: sha512-
T04e4iTurVy7I8Sw4+c5OSN9/RkPlo1uKxAomtxQNLq8j1uPAqnsqG1bqvY3Jv7c13gyr6dui0zmh/l3+f/JaQ==
}
    dev: false

  /@noble/secp256k1@2.0.0:
    resolution: {integrity: sha512-
rUGBd95e2a45rlmFTqQJYEFA4/gdIARFfuTuTqLglz0PZ6AKyzyXsEZZq7UZn8hZsvaBgpCzKKBJizT2cJER
Xw==}
    dev: false

  /@nodelib/fs.scandir@2.1.5:
    resolution: {integrity: sha512-
vq24Bq3ym5HEQm2NKCr3yXDwjc7vTsEThRDnkp2DK9p1uqLR+DHurm/NOTo0KG7HYHU7eppKZj3MyqY
uMBf62g==}
    engines: {node: '>= 8'}
    dependencies:
      '@nodelib/fs.stat': 2.0.5
      run-parallel: 1.2.0

  /@nodelib/fs.stat@2.0.5:
    resolution: {integrity: sha512-
RkhPPp2zrqDAQA/2jNhnztcPAlv64XdhIp7a7454A5ovI7Bukxgt7MX7udwAu3zg1DcpPU0rz3VV1SeaqvY4+
A==}
    engines: {node: '>= 8'}

  /@nodelib/fs.walk@1.2.8:
    resolution: {integrity: sha512-
oGB+UxlgWcgQkgwo8GcEGwemoTFt3FIO9ababBmaGwXIoBKZ+GTy0pP185beGg7Llih/NSHSV2XAs1lnz
nocSg==}
    engines: {node: '>= 8'}
    dependencies:
      '@nodelib/fs.scandir': 2.1.5
      fastq: 1.15.0

  /@octokit/auth-token@3.0.4:
    resolution: {integrity: sha512-
TWFX7cZF2LXoCvdmJWY7XVPi74aSY0+FfBZNSXEXFkMpjcqsQwDSYVv5FhRFaI0V1ECnwbz4j59T/G+r
XNWaIQ==}
    engines: {node: '>= 14'}
    dev: true

  /@octokit/core@4.2.4:
    resolution: {integrity: sha512-
rYKilwgzQ7/imScn3M9/pFfUf4I1AZEH3KhyJmtPdE2zfaXAn2mFfUy4FbKewzc2We5y/LlKLj36fWJLKC2SIQ
==}
    engines: {node: '>= 14'}
    dependencies:
      '@octokit/auth-token': 3.0.4
      '@octokit/graphql': 5.0.6
      '@octokit/request': 6.2.8
      '@octokit/request-error': 3.0.3
      '@octokit/types': 9.3.2
      before-after-hook: 2.2.3
      universal-user-agent: 6.0.0
    transitivePeerDependencies:
      - encoding
    dev: true

  /@octokit/endpoint@7.0.6:
    resolution: {integrity: sha512-
5L4fseVRUsDFGR00tMWD/Trdeeihn999rTMGRMC1G/Ldi1uWlWJzI98H4Iak5DB/RVvQuyMYKqSK/R6mbS
OQyg==}
    engines: {node: '>= 14'}
    dependencies:
      '@octokit/types': 9.3.2
      is-plain-object: 5.0.0
      universal-user-agent: 6.0.0
    dev: true

  /@octokit/graphql@5.0.6:
    resolution: {integrity: sha512-
Fxyxdy/JH0MnIB5h+UQ3yCoh1FG4kWXfFKkpWqjZHw/p+Kc8Y44Hu/kCgNBT6nU1shNumEchmW/sUO1Ju
QnPcw==}
    engines: {node: '>= 14'}
    dependencies:
      '@octokit/request': 6.2.8
      '@octokit/types': 9.3.2
      universal-user-agent: 6.0.0
    transitivePeerDependencies:
      - encoding
    dev: true

  /@octokit/openapi-types@18.1.1:
    resolution: {integrity: sha512-
VRaeH8nCDtF5aXWnjPuEMIYf1itK/s3JYyJcWFJT8X9pSNnBtriDf7wlEWsGuhPLl4QIH4xM8fqTXDwJ3Mu6
sw==}
    dev: true

  /@octokit/plugin-paginate-rest@6.1.2(@octokit/core@4.2.4):
    resolution: {integrity: sha512-
qhrmtQeHU/IivxucOV1bbI/xZyC/iOBhclokv7Sut5vnejAIAEXVcGQeRpQlU39E0WwK9lNvJHphHri/DB6lbQ==
}
    engines: {node: '>= 14'}
    peerDependencies:
      '@octokit/core': '>=4'
    dependencies:
      '@octokit/core': 4.2.4
      '@octokit/tsconfig': 1.0.2
      '@octokit/types': 9.3.2
    dev: true

  /@octokit/plugin-request-log@1.0.4(@octokit/core@4.2.4):
    resolution: {integrity: sha512-
mLUsMkgP7K/cnFEw07kWqXGF5LKrOkD+lhCrKvPHXWDywAwuDUeDwWBpc69XK3pNX0uKiVt8g5z96PJ
6z9xCFA==}
    peerDependencies:
      '@octokit/core': '>=3'
    dependencies:
      '@octokit/core': 4.2.4
    dev: true

  /@octokit/plugin-rest-endpoint-methods@7.2.3(@octokit/core@4.2.4):
    resolution: {integrity: sha512-
I5Gml6kTAkzVlN7KCtjOM+Ruwe/rQppp0QU372K1GP7kNOYEKe8Xn5BW4sE62JAHdwpq95OQK/qGNyK
QMUzVgA==}
    engines: {node: '>= 14'}
    peerDependencies:
      '@octokit/core': '>=3'
    dependencies:
      '@octokit/core': 4.2.4
      '@octokit/types': 10.0.0
    dev: true

  /@octokit/request-error@3.0.3:
    resolution: {integrity: sha512-
crqw3V5Iy2uOU5Np+8M/YexTlT8zxCfI+qu+LxUB7SZpje4Qmx3mub5DfEKSO8Ylyk0aogi6TYdf6kxzh2Bgu
Q==}
    engines: {node: '>= 14'}
    dependencies:
      '@octokit/types': 9.3.2
      deprecation: 2.3.1
      once: 1.4.0
    dev: true

  /@octokit/request@6.2.8:
    resolution: {integrity: sha512-
ow4+pkVQ+6XVVsekSYBzJC0VTVvh/FCTUUgTsboGq+DTeWdyIFV8WSCdo0RIxk6wSkBTHqIK1mYuY7n
OBXOchw==}
    engines: {node: '>= 14'}
    dependencies:
      '@octokit/endpoint': 7.0.6
      '@octokit/request-error': 3.0.3
      '@octokit/types': 9.3.2
      is-plain-object: 5.0.0
      node-fetch: 2.7.0
      universal-user-agent: 6.0.0
    transitivePeerDependencies:
      - encoding
    dev: true

  /@octokit/rest@19.0.13:
    resolution: {integrity: sha512-
/EzVox5V9gYGdbAI+ovYj3nXQT1TtTHRT+0eZPcuC05UFSWO3mdO9UY1C0i2eLF9Un1ONJkAk+IEtYGA
C+TahA==}
    engines: {node: '>= 14'}
    dependencies:
      '@octokit/core': 4.2.4
      '@octokit/plugin-paginate-rest': 6.1.2(@octokit/core@4.2.4)
      '@octokit/plugin-request-log': 1.0.4(@octokit/core@4.2.4)
      '@octokit/plugin-rest-endpoint-methods': 7.2.3(@octokit/core@4.2.4)
    transitivePeerDependencies:
      - encoding
    dev: true

  /@octokit/tsconfig@1.0.2:
    resolution: {integrity: sha512-
I0vDR0rdtP8p2lGMzvsJzbhdOWy405HcGovrspJ8RRibHnyRgggUSNO5AIox5LmqiwmatHKYsvj6VGFHkqS
7lA==}
    dev: true

  /@octokit/types@10.0.0:
    resolution: {integrity: sha512-
Vm8IddVmhCgU1fxC1eyinpwqzXPEYu0NrYzD3YZjlGjyftdLBTeqNblRC0jmJmgxbJIsQlyogVeGnrNaaMVzIg
==}
    dependencies:
      '@octokit/openapi-types': 18.1.1
    dev: true

  /@octokit/types@9.3.2:
    resolution: {integrity: sha512-D4iHGTdAnEEVsB8fl95m1hiz7D5YiRdQ9b/OEb3BYRVwbLsGHcRVPz+u+BgRLNk0Q0/4iZCBqDN96j2XNxfXrA==}
    dependencies:
      '@octokit/openapi-types': 18.1.1
    dev: true

  /@openzeppelin/contracts-upgradeable@4.9.3:
    resolution: {integrity: sha512-jjaHAVRMrE4UuZNfDwjlLGDxTHWIOwTJS2ldnc278a0gevfXfPr8hxKEVBGFBE96kl2G3VHDZhUimw/+G3TG2A==}
    dev: false

  /@openzeppelin/contracts@4.9.3:
    resolution: {integrity: sha512-He3LieZ1pP2TNt5JbkPA4PNT9WC3gOTOlDcFGJW4Le4QKqwmiNJCRt44APfxMxvq7OugU/cqYuPcSBzOw38DAg==}
    dev: false

  /@pkgr/utils@2.4.2:
    resolution: {integrity: sha512-POgTXhjrTfbTV63DiFXav4lBHiICLKKwDeaKn9Nphwj7WH6m0hMMCaJkMyRWjgtPFyRKRVoMXXjczsTQRDEhYw==}
    engines: {node: ^12.20.0 || ^14.18.0 || >=16.0.0}
    dependencies:
      cross-spawn: 7.0.3
      fast-glob: 3.3.1
      is-glob: 4.0.3
      open: 9.1.0
      picocolors: 1.0.0
      tslib: 2.6.2
    dev: true

  /@rushstack/eslint-patch@1.5.1:
    resolution: {integrity: sha512-6i/8UoL0P5y4leBIGzvkZdS85RDMG9y1ihZzmTZQ5LdHUYmZ7pKFoj8X0236s3lusPs1Fa5HTQUpwI+UfTcmeA==}
    dev: true

  /@safe-global/safe-apps-provider@0.17.1(typescript@5.2.2):
    resolution: {integrity: sha512-lYfRqrbbK1aKU1/UGkYWc/X7PgySYcumXKc5FB2uuwAs2Ghj8uETuW5BrwPqyjBknRxutFbTv+gth/JzjxAhdQ==}
    dependencies:
      '@safe-global/safe-apps-sdk': 8.0.0(typescript@5.2.2)
      events: 3.3.0
    transitivePeerDependencies:
      - bufferutil
      - typescript
      - utf-8-validate
      - zod
    dev: false

  /@safe-global/safe-apps-sdk@8.0.0(typescript@5.2.2):
    resolution: {integrity: sha512-
gYw0ki/EAuV1oSyMxpqandHjnthZjYYy+YWpTAzf8BqfXM3ItcZLpjxfg+3+mXW8HIO+3jw6T9iiqEXsqHaMM
w==}
    dependencies:
      '@safe-global/safe-gateway-typescript-sdk': 3.12.0
      viem: 1.17.2(typescript@5.2.2)
    transitivePeerDependencies:
      - bufferutil
      - typescript
      - utf-8-validate
      - zod
    dev: false

  /@safe-global/safe-apps-sdk@8.1.0(typescript@5.2.2):
    resolution: {integrity: sha512-
XJbEPuaVc7b9n23MqlF6c+ToYIS3f7P2Sel8f3cSBQ9WORE4xrSuvhMpK9fDSFqJ7by/brc+rmJR/5HViRr0/w
==}
    dependencies:
      '@safe-global/safe-gateway-typescript-sdk': 3.12.0
      viem: 1.17.2(typescript@5.2.2)
    transitivePeerDependencies:
      - bufferutil
      - typescript
      - utf-8-validate
      - zod
    dev: false

  /@safe-global/safe-gateway-typescript-sdk@3.12.0:
    resolution: {integrity: sha512-
hExCo62lScVC9/ztVqYEYL2pFxcqLTvB8fj0WtdP5FWrvbtEgD0pbVolchzD5bf85pbzvEwdAxSVS7EdCZxTN
w==}
    engines: {node: '>=16'}
    dev: false

  /@scure/base@1.1.3:
    resolution: {integrity: sha512-
/+SgoRjLq7Xlf0CWuLHq2LUZeL/w65kfzAPG5NH9pcmBhs+nunQTn4gvdwgMTIXnt9b2C/1SeL2XiysZEyIC9
Q==}
    dev: false

  /@scure/bip32@1.1.0:
    resolution: {integrity: sha512-
ftTW3kKX54YXLCxH6BB7oEEoJfoE2pIgw7MINKAs5PsS6nqKPuKk1haTF/EuHmYqG330t5GSrdmtRuHaY
1a62Q==}
    dependencies:
      '@noble/hashes': 1.1.2
      '@noble/secp256k1': 1.6.3
      '@scure/base': 1.1.3
    dev: false

  /@scure/bip32@1.3.0:
    resolution: {integrity: sha512-
bcKpo1oj54hGholplGLpqPHRbIsnbixFtc06nwuNM5/dwSXOq/AAYoIBRsBmnZJSdfeNW5rnff7NTAz3ZCqR9
Q==}
    dependencies:
      '@noble/curves': 1.0.0
      '@noble/hashes': 1.3.2
      '@scure/base': 1.1.3
    dev: false

  /@scure/bip32@1.3.2:
    resolution: {integrity: sha512-
N1ZhksgwD3OBlwTv3R6KFEcPojl/W4ElJOeCZdi+vuI5QmTFwLq3OFf2zd2ROpKvxFdgZ6hUpb0dx9bVNE
wYCA==}
    dependencies:
      '@noble/curves': 1.2.0
      '@noble/hashes': 1.3.2
      '@scure/base': 1.1.3
    dev: false

  /@scure/bip39@1.1.0:
    resolution: {integrity: sha512-
pwrPOS16VeTKg98dYXQyIjJEcWfz7/1YJIwxUEPFfQPtc86Ym/1sVgQ2RLoD43AazMk2l/unK4ITySSpW2+8
2w==}
    dependencies:
      '@noble/hashes': 1.1.2
      '@scure/base': 1.1.3
    dev: false

  /@scure/bip39@1.2.0:
    resolution: {integrity: sha512-
SX/uKq52cuxm4YFXWFaVByaSHJh2w3BnokVSeUJVCv6K7WulT9u2BuNRBhuFl8vAuYnzx9bEu9WgpcNY
TrYieg==}
    dependencies:
      '@noble/hashes': 1.3.2
      '@scure/base': 1.1.3
    dev: false

  /@scure/bip39@1.2.1:
    resolution: {integrity: sha512-
Z3/Fsz1yr904dduJD0NpiyRHhRYHdcnyh73FZWiV+/qhWi83wNJ3NWolYqCEN+ZWsUz2TWwajJggcRE9r1z
UYg==}
    dependencies:
      '@noble/hashes': 1.3.2
      '@scure/base': 1.1.3
    dev: false

  /@sindresorhus/is@4.6.0:
    resolution: {integrity: sha512-
t09vSN3MdfsyCHoFcTRCH/iUtG7OJ0CsjzB8cjAmKc/va/kIgeDI/TxsigdncE/4be734m0cvIYwNaV4i2XqAw==
}
    engines: {node: '>=10'}
    dev: false

  /@solana/buffer-layout@4.0.0:
    resolution: {integrity: sha512-
IR0EMP2HC3+Mxwd4YcnZb0smnaDw7Bl2IQWZiTevRH5ZZBZn6VRWn3/92E3qdU4SSImJkA6IDHawOHA
nx/qUvQ==}
    engines: {node: '>=5.10'}
    dependencies:
      buffer: 6.0.3
    dev: false

  /@solana/web3.js@1.87.3:
    resolution: {integrity: sha512-
WGLzTZpi00vP443qGK3gL+LZXQJwaWkh6bzNXYpMTCAH2Z102y3YbPWOoQzJUeRSZWSXKh7MFkA3
vDMFlMvGZQ==}
    dependencies:
      '@babel/runtime': 7.23.2
      '@noble/curves': 1.2.0
      '@noble/hashes': 1.3.2
      '@solana/buffer-layout': 4.0.0
      agentkeepalive: 4.5.0
      bigint-buffer: 1.1.5
      bn.js: 5.2.1
      borsh: 0.7.0
      bs58: 4.0.1
      buffer: 6.0.3
      fast-stable-stringify: 1.0.0
      jayson: 4.1.0
      node-fetch: 2.7.0
      rpc-websockets: 7.6.1
      superstruct: 0.14.2
    transitivePeerDependencies:
      - bufferutil
      - encoding
      - utf-8-validate
    dev: false

  /@spruceid/siwe-parser@2.0.2:
    resolution: {integrity: sha512-
9WuA0ios2537cWYu39MMeH0O2KdrMKgKlOBUTWRTXQjCYu5B+mHCA0JkCbFaJ/0EjxoVIcYCXIW/DoP
Epw+PqA==}
    dependencies:
      '@noble/hashes': 1.3.2
      apg-js: 4.2.1
      uri-js: 4.4.1
      valid-url: 1.0.9
    dev: false

  /@stablelib/aead@1.0.1:
    resolution: {integrity: sha512-q39ik6sxGHewqtO0nP4BuSe3db5G1fEJE8ukvngS2gLkBXyy6E7pLubhbYgnkDFv6V8cWaxcE4Xn0t6LWcJkyg==}
    dev: false

  /@stablelib/binary@1.0.1:
    resolution: {integrity: sha512-ClJWvmL6UBM/wjkvv/7m5VP3GMr9t0osr4yVgLZsLCOz4hGN9gIAFEqnJ0TsSMAN+n840nf2cHZnA5/KFqHC7Q==}
    dependencies:
      '@stablelib/int': 1.0.1
    dev: false

  /@stablelib/bytes@1.0.1:
    resolution: {integrity: sha512-Kre4Y4kdwuqL8BR2E9hV/R5sOrUj6NanZaZis0V6lX5yzqC3hBuVSDXUIBqQv/sCpmuWRiHLwqiT1pqqjuBXoQ==}
    dev: false

  /@stablelib/chacha20poly1305@1.0.1:
    resolution: {integrity: sha512-MmViqnqHd1ymwjOQfghRKw2R/jMIGT3wySN7cthjXCBdO+qErNPUBnRzqNpnvIwg7JBCg3LdeCZZO4de/yEhVA==}
    dependencies:
      '@stablelib/aead': 1.0.1
      '@stablelib/binary': 1.0.1
      '@stablelib/chacha': 1.0.1
      '@stablelib/constant-time': 1.0.1
      '@stablelib/poly1305': 1.0.1
      '@stablelib/wipe': 1.0.1
    dev: false

  /@stablelib/chacha@1.0.1:
    resolution: {integrity: sha512-Pmlrswzr0pBzDofdFuVe1q7KdsHKhhU24e8gkEwnTGOmlC7PADzLVxGdn2PoNVBBabdg0l/IfLKg6sHAbTQugg==}
    dependencies:
      '@stablelib/binary': 1.0.1
      '@stablelib/wipe': 1.0.1
    dev: false

  /@stablelib/constant-time@1.0.1:
    resolution: {integrity: sha512-tNOs3uD0vSJcK6z1fvef4Y+buN7DXhzHDPqRLSXUel1UfqMB1PWNsnnAezrKfEwTLpN0cGH2p9NNjs6IqeD0eg==}
    dev: false

  /@stablelib/ed25519@1.0.3:
    resolution: {integrity: sha512-puIMWaX9QlRsbhxfDc5i+mNPMY+0TmQEskunY1rZEBPi1acBCVQAhnsk/1Hk50DGPtVsZtAWQg4NHGlVaO9Hqg==}
    dependencies:
      '@stablelib/random': 1.0.2
      '@stablelib/sha512': 1.0.1
      '@stablelib/wipe': 1.0.1
    dev: false

  /@stablelib/hash@1.0.1:
    resolution: {integrity: sha512-
eTPJc/stDkdtOcrNMZ6mcMK1e6yBbqRBaNW55XA1jU8w/7QdnCF0CmMmOD1m7VSkBR44PWrMHU2l6r8
YEQHMgg==}
    dev: false

  /@stablelib/hkdf@1.0.1:
    resolution: {integrity: sha512-
SBEHYE16ZXlHuaW5RcGk533YlBj4grMeg5TooN80W3NpcHRtLZLLXvKyX0qcRFxf+BGDobJLnwkvgEwHI
DBR6g==}
    dependencies:
      '@stablelib/hash': 1.0.1
      '@stablelib/hmac': 1.0.1
      '@stablelib/wipe': 1.0.1
    dev: false

  /@stablelib/hmac@1.0.1:
    resolution: {integrity: sha512-
V2APD9NSnhVpV/QMYgCVMIYKiYG6LSqw1S65wxVoirhU/51ACio6D4yDVSwMzuTJXWZoVHbDdINioBw
Ky5kVmA==}
    dependencies:
      '@stablelib/constant-time': 1.0.1
      '@stablelib/hash': 1.0.1
      '@stablelib/wipe': 1.0.1
    dev: false

  /@stablelib/int@1.0.1:
    resolution: {integrity: sha512-
byr69X/sDtDiIjIV6m4roLVWnNNlRGzsvxw+agj8CIEazqWGOQp2dTYgQhtyVXV9wpO6WyXRQUzLV/JRNu
mT2w==}
    dev: false

  /@stablelib/keyagreement@1.0.1:
    resolution: {integrity: sha512-
VKL6xBwgJnI6l1jKrBAfn265cspaWBPAPEc62VBQrWHLqVgNRE09gQ/AnOEyKUWrrqfD+xSQ3u42gJjLDd
MDQg==}
    dependencies:
      '@stablelib/bytes': 1.0.1
    dev: false

  /@stablelib/poly1305@1.0.1:
    resolution: {integrity: sha512-
1HlG3oTSuQDOhSnLwJRKeTRSAdFNVB/1djy2ZbS35rBSJ/PFqx9cf9qatinWghC2UbfOYD8AcrtbUQl8Wox
abA==}
    dependencies:
      '@stablelib/constant-time': 1.0.1
      '@stablelib/wipe': 1.0.1
    dev: false

  /@stablelib/random@1.0.2:
    resolution: {integrity: sha512-
rIsE83Xpb7clHPVRlBj8qNe5L8ISQOzjghYQm/dZ7VaM2KHYwMW5adjQjrzTZCchFnNCNhkwtnOBa9HTMJ
CI8w==}
    dependencies:
      '@stablelib/binary': 1.0.1
      '@stablelib/wipe': 1.0.1
    dev: false

  /@stablelib/sha256@1.0.1:
    resolution: {integrity: sha512-
GIIH3e6KH+91FqGV42Kcj71Uefd/QEe7Dy42sBTeqppXV95ggCcxLTk39bEr+lZfJmp+ghsR07J++ORkRELs
BQ==}
    dependencies:
      '@stablelib/binary': 1.0.1
      '@stablelib/hash': 1.0.1
      '@stablelib/wipe': 1.0.1
    dev: false

  /@stablelib/sha512@1.0.1:
    resolution: {integrity: sha512-
13gl/iawHV9zvDKciLo1fQ8Bgn2Pvf7OV6amaRVKiq3pjQ3UmEpXxWiAfV8tYjUpeZroBxtyrwtdooQT/i3hzw==
}
    dependencies:
      '@stablelib/binary': 1.0.1
      '@stablelib/hash': 1.0.1
      '@stablelib/wipe': 1.0.1
    dev: false

  /@stablelib/wipe@1.0.1:
    resolution: {integrity: sha512-
WfqfX/eXGiAd3RJe4VU2snh/ZPwtSjLG4ynQ/vYzvghTh7dHFcI1wl+nrkWG6lGhukOxOsUHfv8dUXr58D0ay
g==}
    dev: false

  /@stablelib/x25519@1.0.3:
    resolution: {integrity: sha512-
KnTbKmUhPhHavzobclVJQG5kuivH+qDLpe84iRqX3CLrKp881cF160JvXJ+hjn1aMyCwYOKeIZefIH/P5cJo
Rw==}
    dependencies:
      '@stablelib/keyagreement': 1.0.1
      '@stablelib/random': 1.0.2
      '@stablelib/wipe': 1.0.1
    dev: false

  /@szmarczak/http-timer@4.0.6:
    resolution: {integrity: sha512-
4BAffykYOgO+5nzBWYwE3W90sBgLJoUPRWWcL8wlyiM8IB8ipJz3UMJ9KXQd1RKQXpKp8Tutn80HZtWs
u2u76w==}
    engines: {node: '>=10'}
    dependencies:
      defer-to-connect: 2.0.1
    dev: false

  /@szmarczak/http-timer@5.0.1:
    resolution: {integrity: sha512-
+PmQX0PiAYPMeVYe237LJAYvOMYW1j2rH5YROyS3b4CTVJum34HfRvKvAzozHAQG0TnHNdUfY9nCe
UyRAs//cw==}
    engines: {node: '>=14.16'}
    dependencies:
      defer-to-connect: 2.0.1
    dev: false

  /@tailwindcss/forms@0.5.6(tailwindcss@3.3.5):
    resolution: {integrity: sha512-
Fw+2BJ0tmAwK/w01tEFL5TiaJBX1NLT1/YbWgvm7ws3Qcn11kiXxzNTEQDMs5V3mQemhB56l3u0i9dwdz
SQldA==}
    peerDependencies:
      tailwindcss: '>=3.0.0 || >= 3.0.0-alpha.1'
    dependencies:
      mini-svg-data-uri: 1.4.4
      tailwindcss: 3.3.5(ts-node@10.9.1)
    dev: false

  /@tanstack/query-core@4.36.1:
    resolution: {integrity: sha512-
DJSilV5+ytBP1FbFcEJovv4rnnm/CokuVvrBEtW/Va9DvuJ3HksbXUJEpI0aV1KtuL4ZoO9AVE6PyNLzF7tLe
A==}
    dev: false

  /@tanstack/query-persist-client-core@4.36.1:
    resolution: {integrity: sha512-
eocgCeI7D7TRv1IUUBMfVwOI0wdSmMkBIbkKhqEdTrnUHUQEeOaYac8oeZk2cumAWJdycu6P/wB+WqG
ynTnzXg==}
    dependencies:
      '@tanstack/query-core': 4.36.1
    dev: false

  /@tanstack/query-sync-storage-persister@4.36.1:
    resolution: {integrity: sha512-
yMEt5hWe2+1eclf1agMtXHnPIkxEida0lYWkfdhR8U6KXk/lO4Vca6piJmhKI85t0NHlx3l/z6zX+t/Fn5O9NA==}
    dependencies:
      '@tanstack/query-persist-client-core': 4.36.1
    dev: false

  /@tanstack/react-query-persist-client@4.36.1(@tanstack/react-query@4.36.1):
    resolution: {integrity: sha512-
32I5b9aAu4NCiXZ7Te/KEQLfHbYeTNriVPrKYcvEThnZ9tlW01vLcSoxpUIsMYRsembvJUUAkzYBAiZHLOd6
pQ==}
    peerDependencies:
      '@tanstack/react-query': ^4.36.1
    dependencies:
      '@tanstack/query-persist-client-core': 4.36.1
      '@tanstack/react-query': 4.36.1(react-dom@18.2.0)(react@18.2.0)
    dev: false

  /@tanstack/react-query@4.36.1(react-dom@18.2.0)(react@18.2.0):
    resolution: {integrity: sha512-
y7ySVHFyyQblPl3J3eQBWpXZkliroki3ARnBKsdJchlgt7yJLRDUcf4B8soufgiYt3pEQIkBWBx1N9/ZPIeUWw==
}
    peerDependencies:
      react: ^16.8.0 || ^17.0.0 || ^18.0.0
      react-dom: ^16.8.0 || ^17.0.0 || ^18.0.0
      react-native: '*'
    peerDependenciesMeta:
      react-dom:
        optional: true
      react-native:
        optional: true
    dependencies:
      '@tanstack/query-core': 4.36.1
      react: 18.2.0
      react-dom: 18.2.0(react@18.2.0)
      use-sync-external-store: 1.2.0(react@18.2.0)
    dev: false

  /@truffle/hdwallet-provider@2.1.15(@babel/core@7.23.2):
    resolution: {integrity: sha512-
I5cSS+5LygA3WFzru9aC5+yDXVowEEbLCx0ckl/RqJ2/SCiYXkzYIR5/DjjDJuCtYhivhrn2RP9AheeFIRF+qw==
}
    engines: {node: ^16.20 || ^18.16 || >=20}
    dependencies:
      '@ethereumjs/common': 2.6.5
      '@ethereumjs/tx': 3.5.2
      '@metamask/eth-sig-util': 4.0.1
      '@truffle/hdwallet': 0.1.4
      '@types/ethereum-protocol': 1.0.4
      '@types/web3': 1.0.20
      '@types/web3-provider-engine': 14.0.3
      ethereum-cryptography: 1.1.2
      ethereum-protocol: 1.0.1
      ethereumjs-util: 7.1.5
      web3: 1.10.0
      web3-provider-engine: 16.0.3(@babel/core@7.23.2)
    transitivePeerDependencies:
      - '@babel/core'
      - bufferutil
      - encoding
      - supports-color
      - utf-8-validate
    dev: false

  /@truffle/hdwallet@0.1.4:
    resolution: {integrity: sha512-
D3SN0iw3sMWUXjWAedP6RJtopo9qQXYi80inzbtcsoso4VhxFxCwFvCErCl4b27AEJ9pkAtgnxEFRaSKdMm
i1Q==}
    engines: {node: ^16.20 || ^18.16 || >=20}
    dependencies:
      ethereum-cryptography: 1.1.2
      keccak: 3.0.2
      secp256k1: 4.0.3
    dev: false

  /@tsconfig/node10@1.0.9:
    resolution: {integrity: sha512-
jNsYVVxU8v5g43Erja32laIDHXeoNvFEpX33OK4d6hljo3jDhCBDhx5dhCCTMWUojscpAagGiRkBKxpdl9fxq
A==}

  /@tsconfig/node12@1.0.11:
    resolution: {integrity: sha512-
cqefuRsh12pWyGsIoBKJA9luFu3mRxCA+ORZvA4ktLSzIuCUtWVxGIuXigEwO5/ywWFMZ2QEGKWvkZG1
zDMTag==}

  /@tsconfig/node14@1.0.3:
    resolution: {integrity: sha512-
ysT8mhdixWK6Hw3i1V2AeRqZ5WfXg1G43mqoYlM2nc6388Fq5jcXyr5mRsqViLx/GJYdoL0bfXD8nmF+Zn/I
ow==}

  /@tsconfig/node16@1.0.4:
    resolution: {integrity: sha512-
vxhUy4J8lyeyinH7Azl1pdd43GJhZH/tP2weN8TntQblOY+A0XbT8DJk1/oCPuOOyg/Ja757rG0CgHcWC8Of
MA==}

  /@typechain/ethers-v6@0.5.1(ethers@6.8.0)(typechain@8.3.2)(typescript@5.2.2):
    resolution: {integrity: sha512-
F+GklO8jBWlsaVV+9oHaPh5NJdd6rAKN4tklGfInX1Q7h0xPgVLP39Jl3eCuIPB5qexI71ZFHwbljx4ZXNfouA==
}
    peerDependencies:
      ethers: 6.x
      typechain: ^8.3.2
      typescript: '>=4.7.0'
    dependencies:
      ethers: 6.8.0
      lodash: 4.17.21
      ts-essentials: 7.0.3(typescript@5.2.2)
      typechain: 8.3.2(typescript@5.2.2)
      typescript: 5.2.2
    dev: false

  /@types/babel__core@7.20.3:
    resolution: {integrity: sha512-
54fjTSeSHwfan8AyHWrKbfBWiEUrNTZsUwPTDSNaaP1QDQIZbeNUg3a59E9D+375MzUw/x1vx2/0F5LBz
+AeYA==}
    dependencies:
      '@babel/parser': 7.23.0
      '@babel/types': 7.23.0
      '@types/babel__generator': 7.6.6
      '@types/babel__template': 7.4.3
      '@types/babel__traverse': 7.20.3
    dev: true

  /@types/babel__generator@7.6.6:
    resolution: {integrity: sha512-
66BXMKb/sUWbMdBNdMvajU7i/44RkrA3z/Yt1c7R5xejt8qh84iU54yUWCtm0QwGJlDcf/gg4zd/x4mpLAlb/w
==}
    dependencies:
      '@babel/types': 7.23.0
    dev: true

  /@types/babel__template@7.4.3:
    resolution: {integrity: sha512-
ciwyCLeuRfxboZ4isgdNZi/tkt06m8Tw6uGbBSBgWrnnZGNXiEyM27xc/PjXGQLqlZ6ylbgHMnm7ccF9tCkOe
Q==}
    dependencies:
      '@babel/parser': 7.23.0
      '@babel/types': 7.23.0
    dev: true

  /@types/babel__traverse@7.20.3:
    resolution: {integrity: sha512-
Lsh766rGEFbaxMIDH7Qa+Yha8cMVI3qAK6CHt3OR0YfxOIn5Z54iHiyDRycHrBqeIiqGa20Kpsv1cavfBKkRS
w==}
    dependencies:
      '@babel/types': 7.23.0
    dev: true

  /@types/bn.js@4.11.6:
    resolution: {integrity: sha512-
pqr857jrp2kPuO9uRjZ3PwnJTjoQy+fcdxvBTvHm6dkmEL9q+hDD/2j/0ELOBPtPnS8LjCX0gI9nbl8lVkadpg==
}
    dependencies:
      '@types/node': 20.8.9
    dev: false

  /@types/bn.js@5.1.3:
    resolution: {integrity: sha512-
wT1B4iIO82ecXkdN6waCK8Ou7E71WU+mP1osDA5Q8c6Ur+ozU2vIKUIhSpUr6uE5L2YHocKS1Z2jG2fBC
1YVeg==}
    dependencies:
      '@types/node': 20.8.9
    dev: false

  /@types/cacheable-request@6.0.3:
    resolution: {integrity: sha512-
IQ3EbTzGxIigb1I3qPZc1rWJnH0BmSKv5QYTalEwweFvyBDLSAe24zP0le/hyi7ecGfZVlIVAg4BZqb8WBwK
qw==}
    dependencies:
      '@types/http-cache-semantics': 4.0.3
      '@types/keyv': 3.1.4
      '@types/node': 20.8.9
      '@types/responselike': 1.0.2
    dev: false

  /@types/connect@3.4.35:
    resolution: {integrity: sha512-
cdeYyv4KWoEgpBISTxWvqYsVy444DOqehiF3fM3ne10AmJ62RSyNkUnxMJXHQWRQQX2eR94m5y1IZyD
wBjV9FQ==}
    dependencies:
      '@types/node': 20.8.9
    dev: false

  /@types/debug@4.1.10:
    resolution: {integrity: sha512-
tOSCru6s732pofZ+sMv9o4o3Zc+Sa8l3bxd/tweTQudFn06vAzb13ZX46Zi6m6EJ+RUbRTHvgQJ1gBtSgkaU
YA==}
    dependencies:
      '@types/ms': 0.7.33
    dev: false

  /@types/elliptic@6.4.16:
    resolution: {integrity: sha512-
MSN6m9BR51W1/umzcsX0K9dAmJ59ECKxOjZ3XsjvsZAt+q0mTmKlzEdwtU+u4i+Om231d8TuY3xK6FAGI
s5MbA==}
    dependencies:
      '@types/bn.js': 5.1.3
    dev: false

  /@types/ethereum-protocol@1.0.4:
    resolution: {integrity: sha512-
qzqIdFigCLH4ZoTjDAAz3kv6+dGXx8UVP9a5RHrJzbnaqSKEMSI/zAMgMH2HPRJrMKKYBOydYgblzS+WZ
G27+Q==}
    dependencies:
      bignumber.js: 7.2.1
    dev: false

  /@types/http-cache-semantics@4.0.3:
    resolution: {integrity: sha512-
V46MYLFp08Wf2mmaBhvgjStM3tPa+2GAdy/iqoX+noX1//zje2x4XmrIU0cAwyClATsTmahbtoQ2EwP7I5WSi
A==}
    dev: false

  /@types/json-schema@7.0.14:
    resolution: {integrity: sha512-
U3PUjAudAdJBeC2pgN8uTIKgxrb4nlDF3SF0++EldXQvQBGkpFZMSnwQiIoDU77tv45VgNkl/L4ouD+rEomu
jw==}
    dev: true

  /@types/json5@0.0.29:
    resolution: {integrity: sha512-
dRLjCWHYg4oaA77cxO64oO+7JwCwnIzkZPdrrC71jQmQtlhM556pwKo5bUzqvZndkVbeFLIIi+9TC40JNF5h
```

```
NQ==}
    dev: true

  /@types/keyv@3.1.4:
    resolution: {integrity: sha512-
BQ5aZNSCpj7D6K2ksrRCTmKRLEpnPvWDiLPfoGyhZ++8YtiK9d/3DBKPJgry359X/P1PfruyYwvnvwFjuEiEI
g==}
    dependencies:
      '@types/node': 20.8.9
    dev: false

  /@types/ms@0.7.33:
    resolution: {integrity: sha512-
AuHIyzR5Hea7ij0P9q7vx7xu4z0C28ucwjAZC0ja7JhINyCnOw8/DnvAPQQ9TfOlCtZAmCERKQX9+o1mgQh
uOQ==}
    dev: false

  /@types/node@12.20.55:
    resolution: {integrity: sha512-
J8xLz7q2OFulZ2cyGTLE1TbbZcjpno7FaN6zdJNrgAdrJ+DZzh/uFR6YrTb4C+nXakvud8Q4+rbhoIWlYQbUF
Q==}
    dev: false

  /@types/node@18.15.13:
    resolution: {integrity: sha512-
N+0kuo9KgrUQ1Sn/ifDXsvg0TTleP7rIy4zOBGECxAljqvqfqpTfzx0Q1NUedOixRMBfe2Whhb056a42cWs26
Q==}
    dev: false

  /@types/node@20.8.9:
    resolution: {integrity: sha512-
UzykFsT3FhHb1h7yD4CA4YhBHq545JC0YnEz41xkipN88eKQtL6rSgocL5tbAP6Ola9Izm/Aw4Ora8He4x0B
Hg==}
    dependencies:
      undici-types: 5.26.5

  /@types/normalize-package-data@2.4.3:
    resolution: {integrity: sha512-
ehPtgRgaULsFG8x0NeYJvmyH1hmlfsNLujHe9dQEia/7MAJYdzMSi19JtchUHjmBA6XC/75dK55mzZH+Ryie
Sg==}
    dev: true

  /@types/pbkdf2@3.1.1:
    resolution: {integrity: sha512-
4HCoGwR3221nOc7G0Z/6KgTNGgaaFGkbGrtUJsB+zlKX2LBVjFHHIUkieMBgHHXgBH5Gq6dZHJKdBYdtl
hBQvw==}
    dependencies:
      '@types/node': 20.8.9
    dev: false

  /@types/prettier@2.7.3:
    resolution: {integrity: sha512-
```

```
+68kP9yzs4LMp7VNh8gdzMSPZFL44MLGqiHWvttYJe+6qnuVr4Ek9wSBQoveqY/r+LwjCcU29kNVkidwim+
kYA==}

  /@types/prop-types@15.7.5:
    resolution: {integrity: sha512-
JCB8C6SnDoQf0cNycqd/35A7MjcnK+ZTqE7judS6o7utxUCg6imJg3QK2qzHKszlTjcj2cn+NwMB2i96ubpj7w
==}

  /@types/react-dom@18.0.9:
    resolution: {integrity: sha512-
qnVvHxASt/H7i+XG1U1xMiY5t+IHcPGUK7TDMDzom08xa7e86eCeKOiLZezwCKVxJn6NEiiy2ekgX8aQssjI
Kg==}
    dependencies:
      '@types/react': 18.0.25
    dev: true

  /@types/react@18.0.25:
    resolution: {integrity: sha512-
xD6c0KDT4m7n9uD4ZHi02lzskaiqcBxf4zi+tXZY98a04wvc0hi/TcCPC2FOESZi51Nd7tlUeOJY8RofL799/g==
}
    dependencies:
      '@types/prop-types': 15.7.5
      '@types/scheduler': 0.16.2
      csstype: 3.1.1

  /@types/responselike@1.0.2:
    resolution: {integrity: sha512-
/4YQT5Kp6HxUDb4yhRkm0bJ7TbjvTddqX7PZ5hz6qV3pxSo72f/6YPRo+Mu2DU307tm9IioO69l7uAwn5XN
cFA==}
    dependencies:
      '@types/node': 20.8.9
    dev: false

  /@types/scheduler@0.16.2:
    resolution: {integrity: sha512-
hppQEBDmlwhFAXKJX2KnWLYu5yMfi91yazPb2l+lbJiwW+wdo1gNeRA+3RgNSO39WYX2euey41KEwnqe
sU2Jew==}

  /@types/secp256k1@4.0.5:
    resolution: {integrity: sha512-
aIonTBMErtE3T9MxDvTZRzcrT/mCqpEZBw3CCY/i+oG9n57N/+7obBkhFgavUAIrX21bU0LHg1XRgtaLdelB
hA==}
    dependencies:
      '@types/node': 20.8.9
    dev: false

  /@types/semver@7.5.4:
    resolution: {integrity: sha512-
MMzuxN3GdFwskAnb6fz0orFvhfqi752yjaXylr0Rp4oDg5H0Zn1IuyRhDVvYOwAXoJirx2xuS16I3WjxnAIHiQ==
}
    dev: true
  /@types/trusted-types@2.0.2:
    resolution: {integrity: sha512-F5DIZ36YVLE+PN+Zwws4kJogq47hNgX3Nx6WyDJ3kcplxyke3XIzB8uK5n/Lpm1HBsbGzd6nmGehL8cPekP+Tg==}
    dev: false

  /@types/underscore@1.11.12:
    resolution: {integrity: sha512-beX81q12OQo809WJ/UYCvUDvJR3YQ4wtehYYQ6eNw5VLyd+KUNBRV4FgzZCHBmACbdPulH9F9ifhxzFFU9TWvQ==}
    dev: false

  /@types/web3-provider-engine@14.0.3:
    resolution: {integrity: sha512-ZuVzhriuFg8xdOsC6JW9krNxOAomPqwcXhyEO5/M4NGn5cMhHoUgQyKiyt++pU6tGiTRaAAIq1mL31nx5/tihw==}
    dependencies:
      '@types/ethereum-protocol': 1.0.4
    dev: false

  /@types/web3@1.0.20:
    resolution: {integrity: sha512-KTDIFuYjzCUIBDGt35Ir5QRtyV9kIF84MMKUsEJK10sTWga/71V+8VYLT7yysjuBjaOx2uFYtIWNGoz3yrNDlg==}
    dependencies:
      '@types/bn.js': 5.1.3
      '@types/underscore': 1.11.12
    dev: false

  /@types/ws@7.4.7:
    resolution: {integrity: sha512-JQbbmxZTZehdc2iszGKs5oC3NFnjeay7mtAWrdt7qNtAVK0g19muApzAy4bm9byz79xa2ZnO/BOBC2R8RC5Lww==}
    dependencies:
      '@types/node': 20.8.9
    dev: false

  /@typescript-eslint/eslint-plugin@6.9.0(@typescript-eslint/parser@6.9.0)(eslint@8.52.0)(typescript@5.2.2):
    resolution: {integrity: sha512-lgX7F0azQwRPB7t7WAyeHWVfW1YJ9NIgd9mvGhfQpRY56X6AVf8mwM8Wol+0z4liE7XX3QOt8MN1rUKCfSjRIA==}
    engines: {node: ^16.0.0 || >=18.0.0}
    peerDependencies:
      '@typescript-eslint/parser': ^6.0.0 || ^6.0.0-alpha
      eslint: ^7.0.0 || ^8.0.0
      typescript: '*'
    peerDependenciesMeta:
      typescript:
        optional: true
    dependencies:
      '@eslint-community/regexpp': 4.10.0
      '@typescript-eslint/parser': 6.9.0(eslint@8.52.0)(typescript@5.2.2)
      '@typescript-eslint/scope-manager': 6.9.0
      '@typescript-eslint/type-utils': 6.9.0(eslint@8.52.0)(typescript@5.2.2)
      '@typescript-eslint/utils': 6.9.0(eslint@8.52.0)(typescript@5.2.2)
      '@typescript-eslint/visitor-keys': 6.9.0
      debug: 4.3.4
      eslint: 8.52.0
      graphemer: 1.4.0
      ignore: 5.2.4
      natural-compare: 1.4.0
      semver: 7.5.4
      ts-api-utils: 1.0.3(typescript@5.2.2)
      typescript: 5.2.2
    transitivePeerDependencies:
      - supports-color
    dev: true

  /@typescript-eslint/parser@6.9.0(eslint@8.52.0)(typescript@5.2.2):
    resolution: {integrity: sha512-GZmjMh4AJ/5gaH4XF2eXA8tMnHWP+Pm1mjQR2QN4Iz+j/zO04b9TOvJYOX2sCNIQHtRStKTxRY1FX7LhpJT4Gw==}
    engines: {node: ^16.0.0 || >=18.0.0}
    peerDependencies:
      eslint: ^7.0.0 || ^8.0.0
      typescript: '*'
    peerDependenciesMeta:
      typescript:
        optional: true
    dependencies:
      '@typescript-eslint/scope-manager': 6.9.0
      '@typescript-eslint/types': 6.9.0
      '@typescript-eslint/typescript-estree': 6.9.0(typescript@5.2.2)
      '@typescript-eslint/visitor-keys': 6.9.0
      debug: 4.3.4
      eslint: 8.52.0
      typescript: 5.2.2
    transitivePeerDependencies:
      - supports-color
    dev: true

  /@typescript-eslint/scope-manager@6.9.0:
    resolution: {integrity: sha512-1R8A9Mc39n4pCCz9o79qRO31HGNDvC7UhPhv26TovDsWPBDx+Sg3rOZdCELIA3ZmNoWAuxaMOT7aWtGRSYkQxw==}
    engines: {node: ^16.0.0 || >=18.0.0}
    dependencies:
      '@typescript-eslint/types': 6.9.0
      '@typescript-eslint/visitor-keys': 6.9.0
    dev: true

  /@typescript-eslint/type-utils@6.9.0(eslint@8.52.0)(typescript@5.2.2):
    resolution: {integrity: sha512-XXeahmfbpuhVbhSOROIzJ+b13krFmgtc4GlEuu1WBT+RpyGPIA4Y/eGnXzjbDj5gZLzpAXO/sj+IF/x2GtTMj
```

```
Q==}
    engines: {node: ^16.0.0 || >=18.0.0}
    peerDependencies:
      eslint: ^7.0.0 || ^8.0.0
      typescript: '*'
    peerDependenciesMeta:
      typescript:
        optional: true
    dependencies:
      '@typescript-eslint/typescript-estree': 6.9.0(typescript@5.2.2)
      '@typescript-eslint/utils': 6.9.0(eslint@8.52.0)(typescript@5.2.2)
      debug: 4.3.4
      eslint: 8.52.0
      ts-api-utils: 1.0.3(typescript@5.2.2)
      typescript: 5.2.2
    transitivePeerDependencies:
      - supports-color
    dev: true

  /@typescript-eslint/types@6.9.0:
    resolution: {integrity: sha512-
+KB0lbkpxBkBSiVCuQvduqMJy+I1FyDbdwSpM3IoBS7APl4Bu15lStPjgBIdykdRqQNYqYNMa8Kuidax6pha
Ew==}
    engines: {node: ^16.0.0 || >=18.0.0}
    dev: true

  /@typescript-eslint/typescript-estree@6.9.0(typescript@5.2.2):
    resolution: {integrity: sha512-
NJM2BnJFZBEAbCfBP00zONKXvMqihZCrmwCaik0UhLr0vAgb6oguXxLX1k00oQyD+vZZ+CJn3kocvv2yxm
4awQ==}
    engines: {node: ^16.0.0 || >=18.0.0}
    peerDependencies:
      typescript: '*'
    peerDependenciesMeta:
      typescript:
        optional: true
    dependencies:
      '@typescript-eslint/types': 6.9.0
      '@typescript-eslint/visitor-keys': 6.9.0
      debug: 4.3.4
      globby: 11.1.0
      is-glob: 4.0.3
      semver: 7.5.4
      ts-api-utils: 1.0.3(typescript@5.2.2)
      typescript: 5.2.2
    transitivePeerDependencies:
      - supports-color
    dev: true

  /@typescript-eslint/utils@6.9.0(eslint@8.52.0)(typescript@5.2.2):
    resolution: {integrity: sha512-
5Wf+Jsqya7WcCO8me504FBigeQKVLAMPmUzYgDbWchINNh1KJbxCgVya3EQ2MjvJMVeXl3pofRmprqX6
```

```
mfQkjQ==}
    engines: {node: ^16.0.0 || >=18.0.0}
    peerDependencies:
      eslint: ^7.0.0 || ^8.0.0
    dependencies:
      '@eslint-community/eslint-utils': 4.4.0(eslint@8.52.0)
      '@types/json-schema': 7.0.14
      '@types/semver': 7.5.4
      '@typescript-eslint/scope-manager': 6.9.0
      '@typescript-eslint/types': 6.9.0
      '@typescript-eslint/typescript-estree': 6.9.0(typescript@5.2.2)
      eslint: 8.52.0
      semver: 7.5.4
    transitivePeerDependencies:
      - supports-color
      - typescript
    dev: true

  /@typescript-eslint/visitor-keys@6.9.0:
    resolution: {integrity: sha512-
dGtAfqjV6RFOtIP8I0B4ZTBRrlTT8NHHlZZSchQx3qReaoDeXhYM++M4So2AgFK9ZB0emRPA6JI1HkafzA2
Ibg==}
    engines: {node: ^16.0.0 || >=18.0.0}
    dependencies:
      '@typescript-eslint/types': 6.9.0
      eslint-visitor-keys: 3.4.3
    dev: true

  /@ungap/structured-clone@1.2.0:
    resolution: {integrity: sha512-
zuVdFrMJiuCDQUMCzQaD6KL28MjnqqN8XnAqiEq9PNm/hCPTSGfrXCOfwj1ow4LFb/tNymJPwsNbVePc1
xFqrQ==}
    dev: true

  /@vitejs/plugin-react@4.1.0(vite@4.5.0):
    resolution: {integrity: sha512-
rM0SqazU9iqPUraQ2JlIvReeaxOoRj6n+PzB1C0cBzIbd8qP336nC39/R9yPi3wVcah7E7j/kdU1uCUqMEU4O
Q==}
    engines: {node: ^14.18.0 || >=16.0.0}
    peerDependencies:
      vite: ^4.2.0
    dependencies:
      '@babel/core': 7.23.2
      '@babel/plugin-transform-react-jsx-self': 7.22.5(@babel/core@7.23.2)
      '@babel/plugin-transform-react-jsx-source': 7.22.5(@babel/core@7.23.2)
      '@types/babel__core': 7.20.3
      react-refresh: 0.14.0
      vite: 4.5.0(@types/node@20.8.9)
    transitivePeerDependencies:
      - supports-color
    dev: true
```

```
 /@wagmi/chains@1.0.0(typescript@5.2.2):
   resolution: {integrity: sha512-
eNbqRWyHbivcMNq5tbXJks4NaOzVLHnNQauHPeE/EDT9AlpqzcrMc+v2T1/2Iw8zN4zgqB86NCsxeJHJs7+
xng==}
   peerDependencies:
     typescript: '>=5.0.4'
   peerDependenciesMeta:
     typescript:
       optional: true
   dependencies:
     typescript: 5.2.2
   dev: false

 /@wagmi/connectors@3.1.3(@types/react@18.0.25)(react@18.2.0)(typescript@5.2.2)(viem@0.3.50):
   resolution: {integrity: sha512-
UgwsQKQDFObJVJMf9pDfFoXTv710o4zrTHyhIWKBTMMkLpCMsMxN5+ZaDhBYt/BgoRinfRYQo8uwuwLh
xE6Log==}
   peerDependencies:
     typescript: '>=5.0.4'
     viem: '>=0.3.35'
   peerDependenciesMeta:
     typescript:
       optional: true
   dependencies:
     '@coinbase/wallet-sdk': 3.7.2
     '@ledgerhq/connect-kit-loader': 1.1.2
     '@safe-global/safe-apps-provider': 0.17.1(typescript@5.2.2)
     '@safe-global/safe-apps-sdk': 8.1.0(typescript@5.2.2)
     '@walletconnect/ethereum-provider': 2.10.2(@walletconnect/modal@2.6.2)
     '@walletconnect/legacy-provider': 2.0.0
     '@walletconnect/modal': 2.6.2(@types/react@18.0.25)(react@18.2.0)
     '@walletconnect/utils': 2.10.2
     abitype: 0.8.7(typescript@5.2.2)
     eventemitter3: 4.0.7
     typescript: 5.2.2
     viem: 0.3.50(typescript@5.2.2)
   transitivePeerDependencies:
     - '@react-native-async-storage/async-storage'
     - '@types/react'
     - bufferutil
     - encoding
     - lokijs
     - react
     - supports-color
     - utf-8-validate
     - zod
   dev: false

 /@wagmi/core@1.4.5(@types/react@18.0.25)(react@18.2.0)(typescript@5.2.2)(viem@0.3.50):
   resolution: {integrity: sha512-
N9luRb1Uk4tBN9kaYcQSWKE9AsRt/rvZaFt5IZech4JPzNN2sQlfhKd9GEjOXYRDqEPHdDvos7qyBKiDNTz
4GA==}
```

```
    peerDependencies:
      typescript: '>=5.0.4'
      viem: '>=0.3.35'
    peerDependenciesMeta:
      typescript:
        optional: true
    dependencies:
      '@wagmi/connectors': 3.1.3(@types/react@18.0.25)(react@18.2.0)(typescript@5.2.2)(viem@0.3.50)
      abitype: 0.8.7(typescript@5.2.2)
      eventemitter3: 4.0.7
      typescript: 5.2.2
      viem: 0.3.50(typescript@5.2.2)
      zustand: 4.4.4(@types/react@18.0.25)(react@18.2.0)
    transitivePeerDependencies:
      - '@react-native-async-storage/async-storage'
      - '@types/react'
      - bufferutil
      - encoding
      - immer
      - lokijs
      - react
      - supports-color
      - utf-8-validate
      - zod
    dev: false

  /@walletconnect/core@2.10.2:
    resolution: {integrity: sha512-
JQz/xp3SLEpTeRQctdck2ugSBVEpMxoSE+lFi2voJkZop1hv6P+uqr6E4PzjFluAjeAnKlT1xvra0aFWjPWVcw
==}
    dependencies:
      '@walletconnect/heartbeat': 1.2.1
      '@walletconnect/jsonrpc-provider': 1.0.13
      '@walletconnect/jsonrpc-types': 1.0.3
      '@walletconnect/jsonrpc-utils': 1.0.8
      '@walletconnect/jsonrpc-ws-connection': 1.0.13
      '@walletconnect/keyvaluestorage': 1.0.2
      '@walletconnect/logger': 2.0.1
      '@walletconnect/relay-api': 1.0.9
      '@walletconnect/relay-auth': 1.0.4
      '@walletconnect/safe-json': 1.0.2
      '@walletconnect/time': 1.0.2
      '@walletconnect/types': 2.10.2
      '@walletconnect/utils': 2.10.2
      events: 3.3.0
      lodash.isequal: 4.5.0
      uint8arrays: 3.1.1
    transitivePeerDependencies:
      - '@react-native-async-storage/async-storage'
      - bufferutil
      - lokijs
      - utf-8-validate
    dev: false

  /@walletconnect/crypto@1.0.3:
    resolution: {integrity: sha512-
+2jdORD7XQs76I2Odgr3wwrtyuLUXD/kprNVsjWRhhhdO9Mt6WqVzOPu0/t7OHSmgal8k7SoBQzUc5hu/8z
L/g==}
    dependencies:
      '@walletconnect/encoding': 1.0.2
      '@walletconnect/environment': 1.0.1
      '@walletconnect/randombytes': 1.0.3
      aes-js: 3.1.2
      hash.js: 1.1.7
      tslib: 1.14.1
    dev: false

  /@walletconnect/encoding@1.0.2:
    resolution: {integrity: sha512-
CrwSBrjqJ7rpGQcTL3kU+Ief+Bcuu9PH6JLOb+wM6NITX1GTxR/MfNwnQfhLKK6xpRAyj2/nM04OOH6wS8I
mag==}
    dependencies:
      is-typedarray: 1.0.0
      tslib: 1.14.1
      typedarray-to-buffer: 3.1.5
    dev: false

  /@walletconnect/environment@1.0.1:
    resolution: {integrity: sha512-
T426LLZtHj8e8rYnKfzsw1aG6+M0BT1ZxayMdv/p8yM0MU+eJDISqNY3/bccxRr4LrF9csq02Rhqt08Ibl0VRg
==}
    dependencies:
      tslib: 1.14.1
    dev: false

  /@walletconnect/ethereum-provider@2.10.2(@walletconnect/modal@2.6.2):
    resolution: {integrity: sha512-
QMYFZ6+rVq2CJLdIPdKK0j1Qm66UA27oQU5V2SrL8EVwl7wFfm0Bq7fnL+qAWeDpn612dNeNErpk/ROa1
zWlWg==}
    peerDependencies:
      '@walletconnect/modal': '>=2'
    peerDependenciesMeta:
      '@walletconnect/modal':
        optional: true
    dependencies:
      '@walletconnect/jsonrpc-http-connection': 1.0.7
      '@walletconnect/jsonrpc-provider': 1.0.13
      '@walletconnect/jsonrpc-types': 1.0.3
      '@walletconnect/jsonrpc-utils': 1.0.8
      '@walletconnect/modal': 2.6.2(@types/react@18.0.25)(react@18.2.0)
      '@walletconnect/sign-client': 2.10.2
      '@walletconnect/types': 2.10.2
      '@walletconnect/universal-provider': 2.10.2
      '@walletconnect/utils': 2.10.2
    events: 3.3.0
   transitivePeerDependencies:
    - '@react-native-async-storage/async-storage'
    - bufferutil
    - encoding
    - lokijs
    - utf-8-validate
   dev: false

  /@walletconnect/events@1.0.1:
   resolution: {integrity: sha512-
NPTqaoi0oPBVNuLv7qPaJazmGHs5JGyO8eEAk5VGKmJzDR7AHzD4k6ilox5kxk1iwiOnFopBOOMLs86Oa
76HpQ==}
   dependencies:
    keyvaluestorage-interface: 1.0.0
    tslib: 1.14.1
   dev: false

  /@walletconnect/heartbeat@1.2.1:
   resolution: {integrity: sha512-
yVzws616xsDLJxuG/28FqtZ5rzrTA4gUjdEMTbWB5Y8V1XHRmqq4efAxCw5ie7WjbXFSUyBHaWlMR+2/Cp
QC5Q==}
   dependencies:
    '@walletconnect/events': 1.0.1
    '@walletconnect/time': 1.0.2
    tslib: 1.14.1
   dev: false

  /@walletconnect/jsonrpc-http-connection@1.0.7:
   resolution: {integrity: sha512-
qlfh8fCfu8LOM9JRR9KE0s0wxP6ZG9/Jom8M0qsoIQeKF3Ni0FyV4V1qy/cc7nfI46SLQLSl4tgWSfLiE1swyQ
==}
   dependencies:
    '@walletconnect/jsonrpc-utils': 1.0.8
    '@walletconnect/safe-json': 1.0.2
    cross-fetch: 3.1.5
    tslib: 1.14.1
   transitivePeerDependencies:
    - encoding
   dev: false

  /@walletconnect/jsonrpc-provider@1.0.13:
   resolution: {integrity: sha512-
K73EpThqHnSR26gOyNEL+acEex3P7VWZe6KE12ZwKzAt2H4e5gldZHbjsu2QR9cLeJ8AXuO7kEMOIcRv1
QEc7g==}
   dependencies:
    '@walletconnect/jsonrpc-utils': 1.0.8
    '@walletconnect/safe-json': 1.0.2
    tslib: 1.14.1
   dev: false

  /@walletconnect/jsonrpc-types@1.0.3:
    resolution: {integrity: sha512-
iIQ8hboBl3o5ufmJ8cuduGad0CQm3ZlsHtujv9Eu16xq89q+BG7Nh5VLxxUgmtpnrePgFkTwXirCTkwJH1v+Y
w==}
    dependencies:
      keyvaluestorage-interface: 1.0.0
      tslib: 1.14.1
    dev: false

  /@walletconnect/jsonrpc-utils@1.0.8:
    resolution: {integrity: sha512-
vdeb03bD8VzJUL6ZtzRYsFMq1eZQcM3EAzT0a3st59dyLfJ0wq+tKMpmGH7HlB7waD858UWgfIcudbPFsbz
Vdw==}
    dependencies:
      '@walletconnect/environment': 1.0.1
      '@walletconnect/jsonrpc-types': 1.0.3
      tslib: 1.14.1
    dev: false

  /@walletconnect/jsonrpc-ws-connection@1.0.13:
    resolution: {integrity: sha512-
mfOM7uFH4lGtQxG+XklYuFBj6dwVvseTt5/ahOkkmpcAEgz2umuzu7fTR+h5EmjQBdrmYyEBOWADbeaFN
xdySg==}
    dependencies:
      '@walletconnect/jsonrpc-utils': 1.0.8
      '@walletconnect/safe-json': 1.0.2
      events: 3.3.0
      tslib: 1.14.1
      ws: 7.5.9
    transitivePeerDependencies:
      - bufferutil
      - utf-8-validate
    dev: false

  /@walletconnect/keyvaluestorage@1.0.2:
    resolution: {integrity: sha512-
U/nNG+VLWoPFdwwKx0oliT4ziKQCEoQ27L5Hhw8YOFGA2Po9A9pULUYNWhDgHkrb0gYDNt//X7wABcE
WWBd3FQ==}
    peerDependencies:
      '@react-native-async-storage/async-storage': 1.x
      lokijs: 1.x
    peerDependenciesMeta:
      '@react-native-async-storage/async-storage':
        optional: true
      lokijs:
        optional: true
    dependencies:
      safe-json-utils: 1.1.1
      tslib: 1.14.1
    dev: false

  /@walletconnect/legacy-client@2.0.0:
    resolution: {integrity: sha512-
```

```
v5L7rYk9loVnfvUf0mF+76bUPFaU5/Vh7mzL6/950CD/yoGdzYZ3Kj+L7mkC6HPMEGeQsBP1+sqBuiVGZ/a
ODA==}
    dependencies:
      '@walletconnect/crypto': 1.0.3
      '@walletconnect/encoding': 1.0.2
      '@walletconnect/jsonrpc-utils': 1.0.8
      '@walletconnect/legacy-types': 2.0.0
      '@walletconnect/legacy-utils': 2.0.0
      '@walletconnect/safe-json': 1.0.2
      '@walletconnect/window-getters': 1.0.1
      '@walletconnect/window-metadata': 1.0.1
      detect-browser: 5.3.0
      query-string: 6.13.5
    dev: false

  /@walletconnect/legacy-modal@2.0.0:
    resolution: {integrity: sha512-
jckNd8lMhm4X7dX9TDdxM3bXKJnaqkRs6K2Mo5j6GmbIF9Eyx40jZ5+q457RVxvM6ciZEDT5s1wBHWdWo
Oo+9Q==}
    dependencies:
      '@walletconnect/legacy-types': 2.0.0
      '@walletconnect/legacy-utils': 2.0.0
      copy-to-clipboard: 3.3.3
      preact: 10.18.1
      qrcode: 1.5.1
    dev: false

  /@walletconnect/legacy-provider@2.0.0:
    resolution: {integrity: sha512-
A8xPebMI1A+50HbWwTpFCbwP7G+1NGKdTKyg8BUUg3h3Y9JucpC1W6w/x0v1Xw7qFEqQnz74LoIN/A3y
tH9xrQ==}
    dependencies:
      '@walletconnect/jsonrpc-http-connection': 1.0.7
      '@walletconnect/jsonrpc-provider': 1.0.13
      '@walletconnect/legacy-client': 2.0.0
      '@walletconnect/legacy-modal': 2.0.0
      '@walletconnect/legacy-types': 2.0.0
      '@walletconnect/legacy-utils': 2.0.0
    transitivePeerDependencies:
      - encoding
    dev: false

  /@walletconnect/legacy-types@2.0.0:
    resolution: {integrity: sha512-
sOVrA7HUdbI1OwKyPOQU0/DdvTSVFlsXWpAk2K2WvP2erTkBWPMTJq6cv2BmKdoJ3p6gLApT7sd+jHi3
OF71uw==}
    dependencies:
      '@walletconnect/jsonrpc-types': 1.0.3
    dev: false

  /@walletconnect/legacy-utils@2.0.0:
    resolution: {integrity: sha512-
```

```
CPWxSVVXw0kgNCxvU126g4GiV3mzXmC8IPJ15twE46aJ1FX+RHEIfAzFMFz2F2+fEhBxL63A7dwNQKD
XorRPcQ==}
    dependencies:
      '@walletconnect/encoding': 1.0.2
      '@walletconnect/jsonrpc-utils': 1.0.8
      '@walletconnect/legacy-types': 2.0.0
      '@walletconnect/safe-json': 1.0.2
      '@walletconnect/window-getters': 1.0.1
      '@walletconnect/window-metadata': 1.0.1
      detect-browser: 5.3.0
      query-string: 6.13.5
    dev: false

  /@walletconnect/logger@2.0.1:
    resolution: {integrity: sha512-
SsTKdsgWm+oDTBeNE/zHxxr5eJfZmE9/5yp/Ku+zJtcTAjELb3DXueWkDXmE9h8uHIbJzIb5wj5lPdzyrjT6hQ
==}
    dependencies:
      pino: 7.11.0
      tslib: 1.14.1
    dev: false

  /@walletconnect/modal-core@2.6.2(@types/react@18.0.25)(react@18.2.0):
    resolution: {integrity: sha512-
cv8ibvdOJQv2B+nyxP9IIFdxvQznMz8OOr/oR/AaUZym4hjXNL/l1a2UlSQBXrVjo3xxbouMxLb3kBsHoYP2C
A==}
    dependencies:
      valtio: 1.11.2(@types/react@18.0.25)(react@18.2.0)
    transitivePeerDependencies:
      - '@types/react'
      - react
    dev: false

  /@walletconnect/modal-ui@2.6.2(@types/react@18.0.25)(react@18.2.0):
    resolution: {integrity: sha512-
rbdstM1HPGvr7jprQkyPggX7rP4XiCG85ZA+zWBEX0dVQg8PpAgRUqpeub4xQKDgY7pY/xLRXSiCVdWGq
vG2HA==}
    dependencies:
      '@walletconnect/modal-core': 2.6.2(@types/react@18.0.25)(react@18.2.0)
      lit: 2.8.0
      motion: 10.16.2
      qrcode: 1.5.3
    transitivePeerDependencies:
      - '@types/react'
      - react
    dev: false

  /@walletconnect/modal@2.6.2(@types/react@18.0.25)(react@18.2.0):
    resolution: {integrity: sha512-
eFopgKi8AjKf/0U4SemvcYw9zlLpx9njVN8sf6DAkowC2Md0gPU/UNEbH1Wwj407pEKnEds98pKWib1NN1A
CoA==}
    dependencies:
```

```
    '@walletconnect/modal-core': 2.6.2(@types/react@18.0.25)(react@18.2.0)
    '@walletconnect/modal-ui': 2.6.2(@types/react@18.0.25)(react@18.2.0)
  transitivePeerDependencies:
    - '@types/react'
    - react
  dev: false

 /@walletconnect/randombytes@1.0.3:
  resolution: {integrity: sha512-
35lpzxcHFbTN3ABefC9W+uBpNZI1GC4Wpx0ed30gibfO/y9oLdy1NznbV96HARQKSBV9J9M/rrtIvf6a23jfYw
==}
  dependencies:
    '@walletconnect/encoding': 1.0.2
    '@walletconnect/environment': 1.0.1
    randombytes: 2.1.0
    tslib: 1.14.1
  dev: false

 /@walletconnect/relay-api@1.0.9:
  resolution: {integrity: sha512-
Q3+rylJOqRkO1D9Su0DPE3mmznbAalYapJ9qmzDgK28mYF9alcP3UwG/og5V7l7CFOqzCLi7B8BvcBUrp
Dj0Rg==}
  dependencies:
    '@walletconnect/jsonrpc-types': 1.0.3
    tslib: 1.14.1
  dev: false

 /@walletconnect/relay-auth@1.0.4:
  resolution: {integrity: sha512-
kKJcS6+WxYq5kshpPaxGHdwf5y98ZwbfuS4EE/NkQzqrDFm5Cj+dP8LofzWvjrrLkZq7Afy7WrQMXdLy8Sx7
HQ==}
  dependencies:
    '@stablelib/ed25519': 1.0.3
    '@stablelib/random': 1.0.2
    '@walletconnect/safe-json': 1.0.2
    '@walletconnect/time': 1.0.2
    tslib: 1.14.1
    uint8arrays: 3.1.1
  dev: false

 /@walletconnect/safe-json@1.0.2:
  resolution: {integrity: sha512-
Ogb7I27kZ3LPC3ibn8ldyUr5544t3/STow9+lzz7Sfo808YD7SBWk7SAsdBFlYgP2zDRy2hS3sKRcuSRM0OT
mA==}
  dependencies:
    tslib: 1.14.1
  dev: false

 /@walletconnect/sign-client@2.10.2:
  resolution: {integrity: sha512-
vviSLV3f92I0bReX+OLr1HmbH0uIzYEQQFd1MzIfDk9PkfFT/LLAHhUnDaIAMkIdippqDcJia+5QEtT4JihL3Q
==}
```

```
    dependencies:
      '@walletconnect/core': 2.10.2
      '@walletconnect/events': 1.0.1
      '@walletconnect/heartbeat': 1.2.1
      '@walletconnect/jsonrpc-utils': 1.0.8
      '@walletconnect/logger': 2.0.1
      '@walletconnect/time': 1.0.2
      '@walletconnect/types': 2.10.2
      '@walletconnect/utils': 2.10.2
      events: 3.3.0
    transitivePeerDependencies:
      - '@react-native-async-storage/async-storage'
      - bufferutil
      - lokijs
      - utf-8-validate
    dev: false

  /@walletconnect/time@1.0.2:
    resolution: {integrity: sha512-uzdd9woDcJ1AaBZRhqy5rNC9laqWGErfc4dxA9a87mPdKOgWMD85mcFo9dIYIts/Jwocfwn07EC6EzclKubk/g==}
    dependencies:
      tslib: 1.14.1
    dev: false

  /@walletconnect/types@2.10.2:
    resolution: {integrity: sha512-luNV+07Wdla4STi9AejseCQY31tzWKQ5a7C3zZZaRK/di+rFaAAb7YW04OP4klE7tw/mJRGPTlekZElmHxO8kQ==}
    dependencies:
      '@walletconnect/events': 1.0.1
      '@walletconnect/heartbeat': 1.2.1
      '@walletconnect/jsonrpc-types': 1.0.3
      '@walletconnect/keyvaluestorage': 1.0.2
      '@walletconnect/logger': 2.0.1
      events: 3.3.0
    transitivePeerDependencies:
      - '@react-native-async-storage/async-storage'
      - lokijs
    dev: false

  /@walletconnect/universal-provider@2.10.2:
    resolution: {integrity: sha512-wFgI0LbQ3D56sgaUMsgOHCM5m8WLxiC71BGuCKQfApgsbNMVKugYVy2zWHyUyi8sqTQHI+uSaVpDev4UHq9LEw==}
    dependencies:
      '@walletconnect/jsonrpc-http-connection': 1.0.7
      '@walletconnect/jsonrpc-provider': 1.0.13
      '@walletconnect/jsonrpc-types': 1.0.3
      '@walletconnect/jsonrpc-utils': 1.0.8
      '@walletconnect/logger': 2.0.1
      '@walletconnect/sign-client': 2.10.2
      '@walletconnect/types': 2.10.2
      '@walletconnect/utils': 2.10.2
      events: 3.3.0
    transitivePeerDependencies:
      - '@react-native-async-storage/async-storage'
      - bufferutil
      - encoding
      - lokijs
      - utf-8-validate
    dev: false

  /@walletconnect/utils@2.10.2:
    resolution: {integrity: sha512-
syxXRpc2yhSknMu3IfiBGobxOY7fLfLTJuw+ppKaeO6WUdZpIit3wfuGOcc0Ms3ZPFCrGfyGOoZsCvgdXtptR
g==}
    dependencies:
      '@stablelib/chacha20poly1305': 1.0.1
      '@stablelib/hkdf': 1.0.1
      '@stablelib/random': 1.0.2
      '@stablelib/sha256': 1.0.1
      '@stablelib/x25519': 1.0.3
      '@walletconnect/relay-api': 1.0.9
      '@walletconnect/safe-json': 1.0.2
      '@walletconnect/time': 1.0.2
      '@walletconnect/types': 2.10.2
      '@walletconnect/window-getters': 1.0.1
      '@walletconnect/window-metadata': 1.0.1
      detect-browser: 5.3.0
      query-string: 7.1.3
      uint8arrays: 3.1.1
    transitivePeerDependencies:
      - '@react-native-async-storage/async-storage'
      - lokijs
    dev: false

  /@walletconnect/window-getters@1.0.1:
    resolution: {integrity: sha512-
vHp+HqzGxORPAN8gY03qnbTMnhqIwjeRJNOMOAzePRg4xVEEE2WvYsI9G2NMjOknA8hnuYbU3/hwLcK
bjhc8+Q==}
    dependencies:
      tslib: 1.14.1
    dev: false

  /@walletconnect/window-metadata@1.0.1:
    resolution: {integrity: sha512-
9koTqyGrM2cqFRW517BPY/iEtUDx2r1+Pwwu5m7sJ7ka79wi3EyqhqcICk/yDmv6jAS1rjKgTKXIEhanYjijcA==
}
    dependencies:
      '@walletconnect/window-getters': 1.0.1
      tslib: 1.14.1
    dev: false
  /@web3modal/core@2.4.1(react@18.2.0):
    resolution: {integrity: sha512-
v6Y/eQJSI2YfUTv8rGqjFabqdk3ZPjx6Fe7j5Q8fw0ZWF1YRGM3mySG457qtKQ7D7E1kNKA3BHbaOZ3pgQ
oG6A==}
    dependencies:
      buffer: 6.0.3
      valtio: 1.10.5(react@18.2.0)
    transitivePeerDependencies:
      - react
    dev: false

  /@web3modal/ethereum@2.4.1(@wagmi/core@1.4.5)(viem@0.3.50):
    resolution: {integrity: sha512-
sp6/viEpL1yQRyQXSP76martUqr+EZ1EFaS79Z1kvdOlHg6A5BEiOOXGI5KwNTlGlaAWK2GOggoytsub1gV
a5A==}
    deprecated: Please use new @web3modal/wagmi package
    peerDependencies:
      '@wagmi/core': '>=1'
      viem: '>=0.3'
    dependencies:
      '@wagmi/core': 1.4.5(@types/react@18.0.25)(react@18.2.0)(typescript@5.2.2)(viem@0.3.50)
      viem: 0.3.50(typescript@5.2.2)
    dev: false

  /@web3modal/react@2.4.1(react-dom@18.2.0)(react@18.2.0):
    resolution: {integrity: sha512-
B0qAtGYgO3gx4ZJyJWDrN1pn32qUr9p4fJeZ+FiorSJWVUzSDEPswIW6qCIGV8cnosXqV3xMqZ2Mq+0ivH
8luw==}
    deprecated: Please use new @web3modal/wagmi package
    peerDependencies:
      react: '>=17'
      react-dom: '>=17'
    dependencies:
      '@web3modal/core': 2.4.1(react@18.2.0)
      '@web3modal/ui': 2.4.1(react@18.2.0)
      react: 18.2.0
      react-dom: 18.2.0(react@18.2.0)
    dev: false

  /@web3modal/ui@2.4.1(react@18.2.0):
    resolution: {integrity: sha512-
x1ceyd3mMJsIHs5UUTLvE+6qyCjhyjL6gB/wVmTDbwASHSQIVyshQJ+s7BwIEMP/pbAsYDg+/M8EiUuE+/E
/kg==}
    dependencies:
      '@web3modal/core': 2.4.1(react@18.2.0)
      lit: 2.7.4
      motion: 10.15.5
      qrcode: 1.5.3
    transitivePeerDependencies:
      - react
    dev: false
  /JSONStream@1.3.5:
    resolution: {integrity: sha512-
E+iruNOY8VV9s4JEbe1aNEm6MiszPRr/UfcHMz0TQh1BXSxHK+ASV1R6W4HpjBhSeS+54PIsAMCBmwD
06LLsqQ==}
    hasBin: true
    dependencies:
      jsonparse: 1.3.1
      through: 2.3.8
    dev: false

  /abbrev@1.1.1:
    resolution: {integrity: sha512-
nne9/IiQ/hzIhY6pdDnbBtz7DjPTKrY00P/zvPSm5pOFkl6xuGrGnXn/VtTNNfNtAfZ9/1RtehkszU9qcTii0Q==}
    dev: true

  /abitype@0.8.7(typescript@5.2.2):
    resolution: {integrity: sha512-
wQ7hV8Yg/yKmGyFpqrNZufCxbszDe5es4AZGYPBitocfSqXtjrTG9JMWFcc4N30ukl2ve48aBTwt7NJxVQdU
3w==}
    peerDependencies:
      typescript: '>=5.0.4'
      zod: ^3 >=3.19.1
    peerDependenciesMeta:
      zod:
        optional: true
    dependencies:
      typescript: 5.2.2
    dev: false

  /abitype@0.9.8(typescript@5.2.2):
    resolution: {integrity: sha512-
puLifILdm+8sjyss4S+fsUN09obiT1g2YW6CtcQF+QDzxR0euzgEB29MZujC6zMk2a6SVmtttq1fc6+YFA7WY
Q==}
    peerDependencies:
      typescript: '>=5.0.4'
      zod: ^3 >=3.19.1
    peerDependenciesMeta:
      typescript:
        optional: true
      zod:
        optional: true
    dependencies:
      typescript: 5.2.2
    dev: false

  /abort-controller@3.0.0:
    resolution: {integrity: sha512-
h8lQ8tacZYnR3vNQTgibj+tODHI5/+l06Au2Pcriv/Gmet0eaj4TwWH41sO9wnHDiQsEj19q0drzdWdeAHtweg==
}
    engines: {node: '>=6.5'}
    dependencies:
      event-target-shim: 5.0.1
    dev: false

  /abortcontroller-polyfill@1.7.5:
    resolution: {integrity: sha512-
JMJ5soJWP18htbbxJjG7bG6yuI6pRhgJ0scHHTfkUjf6wjP912xZWvM+A4sJK3gqd9E8fcPbDnOefbA9Th/FIQ
==}
    dev: false

  /abstract-leveldown@2.6.3:
    resolution: {integrity: sha512-
2++wDf/DYqkPR3o5tbfdhF96EfMApo1GpPfzOsR/ZYXdkSmELlvOOEAl9iKkRsktMPHdGjO4rtkBpf2I7TiTeA
==}
    dependencies:
      xtend: 4.0.2
    dev: false

  /abstract-leveldown@2.7.2:
    resolution: {integrity: sha512-
+OVvxH2rHVEhWLdbudP6p0+dNMXu8JA1CbhP19T8paTYAcX7oJ4OVjT+ZUVpv7mITxXHqDMej+GdqXB
mXkw09w==}
    dependencies:
      xtend: 4.0.2
    dev: false

  /abstract-logging@2.0.1:
    resolution: {integrity: sha512-
2BjRTZxTPvheOvGbBslFSYOUkr+SjPtOnrLP33f+VIWLzezQpZcqVg7ja3L4dBXmzzgwT+a029jRx5PCi3Jui
A==}
    dev: false

  /accepts@1.3.8:
    resolution: {integrity: sha512-
PYAthTa2m2VKxuvSD3DPC/Gy+U+sOA1LAuT8mkmRuvw+NACSaeXEQ+NHcVF7rONl6qcaxV3Uuemwa
wk+7+SJLw==}
    engines: {node: '>= 0.6'}
    dependencies:
      mime-types: 2.1.35
      negotiator: 0.6.3
    dev: false

  /acorn-jsx@5.3.2(acorn@8.11.2):
    resolution: {integrity: sha512-
rq9s+JNhf0IChjtDXxllJ7g41oZk5SlXtp0LHwyA5cejwn7vKmKp4pPri6YEePv2PU65sAsegbXtIinmDFDXgQ==
}
    peerDependencies:
      acorn: ^6.0.0 || ^7.0.0 || ^8.0.0
    dependencies:
      acorn: 8.11.2
    dev: true

  /acorn-walk@8.3.0:
    resolution: {integrity: sha512-
FS7hV565M5l1R08MXqo8odwMTB02C2UqzB17RVgu9EyuYFBqJZ3/ZY97sQD5FewVu1UyDFc1yztUDrAw
T0EypA==}
    engines: {node: '>=0.4.0'}

  /acorn@8.11.2:
    resolution: {integrity: sha512-
nc0Axzp/0FILLEVsm4fNwLCwMttvhEI263QtVPQcbpfZZ3ts0hLsZGOpE6czNlid7CJ9MlyH8reXkpsf3YUY4w
==}
    engines: {node: '>=0.4.0'}
    hasBin: true

  /aes-js@3.1.2:
    resolution: {integrity: sha512-
e5pEa2kBnBOgR4Y/p20pskXI74UEz7de8ZGVo58asOtvSVG5YAbJeELPZxOmt+Bnz3rX753YKhfIn4X4l1P
PRQ==}
    dev: false

  /aes-js@4.0.0-beta.5:
    resolution: {integrity: sha512-
G965FqalsNyrPqgEGON7nIx1e/OVENSgiEIzyC63haUMuvNnwIgIjMs52hlTCKhkBny7A2ORNlfY9Zu+jmGk1
Q==}
    dev: false

  /agentkeepalive@4.5.0:
    resolution: {integrity: sha512-
5GG/5IbQQpC9FpkRGsSvZI5QYeSCzlJHdpBQntCsuTOxhKD8lqKhrleg2Yi7yvMIf82Ycmmqln9U8V9qwEiJe
w==}
    engines: {node: '>= 8.0.0'}
    dependencies:
      humanize-ms: 1.2.1
    dev: false

  /ajv-formats@2.1.1(ajv@8.12.0):
    resolution: {integrity: sha512-
Wx0Kx52hxE7C18hkMEggYlEifqWZtYaRgouJor+WMdPnQyEK13vgEWyVNup7SoeeoLMsr4kf5h6dOW11I1
5MUA==}
    peerDependencies:
      ajv: ^8.0.0
    peerDependenciesMeta:
      ajv:
        optional: true
    dependencies:
      ajv: 8.12.0
    dev: false

  /ajv@6.12.6:
    resolution: {integrity: sha512-
j3fVLgvTo527anyYyJOGTYJbG+vnnQYvE0m5mmkc1TK+nxAppkCLMIL0aZ4dblVCNoGShhm+kzE4ZUykB
oMg4g==}
    dependencies:
      fast-deep-equal: 3.1.3
      fast-json-stable-stringify: 2.1.0
      json-schema-traverse: 0.4.1
      uri-js: 4.4.1

  /ajv@8.12.0:
    resolution: {integrity: sha512-
sRu1kpcO9yLtYxBKvqfTeh9KzZEwO3STyX1HT+4CaDzC6HpTGYhIhPIzj9XuKU7KYDwnaeh5hcOwjy1QuJ
zBPA==}
    dependencies:
      fast-deep-equal: 3.1.3
      json-schema-traverse: 1.0.0
      require-from-string: 2.0.2
      uri-js: 4.4.1
    dev: false

  /ansi-regex@5.0.1:
    resolution: {integrity: sha512-
quJQXlTSUGL2LH9SUXo8VwsY4soanhgo6LNSm84E1LBcE8s3O0wpdiRzyR9z/ZZJMlMWv37qOOb9pdJl
MUEKFQ==}
    engines: {node: '>=8'}

  /ansi-styles@3.2.1:
    resolution: {integrity: sha512-
VT0ZI6kZRdTh8YyJw3SMbYm/u+NqfsAxEpWO0Pf9sq8/e94WxxOpPKx9FR1FlyCtOVDNOQ+8ntlqFxiRc+r
5qA==}
    engines: {node: '>=4'}
    dependencies:
      color-convert: 1.9.3

  /ansi-styles@4.3.0:
    resolution: {integrity: sha512-
zbB9rCJAT1rbjiVDb2hqKFHNYLxgtk8NURxZ3IZwD3F6NtxbXZQCnnSi1Lkx+IDohdPlFp222wVALIheZJQS
Eg==}
    engines: {node: '>=8'}
    dependencies:
      color-convert: 2.0.1

  /any-promise@1.3.0:
    resolution: {integrity: sha512-
7UvmKalWRt1wgjL1RrGxoSJW/0QZFIegpeGvZG9kjp8vrRu55XTHbwnqq2GpXm9uLbcuhxm3IqX9OB4MZ
R1b2A==}
    dev: false

  /anymatch@3.1.3:
    resolution: {integrity: sha512-
KMReFUr0B4t+D+OBkjR3KYqvocp2XaSzO55UcB6mgQMd3KbcE+mWTyvVV7D/zsdEbNnV6acZUutkiHQX
vTr1Rw==}
    engines: {node: '>= 8'}
    dependencies:
      normalize-path: 3.0.0
      picomatch: 2.3.1

  /apg-js@4.2.1:
    resolution: {integrity: sha512-
S3unf9jb9XrhmZYC27L5yits1p/+HqN6lksrkqKEv1U0Ytu+ENkIkOMV5+iutFjBaY9JTUCDQFS1svHsy3uNZA
==}
    dev: false

  /archy@1.0.0:
    resolution: {integrity: sha512-
Xg+9RwCg/0p32teKdGMPTPnVXKD0w3DfHnFTficozsAgsvq2XenPJq/MYpzzQ/v8zrOyJn6Ds39VA4JIDwFf
qw==}
    dev: false

  /arg@4.1.3:
    resolution: {integrity: sha512-
58S9QDqG0Xx27YwPSt9fJxivjYl432YCwfDMfZ+71RAqUrZef7LrKQZ3LHLOwCS4FLNBplP533Zx895SeOC
HvA==}

  /arg@5.0.2:
    resolution: {integrity: sha512-
PYjyFOLKQ9y57JvQ6QLo8dAgNqswh8M1RMJYdQduT6xbWSgK36P/Z/v+p888pM69jMMfS8Xd8F6I1kQ/I9
HUGg==}
    dev: false

  /argparse@2.0.1:
    resolution: {integrity: sha512-
8+9WqebbFzpX9OR+Wa6O29asIogeRMzcGtAINdpMHHyAg10f05aSFVBbcEqGf/PXw1EjAZ+q2/bEBg3Dv
urK3Q==}
    dev: true

  /aria-query@5.3.0:
    resolution: {integrity: sha512-
b0P0sZPKtyu8HkeRAfCq0IfURZK+SuwMjY1UXGBU27wpAiTwQAIlq56IbIO+ytk/JjS1fMR14ee5WBBfKi5J6
A==}
    dependencies:
      dequal: 2.0.3
    dev: true

  /array-back@3.1.0:
    resolution: {integrity: sha512-
TkuxA4UCOvxuDK6NZYXCalszEzj+TLszyASooky+i742l9TqsOdYCMJJupxRic61hwquNtppB3hgcuq9SVSH
1Q==}
    engines: {node: '>=6'}

  /array-back@4.0.2:
    resolution: {integrity: sha512-
NbdMezxqf94cnNfWLL7V/im0Ub+Anbb0IoZhvzie8+4HJ4nMQuzHuy49FkGYCJK2yAloZ3meiB6AVMClbrI1v
g==}
    engines: {node: '>=8'}

  /array-buffer-byte-length@1.0.0:
    resolution: {integrity: sha512-
LPuwb2P+NrQw3XhxGc36+XSvuBPopovXYTR9Ew++Du9Yb/bx5AzBfrIsBoj0EZUifjQU+sHL21sseZ3jerWO
/A==}
```

```
    dependencies:
      call-bind: 1.0.5
      is-array-buffer: 3.0.2
    dev: true

  /array-flatten@1.1.1:
    resolution: {integrity: sha512-
PCVAQswWemu6UdxsDFFX/+gVeYqKAod3D3UVm91jHwynguOwAvYPhx8nNlM++NqRcK6CxxpUafjmhId
KiHibqg==}
    dev: false

  /array-includes@3.1.7:
    resolution: {integrity: sha512-
dlcsNBIiWhPkHdOEEKnehA+RNUWDc4UqFtnIXU4uuYDPtA4LDkr7qip2p0VvFAEXNDr0yWZ9PJyIRiGjRL
QzwQ==}
    engines: {node: '>= 0.4'}
    dependencies:
      call-bind: 1.0.2
      define-properties: 1.2.1
      es-abstract: 1.22.3
      get-intrinsic: 1.2.2
      is-string: 1.0.7
    dev: true

  /array-union@2.1.0:
    resolution: {integrity: sha512-
HGyxoOTYUyCM6stUe6EJgnd4EoewAI7zMdfqO+kGjnlZmBDz/cR5pf8r/cR4Wq60sL/p0IkcjUEEPwS3GFrIy
w==}
    engines: {node: '>=8'}
    dev: true

  /array.prototype.findlastindex@1.2.3:
    resolution: {integrity: sha512-
LzLoiOMAxvy+Gd3BAq3B7VeIgPdo+Q8hthvKtXybMvRV0jrXfJM/t8mw7nNlpEcVlVUnCnM2KSX4XU5Hmpo
dOA==}
    engines: {node: '>= 0.4'}
    dependencies:
      call-bind: 1.0.2
      define-properties: 1.2.1
      es-abstract: 1.22.3
      es-shim-unscopables: 1.0.2
      get-intrinsic: 1.2.2
    dev: true

  /array.prototype.flat@1.3.2:
    resolution: {integrity: sha512-
djYB+Zx2vLewY8RWlNCUdHjDXs2XOgm602S9E7P/UpHgfeHL00cRiIF+IN/G/aUJ7kGPb6yO/ErDI5V2s8iyc
A==}
    engines: {node: '>= 0.4'}
    dependencies:
      call-bind: 1.0.2
      define-properties: 1.2.1
      es-abstract: 1.22.3
      es-shim-unscopables: 1.0.2
    dev: true

  /array.prototype.flatmap@1.3.2:
    resolution: {integrity: sha512-
Ewyx0c9PmpcsByhSW4r+9zDU7sGjFc86qf/kKtuSCRdhfbk0SNLLkaT5qvcHnRGgc5NP/ly/y+qkXkqONX54
CQ==}
    engines: {node: '>= 0.4'}
    dependencies:
      call-bind: 1.0.2
      define-properties: 1.2.1
      es-abstract: 1.22.3
      es-shim-unscopables: 1.0.2
    dev: true

  /array.prototype.tosorted@1.1.2:
    resolution: {integrity: sha512-
HuQCHOlk1Weat5jzStICBCd83NxiIMwqDg/dHEsoefabn/hJRj5pVdWcPUSpRrwhwxZOsQassMpgN/xRYFB
MIg==}
    dependencies:
      call-bind: 1.0.2
      define-properties: 1.2.1
      es-abstract: 1.22.3
      es-shim-unscopables: 1.0.2
      get-intrinsic: 1.2.2
    dev: true

  /arraybuffer.prototype.slice@1.0.2:
    resolution: {integrity: sha512-
yMBKppFur/fbHu9/6USUe03bZ4knMYiwFBcyiaXB8Go0qNehwX6inYPzK9U0NeQvGxKthcmHcaR8P5MStS
RBAw==}
    engines: {node: '>= 0.4'}
    dependencies:
      array-buffer-byte-length: 1.0.0
      call-bind: 1.0.5
      define-properties: 1.2.1
      es-abstract: 1.22.3
      get-intrinsic: 1.2.2
      is-array-buffer: 3.0.2
      is-shared-array-buffer: 1.0.2
    dev: true

  /asap@2.0.6:
    resolution: {integrity: sha512-
BSHWgDSAiKs50o2Re8ppvp3seVHXSRM44cdSsT9FfNEUUZLOGWVCsiWaRPWM1Znn+mqZ1OfVZ3z3D
WEzSp7hRA==}
    dev: false

  /asn1.js@5.4.1:
    resolution: {integrity: sha512-
+I//4cYPccV8LdmBLiX8CYvf9Sp3vQsrqu2QNXRcrbiWvcx/UdlFiqUJJzxRQxgsZmvhXhn4cSKeSmoFjVdupA
==}
    dependencies:
      bn.js: 4.12.0
      inherits: 2.0.4
      minimalistic-assert: 1.0.1
      safer-buffer: 2.1.2
    dev: false

  /asn1@0.2.6:
    resolution: {integrity: sha512-
ix/FxPn0MDjeyJ7i/yoHGFt/EX6LyNbxSEhPPXODPL+KB0VPk86UYfL0lMdy+KCnv+fmvIzySwaK5COwqVb
WTQ==}
    dependencies:
      safer-buffer: 2.1.2
    dev: false

  /assert-plus@1.0.0:
    resolution: {integrity: sha512-
NfJ4UzBCcQGLDlQq7nHxH+tv3kyZ0hHQqF5BO6J7tNJeP5do1llPr8dZ8zHonfhAu0PHAdMkSo+8o0wxg9lZ
Ww==}
    engines: {node: '>=0.8'}
    dev: false

  /ast-types-flow@0.0.7:
    resolution: {integrity: sha512-
eBvWn1lvIApYMhzQMsu9ciLfkBY499mFZlNqG+/9WR7PVlroQw0vG30cOQQbaKz3sCEc44TAOu2ykzqXS
Nnwag==}
    dev: true

  /async-eventemitter@0.2.4:
    resolution: {integrity: sha512-
pd20BwL7Yt1zwDFy+8MX8F1+WCT8aQeKj0kQnTrH9WaeRETlRamVhD0JtRPmrV4GfOJ2F9CvdQkZeZhn
h2TuHw==}
    dependencies:
      async: 2.6.4
    dev: false

  /async-limiter@1.0.1:
    resolution: {integrity: sha512-
csOlWGAcRFJaI6m+F2WKdnMKr4HhdhFVBk0H/QbJFMCr+uO2kwohwXQPxw/9OCxp05r5ghVBFSyioixx3
gfkNQ==}
    dev: false

  /async-mutex@0.2.6:
    resolution: {integrity: sha512-
Hs4R+4SPgamu6rSGW8C7cV9gaWUKEHykfzCCvIRuaVv636Ju10ZdeUbvb4TBEW0INuq2DHZqXbK4Nd3
yG4RaRw==}
    dependencies:
      tslib: 2.4.1
    dev: false

  /async@1.5.2:
    resolution: {integrity: sha512-
nSVgobk4rv61R9PUSDtYt7mPVB2olxNR5RWJcAsH676/ef11bUZwvu7+RGYrYauVdDPcO519v68wRhXQt
xsV9w==}
    dev: false

  /async@2.6.4:
    resolution: {integrity: sha512-
mzo5dfJYwAn29PeiJ0zvwTo04zj8HDJj0Mn8TD7sno7q12prdbnasKJHhkm2c1LgrhlJ0teaea8860oxi51mGA==
}
    dependencies:
      lodash: 4.17.21
    dev: false

  /asynciterator.prototype@1.0.0:
    resolution: {integrity: sha512-
wwHYEIS0Q80f5mosx3L/dfG5t5rjEa9Ft51GTaNt862EnpyGHpgz2RkZvLPp1oF5TnAiTohkEKVEu8pQPJI7V
g==}
    dependencies:
      has-symbols: 1.0.3
    dev: true

  /asynckit@0.4.0:
    resolution: {integrity: sha512-
Oei9OH4tRh0YqU3GxhX79dM/mwVgvbZJaSNaRk+bshkj0S5cfHcgYakreBjrHwatXKbz+IoIdYLxrKim2MjW0
Q==}

  /atomic-sleep@1.0.0:
    resolution: {integrity: sha512-
kNOjDqAh7px0XWNI+4QbzoiR/nTkHAWNud2uvnJquD1/x5a7EQZMJT0AczqK0Qn67oY/TTQ1LbUKajZpp3I
9tQ==}
    engines: {node: '>=8.0.0'}
    dev: false

  /autoprefixer@10.4.16(postcss@8.4.31):
    resolution: {integrity: sha512-
7vd3UC6xKp0HLfua5IjZlcXvGAGy7cBAXTg2lyQ/8WpNhd6SiZ8Be+xm3FyBSYJx5GKcpRCzBh7RH4/0dnY
+uQ==}
    engines: {node: ^10 || ^12 || >=14}
    hasBin: true
    peerDependencies:
      postcss: ^8.1.0
    dependencies:
      browserslist: 4.22.1
      caniuse-lite: 1.0.30001554
      fraction.js: 4.3.7
      normalize-range: 0.1.2
      picocolors: 1.0.0
      postcss: 8.4.31
      postcss-value-parser: 4.2.0
    dev: false

  /available-typed-arrays@1.0.5:
    resolution: {integrity: sha512-DMD0KiN46eipeziST1LPP/STfDU0sufISXmjSgvVsoU2tqxctQeASejWcfNtxYKqETM1UxQ8sp2OrSBWpHY6sw==}
    engines: {node: '>= 0.4'}

  /avvio@8.2.1:
    resolution: {integrity: sha512-TAlMYvOuwGyLK3PfBb5WKBXZmXz2fVCgv23d6zZFdle/q3gPjmxBaeuC0pY0Dzs5PWMSgfqqEZkrye19GlDTgw==}
    dependencies:
      archy: 1.0.0
      debug: 4.3.4
      fastq: 1.15.0
    transitivePeerDependencies:
      - supports-color
    dev: false

  /aws-sign2@0.7.0:
    resolution: {integrity: sha512-08kcGqnYf/YmjoRhfxyu+CLxBjUtHLXLXX/vUfx9l2LYzG3c1m61nrpyFUZI6zeS+Li/wWMMidD9KgrqtGq3mA==}
    dev: false

  /aws4@1.12.0:
    resolution: {integrity: sha512-NmWvPnx0F1SfrQbYwOi7OeaNGokp9XhzNioJ/CSBs8Qa4vxug81mhJEAVZwxXuBmYB5KDRfMq/F3RR0bIU7sWg==}
    dev: false

  /axe-core@4.8.2:
    resolution: {integrity: sha512-/dlp0fxyM3R8YW7MFzaHWXrf4zzbr0vaYb23VBFCl83R7nWNPg/yaQw2Dc8jzCMmDVLhSdzH8MjrsuIUuvX+6g==}
    engines: {node: '>=4'}
    dev: true

  /axios@1.6.0(debug@4.3.4):
    resolution: {integrity: sha512-EZ1DYihju9pwVB+jg67ogm+Tmqc6JmhamRN6I4Zt8DfZu5lbcQGw3ozH9lFejSJgs/ibaef3A9PMXPLeefFGJg==}
    dependencies:
      follow-redirects: 1.15.3(debug@4.3.4)
      form-data: 4.0.0
      proxy-from-env: 1.1.0
    transitivePeerDependencies:
      - debug
    dev: true

  /axobject-query@3.2.1:
    resolution: {integrity: sha512-jsyHu61e6N4Vbz/v18DHwWYKK0bSWLqn47eeDSKPB7m8tqMHF9YJ+mhIk2lVteyZrY8tnSj/jHOv4YiTCuCJgg==}
    dependencies:
      dequal: 2.0.3
    dev: true

  /babel-plugin-polyfill-corejs2@0.4.6(@babel/core@7.23.2):
    resolution: {integrity: sha512-
jhHiWVZIlnPbEUKSSNb9YoWcQGdlTLq7z1GHL4AjFxaoOUMuuEVJ+Y4pAaQUGOGk93YsVCKPbqbfw3m
0SM6H8Q==}
    peerDependencies:
      '@babel/core': ^7.4.0 || ^8.0.0-0 <8.0.0
    dependencies:
      '@babel/compat-data': 7.23.2
      '@babel/core': 7.23.2
      '@babel/helper-define-polyfill-provider': 0.4.3(@babel/core@7.23.2)
      semver: 6.3.1
    transitivePeerDependencies:
      - supports-color
    dev: false

  /babel-plugin-polyfill-corejs3@0.8.6(@babel/core@7.23.2):
    resolution: {integrity: sha512-
leDIc4l4tUgU7str5BWLS2h8q2N4Nf6lGZP6UrNDxdtfF2g69eJ5L0H7S8A5Ln/arfFAfHor5InAdZuIOwZdgQ==
}
    peerDependencies:
      '@babel/core': ^7.4.0 || ^8.0.0-0 <8.0.0
    dependencies:
      '@babel/core': 7.23.2
      '@babel/helper-define-polyfill-provider': 0.4.3(@babel/core@7.23.2)
      core-js-compat: 3.33.1
    transitivePeerDependencies:
      - supports-color
    dev: false

  /babel-plugin-polyfill-regenerator@0.5.3(@babel/core@7.23.2):
    resolution: {integrity: sha512-
8sHeDOmXC8csczMrYEOf0UTNa4yE2SxV5JGeT/LP1n0OYVDUUFPxG9vdk2AlDlIit4t+Kf0xCtpgXPBwnn/9
pw==}
    peerDependencies:
      '@babel/core': ^7.4.0 || ^8.0.0-0 <8.0.0
    dependencies:
      '@babel/core': 7.23.2
      '@babel/helper-define-polyfill-provider': 0.4.3(@babel/core@7.23.2)
    transitivePeerDependencies:
      - supports-color
    dev: false

  /backoff@2.5.0:
    resolution: {integrity: sha512-
wC5ihrnUXmR2douXmXLCe5O3zg3GKIyvRi/hi58a/XyRxVI+3/yM0PYueQOZXPXQ9pxBisIYkw+sF9b7C/Ru
MA==}
    engines: {node: '>= 0.6'}
    dependencies:
      precond: 0.2.3
    dev: false

  /balanced-match@1.0.2:
    resolution: {integrity: sha512-
3oSeUO0TMV67hN1AmbXsK4yaqU7tjiHlbxRDZOpH0KW9+CeX4bRAaX0Anxt0tx2MrpRpWwQaPwIlISEJh
YU5Pw==}

  /base-x@3.0.9:
    resolution: {integrity: sha512-
H7JU6iBHTal1gp56aKoaa//YUxEaAOUiydvrV/pILqIHXTtqxSkATOnDA2u+jZ/61sD+L/412+7kzXRtWukhpQ==
}
    dependencies:
      safe-buffer: 5.2.1
    dev: false

  /base16@1.0.0:
    resolution: {integrity: sha512-
pNdYkNPiJUnEhnfXV56+sQy8+AaPcG3POZAUnwr4EeqCUZFz4u2PePbo3e5Gj4ziYPCWGUZT9RHisvJKn
wFuBQ==}
    dev: false

  /base64-js@1.5.1:
    resolution: {integrity: sha512-
AKpaYlHn8t4SVbOHCy+b5+KKgvR4vrsD8vbvrbiQJps7fKDTkjkDry6ji0rUJjC0kzbNePLwzxq8iypo41qeWA==
}
    dev: false

  /base64url@3.0.1:
    resolution: {integrity: sha512-
ir1UPr3dkwexU7FdV8qBBbNDRUhMmIekYMFZfi+C/sLNnRESKPl23nB9b2pltqfOQNnGzsDdId90AEtG5tCx
4A==}
    engines: {node: '>=6.0.0'}
    dev: false

  /bcrypt-pbkdf@1.0.2:
    resolution: {integrity: sha512-
qeFIXtP4MSoi6NLqO12WfqARWWuCKi2Rn/9hJLEmtB5yTNr9DqFWkJRCf2qShWzPeAMRnOgCrq0sg/KLv
5ES9w==}
    dependencies:
      tweetnacl: 0.14.5
    dev: false

  /before-after-hook@2.2.3:
    resolution: {integrity: sha512-
NzUnlZexiaH/46WDhANlyR2bXRopNg4F/zuSA3OpZnllCUgRaOF2znDioDWrmbNVsuZk6l9pMquQB38cfBZ
wkQ==}
    dev: true

  /big-integer@1.6.51:
    resolution: {integrity: sha512-
GPEid2Y9QU1Exl1rpO9B2IPJGHPSupF5GnVIP0blYvNOMer2bTvSWs1jGOUg04hTmu67nmLsQ9TBo1pua
otBHg==}
    engines: {node: '>=0.6'}
    dev: true

  /bigint-buffer@1.1.5:
    resolution: {integrity: sha512-
trfYco6AoZ+rKhKnxA0hgX0HAbVP/s808/EuDSe2JDzUnCp/xAsli35Orvk67UrTEcwuxZqYZDmfA2RXJgxVv
A==}
    engines: {node: '>= 10.0.0'}
    requiresBuild: true
    dependencies:
      bindings: 1.5.0
    dev: false

  /bignumber.js@7.2.1:
    resolution: {integrity: sha512-
S4XzBk5sMB+Rcb/LNcpzXr57VRTxgAvaAEDAl1AwRx27j00hT84O6OkteE7u8UB3NuaaygCRrEpqox4uDO
rbdQ==}
    dev: false

  /bignumber.js@9.1.2:
    resolution: {integrity: sha512-
2/mKyZH9K85bzOEfhXDBFZTGd1CTs+5IHpeFQo9luiBG7hghdC851Pj2WAhb6E3R6b9tZj/XKhbg4fum+Ke
pug==}
    dev: false

  /binary-extensions@2.2.0:
    resolution: {integrity: sha512-
jDctJ/IVQbZoJykoeHbhXpOlNBqGNcwXJKJog42E5HDPUwQTSdjCHdihjj0DlnheQ7blbT6dHOafNAiS8ooQK
A==}
    engines: {node: '>=8'}

  /bind-decorator@1.0.11:
    resolution: {integrity: sha512-
yzkH0uog6Vv/vQ9+rhSKxecnqGUZHYncg7qS7voz3Q76+TAi1SGiOKk2mlOvusQnFz9Dc4BC/NMkeXu11Y
gjJg==}
    dev: false

  /bindings@1.5.0:
    resolution: {integrity: sha512-
p2q/t/mhvuOj/UeLlV6566GD/guowlr0hHxClI0W9m7MWYkL1F0hLo+0Aexs9HSPCtR1SXQ0TD3MMKrXZajb
iQ==}
    dependencies:
      file-uri-to-path: 1.0.0
    dev: false

  /blakejs@1.2.1:
    resolution: {integrity: sha512-
QXUSXI3QVc/gJME0dBpXrag1kbzOqCjCX8/b54ntNyW6sjtoqxqRk3LTmXzaJoh71zMsDCjM+47jS7XiwN/+f
Q==}
    dev: false
  /bluebird@3.7.2:
    resolution: {integrity: sha512-XpNj6GDQzdfW+r2Wnn7xiSAd7TM3jzkxGXBGTtWKuSXv1xUV+azxAm8jdWZN06QTQk+2N2XB9jRDkvbmQmcRtg==}
    dev: false

  /bn.js@4.11.6:
    resolution: {integrity: sha512-XWwnNNFCuuSQ0m3r3C4LE3EiORltHd9M05pq6FOlVeiophzRbMo50Sbz1ehl8K3Z+jw9+vmgnXefY1hz8X+2wA==}
    dev: false

  /bn.js@4.12.0:
    resolution: {integrity: sha512-c98Bf3tPniI+scsdk237ku1Dc3ujXQTSgyiPUDEOe7tRkhrqridvh8klBv0HCEso1OLOYcHuCv/cS6DNxKH+ZA==}
    dev: false

  /bn.js@5.2.1:
    resolution: {integrity: sha512-eXRvHzWyYPBuB4NBy0cmYQjGitUrtqwbvlzP3G6VFnNRbsZQIxQ10PbKKHt8gZ/HW/D/747aDl+QkDqg3KQLMQ==}
    dev: false

  /body-parser@1.20.1:
    resolution: {integrity: sha512-jWi7abTbYwajOytWCQc37VulmWiRae5RyTpaCyDcS5/lMdtwSz5lOpDE67srw/HYe35f1z3fDQw+3txg7gNtWw==}
    engines: {node: '>= 0.8', npm: 1.2.8000 || >= 1.4.16}
    dependencies:
      bytes: 3.1.2
      content-type: 1.0.5
      debug: 2.6.9
      depd: 2.0.0
      destroy: 1.2.0
      http-errors: 2.0.0
      iconv-lite: 0.4.24
      on-finished: 2.4.1
      qs: 6.11.0
      raw-body: 2.5.1
      type-is: 1.6.18
      unpipe: 1.0.0
    transitivePeerDependencies:
      - supports-color
    dev: false

  /body-parser@1.20.2:
    resolution: {integrity: sha512-ml9pReCu3M61kGlqoTm2umSXTlRTuGTx0bfYj+uIUKKYycG5NtSbeetV3faSU6R7ajOPw0g/J1PvK4qNy7s5bA==}
    engines: {node: '>= 0.8', npm: 1.2.8000 || >= 1.4.16}
    dependencies:
      bytes: 3.1.2
      content-type: 1.0.5
      debug: 2.6.9
      depd: 2.0.0
      destroy: 1.2.0
      http-errors: 2.0.0
      iconv-lite: 0.4.24
      on-finished: 2.4.1
      qs: 6.11.0
      raw-body: 2.5.2
      type-is: 1.6.18
      unpipe: 1.0.0
    transitivePeerDependencies:
      - supports-color
    dev: false

  /borsh@0.7.0:
    resolution: {integrity: sha512-CLCsZGIBCFnPtkNnieW/a8wmreDmfUtjU2m9yHrzPXIlNbqVs0AQrSatSG6vdNYUqdc83tkQi2eHfF98ubzQLA==}
    dependencies:
      bn.js: 5.2.1
      bs58: 4.0.1
      text-encoding-utf-8: 1.0.2
    dev: false

  /bplist-parser@0.2.0:
    resolution: {integrity: sha512-z0M+byMThzQmD9NILRniCUXYsYpjwnlO8N5uCFaCqIOpqRsJCrQL9NK3JsD67CN5a08nF5oIL2bD6loTdHOuKw==}
    engines: {node: '>= 5.10.0'}
    dependencies:
      big-integer: 1.6.51
    dev: true

  /brace-expansion@1.1.11:
    resolution: {integrity: sha512-iCuPHDFgrHX7H2vEI/5xpz07zSHB00TpugqhmYtVmMO6518mCuRMoOYFldEBl0g187ufozdaHgWKcYFb61qGiA==}
    dependencies:
      balanced-match: 1.0.2
      concat-map: 0.0.1

  /brace-expansion@2.0.1:
    resolution: {integrity: sha512-XnAIvQ8eM+kC6aULx6wuQiwVsnzsi9d3WxzV3FpWTGA19F621kwdbsAcFKXgKUHZWsy+mY6iL1sHTxWEFCytDA==}
    dependencies:
      balanced-match: 1.0.2
    dev: false

  /braces@3.0.2:
    resolution: {integrity: sha512-
b8um+L1RzM3WDSzvhm6gIz1yfTbBt6YTlcEKAvsmqCZZFw46z626lVj9j1yEPW33H5H+lBQpZMP1k8l+78H
a0A==}
    engines: {node: '>=8'}
    dependencies:
      fill-range: 7.0.1

  /brorand@1.1.0:
    resolution: {integrity: sha512-
cKV8tMCEpQs4hK/ik71d6LrPOnpkpGBR0wzxqr68g2m/LB2GxVYQroAjMJZRVM1Y4BCjCKc3vAamxSzOY
2RP+w==}
    dev: false

  /browserify-aes@1.2.0:
    resolution: {integrity: sha512-
+7CHXqGuspUn/Sl5aO7Ea0xWGAtETPXNSAjHo48JfLdPWcMng33Xe4znFvQweqc/uzk5zSOI3H52CYnjCf
b5hA==}
    dependencies:
      buffer-xor: 1.0.3
      cipher-base: 1.0.4
      create-hash: 1.2.0
      evp_bytestokey: 1.0.3
      inherits: 2.0.4
      safe-buffer: 5.2.1
    dev: false

  /browserslist@4.22.1:
    resolution: {integrity: sha512-
FEVc202+2iuClEhZhrWy6ZiAcRLvNMyYcxZ8raemul1DYVOVdFsbqckWLdsixQZCpJlwe77Z3UTalE7jsjnKf
Q==}
    engines: {node: ^6 || ^7 || ^8 || ^9 || ^10 || ^11 || ^12 || >=13.7}
    hasBin: true
    dependencies:
      caniuse-lite: 1.0.30001554
      electron-to-chromium: 1.4.568
      node-releases: 2.0.13
      update-browserslist-db: 1.0.13(browserslist@4.22.1)

  /bs58@4.0.1:
    resolution: {integrity: sha512-
Ok3Wdf5vOIlBrgCvTq96gBkJw+JUEzdBgyaza5HLtPm7yTHkjRy8+JzNyHF7BHa0bNWOQIp3m5YF0nnFcO
IKLw==}
    dependencies:
      base-x: 3.0.9
    dev: false

  /bs58check@2.1.2:
    resolution: {integrity: sha512-
0TS1jicxdU09dwJMNZtVAfzPi6Q6QeN0pM1Fkzrjn+XYHvzMKPU3pHVpva+769iNVSfIYWf7LJ6WR+BuuMf
8cA==}
    dependencies:
      bs58: 4.0.1
      create-hash: 1.2.0
      safe-buffer: 5.2.1
    dev: false

  /btoa@1.2.1:
    resolution: {integrity: sha512-SB4/MIGlsiVkMcHmT+pSmIPoNDoHg+7cMzmt3Uxt628MTz2487DKSqK/fuhFBrkuqrYv5UCEnACpF4dTFNKc/g==}
    engines: {node: '>= 0.4.0'}
    hasBin: true
    dev: false

  /buffer-to-arraybuffer@0.0.5:
    resolution: {integrity: sha512-3dthu5CYiVB1DEJp61FtApNnNndTckcqe4pFcLdvHtrpG+kcyekCJKg4MRiDcFW7A6AODnXB9U4dwQiCW5kzJQ==}
    dev: false

  /buffer-xor@1.0.3:
    resolution: {integrity: sha512-571s0T7nZWK6vB67HI5dyUF7wXiNcfaPPPTl6zYCNApANjIvYJTg7hlud/+cJpdAhS7dVzqMLmfhfHR3rAcOjQ==}
    dev: false

  /buffer@5.7.1:
    resolution: {integrity: sha512-EHcyIPBQ4BSGlvjB16k5KgAJ27CIsHY/2JBmCRReo48y9rQ3MaUzWX3KVlBa4U7MyX02HdVj0K7C3WaB3ju7FQ==}
    dependencies:
      base64-js: 1.5.1
      ieee754: 1.2.1
    dev: false

  /buffer@6.0.3:
    resolution: {integrity: sha512-FTiCpNxtwiZZHEZbcbTIcZjERVICn9yq/pDFkTl95/AxzD1naBctN7YO68riM/gLSDY7sdrMby8hofADYuuqOA==}
    dependencies:
      base64-js: 1.5.1
      ieee754: 1.2.1
    dev: false

  /bufferutil@4.0.7:
    resolution: {integrity: sha512-kukuqc39WOHtdxtw4UScxF/WVnMFVSQVKhtx3AjZJzhd0RGZZldcrfSEbVsWWe6KNH253574cq5F+wpv0G9pJw==}
    engines: {node: '>=6.14.2'}
    requiresBuild: true
    dependencies:
      node-gyp-build: 4.5.0
    dev: false
  /builtin-modules@3.3.0:
    resolution: {integrity: sha512-zhaCDicdLuWN5UbN5IMnFqNMhNfo919sH85y2/ea+5Yg9TsTkeZxpL+JLbp6cgYFS4sRLp3YV4S6yDuqVWHYOw==}
    engines: {node: '>=6'}
    dev: true

  /bundle-name@3.0.0:
    resolution: {integrity: sha512-PKA4BeSvBpQKQ8iPOGCSiell+N8P+Tf1DlwqmYhpe2gAhKPHn8EYOxVT+ShuGmhg8lN8XiSlS80yiExKXrURlw==}
    engines: {node: '>=12'}
    dependencies:
      run-applescript: 5.0.0
    dev: true

  /bytes@3.1.2:
    resolution: {integrity: sha512-/Nf7TyzTx6S3yRJObOAV7956r8cr2+Oj8AC5dt8wSP3BQAoeX58NoHyCU8P8zGkNXStjTSi6fzO6F0pBdcYbEg==}
    engines: {node: '>= 0.8'}
    dev: false

  /cacheable-lookup@5.0.4:
    resolution: {integrity: sha512-2/kNscPhpcxrOigMZzbiWF7dz8ilhb/nIHU3EyZiXWXpeq/au8qJ8VhdftMkty3n7Gj6HIGalQG8oiBNB3AJgA==}
    engines: {node: '>=10.6.0'}
    dev: false

  /cacheable-lookup@6.1.0:
    resolution: {integrity: sha512-KJ/Dmo1lDDhmW2XDPMo+9oiy/CeqosPguPCrgcVzKyZrL6pM1gU2GmPY/xo6OQPTUaA/c0kwHuywB4E6nmT9ww==}
    engines: {node: '>=10.6.0'}
    dev: false

  /cacheable-request@7.0.4:
    resolution: {integrity: sha512-v+p6ongsrp0yTGbJXjgxPow2+DL93DASP4kXCDKb8/bwRtt9OEF3whggkkDkGNzgcWy2XaF4a8nZglC7uElscg==}
    engines: {node: '>=8'}
    dependencies:
      clone-response: 1.0.3
      get-stream: 5.2.0
      http-cache-semantics: 4.1.1
      keyv: 4.5.4
      lowercase-keys: 2.0.0
      normalize-url: 6.1.0
      responselike: 2.0.1
    dev: false

  /call-bind@1.0.2:
    resolution: {integrity: sha512-
7O+FbCihrB5WGbFYesctwmTKae6rOiIzmz1icreWJ+0aA7LJfuqhEso2T9ncpcFtzMQtzXf2QGGueWJGTYsq
rA==}
    dependencies:
      function-bind: 1.1.1
      get-intrinsic: 1.1.3

  /call-bind@1.0.5:
    resolution: {integrity: sha512-
C3nQxfFZxFRVoJoGKKI8y3MOEo129NQ+FgQ08iye+Mk4zNZZGdjfs06bVTr+DBSlA66Q2VEcMki/cUCP4S
ercQ==}
    dependencies:
      function-bind: 1.1.2
      get-intrinsic: 1.2.2
      set-function-length: 1.1.1
    dev: true

  /callsites@3.1.0:
    resolution: {integrity: sha512-
P8BjAsXvZS+VIDUI11hHCQEv74YT67YUi5JJFNWIqL235sBmjX4+qx9Muvls5ivyNENctx46xQLQ3aTuE7ss
aQ==}
    engines: {node: '>=6'}
    dev: true

  /camelcase-css@2.0.1:
    resolution: {integrity: sha512-
QOSvevhslijgYwRx6Rv7zKdMF8lbRmx+uQGx2+vDc+KI/eBnsy9kit5aj23AgGu3pa4t9AgwbnXWqS+iOY+2a
A==}
    engines: {node: '>= 6'}
    dev: false

  /camelcase@5.3.1:
    resolution: {integrity: sha512-
L28STB170nwWS63UjtlEOE3dldQApaJXZkOI1uMFfzf3rRuPegHaHesyee+YxQ+W6SvRDQV6UrdOdRiR15
3wJg==}
    engines: {node: '>=6'}
    dev: false

  /caniuse-lite@1.0.30001554:
    resolution: {integrity: sha512-
A2E3U//MBwbJVzebddm1YfNp7Nud5Ip+IPn4BozBmn4KqVX7AvluoIDFWjsv5OkGnKUXQVmMSoMKLa3S
cCblcQ==}

  /caseless@0.12.0:
    resolution: {integrity: sha512-
4tYFyifaFfGacoiObjJegolkwSU4xQNGbVgUiNYVUxbQ2x2lUsFvY4hVgVzGiIe6WLOPqycWXA40l+PWsxth
Uw==}
    dev: false

  /chalk@2.4.2:
    resolution: {integrity: sha512-
Mti+f9lpJNcwF4tWV8/OrTTtF1gZi+f8FqlyAdouralcFWFQWF2+NgCHShjkCb+IFBLq9buZwE1xckQU4peSu
Q==}
    engines: {node: '>=4'}
    dependencies:
      ansi-styles: 3.2.1
      escape-string-regexp: 1.0.5
      supports-color: 5.5.0

  /chalk@4.1.2:
    resolution: {integrity: sha512-
oKnbhFyRIXpUuez8iBMmyEa4nbj4IOQyuhc/wy9kY7/WVPcwIO9VA668Pu8RkO7+0G76SLROeyw9CpQ06
1i4mA==}
    engines: {node: '>=10'}
    dependencies:
      ansi-styles: 4.3.0
      supports-color: 7.2.0

  /checkpoint-store@1.1.0:
    resolution: {integrity: sha512-
J/NdY2WvIx654cc6LWSq/IYFFCUf75fFTgwzFnmbqyORH4MwgiQCgswLLKBGzmsyTI5V7i5bp/So6sMbDW
hedg==}
    dependencies:
      functional-red-black-tree: 1.0.1
    dev: false

  /chokidar@3.5.3:
    resolution: {integrity: sha512-
Dr3sfKRP6oTcjf2JmUmFJfeVMvXBdegxB0iVQ5eb2V10uFJUCAS8OByZdVAyVb8xXNz3GjjTgj9kLWsZTqE
6kw==}
    engines: {node: '>= 8.10.0'}
    dependencies:
      anymatch: 3.1.3
      braces: 3.0.2
      glob-parent: 5.1.2
      is-binary-path: 2.1.0
      is-glob: 4.0.3
      normalize-path: 3.0.0
      readdirp: 3.6.0
    optionalDependencies:
      fsevents: 2.3.2

  /chownr@1.1.4:
    resolution: {integrity: sha512-
jJ0bqzaylmJtVnNgzTeSOs8DPavpbYGEr/b0YL8/2GO3xJEhInFmhKMUnEJQjZumK7KXGFhUy89PrsJWlak
BVg==}
    dev: false

  /ci-info@3.9.0:
    resolution: {integrity: sha512-
NIxF55hv4nSqQswkAeiOi1r83xy8JldOFDTWiug55KBu9Jnblncd2U6ViHmYgHf01TPZS77NJBhBMKdWj9H
QMQ==}
    engines: {node: '>=8'}
    dev: true
```

```
 /cids@0.7.5:
   resolution: {integrity: sha512-
zT7mPeghoWAu+ppn8+BS1tQ5qGmbMfB4AregnQjA/qHY3GC1m1ptI9GkWNlgeu38r7CuRdXB47uY2XgAY
t6QVA==}
   engines: {node: '>=4.0.0', npm: '>=3.0.0'}
   deprecated: This module has been superseded by the multiformats module
   dependencies:
     buffer: 5.7.1
     class-is: 1.1.0
     multibase: 0.6.1
     multicodec: 1.0.4
     multihashes: 0.4.21
   dev: false

 /cipher-base@1.0.4:
   resolution: {integrity: sha512-
Kkht5ye6ZGmwv40uUDZztayT2ThLQGfnj/T71N/XzeZeo3nf8foyW7zGTsPYkEya3m5f3cAypH+qe7YOrM1U
2Q==}
   dependencies:
     inherits: 2.0.4
     safe-buffer: 5.2.1
   dev: false

 /class-is@1.1.0:
   resolution: {integrity: sha512-
rhjH9AG1fvabIDoGRVH587413LPjTZgmDF9fOFCbFJQV4yuocX1mHxxvXI4g3cGwbVY9wAYIoKlg1N79frJ
KQw==}
   dev: false

 /clean-regexp@1.0.0:
   resolution: {integrity: sha512-
GfisEZEJvzKrmGWkvfhgzcz/BllN1USeqD2V6tg14OAOgaCD2Z/PUEuxnAZ/nPvmaHRG7a8y77p1T/IRQ4D1
Hw==}
   engines: {node: '>=4'}
   dependencies:
     escape-string-regexp: 1.0.5
   dev: true

 /cliui@6.0.0:
   resolution: {integrity: sha512-
t6wbgtoCXvAzst7QgXxJYqPt0usEfbgQdftEPbLL/cvv6HPE5VgvqCuAIDR0NgU52ds6rFwqrgakNLrHEjCbrQ
==}
   dependencies:
     string-width: 4.2.3
     strip-ansi: 6.0.1
     wrap-ansi: 6.2.0
   dev: false

 /cliui@8.0.1:
   resolution: {integrity: sha512-
BSeNnyus75C4//NQ9gQt1/csTXyo/8Sb+afLAkzAptFuMsod9HFokGNudZpi/oQV73hnVK+sR+5PVRMd+Dr7
```

```
YQ==}
    engines: {node: '>=12'}
    dependencies:
      string-width: 4.2.3
      strip-ansi: 6.0.1
      wrap-ansi: 7.0.0
    dev: true

  /clone-response@1.0.3:
    resolution: {integrity: sha512-
ROoL94jJH2dUVML2Y/5PEDNaSHgeOdSDicUyS7izcF63G6sTc/FTjLub4b8Il9S8S0beOfYt0TaA5qvFK+w0
wA==}
    dependencies:
      mimic-response: 1.0.1
    dev: false

  /clone@2.1.2:
    resolution: {integrity: sha512-
3Pe/CF1Nn94hyhIYpjtiLhdCoEoz0DqQ+988E9gmeEdQZlojxnOb74wctFyuwWQHzqyf9X7C7MG8juUpqBJT
8w==}
    engines: {node: '>=0.8'}
    dev: false

  /clsx@1.2.1:
    resolution: {integrity: sha512-
EcR6r5a8bj6pu3ycsa/E/cKVGuTgZJZdsyUYHOksG/UHIiKfjxzRxYJpyVBwYaQeOvghal9fcc4PidlgzugAQg==
}
    engines: {node: '>=6'}
    dev: false

  /color-convert@1.9.3:
    resolution: {integrity: sha512-
QfAUtd+vFdAtFQcC8CCyYt1fYWxSqAiK2cSD6zDB8N3cpsEBAvRxp9zOGg6G/SHHJYAT88/az/IuDGALsN
VbGg==}
    dependencies:
      color-name: 1.1.3

  /color-convert@2.0.1:
    resolution: {integrity: sha512-
RRECPsj7iu/xb5oKYcsFHSppFNnsj/52OVTRKb4zP5onXwVF3zVmmToNcOfGC+CRDpfK/U584fMg38ZHC
aElKQ==}
    engines: {node: '>=7.0.0'}
    dependencies:
      color-name: 1.1.4

  /color-name@1.1.3:
    resolution: {integrity: sha512-
72fSenhMw2HZMTVHeCA9KCmpEIbzWiQsjN+BHcBbS9vr1mtt+vJjPdksIBNUmKAW8TFUDPJK5SUU3Qh
E9NEXDw==}

  /color-name@1.1.4:
    resolution: {integrity: sha512-
```

```
dOy+3AuW3a2wNbZHIuMZpTcgjGuLU/uBL/ubcZF9OXbDo8ff4O8yVp5Bf0efS8uEoYo5q4Fx7dY9OgQGXg
AsQA==}

  /combined-stream@1.0.8:
    resolution: {integrity: sha512-
FQN4MRfuJeHf7cBbBMJFXhKSDq+2kAArBlmRBvcvFE5BB1HZKXtSFASDhdlz9zOYwxh8lDdnvmMOe/+5c
doEdg==}
    engines: {node: '>= 0.8'}
    dependencies:
      delayed-stream: 1.0.0

  /command-line-args@5.2.1:
    resolution: {integrity: sha512-
H4UfQhZyakIjC74I9d34fGYDwk3XpSr17QhEd0Q3I9Xq1CETHo4Hcuo87WyWHpAF1aSLjLRf5lD9ZGX2qSt
Uvg==}
    engines: {node: '>=4.0.0'}
    dependencies:
      array-back: 3.1.0
      find-replace: 3.0.0
      lodash.camelcase: 4.3.0
      typical: 4.0.0

  /command-line-usage@6.1.3:
    resolution: {integrity: sha512-
sH5ZSPr+7UStsloltmDh7Ce5fb8XPlHyoPzTpyyMuYCtervL65+ubVZ6Q61cFtFl62UyJlc8/JwERRbAFPUqgw
==}
    engines: {node: '>=8.0.0'}
    dependencies:
      array-back: 4.0.2
      chalk: 2.4.2
      table-layout: 1.0.2
      typical: 5.2.0

  /commander@2.20.3:
    resolution: {integrity: sha512-
GpVkmM8vF2vQUkj2LvZmD35JxeJOLCwJ9cUkugyk2nuhbv3+mJvpLYYt+0+USMxE+oj+ey/lJEnhZw75x/O
McQ==}
    dev: false

  /commander@4.1.1:
    resolution: {integrity: sha512-
NOKm8xhkzAjzFx8B2v5OAHT+u5pRQc2UCa2Vq9jYL/31o2wi9mxBA7LlFs3sV5VSC49z6pEhfbMULvShKj2
6WA==}
    engines: {node: '>= 6'}
    dev: false

  /concat-map@0.0.1:
    resolution: {integrity: sha512-
/Srv4dswyQNBfohGpz9o6Yb3Gz3SrUDqBH5rTuhGR7ahtlbYKnVxw2bCFMRljaA7EXHaXZ8wsHdodFvbkhK
mqg==}

  /concurrently@8.2.2:
    resolution: {integrity: sha512-
1dP4gpXFhei8IOtlXRE/T/4H88ElHgTiUzh71YUmtjTEHMSRS2Z/fgOxHSxxusGHogsRfxNq1vyAwxSC+EVy
Dg==}
    engines: {node: ^14.13.0 || >=16.0.0}
    hasBin: true
    dependencies:
      chalk: 4.1.2
      date-fns: 2.30.0
      lodash: 4.17.21
      rxjs: 7.8.1
      shell-quote: 1.8.1
      spawn-command: 0.0.2
      supports-color: 8.1.1
      tree-kill: 1.2.2
      yargs: 17.7.2
    dev: true

  /content-disposition@0.5.4:
    resolution: {integrity: sha512-
FveZTNuGw04cxlAiWbzi6zTAL/lhehaWbTtgluJh4/E95DqMwTmha3KZN1aAWA8cFIhHzMZUvLevkw5Rqk+t
SQ==}
    engines: {node: '>= 0.6'}
    dependencies:
      safe-buffer: 5.2.1
    dev: false

  /content-hash@2.5.2:
    resolution: {integrity: sha512-
FvIQKy0S1JaWV10sMsA7TRx8bpU+pqPkhbsfvOJAdjRXvYxEckAwQWGwtRjiaJfh+E0DvcWUGqcdjwMGFj
sSdw==}
    dependencies:
      cids: 0.7.5
      multicodec: 0.5.7
      multihashes: 0.4.21
    dev: false

  /content-type@1.0.5:
    resolution: {integrity: sha512-
nTjqfcBFEipKdXCv4YDQWCfmcLZKm81ldF0pAopTvyrFGVbcR6P/VAAd5G7N+0tTr8QqiU0tFadD6FK4NtJ
wOA==}
    engines: {node: '>= 0.6'}
    dev: false

  /convert-source-map@2.0.0:
    resolution: {integrity: sha512-
Kvp459HrV2FEJ1CAsi1Ku+MY3kasH19TFykTz2xWmMeq6bk2NU3XXvfJ+Q61m0xktWwt+1HSYf3JZsTms
3aRJg==}

  /cookie-signature@1.0.6:
    resolution: {integrity: sha512-
QADzlaHc8icV8I7vbaJXJwod9HWYp8uCqf1xa4OfNu1T7JVxQIrUgOWtHdNDtPiywmFbiS12VjotIXLrKM3or
Q==}
```

```
    dev: false

  /cookie@0.5.0:
    resolution: {integrity: sha512-
YZ3GUyn/o8gfKJlnlX7g7xq4gyO6OSuhGPKaaGssGB2qgDUS0gPgtTvoyZLTt9Ab6dC4hfc9dV5arkvc/OCmr
w==}
    engines: {node: '>= 0.6'}
    dev: false

  /copy-to-clipboard@3.3.3:
    resolution: {integrity: sha512-
2KV8NhB5JqC3ky0r9PMCAZKbUHSwtEo4CwCs0KXgruG43gX5PMqDEBbVU4OUzw2MuAWUfsuFmWvE
KG5QRfSnJA==}
    dependencies:
      toggle-selection: 1.0.6
    dev: false

  /core-js-compat@3.33.1:
    resolution: {integrity: sha512-
6pYKNOgD/j/bkC5xS5IIg6bncid3rfrI42oBH1SQJbsmYPKF7rhzcFzYCcxYMmNQQ0rCEB8WqpW7QHndOg
gaeQ==}
    dependencies:
      browserslist: 4.22.1
    dev: false

  /core-util-is@1.0.2:
    resolution: {integrity: sha512-
3lqz5YjWTYnW6dlDa5TLaTCcShfar1e40rmcJVwCBJC6mWlFuj0eCHIElmG1g5kyuJ/GD+8Wn4FFCcz4gJP
faQ==}
    dev: false

  /core-util-is@1.0.3:
    resolution: {integrity: sha512-
ZQBvi1DcpJ4GDqanjucZ2Hj3wEO5pZDS89BWbkcrvdxksJorwUDDZamX9ldFkp9aw2lmBDLgkObEA4DWN
J9FYQ==}
    dev: false

  /cors@2.8.5:
    resolution: {integrity: sha512-
KIHbLJqu73RGr/hnbrO9uBeixNGuvSQjul/jdFvS/KFSIH1hWVd1ng7zOHx+YrEfInLG7q4n6GHQ9cDtxv/P6g=
}
    engines: {node: '>= 0.10'}
    dependencies:
      object-assign: 4.1.1
      vary: 1.1.2
    dev: false

  /crc-32@1.2.2:
    resolution: {integrity: sha512-
ROmzCKrTnOwybPcJApAA6WBWij23HVfGVNKqqrZpuyZOHqK2CwHSvpGuyt/UNNvaIjEd8X5IFGp4Mh+Ie
1IHJQ==}
    engines: {node: '>=0.8'}
    hasBin: true
    dev: false

  /create-hash@1.2.0:
    resolution: {integrity: sha512-
z00bCGNHDG8mHAkP7CtT1qVu+bFQUPjYq/4Iv3C3kWjTFV10zIjfSoeqXo9Asws8gwSHDGj/hl2u4OGIjape
Cg==}
    dependencies:
      cipher-base: 1.0.4
      inherits: 2.0.4
      md5.js: 1.3.5
      ripemd160: 2.0.2
      sha.js: 2.4.11
    dev: false

  /create-hmac@1.1.7:
    resolution: {integrity: sha512-
MJG9liiZ+ogc4TzUwuvbER1JRdgvUFSB5+VR/g5h82fGaIRWMWddtKBHi7/sVhfjQZ6SehlyhvQYrcYkaUIpL
g==}
    dependencies:
      cipher-base: 1.0.4
      create-hash: 1.2.0
      inherits: 2.0.4
      ripemd160: 2.0.2
      safe-buffer: 5.2.1
      sha.js: 2.4.11
    dev: false

  /create-require@1.1.1:
    resolution: {integrity: sha512-
dcKFX3jn0MpIaXjisoRvexIJVEKzaq7z2rZKxf+MSr9TkdmHmsU4m2lcLojrj/FHl8mk5VxMmYA+ftRkP/3oKQ==
}

  /cross-fetch@2.2.6:
    resolution: {integrity: sha512-
9JZz+vXCmfKUZ68zAptS7k4Nu8e2qcibe7WVZYps7sAgk5R8GYTc+T1WR0v1rlP9HxgARmOX1UTIJZFytaj
pNA==}
    dependencies:
      node-fetch: 2.7.0
      whatwg-fetch: 2.0.4
    transitivePeerDependencies:
      - encoding
    dev: false

  /cross-fetch@3.1.5:
    resolution: {integrity: sha512-
lvb1SBsI0Z7GDwmuid+mU3kWVBwTVUbe7S0H52yaaAdQOXq2YktTCZdlAcNKFzE6QtRz0snpw9bNiPeOI
kkQvw==}
    dependencies:
      node-fetch: 2.6.7
    transitivePeerDependencies:
      - encoding
    dev: false

  /cross-spawn@7.0.3:
    resolution: {integrity: sha512-
iRDPJKUPVEND7dHPO8rkbOnPpyDygcDFtWjpeWNCgy8WP2rXcxXL8TskReQl6OrB2G7+UJrags1q15Fud
c7G6w==}
    engines: {node: '>= 8'}
    dependencies:
      path-key: 3.1.1
      shebang-command: 2.0.0
      which: 2.0.2
    dev: true

  /cssesc@3.0.0:
    resolution: {integrity: sha512-
/Tb/JcjK111nNScGob5MNtsntNM1aCNUDipB/TkwZFhyDrrE47SOx/18wF2bbjgc3ZzCSKW1T5nt5EbFoAz/V
g==}
    engines: {node: '>=4'}
    hasBin: true
    dev: false

  /csstype@3.1.1:
    resolution: {integrity: sha512-
DJR/VvkAvSZW9bTouZue2sSxDwdTN92uHjqeKVm+0dAqdfNykRzQ95tay8aXMBAAPpUiq4Qcug2L7neoR
h2Egw==}

  /d@1.0.1:
    resolution: {integrity: sha512-
m62ShEObQ39CfralilEQRjH6oAMtNCV1xJyEx5LpRYUVN+EviphDgUc/F3hnYbADmkiNs67Y+3ylmlG7Lnu+
FA==}
    dependencies:
      es5-ext: 0.10.62
      type: 1.2.0
    dev: false

  /damerau-levenshtein@1.0.8:
    resolution: {integrity: sha512-
sdQSFB7+llfUcQHUQO3+B8ERRj0Oa4w9POWMI/puGtuf7gFywGmkaLCElnudfTiKZV+NvHqL0ifzdrI8Ro7E
SA==}
    dev: true

  /dashdash@1.14.1:
    resolution: {integrity: sha512-
jRFi8UDGo6j+odZiEpjazZaWqEal3w/basFjQHQEwVtZJGDpxbH1MeYluwCS8Xq5wmLJooDlMgvVarmWfG
M44g==}
    engines: {node: '>=0.10'}
    dependencies:
      assert-plus: 1.0.0
    dev: false

  /date-fns@2.30.0:
    resolution: {integrity: sha512-
fnULvOpxnC5/Vg3NCiWelDsLiUc9bRwAPs/+LfTLNvetFCtCTN+yQz15C/fs4AwX1R9K5GLtLfn8QW+dWisa
Aw==}
    engines: {node: '>=0.11'}
    dependencies:
      '@babel/runtime': 7.23.2
    dev: true

  /debug@2.6.9:
    resolution: {integrity: sha512-
bC7ElrdJaJnPbAP+1EotYvqZsb3ecl5wi6Bfi6BJTUcNowp6cvspg0jXznRTKDjm/E7AdgFBVeAPVMNcKGsH
MA==}
    peerDependencies:
      supports-color: '*'
    peerDependenciesMeta:
      supports-color:
        optional: true
    dependencies:
      ms: 2.0.0
    dev: false

  /debug@3.2.7(supports-color@5.5.0):
    resolution: {integrity: sha512-
CFjzYYAi4ThfiQvizrFQevTTXHtnCqWfe7x1AhgEscTz6ZbLbfoLRLPugTQyBth6f8ZERVUSyWHFD/7Wu4t1X
Q==}
    peerDependencies:
      supports-color: '*'
    peerDependenciesMeta:
      supports-color:
        optional: true
    dependencies:
      ms: 2.1.3
      supports-color: 5.5.0
    dev: true

  /debug@4.3.4:
    resolution: {integrity: sha512-
PRWFHuSU3eDtQJPvnNY7Jcket1j0t5OuOsFzPPzsekD52Zl8qUfFIPEiswXqIvHWGVHOgX+7G/vCNNhehw
xfkQ==}
    engines: {node: '>=6.0'}
    peerDependencies:
      supports-color: '*'
    peerDependenciesMeta:
      supports-color:
        optional: true
    dependencies:
      ms: 2.1.2

  /decamelize@1.2.0:
    resolution: {integrity: sha512-
z2S+W9X73hAUUki+N+9Za2lBlun89zigOyGrsax+KUQ6wKW4ZoWpEYBkGhQjwAjjDCkWxhY0VKEhk8wzY
7F5cA==}
    engines: {node: '>=0.10.0'}
    dev: false

  /decode-uri-component@0.2.0:
    resolution: {integrity: sha512-hjf+xovcEn31w/EUYdTXQh/8smFL/dzYjohQGEIgjyNavaJfBY2p5F527Bo1VPATxv0VYTUC2bOcXvqFwk78Og==}
    engines: {node: '>=0.10'}
    dev: false

  /decode-uri-component@0.2.2:
    resolution: {integrity: sha512-FqUYQ+8o158GyGTrMFJms9qh3CqTKvAqgqsTnkLI8sKu0028orqBhxNMFkFen0zGyg6epACD32pjVk58ngIErQ==}
    engines: {node: '>=0.10'}
    dev: false

  /decompress-response@3.3.0:
    resolution: {integrity: sha512-BzRPQuY1ip+qDonAOz42gRm/pg9F768C+npV/4JOsxRC2sq+Rlk+Q4ZCAsOhnIaMrgarILY+RMUIvMmmX1qAEA==}
    engines: {node: '>=4'}
    dependencies:
      mimic-response: 1.0.1
    dev: false

  /decompress-response@6.0.0:
    resolution: {integrity: sha512-aW35yZM6Bb/4oJlZncMH2LCoZtJXTRxES17vE3hoRiowU2kWHaJKFkSBDnDR+cm9J+9QhXmREyIfv0pji9ejCQ==}
    engines: {node: '>=10'}
    dependencies:
      mimic-response: 3.1.0
    dev: false

  /deep-extend@0.6.0:
    resolution: {integrity: sha512-LOHxIOaPYdHlJRtCQfDIVZtfw/ufM8+rVj649RIHzcm/vGwQRXFt6OPqIFWsm2XEMrNIEtWR64sY1LEKD2vAOA==}
    engines: {node: '>=4.0.0'}

  /deep-is@0.1.4:
    resolution: {integrity: sha512-oIPzksmTg4/MriiaYGO+okXDT7ztn/w3Eptv/+gSIdMdKsJo0u4CfYNFJPy+4SKMuCqGw2wxnA+URMg3t8a/bQ==}
    dev: true

  /default-browser-id@3.0.0:
    resolution: {integrity: sha512-OZ1y3y0SqSICtE8DE4S8YOE9UZOJ8wO16fKWVP5J1Qz42kV9jcnMVFrEE/noXb/ss3Q4pZIH79kxofzyNNtUNA==}
    engines: {node: '>=12'}
    dependencies:
```

```
    bplist-parser: 0.2.0
    untildify: 4.0.0
  dev: true

 /default-browser@4.0.0:
  resolution: {integrity: sha512-
wX5pXO1+BrhMkSbROFsyxUm0i/cJEScyNhA4PPxc41ICuv05ZZB/MX28s8aZx6xjmatvebIapF6hLEKEcpne
UA==}
  engines: {node: '>=14.16'}
  dependencies:
    bundle-name: 3.0.0
    default-browser-id: 3.0.0
    execa: 7.2.0
    titleize: 3.0.0
  dev: true

 /defer-to-connect@2.0.1:
  resolution: {integrity: sha512-
4tvttepXG1VaYGrRibk5EwJd1t4udunSOVMdLSAL6mId1ix438oPwPZMALY41FCijukO1L0twNcGsdzS7dHg
Dg==}
  engines: {node: '>=10'}
  dev: false

 /deferred-leveldown@1.2.2:
  resolution: {integrity: sha512-
uukrWD2bguRtXilKt6cAWKyoXrTSMo5m7crUdLfWQmu8kIm88w3QZoUL+6nhpfKVmhHANER6Re3sKoNo
Z3IKMA==}
  dependencies:
    abstract-leveldown: 2.6.3
  dev: false

 /define-data-property@1.1.1:
  resolution: {integrity: sha512-
E7uGkTzkk1d0ByLeSc6ZsFS79Axg+m1P/VsgYsxHgiuc3tFSj+MjMIwe90FC4lOAZzNBdY7kkO2P2wKdsQ1
vgQ==}
  engines: {node: '>= 0.4'}
  dependencies:
    get-intrinsic: 1.2.2
    gopd: 1.0.1
    has-property-descriptors: 1.0.1
  dev: true

 /define-lazy-prop@3.0.0:
  resolution: {integrity: sha512-
N+MeXYoqr3pOgn8xfyRPREN7gHakLYjhsHhWGT3fWAiL4IkAt0iDw14QiiEm2bE30c5XX5q0FtAA3CK5f9/B
Ug==}
  engines: {node: '>=12'}
  dev: true

 /define-properties@1.2.1:
  resolution: {integrity: sha512-
8QmQKqEASLd5nx0U1B1okLElbUuuttJ/AnYmRXbbbGDWh6uS208EjD4Xqq/I9wK7u0v6O08XhTWnt5XtEb
R6Dg==}
  engines: {node: '>= 0.4'}
  dependencies:
    define-data-property: 1.1.1
    has-property-descriptors: 1.0.1
    object-keys: 1.1.1
  dev: true

 /delay@5.0.0:
  resolution: {integrity: sha512-
ReEBKkIfe4ya47wlPYf/gu5ib6yUG0/Aez0JQZQz94kiWtRQvZIQbTiehsnwHvLSWJnQdhVeqYue7Id1dKr0qw
==}
  engines: {node: '>=10'}
  dev: false

 /delayed-stream@1.0.0:
  resolution: {integrity: sha512-
ZySD7Nf91aLB0RxL4KGrKHBXl7Eds1DAmEdcoVawXnLD7SDhpNgtuII2aAkg7a7QS41jxPSZ17p4VdGnM
Hk3MQ==}
  engines: {node: '>=0.4.0'}

 /depd@2.0.0:
  resolution: {integrity: sha512-
g7nH6P6dyDioJogAAGprGpCtVImJhpPk/roCzdb3fIh61/s/nPsfR6onyMwkCAR/OlC3yBC0lESvUoQEAssIrw
==}
  engines: {node: '>= 0.8'}
  dev: false

 /deprecation@2.3.1:
  resolution: {integrity: sha512-
xmHIy4F3scKVwMsQ4WnVaS8bHOx0DmVwRywosKhaILI0ywMDWPtBSku2HNxRvF7jtwDRsoEwYQSfbxj
8b7RlJQ==}
  dev: true

 /dequal@2.0.3:
  resolution: {integrity: sha512-
0je+qPKHEMohvfRTCEo3CrPG6cAzAYgmzKyxRiYSSDkS6eGJdyVJm7WaYA5ECaAD9wLB2T4EEeymA5
aFVcYXCA==}
  engines: {node: '>=6'}
  dev: true

 /destroy@1.2.0:
  resolution: {integrity: sha512-
2sJGJTaXIIaR1w4iJSNoN0hnMY7Gpc/n8D4qSCJw8QqFWXf7cuAgnEHxBpweaVcPevC2l3KpjYCx3NypQ
QgaJg==}
  engines: {node: '>= 0.8', npm: 1.2.8000 || >= 1.4.16}
  dev: false

 /detect-browser@5.3.0:
  resolution: {integrity: sha512-
53rsFbGdwMwlF7qvCt0ypLM5V5/Mbl0szB7GPN8y9NCcbknYOeVVXdrXEq+90IwAfrrzt6Hd+u2E2ntakICU8
w==}
```

```
    dev: false

  /didyoumean@1.2.2:
    resolution: {integrity: sha512-
gxtyfqMg7GKyhQmb056K7M3xszy/myH8w+B4RT+QXBQsvAOdc3XymqDDPHx1BgPgsdAA5SIifona89YtR
ATDzw==}
    dev: false

  /diff@4.0.2:
    resolution: {integrity: sha512-
58lmxKSA4BNyLz+HHMUzlOEpg09FV+ev6ZMe3vJihgdxzgcwZ8VoEEPmALCZG9LmqfVoNMMKpttIYTVG
6uDY7A==}
    engines: {node: '>=0.3.1'}

  /dijkstrajs@1.0.2:
    resolution: {integrity: sha512-
QV6PMaHTCNmKSeP6QoXhVTw9snc9VD8MulTT0Bd99Pacp4SS1cjcrYPgBPmibqKVtMJJfqC6XvOXgPM
EEPH/fg==}
    dev: false

  /dir-glob@3.0.1:
    resolution: {integrity: sha512-
WkrWp9GR4KXfKGYzOLmTuGVi1UWFfws377n9cc55/tb6DuqyF6pcQ5AbiHEshaDpY9v6oaSr2XCDidGmM
wdzIA==}
    engines: {node: '>=8'}
    dependencies:
      path-type: 4.0.0
    dev: true

  /dlv@1.1.3:
    resolution: {integrity: sha512-
+HlytyjIPKnIG8XuRG8WvmBP8xs8P71y+SKKS6ZXWoEgLuePxtDoUEiH7WkdePWrQ5JBpE6aoVqfZfJUQk
jXwA==}
    dev: false

  /doctrine@2.1.0:
    resolution: {integrity: sha512-
35mSku4ZXK0vfCuHEDAwt55dg2jNajHZ1odvF+8SSr82EsZY4QmXfuWso8oEd8zRhVObSN18aM0CjSdoB
X7zIw==}
    engines: {node: '>=0.10.0'}
    dependencies:
      esutils: 2.0.3
    dev: true

  /doctrine@3.0.0:
    resolution: {integrity: sha512-
yS+Q5i3hBf7GBkd4KG8a7eBNNWNGLTaEwwYWUijIYM7zrlYDM0BFXHjjPWlWZ1Rg7UaddZeIDmi9jF3H
mqiQ2w==}
    engines: {node: '>=6.0.0'}
    dependencies:
      esutils: 2.0.3
    dev: true
  /dom-walk@0.1.2:
    resolution: {integrity: sha512-6QvTW9mrGeIegrFXdtQi9pk7O/nSK6lSdXW2eqUspN5LWD7UTji2Fqw5V2YLjBpHEoU9Xl/eUWNpDeZvoyOv2w==}
    dev: false

  /dotenv@16.3.1:
    resolution: {integrity: sha512-IPzF4w4/Rd94bA9imS68tZBaYyBWSCE47V1RGuMrB94iyTOIEwRmVL2x/4An+6mETpLrKJ5hQkB8W4kFAadeIQ==}
    engines: {node: '>=12'}
    dev: false

  /duplexify@4.1.2:
    resolution: {integrity: sha512-fz3OjcNCHmRP12MJoZMPglx8m4rrFP8rovnk4vT8Fs+aonZoCwGg10dSsQsfP/E62eZcPTMSMP6686fu9Qlqtw==}
    dependencies:
      end-of-stream: 1.4.4
      inherits: 2.0.4
      readable-stream: 3.6.0
      stream-shift: 1.0.1
    dev: false

  /ecc-jsbn@0.1.2:
    resolution: {integrity: sha512-eh9O+hwRHNbG4BLTjEl3nw044CkGm5X6LoaCf7LPp7UU8Qrt47JYNi6nPX8xjW97TKGKm1ouctg0QSpZe9qrnw==}
    dependencies:
      jsbn: 0.1.1
      safer-buffer: 2.1.2
    dev: false

  /ee-first@1.1.1:
    resolution: {integrity: sha512-WMwm9LhRUo+WUaRN+vRuETqG89IgZphVSNkdFgeb6sS/E4OrDIN7t48CAewSHXc6C8lefD8KKfr5vY61brQlow==}
    dev: false

  /electron-to-chromium@1.4.568:
    resolution: {integrity: sha512-3TCOv8+BY6Ltpt1/CmGBMups2IdKOyfEmz4J8yIS4xLSeMm0Rf+psSaxLuswG9qMKt+XbNbmADybtXGpTFlbDg==}

  /elliptic@6.5.4:
    resolution: {integrity: sha512-iLhC6ULemrljPZb+QutR5TQGB+pdW6KGD5RSegS+8sorOZT+rdQFbsQFJgvN3eRqNALqJer4oQ16YvJHlU8hzQ==}
    dependencies:
      bn.js: 4.12.0
      brorand: 1.1.0
      hash.js: 1.1.7
      hmac-drbg: 1.0.1
      inherits: 2.0.4
      minimalistic-assert: 1.0.1
      minimalistic-crypto-utils: 1.0.1
    dev: false

  /emoji-regex@8.0.0:
    resolution: {integrity: sha512-
MSjYzcWNOA0ewAHpz0MxpYFvwg6yjy1NG3xteoqz644VCo/RPgnr1/GGt+ic3iJTzQ8Eu3TdM14SawnVUm
GE6A==}

  /emoji-regex@9.2.2:
    resolution: {integrity: sha512-
L18DaJsXSUk2+42pv8mLs5jJT2hqFkFE4j21wOmgbUqsZ2hL72NsUU785g9RXgo3s0ZNgVl42TiHp3ZtOv/V
yg==}
    dev: true

  /encode-utf8@1.0.3:
    resolution: {integrity: sha512-
ucAnuBEhUK4boH2HjVYG5Q2mQyPorvv0u/ocS+zhdw0S8AlHYY+GOFhP1Gio5z4icpP2ivFSvhtFjQi8+T9p
pw==}
    dev: false

  /encodeurl@1.0.2:
    resolution: {integrity: sha512-
TPJXq8JqFaVYm2CWmPvnP2Iyo4ZSM7/QKcSmuMLDObfpH5fi7RUGmd/rTDf+rut/saiDiQEeVTNgAmJEdA
Ox0w==}
    engines: {node: '>= 0.8'}
    dev: false

  /end-of-stream@1.4.4:
    resolution: {integrity: sha512-
+uw1inIHVPQoaVuHzRyXd21icM+cnt4CzD5rW+NC1wjOUSTOs+Te7FOv7AhN7vS9x/oIyhLP5PR1H+phQ
AHu5Q==}
    dependencies:
      once: 1.4.0
    dev: false

  /enhanced-resolve@5.15.0:
    resolution: {integrity: sha512-
LXYT42KJ7lpIKECr2mAXIaMldcNCh/7E0KBKOu4KSfkHmP+mZmSs+8V5gBAqisWBy0OO4W5Oyys0GO1
Y8KtdKg==}
    engines: {node: '>=10.13.0'}
    dependencies:
      graceful-fs: 4.2.11
      tapable: 2.2.1
    dev: true

  /errno@0.1.8:
    resolution: {integrity: sha512-
dJ6oBr5SQ1VSd9qkk7ByRgb/1SH4JZjCHSW/mr63/QcXO9zLVxvJ6Oy13nio03rxpSnVDDjFor75SjVeZWP
W/A==}
    hasBin: true
    dependencies:
      prr: 1.0.1
    dev: false

  /error-ex@1.3.2:
    resolution: {integrity: sha512-7dFHNmqeFSEt2ZBsCriorKnn3Z2pj+fd9kmI6QoWw4//DL+icEBfc0U7qJCisqrTsKTjw4fNFy2pW9OqStD84g==}
    dependencies:
      is-arrayish: 0.2.1
    dev: true

  /es-abstract@1.22.3:
    resolution: {integrity: sha512-eiiY8HQeYfYH2Con2berK+To6GrK2RxbPawDkGq4UiCQQfZHb6wX9qQqkbpPqaxQFcl8d9QzZqo0tGE0VcrdwA==}
    engines: {node: '>= 0.4'}
    dependencies:
      array-buffer-byte-length: 1.0.0
      arraybuffer.prototype.slice: 1.0.2
      available-typed-arrays: 1.0.5
      call-bind: 1.0.5
      es-set-tostringtag: 2.0.2
      es-to-primitive: 1.2.1
      function.prototype.name: 1.1.6
      get-intrinsic: 1.2.2
      get-symbol-description: 1.0.0
      globalthis: 1.0.3
      gopd: 1.0.1
      has-property-descriptors: 1.0.1
      has-proto: 1.0.1
      has-symbols: 1.0.3
      hasown: 2.0.0
      internal-slot: 1.0.6
      is-array-buffer: 3.0.2
      is-callable: 1.2.7
      is-negative-zero: 2.0.2
      is-regex: 1.1.4
      is-shared-array-buffer: 1.0.2
      is-string: 1.0.7
      is-typed-array: 1.1.12
      is-weakref: 1.0.2
      object-inspect: 1.13.1
      object-keys: 1.1.1
      object.assign: 4.1.4
      regexp.prototype.flags: 1.5.1
      safe-array-concat: 1.0.1
      safe-regex-test: 1.0.0
      string.prototype.trim: 1.2.8
      string.prototype.trimend: 1.0.7
      string.prototype.trimstart: 1.0.7
      typed-array-buffer: 1.0.0
      typed-array-byte-length: 1.0.0
      typed-array-byte-offset: 1.0.0
      typed-array-length: 1.0.4
      unbox-primitive: 1.0.2
      which-typed-array: 1.1.13
    dev: true

  /es-iterator-helpers@1.0.15:
    resolution: {integrity: sha512-
GhoY8uYqd6iwUl2kgjTm4CZAf6oo5mHK7BPqx3rKgx893YSsy0LGHV6gfqqQvZt/8xM8xeOnfXBCfqclMKkJ
5g==}
    dependencies:
      asynciterator.prototype: 1.0.0
      call-bind: 1.0.2
      define-properties: 1.2.1
      es-abstract: 1.22.3
      es-set-tostringtag: 2.0.2
      function-bind: 1.1.2
      get-intrinsic: 1.2.2
      globalthis: 1.0.3
      has-property-descriptors: 1.0.1
      has-proto: 1.0.1
      has-symbols: 1.0.3
      internal-slot: 1.0.6
      iterator.prototype: 1.1.2
      safe-array-concat: 1.0.1
    dev: true

  /es-set-tostringtag@2.0.2:
    resolution: {integrity: sha512-
BuDyupZt65P9D2D2vA/zqcI3G5xRsklm5N3xCwuiy+/vKy8i0ifdsQP1sLgO4tZDSCaQUSnmC48khknGMV3D
2Q==}
    engines: {node: '>= 0.4'}
    dependencies:
      get-intrinsic: 1.2.2
      has-tostringtag: 1.0.0
      hasown: 2.0.0
    dev: true

  /es-shim-unscopables@1.0.2:
    resolution: {integrity: sha512-
J3yBRXCzDu4ULnQwxyToo/OjdMx6akgVC7K6few0a7F/0wLtmKKN7I73AH5T2836UuXRqN7Qg+IIUw/+YJ
ksRw==}
    dependencies:
      hasown: 2.0.0
    dev: true

  /es-to-primitive@1.2.1:
    resolution: {integrity: sha512-
QCOllgZJtaUo9miYBcLChTUaHNjJF3PYs1VidD7AwiEj1kYxKeQTctLAezAOH5ZKRH0g2IgPn6KwB4IT8iRp
vA==}
    engines: {node: '>= 0.4'}
    dependencies:
      is-callable: 1.2.7
      is-date-object: 1.0.5
      is-symbol: 1.0.4
    dev: true

  /es5-ext@0.10.62:
    resolution: {integrity: sha512-
BHLqn0klhEpnOKSrzn/Xsz2UIW8j+cGmo9JLzr8BiUapV8hPL9+FliFqjwr9ngW7jWdnxv6eO+/LqyhJVqgrjA==
}
    engines: {node: '>=0.10'}
    requiresBuild: true
    dependencies:
      es6-iterator: 2.0.3
      es6-symbol: 3.1.3
      next-tick: 1.1.0
    dev: false

  /es6-iterator@2.0.3:
    resolution: {integrity: sha512-
zw4SRzoUkd+cl+ZoE15A9o1oQd920Bb0iOJMQkQhl3jNc03YqVjAhG7scf9C5KWRU/R13Orf588uCC6525o
02g==}
    dependencies:
      d: 1.0.1
      es5-ext: 0.10.62
      es6-symbol: 3.1.3
    dev: false

  /es6-promise@4.2.8:
    resolution: {integrity: sha512-
HJDGx5daxeIvxdBxvG2cb9g4tEvwIk3i8+nhX0yGrYmZUzbkdg8QbDevheDB8gd0//uPj4c1EQua8Q+MViT0/
w==}
    dev: false

  /es6-promisify@5.0.0:
    resolution: {integrity: sha512-
C+d6UdsYDk0lMebHNR4S2NybQMMngAOnOwYBQjTOiv0MkoJMP0Myw2mgpDLBcpfCmRLxyFqYhS/CfO
ENq4SJhQ==}
    dependencies:
      es6-promise: 4.2.8
    dev: false

  /es6-symbol@3.1.3:
    resolution: {integrity: sha512-
NJ6Yn3FuDinBaBRWl/q5X/s4koRHBrgKAu+yGI6JCBeiu3qrcbJhwT2GeR/EXVfylRk8dpQVJoLEFhK+Mu31
NA==}
    dependencies:
      d: 1.0.1
      ext: 1.7.0
    dev: false
  /esbuild@0.18.20:
    resolution: {integrity: sha512-
ceqxoedUrcayh7Y7ZX6NdbbDzGROiyVBgC4PriJThBKSVPWnnFHZAkfI1lJT8QFkOwH4qOS2SJkS4wvpGl
8BpA==}
    engines: {node: '>=12'}
    hasBin: true
    requiresBuild: true
    optionalDependencies:
      '@esbuild/android-arm': 0.18.20
      '@esbuild/android-arm64': 0.18.20
      '@esbuild/android-x64': 0.18.20
      '@esbuild/darwin-arm64': 0.18.20
      '@esbuild/darwin-x64': 0.18.20
      '@esbuild/freebsd-arm64': 0.18.20
      '@esbuild/freebsd-x64': 0.18.20
      '@esbuild/linux-arm': 0.18.20
      '@esbuild/linux-arm64': 0.18.20
      '@esbuild/linux-ia32': 0.18.20
      '@esbuild/linux-loong64': 0.18.20
      '@esbuild/linux-mips64el': 0.18.20
      '@esbuild/linux-ppc64': 0.18.20
      '@esbuild/linux-riscv64': 0.18.20
      '@esbuild/linux-s390x': 0.18.20
      '@esbuild/linux-x64': 0.18.20
      '@esbuild/netbsd-x64': 0.18.20
      '@esbuild/openbsd-x64': 0.18.20
      '@esbuild/sunos-x64': 0.18.20
      '@esbuild/win32-arm64': 0.18.20
      '@esbuild/win32-ia32': 0.18.20
      '@esbuild/win32-x64': 0.18.20
    dev: true

  /escalade@3.1.1:
    resolution: {integrity: sha512-
k0er2gUkLf8O0zKJiAhmkTnJlTvINGv7ygDNPbeIsX/TJjGJZHuh9B2UxbsaEkmlEo9MfhrSzmhIlhRlI2GXnw==
}
    engines: {node: '>=6'}

  /escape-html@1.0.3:
    resolution: {integrity: sha512-
NiSupZ4OeuGwr68lGIeym/ksIZMJodUGOSCZ/FSnTxcrekbvqrgdUxlJOMpijaKZVjAJrWrGs/6Jy8OMuyj9ow
==}
    dev: false

  /escape-string-regexp@1.0.5:
    resolution: {integrity: sha512-
vbRorB5FUQWvla16U8R/qgaFIya2qGzwDrNmCZuYKrbdSUMG6I1ZCGQRefkRVhuOkIGVne7BQ35DSfo1
qvJqFg==}
    engines: {node: '>=0.8.0'}

  /escape-string-regexp@4.0.0:
```

```yaml
    resolution: {integrity: sha512-TtpcNJ3XAzx3Gq8sWRzJaVajRs0uVxA2YAkdb1jm2YkPz4G6egUFAyA3n5vtEIZefPk5Wa4UXbKuS5fKkJWdgA==}
    engines: {node: '>=10'}
    dev: true

  /eslint-config-next@14.0.0(eslint@8.52.0)(typescript@5.2.2):
    resolution: {integrity: sha512-jtXeE+/pGQ3h9n11QyyuPN50kO13GO5XvjU5ZRq6W+XTpOMjyobWmK2s7aowy0FtzA49krJzYzEU9s1RMwoJ6g==}
    peerDependencies:
      eslint: ^7.23.0 || ^8.0.0
      typescript: '>=3.3.1'
    peerDependenciesMeta:
      typescript:
        optional: true
    dependencies:
      '@next/eslint-plugin-next': 14.0.0
      '@rushstack/eslint-patch': 1.5.1
      '@typescript-eslint/parser': 6.9.0(eslint@8.52.0)(typescript@5.2.2)
      eslint: 8.52.0
      eslint-import-resolver-node: 0.3.9
      eslint-import-resolver-typescript: 3.6.1(@typescript-eslint/parser@6.9.0)(eslint-import-resolver-node@0.3.9)(eslint-plugin-import@2.29.0)(eslint@8.52.0)
      eslint-plugin-import: 2.29.0(@typescript-eslint/parser@6.9.0)(eslint-import-resolver-typescript@3.6.1)(eslint@8.52.0)
      eslint-plugin-jsx-a11y: 6.7.1(eslint@8.52.0)
      eslint-plugin-react: 7.33.2(eslint@8.52.0)
      eslint-plugin-react-hooks: 4.6.0(eslint@8.52.0)
      typescript: 5.2.2
    transitivePeerDependencies:
      - eslint-import-resolver-webpack
      - supports-color
    dev: true

  /eslint-config-prettier@9.0.0(eslint@8.52.0):
    resolution: {integrity: sha512-IcJsTkJae2S35pRsRAwoCE+925rJJStOdkKnLVgtE+tEpqU0EVVM7OqrwxqgptKdX29NUwC82I5pXsGFIgSevw==}
    hasBin: true
    peerDependencies:
      eslint: '>=7.0.0'
    dependencies:
      eslint: 8.52.0
    dev: true

  /eslint-import-resolver-node@0.3.9:
    resolution: {integrity: sha512-WFj2isz22JahUv+B788TlO3N6zL3nNJGU8CcZbPZvVEkBPaJdCV4vy5wyghty5ROFbCRnm132v8BScu5/1BQ8g==}
    dependencies:
      debug: 3.2.7(supports-color@5.5.0)
      is-core-module: 2.13.1
      resolve: 1.22.8
    transitivePeerDependencies:
      - supports-color
    dev: true

  /eslint-import-resolver-typescript@3.6.1(@typescript-eslint/parser@6.9.0)(eslint-import-resolver-node@0.3.9)(eslint-plugin-import@2.29.0)(eslint@8.52.0):
    resolution: {integrity: sha512-xgdptdoi5W3niYeuQxKmzVDTATvLYqhpwmykwsh7f6HIOStGWEIL9iqZgQDF9u9OEzrRwR8no5q2VT+bjAujTg==}
    engines: {node: ^14.18.0 || >=16.0.0}
    peerDependencies:
      eslint: '*'
      eslint-plugin-import: '*'
    dependencies:
      debug: 4.3.4
      enhanced-resolve: 5.15.0
      eslint: 8.52.0
      eslint-module-utils: 2.8.0(@typescript-eslint/parser@6.9.0)(eslint-import-resolver-node@0.3.9)(eslint-import-resolver-typescript@3.6.1)(eslint@8.52.0)
      eslint-plugin-import: 2.29.0(@typescript-eslint/parser@6.9.0)(eslint-import-resolver-typescript@3.6.1)(eslint@8.52.0)
      fast-glob: 3.3.1
      get-tsconfig: 4.7.2
      is-core-module: 2.13.1
      is-glob: 4.0.3
    transitivePeerDependencies:
      - '@typescript-eslint/parser'
      - eslint-import-resolver-node
      - eslint-import-resolver-webpack
      - supports-color
    dev: true

  /eslint-module-utils@2.8.0(@typescript-eslint/parser@6.9.0)(eslint-import-resolver-node@0.3.9)(eslint-import-resolver-typescript@3.6.1)(eslint@8.52.0):
    resolution: {integrity: sha512-aWajIYfsqCKRDgUfjEXNN/JlrzauMuSEy5sbd7WXbtW3EH6A6MpwEh42c7qD+MqQo9QMJ6fWLAeIJynx0g6OAw==}
    engines: {node: '>=4'}
    peerDependencies:
      '@typescript-eslint/parser': '*'
      eslint: '*'
      eslint-import-resolver-node: '*'
      eslint-import-resolver-typescript: '*'
      eslint-import-resolver-webpack: '*'
    peerDependenciesMeta:
      '@typescript-eslint/parser':
        optional: true
      eslint:
        optional: true
      eslint-import-resolver-node:
        optional: true
      eslint-import-resolver-typescript:
        optional: true
      eslint-import-resolver-webpack:
        optional: true
    dependencies:
      '@typescript-eslint/parser': 6.9.0(eslint@8.52.0)(typescript@5.2.2)
      debug: 3.2.7(supports-color@5.5.0)
      eslint: 8.52.0
      eslint-import-resolver-node: 0.3.9
      eslint-import-resolver-typescript: 3.6.1(@typescript-eslint/parser@6.9.0)(eslint-import-resolver-node@0.3.9)(eslint-plugin-import@2.29.0)(eslint@8.52.0)
    transitivePeerDependencies:
      - supports-color
    dev: true

  /eslint-plugin-import@2.29.0(@typescript-eslint/parser@6.9.0)(eslint-import-resolver-typescript@3.6.1)(eslint@8.52.0):
    resolution: {integrity: sha512-QPOO5NO6Odv5lpoTkddtutccQjysJuFxoPS7fAHO+9m9udNHvTCPSAMW9zGAYj8lAIdr40I8yPCdUYrncXtrwg==}
    engines: {node: '>=4'}
    peerDependencies:
      '@typescript-eslint/parser': '*'
      eslint: ^2 || ^3 || ^4 || ^5 || ^6 || ^7.2.0 || ^8
    peerDependenciesMeta:
      '@typescript-eslint/parser':
        optional: true
    dependencies:
      '@typescript-eslint/parser': 6.9.0(eslint@8.52.0)(typescript@5.2.2)
      array-includes: 3.1.7
      array.prototype.findlastindex: 1.2.3
      array.prototype.flat: 1.3.2
      array.prototype.flatmap: 1.3.2
      debug: 3.2.7(supports-color@5.5.0)
      doctrine: 2.1.0
      eslint: 8.52.0
      eslint-import-resolver-node: 0.3.9
      eslint-module-utils: 2.8.0(@typescript-eslint/parser@6.9.0)(eslint-import-resolver-node@0.3.9)(eslint-import-resolver-typescript@3.6.1)(eslint@8.52.0)
      hasown: 2.0.0
      is-core-module: 2.13.1
      is-glob: 4.0.3
      minimatch: 3.1.2
      object.fromentries: 2.0.7
      object.groupby: 1.0.1
      object.values: 1.1.7
      semver: 6.3.1
      tsconfig-paths: 3.14.2
    transitivePeerDependencies:
      - eslint-import-resolver-typescript
      - eslint-import-resolver-webpack
      - supports-color
    dev: true

  /eslint-plugin-jsx-a11y@6.7.1(eslint@8.52.0):
    resolution: {integrity: sha512-
63Bog4iIethyo8smBklORknVjB0T2dwB8Mr/hIC+fBS0uyHdYYpzM/Ed+YC8VxTjlXHEWFOdmgwcDn1U2L9
VCA==}
    engines: {node: '>=4.0'}
    peerDependencies:
      eslint: ^3 || ^4 || ^5 || ^6 || ^7 || ^8
    dependencies:
      '@babel/runtime': 7.23.2
      aria-query: 5.3.0
      array-includes: 3.1.7
      array.prototype.flatmap: 1.3.2
      ast-types-flow: 0.0.7
      axe-core: 4.8.2
      axobject-query: 3.2.1
      damerau-levenshtein: 1.0.8
      emoji-regex: 9.2.2
      eslint: 8.52.0
      has: 1.0.3
      jsx-ast-utils: 3.3.5
      language-tags: 1.0.5
      minimatch: 3.1.2
      object.entries: 1.1.7
      object.fromentries: 2.0.7
      semver: 6.3.1
    dev: true

  /eslint-plugin-prettier@5.0.1(eslint-config-prettier@9.0.0)(eslint@8.52.0)(prettier@3.0.3):
    resolution: {integrity: sha512-
m3u5RnR56asrwV/lDC4GHorlW75DsFfmUcjfCYylTUs85dBRnB7VM6xG8eCMJdeDRnppzmxZVf1GEPJvl1
JmNg==}
    engines: {node: ^14.18.0 || >=16.0.0}
    peerDependencies:
      '@types/eslint': '>=8.0.0'
      eslint: '>=8.0.0'
      eslint-config-prettier: '*'
      prettier: '>=3.0.0'
    peerDependenciesMeta:
      '@types/eslint':
        optional: true
      eslint-config-prettier:
        optional: true
    dependencies:
      eslint: 8.52.0
      eslint-config-prettier: 9.0.0(eslint@8.52.0)
      prettier: 3.0.3
      prettier-linter-helpers: 1.0.0
      synckit: 0.8.5
    dev: true
  /eslint-plugin-react-hooks@4.6.0(eslint@8.52.0):
    resolution: {integrity: sha512-oFc7Itz9Qxh2x4gNHStv3BqJq54ExXmfC+a1NjAta66IAN87Wu0R/QArgIS9qKzX3dXKPI9H5crl9QchNMY9+g==}
    engines: {node: '>=10'}
    peerDependencies:
      eslint: ^3.0.0 || ^4.0.0 || ^5.0.0 || ^6.0.0 || ^7.0.0 || ^8.0.0-0
    dependencies:
      eslint: 8.52.0
    dev: true

  /eslint-plugin-react@7.33.2(eslint@8.52.0):
    resolution: {integrity: sha512-73QQMKALArI8/7xGLNI/3LylrEYrlKZSb5C9+q3OtOewTnMQi5cT+aE9E41sLCmli3I9PGGmD1yiZydyo4FEPw==}
    engines: {node: '>=4'}
    peerDependencies:
      eslint: ^3 || ^4 || ^5 || ^6 || ^7 || ^8
    dependencies:
      array-includes: 3.1.7
      array.prototype.flatmap: 1.3.2
      array.prototype.tosorted: 1.1.2
      doctrine: 2.1.0
      es-iterator-helpers: 1.0.15
      eslint: 8.52.0
      estraverse: 5.3.0
      jsx-ast-utils: 3.3.5
      minimatch: 3.1.2
      object.entries: 1.1.7
      object.fromentries: 2.0.7
      object.hasown: 1.1.3
      object.values: 1.1.7
      prop-types: 15.8.1
      resolve: 2.0.0-next.5
      semver: 6.3.1
      string.prototype.matchall: 4.0.10
    dev: true

  /eslint-plugin-unicorn@48.0.1(eslint@8.52.0):
    resolution: {integrity: sha512-FW+4r20myG/DqFcCSzoumaddKBicIPeFnTrifon2mWIzlfyvzwyqZjqVP7m4Cqr/ZYisS2aiLghkUWaPg6vtCw==}
    engines: {node: '>=16'}
    peerDependencies:
      eslint: '>=8.44.0'
    dependencies:
      '@babel/helper-validator-identifier': 7.22.20
      '@eslint-community/eslint-utils': 4.4.0(eslint@8.52.0)
      ci-info: 3.9.0
      clean-regexp: 1.0.0
      eslint: 8.52.0
      esquery: 1.5.0
      indent-string: 4.0.0
      is-builtin-module: 3.2.1
      jsesc: 3.0.2
      lodash: 4.17.21
      pluralize: 8.0.0
      read-pkg-up: 7.0.1
      regexp-tree: 0.1.27
      regjsparser: 0.10.0
      semver: 7.5.4
      strip-indent: 3.0.0
    dev: true

  /eslint-scope@7.2.2:
    resolution: {integrity: sha512-dOt21O7lTMhDM+X9mB4GX+DZrZtCUJPL/wlcTqxyrx5IvO0IYtILdtrQGQp+8n5S0gwSVmOf9NQrjMOgfQZlIg==}
    engines: {node: ^12.22.0 || ^14.17.0 || >=16.0.0}
    dependencies:
      esrecurse: 4.3.0
      estraverse: 5.3.0
    dev: true

  /eslint-visitor-keys@3.4.3:
    resolution: {integrity: sha512-wpc+LXeiyiisxPlEkUzU6svyS1frIO3Mgxj1fdy7Pm8Ygzguax2N3Fa/D/ag1WqbOprdI+uY6wMUl8/a2G+iag==}
    engines: {node: ^12.22.0 || ^14.17.0 || >=16.0.0}
    dev: true

  /eslint@8.52.0:
    resolution: {integrity: sha512-zh/JHnaixqHZsolRB/w9/02akBk9EPrOs9JwcTP2ek7yL5bVvXuRariiaAjjoJ5DvuwQ1WAE/HsMz+w17YgBCg==}
    engines: {node: ^12.22.0 || ^14.17.0 || >=16.0.0}
    hasBin: true
    dependencies:
      '@eslint-community/eslint-utils': 4.4.0(eslint@8.52.0)
      '@eslint-community/regexpp': 4.10.0
      '@eslint/eslintrc': 2.1.2
      '@eslint/js': 8.52.0
      '@humanwhocodes/config-array': 0.11.13
      '@humanwhocodes/module-importer': 1.0.1
      '@nodelib/fs.walk': 1.2.8
      '@ungap/structured-clone': 1.2.0
      ajv: 6.12.6
      chalk: 4.1.2
      cross-spawn: 7.0.3
      debug: 4.3.4
      doctrine: 3.0.0
      escape-string-regexp: 4.0.0
      eslint-scope: 7.2.2
      eslint-visitor-keys: 3.4.3
      espree: 9.6.1
      esquery: 1.5.0
      esutils: 2.0.3
      fast-deep-equal: 3.1.3
      file-entry-cache: 6.0.1
      find-up: 5.0.0
      glob-parent: 6.0.2
      globals: 13.23.0
      graphemer: 1.4.0
      ignore: 5.2.4
      imurmurhash: 0.1.4
      is-glob: 4.0.3
      is-path-inside: 3.0.3
      js-yaml: 4.1.0
      json-stable-stringify-without-jsonify: 1.0.1
      levn: 0.4.1
      lodash.merge: 4.6.2
      minimatch: 3.1.2
      natural-compare: 1.4.0
      optionator: 0.9.3
      strip-ansi: 6.0.1
      text-table: 0.2.0
    transitivePeerDependencies:
      - supports-color
    dev: true

  /espree@9.6.1:
    resolution: {integrity: sha512-
oruZaFkjorTpF32kDSI5/75ViwGeZginGGy2NoOSg3Q9bnwlnmDm4HLnkl0RE3n+njDXR037aY1+x58Z/zFd
wQ==}
    engines: {node: ^12.22.0 || ^14.17.0 || >=16.0.0}
    dependencies:
      acorn: 8.11.2
      acorn-jsx: 5.3.2(acorn@8.11.2)
      eslint-visitor-keys: 3.4.3
    dev: true

  /esquery@1.5.0:
    resolution: {integrity: sha512-
YQLXUplAwJgCydQ78IMJywZCceoqk1oH01OERdSAJc/7U2AylwjhSCLDEtqwg811idIS/9fIU5GjG73IgjKMV
g==}
    engines: {node: '>=0.10'}
    dependencies:
      estraverse: 5.3.0
    dev: true

  /esrecurse@4.3.0:
    resolution: {integrity: sha512-
KmfKL3b6G+RXvP8N1vr3Tq1kL/oCFgn2NYXEtqP8/L3pKapUA4G8cFVaoF3SU323CD4XypR/ffioHmkti6/Ta
g==}
    engines: {node: '>=4.0'}
    dependencies:
```

```
    estraverse: 5.3.0
   dev: true

  /estraverse@5.3.0:
   resolution: {integrity: sha512-
MMdARuVEQziNTeJD8DgMqmhwR11BRQ/cBP+pLtYdSTnf3MIO8fFeiINEbX36ZdNlfU/7A9f3gUw49B3oQs
vwBA==}
   engines: {node: '>=4.0'}
   dev: true

  /esutils@2.0.3:
   resolution: {integrity: sha512-
kVscqXk4OCp68SZ0dkgEKVi6/8ij300KBWTJq32P/dYeWTSwK41WyTxalN1eRmA5Z9UU/LX9D7FWSmV9
SAYx6g==}
   engines: {node: '>=0.10.0'}
   dev: true

  /etag@1.8.1:
   resolution: {integrity: sha512-
aIL5Fx7mawVa300al2BnEE4iNvo1qETxLrPI/o05L7z6go7fCw1J6EQmbK4FmJ2AS7kgVF/KEZWufBfdClMc
Pg==}
   engines: {node: '>= 0.6'}
   dev: false

  /eth-block-tracker@4.4.3(@babel/core@7.23.2):
   resolution: {integrity: sha512-
A8tG4Z4iNg4mw5tP1Vung9N9IjgMNqpiMoJ/FouSFwNCGHv2X0mmOYwtQOJzki6XN7r7Tyo01S29p7b224I
4jw==}
   dependencies:
    '@babel/plugin-transform-runtime': 7.23.2(@babel/core@7.23.2)
    '@babel/runtime': 7.23.2
    eth-query: 2.1.2
    json-rpc-random-id: 1.0.1
    pify: 3.0.0
    safe-event-emitter: 1.0.1
   transitivePeerDependencies:
    - '@babel/core'
    - supports-color
   dev: false

  /eth-block-tracker@6.1.0:
   resolution: {integrity: sha512-
K9SY8+/xMBi4M5HHTDdxnpEqEEGjbNpzHFqvxyjMZej8InV/B+CkFRKM6W+uvrFJ7m8Zd1E0qUkseU3vdI
DFYQ==}
   engines: {node: '>=14.0.0'}
   dependencies:
    '@metamask/safe-event-emitter': 2.0.0
    '@metamask/utils': 3.6.0
    json-rpc-random-id: 1.0.1
    pify: 3.0.0
   transitivePeerDependencies:
    - supports-color
    dev: false

  /eth-ens-namehash@2.0.8:
    resolution: {integrity: sha512-VWEI1+KJfz4Km//dadyvBBoBeSQ0MHTXPvr8UIXiLW6IanxvAV+DmlZAijZwAyggqGUfwQBeHf7tc9wzc1piSw==}
    dependencies:
      idna-uts46-hx: 2.3.1
      js-sha3: 0.5.7
    dev: false

  /eth-json-rpc-filters@4.2.2:
    resolution: {integrity: sha512-DGtqpLU7bBg63wPMWg1sCpkKCf57dJ+hj/k3zF26anXMzkmtSBDExL8IhUu7LUd34f0Zsce3PYNO2vV2GaTzaw==}
    dependencies:
      '@metamask/safe-event-emitter': 2.0.0
      async-mutex: 0.2.6
      eth-json-rpc-middleware: 6.0.0
      eth-query: 2.1.2
      json-rpc-engine: 6.1.0
      pify: 5.0.0
    transitivePeerDependencies:
      - encoding
    dev: false

  /eth-json-rpc-filters@5.1.0:
    resolution: {integrity: sha512-fos+9xmoa1A2Ytsc9eYof17r81BjdJOUcGcgZn4K/tKdCCTb+a8ytEtwlu1op5qsXFDlgGmstTELFrDEc89qEQ==}
    engines: {node: '>=14.0.0'}
    dependencies:
      '@metamask/safe-event-emitter': 2.0.0
      async-mutex: 0.2.6
      eth-query: 2.1.2
      json-rpc-engine: 6.1.0
      pify: 5.0.0
    dev: false

  /eth-json-rpc-infura@5.1.0:
    resolution: {integrity: sha512-THzLye3PHUSGn1EXMhg6WTLW9uim7LQZKeKaeYsS9+wOBcamRiCQVGHa6D2/4P0oS0vSaxsBnU/J6qvn0MPdow==}
    deprecated: Package no longer supported. Contact Support at https://www.npmjs.com/support for more info.
    dependencies:
      eth-json-rpc-middleware: 6.0.0
      eth-rpc-errors: 3.0.0
      json-rpc-engine: 5.4.0
      node-fetch: 2.7.0
    transitivePeerDependencies:
      - encoding
    dev: false

  /eth-json-rpc-middleware@6.0.0:
    resolution: {integrity: sha512-
qqBfLU2Uq1Ou15Wox1s+NX05S9OcAEL4JZ04VZox2NS0U+RtCMjSxzXhLFWekdShUPZ+P8ax3zCO2xcP
rp6XJQ==}
    dependencies:
      btoa: 1.2.1
      clone: 2.1.2
      eth-query: 2.1.2
      eth-rpc-errors: 3.0.0
      eth-sig-util: 1.4.2
      ethereumjs-util: 5.2.1
      json-rpc-engine: 5.4.0
      json-stable-stringify: 1.0.2
      node-fetch: 2.7.0
      pify: 3.0.0
      safe-event-emitter: 1.0.1
    transitivePeerDependencies:
      - encoding
    dev: false

  /eth-lib@0.1.29:
    resolution: {integrity: sha512-
bfttrr3/7gG4E02HoWTDUcDDslN003OlOoBxk9virpAZQ1ja/jDgwkWB8QfJF7ojuEowrqy+lzp9VcJG7/k5bQ==
}
    dependencies:
      bn.js: 4.12.0
      elliptic: 6.5.4
      nano-json-stream-parser: 0.1.2
      servify: 0.1.12
      ws: 3.3.3
      xhr-request-promise: 0.1.3
    transitivePeerDependencies:
      - bufferutil
      - supports-color
      - utf-8-validate
    dev: false

  /eth-lib@0.2.8:
    resolution: {integrity: sha512-
ArJ7x1WcWOlSpzdoTBX8vkwlkSQ85CjjifSZtV4co64vWxSV8geWfPI9x4SVYu3DSxnX4yWFVTtGL+j9DUFL
Nw==}
    dependencies:
      bn.js: 4.12.0
      elliptic: 6.5.4
      xhr-request-promise: 0.1.3
    dev: false

  /eth-query@2.1.2:
    resolution: {integrity: sha512-
srES0ZcvwkR/wd5OQBRA1bIJMww1skfGS0s8wlwK3/oNP4+wnds60krvu5R1QbpRQjMmpG5OMIWro5s7g
vDPsA==}
    dependencies:
      json-rpc-random-id: 1.0.1
      xtend: 4.0.2
    dev: false

  /eth-rpc-errors@3.0.0:
    resolution: {integrity: sha512-
iPPNHPrLwUlR9xCSYm7HHQjWBasor3+KZfRvwEWxMz3ca0yqnlBeJrnyphkGIXZ4J7AMAaOLmwy4AWhn
xOiLxg==}
    dependencies:
      fast-safe-stringify: 2.1.1
    dev: false

  /eth-rpc-errors@4.0.2:
    resolution: {integrity: sha512-
n+Re6Gu8XGyfFy1it0AwbD1x0MUzspQs0D5UiPs1fFPCr6WAwZM+vbIhXheBFrpgosqN9bs5PqlB4Q61U/Q
ytQ==}
    dependencies:
      fast-safe-stringify: 2.1.1
    dev: false

  /eth-sig-util@1.4.2:
    resolution: {integrity: sha512-
iNZ576iTOGcfllftB73cPB5AN+XUQAT/T8xzsILsghXC1o8gJUqe3RHlcDqagu+biFpYQ61KQrZZJza8eRSYq
w==}
    deprecated: Deprecated in favor of '@metamask/eth-sig-util'
    dependencies:
      ethereumjs-abi: github.com/ethereumjs/ethereumjs-
abi/ee3994657fa7a427238e6ba92a84d0b529bbcde0
      ethereumjs-util: 5.2.1
    dev: false

  /ethereum-bloom-filters@1.0.10:
    resolution: {integrity: sha512-
rxJ5OFN3RwjQxDcFP2Z5+Q9ho4eIdEmSc2ht0fCu8Se9nbXjZ7/031uXoUYJ87KHCOdVeiUuwSnoS7hmYA
GVHA==}
    dependencies:
      js-sha3: 0.8.0
    dev: false

  /ethereum-common@0.0.18:
    resolution: {integrity: sha512-
EoltVQTRNg2Uy4o84qpa2aXymXDJhxm7eos/ACOg0DG4baAbMjhbdAEsx9GeE8sC3XCxnYvrrzZDH8D8M
tA2iQ==}
    dev: false

  /ethereum-common@0.2.0:
    resolution: {integrity: sha512-
XOnAR/3rntJgbCdGhqdaLIxDLWKLmsZOGhHdBKadEr6gEnJLH52k93Ou+TUdFaPN3hJc3isBZBal3U/XZ1
5abA==}
    dev: false
  /ethereum-cryptography@0.1.3:
    resolution: {integrity: sha512-
w8/4x1SGGzc+tO97TASLja6SLd3fRIK2tLVcV2Gx4IB21hE19atll5Cq9o3d0ZmAYC/8aw0ipieTSiekAea4SQ==
}
    dependencies:
      '@types/pbkdf2': 3.1.1
      '@types/secp256k1': 4.0.5
      blakejs: 1.2.1
      browserify-aes: 1.2.0
      bs58check: 2.1.2
      create-hash: 1.2.0
      create-hmac: 1.1.7
      hash.js: 1.1.7
      keccak: 3.0.2
      pbkdf2: 3.1.2
      randombytes: 2.1.0
      safe-buffer: 5.2.1
      scrypt-js: 3.0.1
      secp256k1: 4.0.3
      setimmediate: 1.0.5
    dev: false

  /ethereum-cryptography@1.1.2:
    resolution: {integrity: sha512-
XDSJlg4BD+hq9N2FjvotwUET9Tfxpxc3kWGE2AqUG5vcbeunnbImVk3cj6e/xT3phdW21mE8R5IugU4fspQ
DcQ==}
    dependencies:
      '@noble/hashes': 1.1.2
      '@noble/secp256k1': 1.6.3
      '@scure/bip32': 1.1.0
      '@scure/bip39': 1.1.0
    dev: false

  /ethereum-protocol@1.0.1:
    resolution: {integrity: sha512-
3KLX1mHuEsBW0dKG+c6EOJS1NBNqdCICvZW9sInmZTt5aY0oxmHVggYRE0lJu1tcnMD1K+AKHdLi6U4
3Awm1Vg==}
    dev: false

  /ethereumjs-abi@0.6.8:
    resolution: {integrity: sha512-
Tx0r/iXI6r+lRsdvkFDlut0N08jWMnKRZ6Gkq+Nmw75lZe4e6o3EkSnkaBP5NF6+m5PTGAr9JP43N3Lyeogls
A==}
    dependencies:
      bn.js: 4.12.0
      ethereumjs-util: 6.2.1
    dev: false

  /ethereumjs-account@2.0.5:
    resolution: {integrity: sha512-
bgDojnXGjhMwo6eXQC0bY6UK2liSFUSMwwylOmQvZbSl/D7NXQ3+vrGO46ZeOgjGfxXmgIeVNDIiHw7fNZ
```

```
M4VA==}
    dependencies:
      ethereumjs-util: 5.2.1
      rlp: 2.2.7
      safe-buffer: 5.2.1
    dev: false

  /ethereumjs-block@1.7.1:
    resolution: {integrity: sha512-
B+sSdtqm78fmKkBq78/QLKJbu/4Ts4P2KFISdgcuZUPDm9x+N7qgBPIIFUGbaakQh8bzuquiRVbdmvPKqbI
LRg==}
    deprecated: 'New package name format for new versions: @ethereumjs/block. Please update.'
    dependencies:
      async: 2.6.4
      ethereum-common: 0.2.0
      ethereumjs-tx: 1.3.7
      ethereumjs-util: 5.2.1
      merkle-patricia-tree: 2.3.2
    dev: false

  /ethereumjs-block@2.2.2:
    resolution: {integrity: sha512-
2p49ifhek3h2zeg/+da6XpdFR3GlqY3BIEiqxGF8j9aSRIgkb7M1Ky+yULBKJOu8PAZxfhsYA+HxUk2aCQp3v
g==}
    deprecated: 'New package name format for new versions: @ethereumjs/block. Please update.'
    dependencies:
      async: 2.6.4
      ethereumjs-common: 1.5.2
      ethereumjs-tx: 2.1.2
      ethereumjs-util: 5.2.1
      merkle-patricia-tree: 2.3.2
    dev: false

  /ethereumjs-common@1.5.2:
    resolution: {integrity: sha512-
hTfZjwGX52GS2jcVO6E2sx4YuFnf0Fhp5ylo4pEPhEffNln7vS59Hr5sLnp3/QCazFLluuBZ+FZ6J5HTp0EqCA
==}
    deprecated: 'New package name format for new versions: @ethereumjs/common. Please update.'
    dev: false

  /ethereumjs-tx@1.3.7:
    resolution: {integrity: sha512-
wvLMxzt1RPhAQ9Yi3/HKZTn0FZYpnsmQdbKYfUUpi4j1SEIcbkd9tndVjcPrufY3V7j2IebOpC00Zp2P/Ay2kA
==}
    deprecated: 'New package name format for new versions: @ethereumjs/tx. Please update.'
    dependencies:
      ethereum-common: 0.0.18
      ethereumjs-util: 5.2.1
    dev: false

  /ethereumjs-tx@2.1.2:
    resolution: {integrity: sha512-
```

```
zZEK1onCeiORb0wyCXUvg94Ve5It/K6GD1K+26KfFKodiBiS6d9lfCXlUKGBBdQ+bv7Day+JK0tj1K+BeNFR
Aw==}
    deprecated: 'New package name format for new versions: @ethereumjs/tx. Please update.'
    dependencies:
      ethereumjs-common: 1.5.2
      ethereumjs-util: 6.2.1
    dev: false

  /ethereumjs-util@5.2.1:
    resolution: {integrity: sha512-
v3kT+7zdyCm1HIqWlLNrHGqHGLpGYIhjeHxQjnDXjLT2FyGJDsd3LWMYUo7pAFRrk86CR3nUJfhC81CCo
JNNGQ==}
    dependencies:
      bn.js: 4.12.0
      create-hash: 1.2.0
      elliptic: 6.5.4
      ethereum-cryptography: 0.1.3
      ethjs-util: 0.1.6
      rlp: 2.2.7
      safe-buffer: 5.2.1
    dev: false

  /ethereumjs-util@6.2.1:
    resolution: {integrity: sha512-
W2Ktez4L01Vexijrm5EB6w7dg4n/TgpoYU4avuT5T3Vmnw/eCRtiBrJfQYS/DCSvDIOLn2k57GcHdeBcgVxA
qw==}
    dependencies:
      '@types/bn.js': 4.11.6
      bn.js: 4.12.0
      create-hash: 1.2.0
      elliptic: 6.5.4
      ethereum-cryptography: 0.1.3
      ethjs-util: 0.1.6
      rlp: 2.2.7
    dev: false

  /ethereumjs-util@7.1.5:
    resolution: {integrity: sha512-
SDl5kKrQAudFBUe5OJM9Ac6WmMyYmXX/6sTmLZ3ffG2eY6ZIGBes3pEDxNN6V72WyOw4CPD5RomKd
sa8DAAwLg==}
    engines: {node: '>=10.0.0'}
    dependencies:
      '@types/bn.js': 5.1.3
      bn.js: 5.2.1
      create-hash: 1.2.0
      ethereum-cryptography: 0.1.3
      rlp: 2.2.7
    dev: false

  /ethereumjs-vm@2.6.0:
    resolution: {integrity: sha512-
r/XIUik/ynGbxS3y+mvGnbOKnuLo40V5Mj1J25+HEO63aWYREIqvWeRO/hnROlMBE5WoniQmPmhiaN0cti
```

```
HaXw==}
    deprecated: 'New package name format for new versions: @ethereumjs/vm. Please update.'
    dependencies:
      async: 2.6.4
      async-eventemitter: 0.2.4
      ethereumjs-account: 2.0.5
      ethereumjs-block: 2.2.2
      ethereumjs-common: 1.5.2
      ethereumjs-util: 6.2.1
      fake-merkle-patricia-tree: 1.0.1
      functional-red-black-tree: 1.0.1
      merkle-patricia-tree: 2.3.2
      rustbn.js: 0.2.0
      safe-buffer: 5.2.1
    dev: false

  /ethers@6.8.0:
    resolution: {integrity: sha512-
zrFbmQRlraM+cU5mE4CZTLBurZTs2gdp2ld0nG/f3ecBK+x6lZ69KSxBqZ4NjclxwfTxl5LeNufcBbMsTdY53Q
==}
    engines: {node: '>=14.0.0'}
    dependencies:
      '@adraffy/ens-normalize': 1.10.0
      '@noble/curves': 1.2.0
      '@noble/hashes': 1.3.2
      '@types/node': 18.15.13
      aes-js: 4.0.0-beta.5
      tslib: 2.4.0
      ws: 8.5.0
    transitivePeerDependencies:
      - bufferutil
      - utf-8-validate
    dev: false

  /ethjs-unit@0.1.6:
    resolution: {integrity: sha512-
/Sn9Y0oKl0uqQuvgFk/zQgR7aw1g36qX/jzSQ5lSwlO0GigPymk4eGQfeNTD03w1dPOqfz8V77Cy43jH56pag
w==}
    engines: {node: '>=6.5.0', npm: '>=3'}
    dependencies:
      bn.js: 4.11.6
      number-to-bn: 1.7.0
    dev: false

  /ethjs-util@0.1.6:
    resolution: {integrity: sha512-
CUnVOQq7gSpDHZVVrQW8ExxUETWrnrvXYvYz55wOU8Uj4VCgw56XC2B/fVqQN+f7gmrnRHSLVnFAws
CuNwji8w==}
    engines: {node: '>=6.5.0', npm: '>=3'}
    dependencies:
      is-hex-prefixed: 1.0.0
      strip-hex-prefix: 1.0.0
    dev: false

  /event-target-shim@5.0.1:
    resolution: {integrity: sha512-i/2XbnSz/uxRCU6+NdVJgKWDTM427+MqYbkQzD321DuCQJUqOuJKIA0IM2+W2xtYHdKOmZ4dR6fExsd4SXL+WQ==}
    engines: {node: '>=6'}
    dev: false

  /eventemitter3@4.0.4:
    resolution: {integrity: sha512-rlaVLnVxtxvoyLsQQFBx53YmXHDxRIzzTLbdfxqi4yocpSjAxXwkU0cScM5JgSKMqEhrZpnvQ2D9gjylR0AimQ==}
    dev: false

  /eventemitter3@4.0.7:
    resolution: {integrity: sha512-8guHBZCwKnFhYdHr2ysuRWErTwhoN2X8XELRlrRwpmfeY2jjuUN4taQMsULKUVo1K4DvZl+0pgfyoysHxvmvEw==}
    dev: false

  /events@3.3.0:
    resolution: {integrity: sha512-mQw+2fkQBALzQ7V0MY0IqdnXNOeTtP4r0lN9z7AAawCXgqea7bDii20AYrIBrFd/Hx0M2Ocz6S111CaFkUcb0Q==}
    engines: {node: '>=0.8.x'}
    dev: false

  /evp_bytestokey@1.0.3:
    resolution: {integrity: sha512-/f2Go4TognH/KvCISP7OUsHn85hT9nUkxxA9BEWxFn+Oj9o8ZNLm/40hdlgSLyuOimsrTKLUMEorQexp/aPQeA==}
    dependencies:
      md5.js: 1.3.5
      safe-buffer: 5.2.1
    dev: false

  /execa@5.1.1:
    resolution: {integrity: sha512-8uSpZZocAZRBAPIEINJj3Lo9HyGitllczc27Eh5YYojjMFMn8yHMDMaUHE2Jqfq05D/wucwI4JGURyXt1vchyg==}
    engines: {node: '>=10'}
    dependencies:
      cross-spawn: 7.0.3
      get-stream: 6.0.1
      human-signals: 2.1.0
      is-stream: 2.0.1
      merge-stream: 2.0.0
      npm-run-path: 4.0.1
      onetime: 5.1.2
      signal-exit: 3.0.7
      strip-final-newline: 2.0.0
    dev: true

  /execa@7.2.0:
    resolution: {integrity: sha512-
UduyVP7TLB5IcAQl+OzLyLcS/l32W/GLg+AhHJ+ow40FOk2U3SAllPwR44v4vmdFwIWqpdwxxpQbF1n5ta9
seA==}
    engines: {node: ^14.18.0 || ^16.14.0 || >=18.0.0}
    dependencies:
      cross-spawn: 7.0.3
      get-stream: 6.0.1
      human-signals: 4.3.1
      is-stream: 3.0.0
      merge-stream: 2.0.0
      npm-run-path: 5.1.0
      onetime: 6.0.0
      signal-exit: 3.0.7
      strip-final-newline: 3.0.0
    dev: true

  /express@4.18.2:
    resolution: {integrity: sha512-
5/PsL6iGPdfQ/lKM1UuielYgv3BUoJfz1aUwU9vHZ+J7gyvwdQXFEBIEIaxeGf0GIcreATNyBExtalisDbuMqQ==
}
    engines: {node: '>= 0.10.0'}
    dependencies:
      accepts: 1.3.8
      array-flatten: 1.1.1
      body-parser: 1.20.1
      content-disposition: 0.5.4
      content-type: 1.0.5
      cookie: 0.5.0
      cookie-signature: 1.0.6
      debug: 2.6.9
      depd: 2.0.0
      encodeurl: 1.0.2
      escape-html: 1.0.3
      etag: 1.8.1
      finalhandler: 1.2.0
      fresh: 0.5.2
      http-errors: 2.0.0
      merge-descriptors: 1.0.1
      methods: 1.1.2
      on-finished: 2.4.1
      parseurl: 1.3.3
      path-to-regexp: 0.1.7
      proxy-addr: 2.0.7
      qs: 6.11.0
      range-parser: 1.2.1
      safe-buffer: 5.2.1
      send: 0.18.0
      serve-static: 1.15.0
      setprototypeof: 1.2.0
      statuses: 2.0.1
      type-is: 1.6.18
      utils-merge: 1.0.1
      vary: 1.1.2
    transitivePeerDependencies:
      - supports-color
    dev: false

  /ext@1.7.0:
    resolution: {integrity: sha512-
6hxeJYaL110a9b5TEJSj0gojyHQAmA2ch5Os+ySCiA1QGdS697XWY1pzsrSjqA9LDEEgdB/KypIlR59RcLu
HYw==}
    dependencies:
      type: 2.7.2
    dev: false

  /extend@3.0.2:
    resolution: {integrity: sha512-
fjquC59cD7CyW6urNXK0FBufkZcoiGG80wTuPujX590cB5Ttln20E2UB4S/WARVqhXffZl2LNgS+gQdPIIim/g
==}
    dev: false

  /extsprintf@1.3.0:
    resolution: {integrity: sha512-
11Ndz7Nv+mvAC1j0ktTa7fAb0vLyGGX+rMHNBYQviQDGU0Hw7lhctJANqbPhu9nV9/izT/IntTgZ7Im/9LJs9g
==}
    engines: {'0': node >=0.6.0}
    dev: false

  /eyes@0.1.8:
    resolution: {integrity: sha512-
GipyPsXO1anza0AOZdy69Im7hGFCNB7Y/NGjDlZGJ3GJJLtwNSb2vrzYrTYJRrRIoVx7pl+bhUaTB8yiccPvF
Q==}
    engines: {node: '> 0.1.90'}
    dev: false

  /fake-merkle-patricia-tree@1.0.1:
    resolution: {integrity: sha512-
Tgq37lkc9pUIgIKw5uitNUKcgcYL3R6JvXtKQbOf/ZSavXbidsksgp/pAY6p//uhw0I4yoMsvTSovvVIsk/qxA==}
    dependencies:
      checkpoint-store: 1.1.0
    dev: false

  /fast-content-type-parse@1.1.0:
    resolution: {integrity: sha512-
fBHHqSTFLVnR61C+gltJuE5GkVQMV0S2nqUO8TJ+5Z3qAKG8vAx4FKai1s5jq/inV1+sREynIWSuQ6HgoS
XpDQ==}
    dev: false

  /fast-decode-uri-component@1.0.1:
    resolution: {integrity: sha512-
WKgKWg5eUxvRZGwW8FvfbaH7AXSh2cL+3j5fMGzUMCxWBJ3dV3a7Wz8y2f/uQ0e3B6WmodD3oS54jT
```

```
Q9HVTIIg==}
    dev: false

  /fast-deep-equal@3.1.3:
    resolution: {integrity: sha512-
f3qQ9oQy9j2AhBe/H9VC91wLmKBCCU/gDOnKNAYG5hswO7BLKj09Hc5HYNz9cGI++xlpDCIgDaitVs03AT
R84Q==}

  /fast-diff@1.3.0:
    resolution: {integrity: sha512-
VxPP4NqbUjj6MaAOafWeUn2cXWLcCtljklUtZf0Ind4XQ+QPtmA0b18zZy0jIQx+ExRVCR/ZQpBmik5lXshNs
w==}
    dev: true

  /fast-glob@3.3.1:
    resolution: {integrity: sha512-
kNFPyjhh5cKjrUltxs+wFx+ZkbRaxxmZ+X0ZU31SOsxCEtP9VPgtq2teZw1DebupL5GmDaNQ6yKMMVcM41
iqDg==}
    engines: {node: '>=8.6.0'}
    dependencies:
      '@nodelib/fs.stat': 2.0.5
      '@nodelib/fs.walk': 1.2.8
      glob-parent: 5.1.2
      merge2: 1.4.1
      micromatch: 4.0.5

  /fast-json-stable-stringify@2.1.0:
    resolution: {integrity: sha512-
lhd/wF+Lk98HZoTCtlVraHtfh5XYijIjalXck7saUtuanSDyLMxnHhSXEDJqHxD7msR8D0uCmqlkwjCV8xvwHw
==}

  /fast-json-stringify@5.8.0:
    resolution: {integrity: sha512-
VVwK8CFMSALIvt14U8AvrSzQAwN/0vaVRiFFUVlpnXSnDGrSkOAO5MtzyN8oQNjLd5AqTW5OZRgyjoNu
AuR3jQ==}
    dependencies:
      '@fastify/deepmerge': 1.3.0
      ajv: 8.12.0
      ajv-formats: 2.1.1(ajv@8.12.0)
      fast-deep-equal: 3.1.3
      fast-uri: 2.2.0
      rfdc: 1.3.0
    dev: false

  /fast-levenshtein@2.0.6:
    resolution: {integrity: sha512-
DCXu6Ifhqcks7TZKY3Hxp3y6qphY5SJZmrWMDrKcERSOXWQdMhU9Ig/PYrzyw/ul9jOIyh0N4M0tbC5hodg
8dw==}
    dev: true

  /fast-querystring@1.1.2:
    resolution: {integrity: sha512-
```

```
g6KuKWmFXc0fID8WWH0jit4g0AGBoJhCkJMb1RmbsSEUNvQ+ZC8D6CUZ+GtF8nMzSPXnhiePyyqqipzN
NEnHjg==}
    dependencies:
      fast-decode-uri-component: 1.0.1
    dev: false

  /fast-redact@3.3.0:
    resolution: {integrity: sha512-
6T5V1QK1u4oF+ATxs1lWUmlEk6P2T9HqJG3e2DnHOdVgZy2rFJBoEnrIedcTXlkAHU/zKC+7KETJ+KGGK
wxgMQ==}
    engines: {node: '>=6'}
    dev: false

  /fast-safe-stringify@2.1.1:
    resolution: {integrity: sha512-
W+KJc2dmILlPplD/H4K9l9LcAHAfPtP6BY84uVLXQ6Evcz9Lcg33Y2z1IVblT6xdY54PXYVHEv+0Wpq8Io6zk
A==}
    dev: false

  /fast-stable-stringify@1.0.0:
    resolution: {integrity: sha512-
wpYMUmFu5f00Sm0cj2pfivpmawLZ0NKdviQ4w9zJeR8JVtOpOxHmLaJuj0vxvGqMJQWyP/COUkF75/57O
KyRag==}
    dev: false

  /fast-uri@2.2.0:
    resolution: {integrity: sha512-
cIusKBIt/R/oI6z/1nyfe2FvGKVTohVRfvkOhvx0nCEW+xf5NoCXjAHcWp93uOUBchzYcsvPlrapAdX1uW+YG
g==}
    dev: false

  /fastify-plugin@4.5.1:
    resolution: {integrity: sha512-
stRHYGeuqpEZTL1Ef0Ovr2ltazUT9g844X5z/zEBFLG8RYlpDiOCIG+ATvYEp+/zmc7sN29mcIMp8gvYplYPI
Q==}
    dev: false

  /fastify@4.24.3:
    resolution: {integrity: sha512-
6HHJ+R2x2LS3y1PqxnwEIjOTZxFl+8h4kSC/TuDPXtA+v2JnV9yEtOsNSKK1RMD7sIR2y1ZsA4BEFaid/cK5
pg==}
    dependencies:
      '@fastify/ajv-compiler': 3.5.0
      '@fastify/error': 3.4.0
      '@fastify/fast-json-stringify-compiler': 4.3.0
      abstract-logging: 2.0.1
      avvio: 8.2.1
      fast-content-type-parse: 1.1.0
      fast-json-stringify: 5.8.0
      find-my-way: 7.7.0
      light-my-request: 5.11.0
      pino: 8.16.1
      process-warning: 2.2.0
      proxy-addr: 2.0.7
      rfdc: 1.3.0
      secure-json-parse: 2.7.0
      semver: 7.5.4
      toad-cache: 3.3.0
    transitivePeerDependencies:
      - supports-color
    dev: false

  /fastq@1.15.0:
    resolution: {integrity: sha512-
wBrocU2LCXXa+lWBt8RoIRD89Fi8OdABODa/kEnyeyjS5aZO5/GNvI5sEINADqP/h8M29UHTHUb53sUu5Ih
qdw==}
    dependencies:
      reusify: 1.0.4

  /fbemitter@3.0.0:
    resolution: {integrity: sha512-
KWKaceCwKQU0+HPoop6gn4eOHk50bBv/VxjJtGMfwmJt3D29JpN4H4eisCtIPA+a8GVBam+ldMMpMjJUv
pDyHw==}
    dependencies:
      fbjs: 3.0.5
    transitivePeerDependencies:
      - encoding
    dev: false

  /fbjs-css-vars@1.0.2:
    resolution: {integrity: sha512-
b2XGFAFdWZWg0phtAWLHCk836A1Xann+I+Dgd3Gk64MHKZO44FfoD1KxyvbSh0qZsIoXQGGlVztIY+oitJ
PpRQ==}
    dev: false

  /fbjs@3.0.5:
    resolution: {integrity: sha512-
ztsSx77JBtkuMrEypfhgc3cI0+0h+svqeie7xHbh1k/IKdcydnvadp/mUaGgjAOXQmQSxsqgaRhS3q9fy+1kxg==
}
    dependencies:
      cross-fetch: 3.1.5
      fbjs-css-vars: 1.0.2
      loose-envify: 1.4.0
      object-assign: 4.1.1
      promise: 7.3.1
      setimmediate: 1.0.5
      ua-parser-js: 1.0.37
    transitivePeerDependencies:
      - encoding
    dev: false

  /file-entry-cache@6.0.1:
    resolution: {integrity: sha512-
7Gps/XWymbLk2QLYK4NzpMOrYjMhdIxXuIvy2QBsLE6ljuodKvdkWs/cpyJJ3CVIVpH0Oi1Hvg1ovbMzLdFB
Bg==}
    engines: {node: ^10.12.0 || >=12.0.0}
    dependencies:
      flat-cache: 3.1.1
    dev: true

  /file-uri-to-path@1.0.0:
    resolution: {integrity: sha512-
0Zt+s3L7Vf1biwWZ29aARiVYLx7iMGnEUl9x33fbB/j3jR81u/O2LbqK+Bm1CDSNDKVtJ/YjwY7TUd5SkeLQL
w==}
    dev: false

  /fill-range@7.0.1:
    resolution: {integrity: sha512-
qOo9F+dMUmC2Lcb4BbVvnKJxTPjCm+RRpe4gDuGrzkL7mEVl/djYSu2OdQ2Pa302N4oqkSg9ir6jaLWJ2U
SVpQ==}
    engines: {node: '>=8'}
    dependencies:
      to-regex-range: 5.0.1

  /filter-obj@1.1.0:
    resolution: {integrity: sha512-
8rXg1ZnX7xzy2NGDVkBVaAy+lSlPNwad13BtgSlLuxfIslyt5Vg64U7tFcCt4WS1R0hvtnQybT/IyCkGZ3DpXQ
==}
    engines: {node: '>=0.10.0'}
    dev: false

  /finalhandler@1.2.0:
    resolution: {integrity: sha512-
5uXcUVftlQMFnWC9qu/svkWv3GTd2PfUhK/3PLkYNAe7FbqJMt3515HaxE6eRL74GdsriiwujiawdaB1BpEIS
g==}
    engines: {node: '>= 0.8'}
    dependencies:
      debug: 2.6.9
      encodeurl: 1.0.2
      escape-html: 1.0.3
      on-finished: 2.4.1
      parseurl: 1.3.3
      statuses: 2.0.1
      unpipe: 1.0.0
    transitivePeerDependencies:
      - supports-color
    dev: false

  /find-my-way@7.7.0:
    resolution: {integrity: sha512-
+SrHpvQ52Q6W9f3wJoJBbAQULJuNEEQwBvlvYwACDhBTLOTMiQ0HYWh4+vC3OivGP2ENcTI1oKlFA2
OepJNjhQ==}
    engines: {node: '>=14'}
    dependencies:
      fast-deep-equal: 3.1.3
      fast-querystring: 1.1.2
      safe-regex2: 2.0.0
    dev: false

  /find-replace@3.0.0:
    resolution: {integrity: sha512-6Tb2myMioCAgv5kfvP5/PkZZ/ntTpVK39fHY7WkWBgvbeE+VHd/tZuZ4mrC+bxh4cfOZeYKVPaJIZtZXV7GNCQ==}
    engines: {node: '>=4.0.0'}
    dependencies:
      array-back: 3.1.0

  /find-up@4.1.0:
    resolution: {integrity: sha512-PpOwAdQ/YlXQ2vj8a3h8IipDuYRi3wceVQQGYWxNINccq40Anw7BlsEXCMbt1Zt+OLA6Fq9suIpIWD0OsnISlw==}
    engines: {node: '>=8'}
    dependencies:
      locate-path: 5.0.0
      path-exists: 4.0.0

  /find-up@5.0.0:
    resolution: {integrity: sha512-78/PXT1wlLLDgTzDs7sjq9hzz0vXD+zn+7wypEe4fXQxCmdmqfGsEPQxmiCSQI3ajFV91bVSsvNtrJRiW6nGng==}
    engines: {node: '>=10'}
    dependencies:
      locate-path: 6.0.0
      path-exists: 4.0.0
    dev: true

  /flat-cache@3.1.1:
    resolution: {integrity: sha512-/qM2b3LUIaIgviBQovTLvijfyOQXPtSRnRK26ksj2J7rzPIecePUIpJsZ4T02Qg+xiAEKIs5K8dsHEd+VaKa/Q==}
    engines: {node: '>=12.0.0'}
    dependencies:
      flatted: 3.2.9
      keyv: 4.5.4
      rimraf: 3.0.2
    dev: true

  /flatted@3.2.9:
    resolution: {integrity: sha512-36yxDn5H7OFZQla0/jFJmbIKTdZAQHngCedGxiMmpNfEZM0sdEeT+WczLQrjK6D7o2aiyLYDnkw0R3JK0Qv1RQ==}
    dev: true

  /flux@4.0.4(react@18.2.0):
    resolution: {integrity: sha512-NCj3XlayA2UsapRpM7va6wU1+9rE5FIL7qoMcmxWHRzbp0yujihMBm9BBHZ1MDIk5h5o2Bl6eGiCe8rYELAmYw==}
    peerDependencies:
      react: ^15.0.2 || ^16.0.0 || ^17.0.0
    dependencies:
      fbemitter: 3.0.0
      fbjs: 3.0.5
      react: 18.2.0
    transitivePeerDependencies:
      - encoding
    dev: false

  /follow-redirects@1.15.3(debug@4.3.4):
    resolution: {integrity: sha512-
1VzOtuEM8pC9SFU1E+8KfTjZyMztRsgEfwQl44z8A25uy13jSzTj6dyK2Df52iV0vgHCfBwLhDWevLn95w5v6
Q==}
    engines: {node: '>=4.0'}
    peerDependencies:
      debug: '*'
    peerDependenciesMeta:
      debug:
        optional: true
    dependencies:
      debug: 4.3.4
    dev: true

  /for-each@0.3.3:
    resolution: {integrity: sha512-
jqYfLp7mo9vIyQf8ykW2v7A+2N4QjeCeI5+Dz9XraiO1ign81wjiH7Fb9vSOWvQfNtmSa4H2RoQTrrXivdUZm
w==}
    dependencies:
      is-callable: 1.2.7

  /forever-agent@0.6.1:
    resolution: {integrity: sha512-
j0KLYPhm6zeac4lz3oJ3o65qvgQCcPubiyotZrXqEaG4hNagNYO8qdlUrX5vwqv9ohqeT/Z3j6+yW067yWWd
Uw==}
    dev: false

  /form-data-encoder@1.7.1:
    resolution: {integrity: sha512-
EFRDrsMm/kyqbTQocNvRXMLjc7Es2Vk+IQFx/YW7hkUH1eBl4J1fqiP34l74Yt0pFLCNpc06fkbVk00008mzjg
==}
    dev: false

  /form-data@2.3.3:
    resolution: {integrity: sha512-
1lLKB2Mu3aGP1Q/2eCOx0fNbRMe7XdwktwOruhfqqd0rIJWwN4Dh+E3hrPSlDCXnSR7UtZ1N38rVXm+6+
MEhJQ==}
    engines: {node: '>= 0.12'}
    dependencies:
      asynckit: 0.4.0
      combined-stream: 1.0.8
      mime-types: 2.1.35
    dev: false
  /form-data@4.0.0:
    resolution: {integrity: sha512-ETEklSGi5t0QMZuiXoA/Q6vcnxcLQP5vdugSpuAyi6SVGi2clPPp+xgEhuMaHC+zGgn31Kd235W35f7Hykkaww==}
    engines: {node: '>= 6'}
    dependencies:
      asynckit: 0.4.0
      combined-stream: 1.0.8
      mime-types: 2.1.35
    dev: true

  /forwarded@0.2.0:
    resolution: {integrity: sha512-buRG0fpBtRHSTCOASe6hD258tEubFoRLb4ZNA6NxMVHNw2gOcwHo9wyablzMzOA5z9xA9L1KNjk/Nt6MT9aYow==}
    engines: {node: '>= 0.6'}
    dev: false

  /fraction.js@4.3.7:
    resolution: {integrity: sha512-ZsDfxO51wGAXREY55a7la9LScWpwv9RxIrYABrlvOFBlH/ShPntsXeuUIfXKKOVicNxQ+o8JTbJvjS4M89yew==}
    dev: false

  /fresh@0.5.2:
    resolution: {integrity: sha512-zJ2mQYM18rEFOudeV4GShTGIQ7RbzA7ozbU9I/XBpm7kqgMywgmylMwXHxZJmkVoYkna9d2pVXVXPdYTP9ej8Q==}
    engines: {node: '>= 0.6'}
    dev: false

  /fs-extra@4.0.3:
    resolution: {integrity: sha512-q6rbdDd1o2mAnQreO7YADIxf/Whx4AHBiRf6d+/cVT8h44ss+lHgxf1FemcqDnQt9X3ct4McHr+JMGlYSsK7Cg==}
    dependencies:
      graceful-fs: 4.2.11
      jsonfile: 4.0.0
      universalify: 0.1.2
    dev: false

  /fs-extra@7.0.1:
    resolution: {integrity: sha512-YJDaCJZEnBmcbw13fvdAM9AwNOJwOzrE4pqMqBq5nFiEqXUqHwlK4B+3pUw6JNvfSPtX05xFHtYy/1ni01eGCw==}
    engines: {node: '>=6 <7 || >=8'}
    dependencies:
      graceful-fs: 4.2.11
      jsonfile: 4.0.0
      universalify: 0.1.2
  /fs-minipass@1.2.7:
    resolution: {integrity: sha512-
GWSSJGFy4e9GUeCcbIkED+bgAoFyj7XF1mV8rma3QW4NIqX9Kyx79N/PF61H5udOV3aY1IaMLs6pGbH7
1nlCTA==}
    dependencies:
      minipass: 2.9.0
    dev: false

  /fs.realpath@1.0.0:
    resolution: {integrity: sha512-
OO0pH2lK6a0hZnAdau5ItzHPI6pUlvI7jMVnxUQRtw4owF2wk8lOSabtGDCTP4Ggrg2MbGnWO9X8K1t4+fG
MDw==}

  /fsevents@2.3.2:
    resolution: {integrity: sha512-
xiqMQR4xAeHTuB9uWm+fFRcIOgKBMiOBP+eXiyT7jsgVCq1bkVygt00oASowB7EdtpOHaaPgKt812P9ab+
DDKA==}
    engines: {node: ^8.16.0 || ^10.6.0 || >=11.0.0}
    os: [darwin]
    requiresBuild: true
    optional: true

  /function-bind@1.1.1:
    resolution: {integrity: sha512-
yIovAzMX49sF8Yl58fSCWJ5svSLuaibPxXQJFLmBObTuCr0Mf1KiPopGM9NiFjiYBCbfaa2Fh6breQ6ANVTI
0A==}

  /function-bind@1.1.2:
    resolution: {integrity: sha512-
7XHNxH7qX9xG5mIwxkhumTox/MIRNcOgDrxWsMt2pAr23WHp6MrRlN7FBSFpCpr+oVO0F744iUgR82nJ
MfG2SA==}

  /function.prototype.name@1.1.6:
    resolution: {integrity: sha512-
Z5kx79swU5P27WEayXM1tBi5Ze/lbIyiNgU3qyXUOf9b2rgXYyF9Dy9Cx+IQv/Lc8WCG6L82zwUPpSS9hGe
hIg==}
    engines: {node: '>= 0.4'}
    dependencies:
      call-bind: 1.0.5
      define-properties: 1.2.1
      es-abstract: 1.22.3
      functions-have-names: 1.2.3
    dev: true

  /functional-red-black-tree@1.0.1:
    resolution: {integrity: sha512-
dsKNQNdj6xA3T+QlADDA7mOSlX0qiMINjn0cgr+eGHGsbSHzTabcIogz2+p/iqP1Xs6EP/sS2SbqH+brGTbq
0g==}
    dev: false

  /functions-have-names@1.2.3:
    resolution: {integrity: sha512-
xckBUXyTIqT97tq2x2AMb+g163b5JFysYk0x4qxNFwbfQkmNZoiRHb6sPzI9/QV33WeuvVYBUIiD4NzNIyqa
RQ==}
    dev: true

  /gensync@1.0.0-beta.2:
    resolution: {integrity: sha512-
3hN7NaskYvMDLQY55gnW3NQ+mesEAepTqlg+VEbj7zzqEMBVNhzcGYYeqFo/TlYz6eQiFcp1HcsCZO+n
GgS8zg==}
    engines: {node: '>=6.9.0'}

  /get-caller-file@2.0.5:
    resolution: {integrity: sha512-
DyFP3BM/3YHTQOCUL/w0OZHR0lpKeGrxotcHWcqNEdnltqFwXVfhEBQ94eIo34AfQpo0rGki4cyIiftY06h2F
g==}
    engines: {node: 6.* || 8.* || >= 10.*}

  /get-intrinsic@1.1.3:
    resolution: {integrity: sha512-
QJVz1Tj7MS099PevUG5jvnt9tSkXN8K14dxQlikJuPt4uD9hHAHjLyLBiLR5zELelBdD9QNRAXZzsJx0WaDL
9A==}
    dependencies:
      function-bind: 1.1.1
      has: 1.0.3
      has-symbols: 1.0.3

  /get-intrinsic@1.2.2:
    resolution: {integrity: sha512-
0gSo4ml/0j98Y3lngkFEot/zhiCeWsbYIlZ+uZOVgzLyLaUw7wxUL+nCTP0XJvJg1AXuIJRI3UJi8GsbDuxdGA
==}
    dependencies:
      function-bind: 1.1.2
      has-proto: 1.0.1
      has-symbols: 1.0.3
      hasown: 2.0.0
    dev: true

  /get-stream@5.2.0:
    resolution: {integrity: sha512-
nBF+F1rAZVCu/p7rjzgA+Yb4lfYXrpl7a6VmJrU8wF9I1CKvP/QwPNZHnOlwbTkY6dvtFIzFMSyQXbLoTQPR
pA==}
    engines: {node: '>=8'}
    dependencies:
      pump: 3.0.0
    dev: false

  /get-stream@6.0.1:
    resolution: {integrity: sha512-
ts6Wi+2j3jQjqi70w5AlN8DFnkSwC+MqmxEzdEALB2qXZYV3X/b1CTfgPLGJNMeAWxdPfU8FO1ms3NUfaH
CPYg==}
    engines: {node: '>=10'}

  /get-symbol-description@1.0.0:
    resolution: {integrity: sha512-
2EmdH1YvIQiZpltCNgkuiUnyukzxM/R6NDJX31Ke3BG1Nq5b0S2PhX59UKi9vZpPDQVdqn+1IcaAwnzTT5v
Cjw==}
    engines: {node: '>= 0.4'}
    dependencies:
      call-bind: 1.0.5
      get-intrinsic: 1.2.2
    dev: true

  /get-tsconfig@4.7.2:
    resolution: {integrity: sha512-
wuMsz4leaj5hbGgg4IvDU0bqJagpftG5l5cXIAvo8uZrqn0NJqwtfupTN00VnkQJPcIRrxYrm1Ue24btpCha2A==
}
    dependencies:
      resolve-pkg-maps: 1.0.0
    dev: true

  /getpass@0.1.7:
    resolution: {integrity: sha512-
0fzj9JxOLfJ+XGLhR8ze3unN0KZCgZwiSSDz168VERjK8Wl8kVSdcu2kspd4s4wtAa1y/qrVRiAA0WclVsu0ng
==}
    dependencies:
      assert-plus: 1.0.0
    dev: false

  /glob-parent@5.1.2:
    resolution: {integrity: sha512-
AOIgSQCepiJYwP3ARnGx+5VnTu2HBYdzbGP45eLw1vr3zB3vZLeyed1sC9hnbcOc9/SrMyM5RPQrkGz4a
S9Zow==}
    engines: {node: '>= 6'}
    dependencies:
      is-glob: 4.0.3

  /glob-parent@6.0.2:
    resolution: {integrity: sha512-
XxwI8EOhVQgWp6iDL+3b0r86f4d6AX6zSU55HfB4ydCEuXLXc5FcYeOu+nnGftS4TEju/11rt4KJPTMgbfmv
4A==}
    engines: {node: '>=10.13.0'}
    dependencies:
      is-glob: 4.0.3

  /glob@7.1.6:
    resolution: {integrity: sha512-
LwaxwyZ72Lk7vZINtNNrywX0ZuLyStrdDtabefZKAY5ZGJhVtgdznluResxNmPitE0SAO+O26sWTHeKSI2wM
BA==}
    dependencies:
      fs.realpath: 1.0.0
      inflight: 1.0.6
      inherits: 2.0.4
      minimatch: 3.1.2
      once: 1.4.0
      path-is-absolute: 1.0.1
    dev: false

  /glob@7.1.7:
    resolution: {integrity: sha512-
OvD9ENzPLbegENnYP5UUfJIirTg4+XwMWGaQfQTY0JenxNvvIKP3U3/tAQSPIu/lHxXYSZmpXlUHeqAIdK
zBLQ==}
    dependencies:
      fs.realpath: 1.0.0
      inflight: 1.0.6
      inherits: 2.0.4
      minimatch: 3.1.2
      once: 1.4.0
      path-is-absolute: 1.0.1

  /glob@8.1.0:
    resolution: {integrity: sha512-
r8hpEjiQEYlF2QU0df3dS+nxxSIreXQS1qRhMJM0Q5NDdR386C7jb7Hwwod8Fgiuex+k0GFjgft18yvxm5XoC
Q==}
    engines: {node: '>=12'}
    dependencies:
      fs.realpath: 1.0.0
      inflight: 1.0.6
      inherits: 2.0.4
      minimatch: 5.1.6
      once: 1.4.0
    dev: false

  /global@4.4.0:
    resolution: {integrity: sha512-
wv/LAoHdRE3BeTGz53FAamhGlPLhlssK45usmGFThIi4XqnBmjKQ16u+RNbP7WvigRZDxUsM0J3gcQ5yic
aL0w==}
    dependencies:
      min-document: 2.19.0
      process: 0.11.10
    dev: false

  /globals@11.12.0:
    resolution: {integrity: sha512-
WOBp/EEGUiIsJSp7wcv/y6MO+lV9UoncWqxuFfm8eBwzWNgyfBd6Gz+IeKQ9jCmyhoH99g15M3T+QaVHF
jizVA==}
    engines: {node: '>=4'}

  /globals@13.23.0:
    resolution: {integrity: sha512-
XAmF0RjlrjY23MA51q3HltdlGxUpXPvg0GioKiD9X6HD28iMjo2dKC8Vqwm7lne4GNr78+RHTfliktR6ZH09wA
==}
    engines: {node: '>=8'}
    dependencies:
      type-fest: 0.20.2
    dev: true

  /globalthis@1.0.3:
    resolution: {integrity: sha512-sFdI5LyBiNTHjRd7cGPWapiHWMOXKyuBNX/cWJ3NfzrZQVa8GI/8cofCl74AOVqq9W5kNmguTIzJ/1s2gyI9wA==}
    engines: {node: '>= 0.4'}
    dependencies:
      define-properties: 1.2.1
    dev: true

  /globby@11.1.0:
    resolution: {integrity: sha512-jhIXaOzy1sb8IyocaruWSn1TjmnBVs8Ayhcy83rmxNJ8q2uWKCAj3CnJY+KpGSXCueAPc0i05kVvVKtP1t9S3g==}
    engines: {node: '>=10'}
    dependencies:
      array-union: 2.1.0
      dir-glob: 3.0.1
      fast-glob: 3.3.1
      ignore: 5.2.4
      merge2: 1.4.1
      slash: 3.0.0
    dev: true

  /gopd@1.0.1:
    resolution: {integrity: sha512-d65bNlIadxvpb/A2abVdlqKqV563juRnZ1Wtk6s1sIR8uNsXR70xqIzVqxVf1eTqDunwT2MkczEeaezCKTZhwA==}
    dependencies:
      get-intrinsic: 1.1.3

  /got@11.8.6:
    resolution: {integrity: sha512-6tfZ91bOr7bOXnK7PRDCGBLa1H4U080YHNaAQ2KsMGlLEzRbk44nsZF2E1IeRc3vtJHPVbKCYgdFbaGO2ljd8g==}
    engines: {node: '>=10.19.0'}
    dependencies:
      '@sindresorhus/is': 4.6.0
      '@szmarczak/http-timer': 4.0.6
      '@types/cacheable-request': 6.0.3
      '@types/responselike': 1.0.2
      cacheable-lookup: 5.0.4
      cacheable-request: 7.0.4
      decompress-response: 6.0.0
      http2-wrapper: 1.0.3
      lowercase-keys: 2.0.0
      p-cancelable: 2.1.1
      responselike: 2.0.1
    dev: false

  /got@12.1.0:
    resolution: {integrity: sha512-hBv2ty9QN2RdbJJMK3hesmSkFTjVIHyIDDbssCKnSmq62edGgImJWD10Eb1k77TiV1bxloxqcFAVK8+9pkhOig==}
    engines: {node: '>=14.16'}
    dependencies:
      '@sindresorhus/is': 4.6.0
      '@szmarczak/http-timer': 5.0.1
      '@types/cacheable-request': 6.0.3
      '@types/responselike': 1.0.2
      cacheable-lookup: 6.1.0
      cacheable-request: 7.0.4
      decompress-response: 6.0.0
      form-data-encoder: 1.7.1
      get-stream: 6.0.1
      http2-wrapper: 2.2.0
      lowercase-keys: 3.0.0
      p-cancelable: 3.0.0
      responselike: 2.0.1
    dev: false

  /graceful-fs@4.2.11:
    resolution: {integrity: sha512-
  RbJ5/jmFcNNCcDV5o9eTnBLJ/HszWV0P73bc+Ff4nS/rJj+YaS6IGyiOL0VoBYX+l1Wrl3k63h/KrH+nhJ0XvQ
  ==}

  /graphemer@1.4.0:
    resolution: {integrity: sha512-
  EtKwoO6kxCL9WO5xipiHTZlSzBm7WLT627TqC/uVRd0HKmq8NXyebnNYxDoBi7wt8eTWrUrKXCOVaFq9
  x1kgag==}
    dev: true

  /har-schema@2.0.0:
    resolution: {integrity: sha512-
  Oqluz6zhGX8cyRaTQlFMPw80bSJVG2x/cFb8ZPhUILGgHka9SsokCCOQgpveePerqidZOrT14ipqfJb7ILcW
  5Q==}
    engines: {node: '>=4'}
    dev: false

  /har-validator@5.1.5:
    resolution: {integrity: sha512-
  nmT2T0lljbxdQZfspsno9hgrG3Uir6Ks5afism62poxqBM6sDnMEuPmzTq8XN0OEwqKLLdh1jQI3qyE66Nzb3
  w==}
    engines: {node: '>=6'}
    deprecated: this library is no longer supported
    dependencies:
      ajv: 6.12.6
      har-schema: 2.0.0
    dev: false

  /has-bigints@1.0.2:
    resolution: {integrity: sha512-
  tSvCKtBr9lkF0Ex0aQiP9N+OpV4zi2r/Nee5VkRDbaqv35RLYMzbwQfFSZZH0kR+Rd6302UJZ2p/bJCEoR3V
  oQ==}
    dev: true
  /has-flag@3.0.0:
    resolution: {integrity: sha512-
sKJf1+ceQBr4SMkvQnBDNDtf4TXpVhVGateu0t918bl30FnbE2m4vNLX+VWe/dpjlb+HugGYzW7uQXH98H
PEYw==}
    engines: {node: '>=4'}

  /has-flag@4.0.0:
    resolution: {integrity: sha512-
EykJT/Q1KjTWctppgIAgfSO0tKVuZUjhgMr17kqTumMl6Afv3EISleU7qZUzoXDFTAHTDC4NOoG/ZxU3EvlM
PQ==}
    engines: {node: '>=8'}

  /has-property-descriptors@1.0.1:
    resolution: {integrity: sha512-
VsX8eaIewvas0xnvinAe9bw4WfIeODpGYikiWYLH+dma0Jw6KHYqWiWfhQlgOVK8D6PvjubK5Uc4P0iIhIcN
Vg==}
    dependencies:
      get-intrinsic: 1.2.2
    dev: true

  /has-proto@1.0.1:
    resolution: {integrity: sha512-
7qE+iP+O+bgF9clE5+UoBFzE65mlBiVj3tKCrlNQ0Ogwm0BjpT/gK4SlLYDMybDh5I3TCTKnPPa0oMG7JDY
rhg==}
    engines: {node: '>= 0.4'}
    dev: true

  /has-symbols@1.0.3:
    resolution: {integrity: sha512-
l3LCuF6MgDNwTDKkdYGEihYjt5pRPbEg46rtlmnSPlUbgmB8LOIrKJbYYFBSbnPaJexMKtiPO8hmeRjRz2T
d+A==}
    engines: {node: '>= 0.4'}

  /has-tostringtag@1.0.0:
    resolution: {integrity: sha512-
kFjcSNhnlGV1kyoGk7OXKSawH5JOb/LzUc5w9B02hOTO0dfFRjbHQKvg1d6cf3HbeUmtU9VbbV3qzZ2Teh
97WQ==}
    engines: {node: '>= 0.4'}
    dependencies:
      has-symbols: 1.0.3

  /has@1.0.3:
    resolution: {integrity: sha512-
f2dvO0VU6Oej7RkWJGrehjbzMAjFp5/VKPp5tTpWIV4JHHZK1/BxbFRtf/siA2SWTe09caDmVtYYzWEIbBS4
zw==}
    engines: {node: '>= 0.4.0'}
    dependencies:
      function-bind: 1.1.1

  /hash-base@3.1.0:
    resolution: {integrity: sha512-
1nmYp/rhMDiE7AYkDw+lLwlAzz0AntGIe51F3RfFfEqyQ3feY2eI/NcwC6umIQVOASPMsWJLJScWKSSvzL9
IVA==}
    engines: {node: '>=4'}
    dependencies:
      inherits: 2.0.4
      readable-stream: 3.6.0
      safe-buffer: 5.2.1
    dev: false

  /hash.js@1.1.7:
    resolution: {integrity: sha512-
taOaskGt4z4SOANNseOviYDvjEJinIkRgmp7LbKP2YTTmVxWBl87s/uzK9r+44BclBSp2X7K1hqeNfz9JbBeX
A==}
    dependencies:
      inherits: 2.0.4
      minimalistic-assert: 1.0.1
    dev: false

  /hasown@2.0.0:
    resolution: {integrity: sha512-
vUptKVTpIJhcczKBbgnS+RtcuYMB8+oNzPK2/Hp3hanz8JmpATdmmgLgSaadVREkDm+e2giHwY3ZRkyjSI
DDFA==}
    engines: {node: '>= 0.4'}
    dependencies:
      function-bind: 1.1.2

  /hey-listen@1.0.8:
    resolution: {integrity: sha512-
COpmrF2NOg4TBWUJ5UVyaCU2A88wEMkUPK4hNqyCkqHbxT92BbvfjoSozkAIIm6XhicGlJHhFdullInrdhw
U8Q==}
    dev: false

  /hmac-drbg@1.0.1:
    resolution: {integrity: sha512-
Tti3gMqLdZfhOQY1Mzf/AanLiqh1WTiJgEj26ZuYQ9fbkLomzGchCws4FyrSd4VkpBfiNhaE1On+lOz894jvXg==
}
    dependencies:
      hash.js: 1.1.7
      minimalistic-assert: 1.0.1
      minimalistic-crypto-utils: 1.0.1
    dev: false

  /hosted-git-info@2.8.9:
    resolution: {integrity: sha512-
mxIDAb9Lsm6DoOJ7xH+5+X4y1LU/4Hi50L9C5sIswK3JzULS4bwk1FvjdBgvYR4bzT4tuUQiC15FE2f5HbLv
Yw==}
    dev: true

  /http-cache-semantics@4.1.1:
    resolution: {integrity: sha512-
er295DKPVsV82j5kw1Gjt+ADA/XYHsajl82cGNQG2eyoPkvgUhX+nDIyelzhIWbbsXP39EHcI6l5tYs2FYqYX
Q==}
    dev: false
```

```
 /http-errors@2.0.0:
   resolution: {integrity: sha512-
FtwrG/euBzaEjYeRqOgly7G0qviiXoJWnvEH2Z1plBdXgbyjv34pHTSb9zoeHMyDy33+DWy5Wt9Wo+TURtO
YSQ==}
   engines: {node: '>= 0.8'}
   dependencies:
     depd: 2.0.0
     inherits: 2.0.4
     setprototypeof: 1.2.0
     statuses: 2.0.1
     toidentifier: 1.0.1
   dev: false

 /http-https@1.0.0:
   resolution: {integrity: sha512-
o0PWwVCSp3O0wS6FvNr6xfBCHgt0m1tvPLFOCc2iFDKTRAXhB7m8klDf7ErowFH8POa6dVdGatKU5I1Y
YwzUyg==}
   dev: false

 /http-signature@1.2.0:
   resolution: {integrity: sha512-
CAbnr6Rz4CYQkLYUtSNXxQPUH2gK8f3iWexVlsnMeD+GjlsQ0Xsy1cOX+mN3dtxYomRy21CiOzU8Uhw6
OwncEQ==}
   engines: {node: '>=0.8', npm: '>=1.3.7'}
   dependencies:
     assert-plus: 1.0.0
     jsprim: 1.4.2
     sshpk: 1.18.0
   dev: false

 /http2-wrapper@1.0.3:
   resolution: {integrity: sha512-
V+23sDMr12Wnz7iTcDeJr3O6AIxlnvT/bmaAAAP/Xda35C90p9599p0F1eHR/N1KILWSoWVAiOMFjBBXaX
SMxg==}
   engines: {node: '>=10.19.0'}
   dependencies:
     quick-lru: 5.1.1
     resolve-alpn: 1.2.1
   dev: false

 /http2-wrapper@2.2.0:
   resolution: {integrity: sha512-
kZB0wxMo0sh1PehyjJUWRFEd99KC5TLjZ2cULC4f9iqJBAmKQQXEICjxl5iPJRwP40dpeHFqqhm7tYCvOD
pqpQ==}
   engines: {node: '>=10.19.0'}
   dependencies:
     quick-lru: 5.1.1
     resolve-alpn: 1.2.1
   dev: false

 /human-signals@2.1.0:
   resolution: {integrity: sha512-
B4FFZ6q/T2jhhksgkbEW3HBvWIfDW85snkQgawt07S7J5QXTk6BkNV+0yAeZrM5QpMAdYlocGoljn0sJ/WQ
kFw==}
   engines: {node: '>=10.17.0'}
   dev: true

 /human-signals@4.3.1:
   resolution: {integrity: sha512-
nZXjEF2nbo7lIw3mgYjItAfgQXog3OjJogSbKa2CQIIvSGWcKgeJnQlNXip6NglNzYH45nSRiEVimMvYL8DDq
Q==}
   engines: {node: '>=14.18.0'}
   dev: true

 /humanize-ms@1.2.1:
   resolution: {integrity: sha512-
Fl70vYtsAFb/C06PTS9dZBo7ihau+Tu/DNCk/OyHhea07S+aeMWpFFkUaXRa8fI+ScZbEI8dfSxwY7gxZ9SA
VQ==}
   dependencies:
     ms: 2.1.3
   dev: false

 /iconv-lite@0.4.24:
   resolution: {integrity: sha512-
v3MXnZAcvnywkTUEZomIActle7RXXeedOR31wwl7VlyoXO4Qi9arvSenNQWne1TcRwhCL1HwLI21bEqdpj
8/rA==}
   engines: {node: '>=0.10.0'}
   dependencies:
     safer-buffer: 2.1.2
   dev: false

 /idna-uts46-hx@2.3.1:
   resolution: {integrity: sha512-
PWoF9Keq6laYdIRwwCdhTPl60xRqAloYNMQLiyUnG42VjT53oW07BXIRM+NK7eQjzXjAk2gUvX9caRxlnF
9TAA==}
   engines: {node: '>=4.0.0'}
   dependencies:
     punycode: 2.1.0
   dev: false

 /ieee754@1.2.1:
   resolution: {integrity: sha512-
dcyqhDvX1C46lXZcVqCpK+FtMRQVdIMN6/Df5js2zouUsqG7I6sFxitIC+7KYK29KdXOLHdu9zL4sFnoVQnq
aA==}
   dev: false

 /ignore-by-default@1.0.1:
   resolution: {integrity: sha512-
Ius2VYcGNk7T90CppJqcIkS5ooHUZyIQK+ClZfMfMNFEF9VSE73Fq+906u/CWu92x4gzZMWOwfFYckPOb
zdEbA==}
   dev: true

 /ignore@5.2.4:
   resolution: {integrity: sha512-
MAb38BcSbH0eHNBxn7ql2NH/kX33OkB3lZ1BNdh7ENeRChHTYsTvWrMubiIAMNS2llXEEgZ1MUOBtXCh
P3kaFQ==}
   engines: {node: '>= 4'}
   dev: true

 /immediate@3.3.0:
   resolution: {integrity: sha512-
HR7EVodfFUdQCTIeySw+WDRFJlPcLOJbXfwwZ7Oom6tjsvZ3bOkCDJHehQC3nxJrv7+f9XecwazynjU8e4
Vw3Q==}
   dev: false

 /import-fresh@3.3.0:
   resolution: {integrity: sha512-
veYYhQa+D1QBKznvhUHxb8faxlrwUnxseDAbAp457E0wLNio2bOSKnjYDhMj+YiAq61xrMGhQk9iXVk5Fzg
QMw==}
   engines: {node: '>=6'}
   dependencies:
     parent-module: 1.0.1
     resolve-from: 4.0.0
   dev: true

 /imurmurhash@0.1.4:
   resolution: {integrity: sha512-
JmXMZ6wuvDmLiHEml9ykzqO6lwFbof0GG4IkcGaENdCRDDmMVnny7s5HsIgHCbaq0w2MyPhDqkhTUgS
2LU2PHA==}
   engines: {node: '>=0.8.19'}
   dev: true

 /indent-string@4.0.0:
   resolution: {integrity: sha512-
EdDDZu4A2OyIK7Lr/2zG+w5jmbuk1DVBnEwREQvBzspBJkCEbRa8GxU1lghYcaGJCnRWibjDXlq779X1/y
5xwg==}
   engines: {node: '>=8'}
   dev: true

 /inflight@1.0.6:
   resolution: {integrity: sha512-
k92I/b08q4wvFscXCLvqfsHCrjrF7yiXsQuIVvVE7N82W3+aqpzuUdBbfhWcy/FZR3/4IgflMgKLOsvPDrGCJA==
}
   dependencies:
     once: 1.4.0
     wrappy: 1.0.2

 /inherits@2.0.4:
   resolution: {integrity: sha512-
k/vGaX4/Yla3WzyMCvTQOXYeIHvqOKtnqBduzTHpzpQZzAskKMhZ2K+EnBiSM9zGSoIFeMpXKxa4dYeZI
QqewQ==}

 /internal-slot@1.0.6:
   resolution: {integrity: sha512-
Xj6dv+PsbtwyPpEflsejS+oIZxmMlV44zAhG479uYu89MsjcYOhCFnNyKrkJrihbsiasQyY0afoCl/9BLR65bg==}
```

```
    engines: {node: '>= 0.4'}
    dependencies:
      get-intrinsic: 1.2.2
      hasown: 2.0.0
      side-channel: 1.0.4
    dev: true

  /ipaddr.js@1.9.1:
    resolution: {integrity: sha512-
0KI/607xoxSToH7GjN1FfSbLoU0+btTicjsQSWQlh/hZykN8KpmMf7uYwPW3R+akZ6R/w18ZlXSHBYXiYUP
O3g==}
    engines: {node: '>= 0.10'}
    dev: false

  /is-arguments@1.1.1:
    resolution: {integrity: sha512-
8Q7EARjzEnKpt/PCD7e1cgUS0a6X8u5tdSiMqXhojOdoV9TsMsiO+9VLC5vAmO8N7/GmXn7yjR8qnA6bVA
EzfA==}
    engines: {node: '>= 0.4'}
    dependencies:
      call-bind: 1.0.2
      has-tostringtag: 1.0.0
    dev: false

  /is-array-buffer@3.0.2:
    resolution: {integrity: sha512-
y+FyyR/w8vfIRq4eQcM1EYgSTnmHXPqaF+IgzgraytCFq5Xh8lllDVmAZolPJiZttZLeFSINPYMaEJ7/vWUa1
w==}
    dependencies:
      call-bind: 1.0.5
      get-intrinsic: 1.2.2
      is-typed-array: 1.1.12
    dev: true

  /is-arrayish@0.2.1:
    resolution: {integrity: sha512-
zz06S8t0ozoDXMG+ube26zeCTNXcKIPJZJi8hBrF4idCLms4CG9QtK7qBl1boi5ODzFpjswb5JPmHCbMpjaY
zg==}
    dev: true

  /is-async-function@2.0.0:
    resolution: {integrity: sha512-
Y1JXKrfykRJGdlDwdKlLpLyMIiWqWvuSd17TvZk68PLAOGOoF4Xyav1z0Xhoi+gCYjZVeC5SI+hYFOfvXmG
RCA==}
    engines: {node: '>= 0.4'}
    dependencies:
      has-tostringtag: 1.0.0
    dev: true

  /is-bigint@1.0.4:
    resolution: {integrity: sha512-
zB9CruMamjym81i2JZ3UMn54PKGsQzsJeo6xvN3HJJ4CAsQNB6iRutp2To77OfCNuoxspsIhzaPoO1zyCEh
FOg==}
    dependencies:
      has-bigints: 1.0.2
    dev: true

  /is-binary-path@2.1.0:
    resolution: {integrity: sha512-
ZMERYes6pDydyuGidse7OsHxtbI7WVeUEozgR/g7rd0xUimYNlvZRE/K2MgZTjWy725IfelLeVcEM97mmtR
GXw==}
    engines: {node: '>=8'}
    dependencies:
      binary-extensions: 2.2.0

  /is-boolean-object@1.1.2:
    resolution: {integrity: sha512-
gDYaKHJmnj4aWxyj6YHyXVpdQawtVLHU5cb+eztPGczf6cjuTdwve5ZIEfgXqH4e57An1D1AKf8CZ3kYrQR
qYA==}
    engines: {node: '>= 0.4'}
    dependencies:
      call-bind: 1.0.2
      has-tostringtag: 1.0.0
    dev: true

  /is-builtin-module@3.2.1:
    resolution: {integrity: sha512-
BSLE3HnV2syZ0FK0iMA/yUGplUeMmNz4AW5fnTunbCIqZi4vG3WjJT9FHMy5D69xmAYBHXQhJdALdpwV
xV501A==}
    engines: {node: '>=6'}
    dependencies:
      builtin-modules: 3.3.0
    dev: true

  /is-callable@1.2.7:
    resolution: {integrity: sha512-
1BC0BVFhS/p0qtw6enp8e+8OD0UrK0oFLztSjNzhcKA3WDuJxxAPXzPuPtKkjEY9UUoEWlX/8fgKeu2S8i9J
TA==}
    engines: {node: '>= 0.4'}

  /is-core-module@2.13.1:
    resolution: {integrity: sha512-
hHrIjvZsftOsvKSn2TRYl63zvxsgE0K+0mYMoH6gD4omR5IWB2KynivBQczo3+wF1cCkjzvptnI9Q0sPU66ilw
==}
    dependencies:
      hasown: 2.0.0

  /is-date-object@1.0.5:
    resolution: {integrity: sha512-
9YQaSxsAiSwcvS33MBk3wTCVnWK+HhF8VZR2jRxehM16QcVOdHqPn4VPHmRK4lSr38n9JriurInLcP90x
sYNfQ==}
    engines: {node: '>= 0.4'}
    dependencies:
      has-tostringtag: 1.0.0
    dev: true

  /is-docker@2.2.1:
    resolution: {integrity: sha512-F+i2BKsFrH66iaUFc0woD8sLy8getkwTwtOBjvs56Cx4CgJDeKQeqfz8wAYiSb8JOprWhHH5p77PbmYCvvUuXQ==}
    engines: {node: '>=8'}
    hasBin: true
    dev: true

  /is-docker@3.0.0:
    resolution: {integrity: sha512-eljcgEDlEns/7AXFosB5K/2nCM4P7FQPkGc/DWLy5rmFEWvZayGrik1d9/QIY5nJ4f9YsVvBkA6kJpHn9rISdQ==}
    engines: {node: ^12.20.0 || ^14.13.1 || >=16.0.0}
    hasBin: true
    dev: true

  /is-extglob@2.1.1:
    resolution: {integrity: sha512-SbKbANkN603Vi4jEZv49LeVJMn4yGwsbzZworEoyEiutsN3nJYdbO36zfhGJ6QEDpOZIFkDtnq5JRxmvl3jsoQ==}
    engines: {node: '>=0.10.0'}

  /is-finalizationregistry@1.0.2:
    resolution: {integrity: sha512-0by5vtUJs8iFQb5TYUHHPudOR+qXYIMKtiUzvLIZITZUjknFmziyBJuLhVRc+Ds0dREFlskDNJKYIdIzu/9pfw==}
    dependencies:
      call-bind: 1.0.2
    dev: true

  /is-fn@1.0.0:
    resolution: {integrity: sha512-XoFPJQmsAShb3jEQRfzf2rqXavq7fIqF/jOekp308JlThqrODnMpweVSGilKTCXELfLhltGP2AGgbQGVP8F1dg==}
    engines: {node: '>=0.10.0'}
    dev: false

  /is-fullwidth-code-point@3.0.0:
    resolution: {integrity: sha512-zymm5+u+sCsSWyD9qNaejV3DFvhCKclKdizYaJUuHA83RLjb7nSuGnddCHGv0hk+KY7BMAlsWeK4Ueg6EV6XQg==}
    engines: {node: '>=8'}

  /is-function@1.0.2:
    resolution: {integrity: sha512-lw7DUp0aWXYg+CBCN+JKkcE0Q2RayZnSvnZBlwgxHBQhqt5pZNVy4Ri7H9GmmXkdu7LUthszM+Tor1u/2iBcpQ==}
    dev: false

  /is-generator-function@1.0.10:
    resolution: {integrity: sha512-jsEjy9l3yiXEQ+PsXdmBwEPcOxaXWLspKdplFUVI9vq1iZgIekeC0L167qeu86czQaxed3q/Uzuw0swL0irL8A==}
    engines: {node: '>= 0.4'}
    dependencies:
      has-tostringtag: 1.0.0

  /is-glob@4.0.3:
    resolution: {integrity: sha512-xelSayHH36ZgE7ZWhli7pW34hNbNl8Ojv5KVmkJD4hBdD3th8Tfk9vYasLM+mXWOZhFkgZfxhLSnrwRr4elSSg==}
    engines: {node: '>=0.10.0'}
    dependencies:
      is-extglob: 2.1.1

  /is-hex-prefixed@1.0.0:
    resolution: {integrity: sha512-WvtOiug1VFrE9v1Cydwm+FnXd3+w9GaeVUss5W4v/SLy3UW00vP+6iNF2SdnfiBoLy4bTqVdkftNGTUeOFVsbA==}
    engines: {node: '>=6.5.0', npm: '>=3'}
    dev: false

  /is-inside-container@1.0.0:
    resolution: {integrity: sha512-KIYLCCJghfHZxqjYBE7rEy0OBuTd5xCHS7tHVgvCLkx7StIoaxwNW3hCALgEUjFfeRk+MG/Qxmp/vtETEF3tRA==}
    engines: {node: '>=14.16'}
    hasBin: true
    dependencies:
      is-docker: 3.0.0
    dev: true

  /is-map@2.0.2:
    resolution: {integrity: sha512-cOZFQQozTha1f4MxLFzlgKYPTyj26picdZTx82hbc/Xf4K/tZOOXSCkMvU4pKioRXGDLJRn0GM7Upe7kR721yg==}
    dev: true

  /is-negative-zero@2.0.2:
    resolution: {integrity: sha512-dqJvarLawXsFbNDeJW7zAz8ItJ9cd28YufuuFzh0G8pNHjJMnY08Dv7sYX2uF5UpQOwieAeOExEYAWWfu7ZZUA==}
    engines: {node: '>= 0.4'}
    dev: true

  /is-number-object@1.0.7:
    resolution: {integrity: sha512-k1U0IRzLMo7ZlYIfzRu23Oh6MiIFasgpb9X76eqfFZAqwH44UI4KTBvBYIZ1dSL9ZzChTB9ShHfLkR4pdW5krQ==}
    engines: {node: '>= 0.4'}
    dependencies:
      has-tostringtag: 1.0.0
    dev: true

  /is-number@7.0.0:
    resolution: {integrity: sha512-
41Cifkg6e8TylSpdtTpeLVMqvSBEVzTttHvERD741+pnZ8ANv0004MRL43QKPDlK9cGvNp6NZWZUBlbGXY
xxng==}
    engines: {node: '>=0.12.0'}

  /is-path-inside@3.0.3:
    resolution: {integrity: sha512-
Fd4gABb+ycGAmKou8eMftCupSir5lRxqf4aD/vd0cD2qc4HL07OjCeuHMr8Ro4CoMaeCKDB0/ECBOVWjTw
UvPQ==}
    engines: {node: '>=8'}
    dev: true

  /is-plain-object@5.0.0:
    resolution: {integrity: sha512-
VRSzKkbMm5jMDoKLbltAkFQ5Qr7VDiTFGXxYFXXowVj387GeGNOCsOH6Msy00SGZ3Fp84b1Naa1psqg
cCIEP5Q==}
    engines: {node: '>=0.10.0'}
    dev: true

  /is-regex@1.1.4:
    resolution: {integrity: sha512-
kvRdxDsxZjhzUX07ZnLydzS1TU/TJlTUHHY4YLL87e37oUA49DfkLqgy+VjFocowy29cKvcSiu+kIv728jTTVg
==}
    engines: {node: '>= 0.4'}
    dependencies:
      call-bind: 1.0.5
      has-tostringtag: 1.0.0
    dev: true

  /is-set@2.0.2:
    resolution: {integrity: sha512-
+2cnTEZeY5z/iXGbLhPrOAaK/Mau5k5eXq9j14CpRTftq0pAJu2MwVRSZhyZWBzx3o6X795Lz6Bpb6R0GKf
37g==}
    dev: true

  /is-shared-array-buffer@1.0.2:
    resolution: {integrity: sha512-
sqN2UDu1/0y6uvXyStCOzyhAjCSlHceFoMKJW8W9EU9cvic/QdsZ0kEU93HEy3IUEFZIiH/3w+AH/UQbPHN
dhA==}
    dependencies:
      call-bind: 1.0.5
    dev: true

  /is-stream@2.0.1:
    resolution: {integrity: sha512-
hFoiJiTl63nn+kstHGBtewWSKnQLpyb155KHheA1l39uvtO9nWIop1p3udqPcUd/xbF1VLMO4n7OI6p7Rbng
Dg==}
    engines: {node: '>=8'}
    dev: true
  /is-stream@3.0.0:
    resolution: {integrity: sha512-LnQR4bZ9IADDRSkvpqMGvt/tEJWclzklNgSw48V5EAaAeDd6qGvN8ei6k5p0tvxSR171VmGyHuTiAOfxAbr8kA==}
    engines: {node: ^12.20.0 || ^14.13.1 || >=16.0.0}
    dev: true

  /is-string@1.0.7:
    resolution: {integrity: sha512-tE2UXzivje6ofPW7l23cjDOMa09gb7xlAqG6jG5ej6uPV32TlWP3NKPigtaGeHNu9fohccRYvIiZMfOOnOYUtg==}
    engines: {node: '>= 0.4'}
    dependencies:
      has-tostringtag: 1.0.0
    dev: true

  /is-symbol@1.0.4:
    resolution: {integrity: sha512-C/CPBqKWnvdcxqIARxyOh4v1UUEOCHpgDa0WYgpKDFMszcrPcffg5uhwSgPCLD2WWxmq6isisz87tzT01tuGhg==}
    engines: {node: '>= 0.4'}
    dependencies:
      has-symbols: 1.0.3
    dev: true

  /is-typed-array@1.1.10:
    resolution: {integrity: sha512-PJqgEHiWZvMpaFZ3uTc8kHPM4+4ADTlDniuQL7cU/UDA0Ql7F70yGfHph3cLNe+c9toaigv+DFzTJKhc2CtO6A==}
    engines: {node: '>= 0.4'}
    dependencies:
      available-typed-arrays: 1.0.5
      call-bind: 1.0.2
      for-each: 0.3.3
      gopd: 1.0.1
      has-tostringtag: 1.0.0

  /is-typed-array@1.1.12:
    resolution: {integrity: sha512-Z14TF2JNG8Lss5/HMqt0//T9JeHXttXy5pH/DBU4vi98ozO2btxzq9MwYDZYnKwU8nRsz/+GVFVRDq3DkVuSPg==}
    engines: {node: '>= 0.4'}
    dependencies:
      which-typed-array: 1.1.13
    dev: true

  /is-typedarray@1.0.0:
    resolution: {integrity: sha512-cyA56iCMHAh5CdzjJIa4aohJyeO1YbwLi3Jc35MmRU6poroFjIGZzUzupGiRPOjgHg9TLu43xbpwXk523fMxKA==}
    dev: false
  /is-weakmap@2.0.1:
    resolution: {integrity: sha512-NSBR4kH5oVj1Uwvv970ruUkCV7O1mzgVFO4/rev2cLRda9Tm9HrL70ZPut4rOHgY0FNrUu9BCbXA2sdQ+x0chA==}
    dev: true

  /is-weakref@1.0.2:
    resolution: {integrity: sha512-qctsuLZmIQ0+vSSMfoVvyFe2+GSEvnmZ2ezTup1SBse9+twCCeial6EEi3Nc2KFcf6+qz2FBPnjXsk8xhKSaPQ==}
    dependencies:
      call-bind: 1.0.5
    dev: true

  /is-weakset@2.0.2:
    resolution: {integrity: sha512-t2yVvttHkQktwnNNmBQ98AhENLdPUTDTE21uPqAQ0ARwQfGeQKRVS0NNurH7bTf7RrvcVn1OOge45CnBeHCSmg==}
    dependencies:
      call-bind: 1.0.2
      get-intrinsic: 1.2.2
    dev: true

  /is-wsl@2.2.0:
    resolution: {integrity: sha512-fKzAra0rGJUUBwGBgNkHZuToZcn+TtXHpeCgmkMJMMYx1sQDYaCSyjJBSCa2nH1DGm7s3n1oBnohoVTBaN7Lww==}
    engines: {node: '>=8'}
    dependencies:
      is-docker: 2.2.1
    dev: true

  /isarray@0.0.1:
    resolution: {integrity: sha512-D2S+3GLxWH+uhrNEcoh/fnmYeP8E8/zHl644d/jdA0g2uyXvy3sb0qxotE+ne0LtccHknQzWwZEzhak7oJ0COQ==}
    dev: false

  /isarray@1.0.0:
    resolution: {integrity: sha512-VLghIWNM6ELQzo7zwmcg0NmTVyWKYjvIeM83yjp0wRDTmUnrM678fQbcKBo6n2CJEF0szoG//ytg+TKla89ALQ==}
    dev: false

  /isarray@2.0.5:
    resolution: {integrity: sha512-xHjhDr3cNBK0BzdUJSPXZntQUx/mwMS5Rw4A7lPJ90XGAO6ISP/ePDNuo0vhqOZU+UD5JoodwCAAoZQd3FeAKw==}
    dev: true

  /isexe@2.0.0:
    resolution: {integrity: sha512-
RHxMLp9lnKHGHRng9QFhRCMbYAcVpn69smSGcq3f36xjgVVWThj4qqLbTLlq7Ssj8B+fIQ1EuCEGI2lKsy
QeIw==}
    dev: true

  /isomorphic-ws@4.0.1(ws@7.5.9):
    resolution: {integrity: sha512-
BhBvN2MBpWTaSHdWRb/bwdZJ1WaehQ2L1KngkCkfLUGF0mAWAT1sQUQacEmQ0jXkFw/czDXPNQSL
5u2/Krsz1w==}
    peerDependencies:
      ws: '*'
    dependencies:
      ws: 7.5.9
    dev: false

  /isomorphic-ws@5.0.0(ws@8.12.0):
    resolution: {integrity: sha512-
muId7Zzn9ywDsyXgTIafTry2sV3nySZeUDe6YedVd1Hvuuep5AsIlqK+XefWpYTyJG5e503F2xIuT2lcU6rCS
w==}
    peerDependencies:
      ws: '*'
    dependencies:
      ws: 8.12.0
    dev: false

  /isows@1.0.3(ws@8.13.0):
    resolution: {integrity: sha512-
2cKei4vlmg2cxEjm3wVSqn8pcoRF/LX/wpifuuNquFO4SQmPwarClT+SUCA2lt+l581tTeZIPIZuIDo2jWN1fg==
}
    peerDependencies:
      ws: '*'
    dependencies:
      ws: 8.13.0(bufferutil@4.0.7)(utf-8-validate@5.0.10)
    dev: false

  /isstream@0.1.2:
    resolution: {integrity: sha512-
Yljz7ffyPbrLpLngrMtZ7NduUgVvi6wG9RJ9IUcyCd59YQ911PBJphODUcbOVbqYfxe1wuYf/LJ8PauMRwsM/
g==}
    dev: false

  /iterator.prototype@1.1.2:
    resolution: {integrity: sha512-
DR33HMMr8EzwuRL8Y9D3u2BMj8+RqSE850jfGu59kS7tbmPLzGkZmVSfyCFSDxuZiEY6Rzt3T2NA/qU+N
wVj1w==}
    dependencies:
      define-properties: 1.2.1
      get-intrinsic: 1.2.2
      has-symbols: 1.0.3
      reflect.getprototypeof: 1.0.4
      set-function-name: 2.0.1
    dev: true
  /jayson@4.1.0:
    resolution: {integrity: sha512-
R6JlbyLN53Mjku329XoRT2zJAE6ZgOQ8f91ucYdMCD4nkGCF9kZSrcGXpHIU4jeKj58zUZke2p+cdQchU7L
y7A==}
    engines: {node: '>=8'}
    hasBin: true
    dependencies:
      '@types/connect': 3.4.35
      '@types/node': 12.20.55
      '@types/ws': 7.4.7
      JSONStream: 1.3.5
      commander: 2.20.3
      delay: 5.0.0
      es6-promisify: 5.0.0
      eyes: 0.1.8
      isomorphic-ws: 4.0.1(ws@7.5.9)
      json-stringify-safe: 5.0.1
      uuid: 8.3.2
      ws: 7.5.9
    transitivePeerDependencies:
      - bufferutil
      - utf-8-validate
    dev: false

  /jiti@1.20.0:
    resolution: {integrity: sha512-
3TV69ZbrvV6U5DfQimop50jE9Dl6J8O1ja1dvBbMba/sZ3YBEQqJ2VZRoQPVnhlzjNtU1vaXRZVrVjU4qtm8y
A==}
    hasBin: true
    dev: false

  /jose@5.0.1:
    resolution: {integrity: sha512-
gRVzy7s3RRdGbXmcTdlOswJOjhwPLx1ijIgAqLY6ktzFpOJxxYn4l0fC2vHaHHi4YBX/5FOL3aY+6W0cvQgp
ug==}
    dev: false

  /js-sha3@0.5.7:
    resolution: {integrity: sha512-
GII20kjaPX0zJ8wzkTbNDYMY7msuZcTWk8S5UOh6806Jq/wz1J8/bnr8uGU0DAUmYDjj2Mr4X1cW8v/GLY
nR+g==}
    dev: false

  /js-sha3@0.8.0:
    resolution: {integrity: sha512-
gF1cRrHhIzNfToc802P800N8PpXS+evLLXfsVpowqmAFR9uwbi89WvXg2QspOmXL8QL86J4T1EpFu+yUk
wJY3Q==}

  /js-tokens@4.0.0:
    resolution: {integrity: sha512-
RdJUflcE3cUzKiMqQgsCu06FPu9UdIJO0beYbPhHN4k6apgJtifcoCtT9bcxOpYBtpD2kCM6Sbzg4CausW/P
```

```
KQ==}

  /js-yaml@4.1.0:
    resolution: {integrity: sha512-
wpxZs9NoxZaJESJGIZTyDEaYpl0FKSA+FB9aJiyemKhMwkxQg63h4T1KJgUGHpTqPDNRcmmYLugrRjJlB
tWvRA==}
    hasBin: true
    dependencies:
      argparse: 2.0.1
    dev: true

  /jsbn@0.1.1:
    resolution: {integrity: sha512-
UVU9dibq2JcFWxQPA6KCqj5O42VOmAY3zQUfEKxU0KpTGXwNoCjkX1e13eHNvw/xPynt6pU0rZ1htjWTN
TSXsg==}
    dev: false

  /jsesc@0.5.0:
    resolution: {integrity: sha512-
uZz5UnB7u4T9LvwmFqXii7pZSouaRPorGs5who1Ip7VO0wxanFvBL7GkM6dTHlgX+jhBApRetaWpnDabOe
TcnA==}
    hasBin: true
    dev: true

  /jsesc@2.5.2:
    resolution: {integrity: sha512-
OYu7XEzjkCQ3C5Ps3QIZsQfNpqoJyZZA99wd9aWd05NCtC5pWOkShK2mkL6HXQR6/Cy2lbNdPlZBpuQH
XE63gA==}
    engines: {node: '>=4'}
    hasBin: true

  /jsesc@3.0.2:
    resolution: {integrity: sha512-
xKqzzWXDttJuOcawBt4KnKHHIf5oQ/Cxax+0PWFG+DFDgHNAdi+TXECADI+RYiFUMmx8792xsMbbgXj4C
wnP4g==}
    engines: {node: '>=6'}
    hasBin: true
    dev: true

  /json-buffer@3.0.1:
    resolution: {integrity: sha512-
4bV5BfR2mqfQTJm+V5tPPdf+ZpuhiIvTuAB5g8kcrXOZpTT/QwwVRWBywX1ozr6lEuPdbHxwaJlm9G6mI2sf
SQ==}

  /json-parse-even-better-errors@2.3.1:
    resolution: {integrity: sha512-
xyFwyhro/JEof6Ghe2iz2NcXoj2sloNsWr/XsERDK/oiPCfaNhl5ONfp+jQdAZRQQ0IJWNzH9zIZF7li91kh2w==
}
    dev: true

  /json-rpc-engine@5.4.0:
    resolution: {integrity: sha512-
```

```
rAffKbPoNDjuRnXkecTjnsE3xLLrb00rEkdgalINhaYVYIxDwWtvYBr9UFbhTvPB1B2qUOLoFd/cV6f4Q7mh7g
==}
    dependencies:
      eth-rpc-errors: 3.0.0
      safe-event-emitter: 1.0.1
    dev: false

  /json-rpc-engine@6.1.0:
    resolution: {integrity: sha512-
NEdLrtrq1jUZyfjkr9OCz9EzCNhnRyWtt1PAnvnhwy6e8XETS0Dtc+ZNCO2gvuAoKsIn2+vCSowXTYE4Ckgn
AQ==}
    engines: {node: '>=10.0.0'}
    dependencies:
      '@metamask/safe-event-emitter': 2.0.0
      eth-rpc-errors: 4.0.2
    dev: false

  /json-rpc-random-id@1.0.1:
    resolution: {integrity: sha512-
RJ9YYNCkhVDBuP4zN5BBtYAzEl03yq/jIIsyif0JY9qyJuQQZNeDK7anAPKKlyEtLSj2s8h6hNh2F8zO5q7Sc
A==}
    dev: false

  /json-schema-traverse@0.4.1:
    resolution: {integrity: sha512-
xbbCH5dCYU5T8LcEhhuh7HJ88HXuW3qsI3Y0zOZFKfZEHcpWiHU/Jxzk629Brsab/mMiHQti9wMP+845RP
e3Vg==}

  /json-schema-traverse@1.0.0:
    resolution: {integrity: sha512-
NM8/P9n3XjXhIZn1lLhkFaACTOURQXjWhV4BA/RnOv8xvgqtqpAX9IO4mRQxSx1Rlo4tqzeqb0sOlruaOy3d
ug==}
    dev: false

  /json-schema@0.4.0:
    resolution: {integrity: sha512-
es94M3nTIfsEPisRafak+HDLfHXnKBhV3vU5eqPcS3flIWqcxJWgXHXiey3YrpaNsanY5ei1VoYEbOzijuq9BA
==}
    dev: false

  /json-stable-stringify-without-jsonify@1.0.1:
    resolution: {integrity: sha512-
Bdboy+l7tA3OGW6FjyFHWkP5LuByj1Tk33Ljyq0axyzdk9//JSi2u3fP1QSmd1KNwq6VOKYGlAu87CisVir6Pw
==}
    dev: true

  /json-stable-stringify@1.0.2:
    resolution: {integrity: sha512-
eunSSaEnxV12z+Z73y/j5N37/In40GK4GmsSy+tEHJMxknvqnA7/djeYtAgW0GsWHUfg+847WJjKaEylk2y09
g==}
    dependencies:
      jsonify: 0.0.1
    dev: false

  /json-stringify-safe@5.0.1:
    resolution: {integrity: sha512-ZClg6AaYvamvYEE82d3Iyd3vSSIjQ+odgjaTzRuO3s7toCdFKczob2i0zCh7JE8kWn17yvAWhUVxvqGwUalsRA==}
    dev: false

  /json5@1.0.2:
    resolution: {integrity: sha512-g1MWMLBiz8FKi1e4w0UyVL3w+iJceWAFBAaBnnGKOpNa5f8TLktkbre1+s6oICydWAm+HRUGTmI+//xv2hvXYA==}
    hasBin: true
    dependencies:
      minimist: 1.2.8
    dev: true

  /json5@2.2.3:
    resolution: {integrity: sha512-XmOWe7eyHYH14cLdVPoyg+GOH3rYX++KpzrylJwSW98t3Nk+U8XOl8FWKOgwtzdb8lXGf6zYwDUzeHMWfxasyg==}
    engines: {node: '>=6'}
    hasBin: true

  /jsonfile@4.0.0:
    resolution: {integrity: sha512-m6F1R3z8jjlf2imQHS2Qez5sjKWQzbuuhuJ/FKYFRZvPE3PuHcSMVZzfsLhGVOkfd20obL5SWEBew5ShlquNxg==}
    optionalDependencies:
      graceful-fs: 4.2.11

  /jsonify@0.0.1:
    resolution: {integrity: sha512-2/Ki0GcmuqSrgFyelQq9M05y7PS0mEwuIzrf3f1fPqkVDVRvZrPZtVSMHxdgo8Aq0sxAOb/cr2aqqA3LeWHVPg==}
    dev: false

  /jsonparse@1.3.1:
    resolution: {integrity: sha512-POQXvpdL69+CluYsillJ7SUhKvytYjW9vG/GKpnf+xP8UWgYEM/RaMzHHofbALDiKbbP1W8UEYmgGl39WkPZsg==}
    engines: {'0': node >= 0.2.0}
    dev: false

  /jsprim@1.4.2:
    resolution: {integrity: sha512-P2bSOMAc/ciLz6DzgjVlGJP9+BrJWu5UDGK70C2iweC5QBIeFf0ZXRvGjEj2uYgrY2MkAAhsSWHDWlFtEroZWw==}
    engines: {node: '>=0.6.0'}
    dependencies:
      assert-plus: 1.0.0
      extsprintf: 1.3.0
      json-schema: 0.4.0
      verror: 1.10.0
    dev: false

  /jsx-ast-utils@3.3.5:
    resolution: {integrity: sha512-ZZow9HBI5O6EPgSJLUb8n2NKgmVWTwCvHGwFuJlMjvLFqlGG6pjirPhtdsseaLZjSibD8eegzmYpUZwoIlj2cQ==}
    engines: {node: '>=4.0'}
    dependencies:
      array-includes: 3.1.7
      array.prototype.flat: 1.3.2
      object.assign: 4.1.4
      object.values: 1.1.7
    dev: true

  /keccak@3.0.2:
    resolution: {integrity: sha512-PyKKjkH53wDMLGrvmRGSNWgmSxZOUqbnXwKL9tmgbFYA1iAYqW21kfR7mZXV0MlESiefxQQE9X9fTa3X+2MPDQ==}
    engines: {node: '>=10.0.0'}
    requiresBuild: true
    dependencies:
      node-addon-api: 2.0.2
      node-gyp-build: 4.5.0
      readable-stream: 3.6.0
    dev: false

  /key-encoder@2.0.3:
    resolution: {integrity: sha512-fgBtpAGIr/Fy5/+ZLQZIPPhsZEcbSlYu/Wu96tNDFNSjSACw5lEIOFeaVdQ/iwrb8oxjlWi6wmWdH76hV6GZjg==}
    dependencies:
      '@types/elliptic': 6.4.16
      asn1.js: 5.4.1
      bn.js: 4.12.0
      elliptic: 6.5.4
    dev: false

  /keyv@4.5.4:
    resolution: {integrity: sha512-oxVHkHR/EJf2CNXnWxRLW6mg7JyCCUcG0DtEGmL2ctUo1PNTin1PUil+r/+4r5MpVgC/fn1kjsx7mjSujKqIpw==}
    dependencies:
      json-buffer: 3.0.1

  /keyvaluestorage-interface@1.0.0:
    resolution: {integrity: sha512-8t6Q3TclQ4uZynJY9IGr2+SsIGwK9JHcO6ootkHCGA0CrQCRy+VkouYNO2xicET6b9al7QKzpebNow+gkpCL8g==}
    dev: false
  /language-subtag-registry@0.3.22:
    resolution: {integrity: sha512-
tN0MCzyWnoz/4nHS6uxdlFWoUZT7ABptwKPQ52Ea7URk6vll88bWBVhodtnlfEuCcKWNGoc+uGbw1cwa9I
Kh/w==}
    dev: true

  /language-tags@1.0.5:
    resolution: {integrity: sha512-
qJhlO9cGXi6hBGKoxEG/sKZDAHD5Hnu9Hs4WbOY3pCWXDhw0N8x1NenNzm2EnNLkLkk7J2SdxAkDSbb
6ftT+UQ==}
    dependencies:
      language-subtag-registry: 0.3.22
    dev: true

  /level-codec@7.0.1:
    resolution: {integrity: sha512-
Ua/R9B9r3RasXdRmOtd+t9TCOEIIlts+TN/7XTT2unhDaL6sJn83S3rUyljbr6lVtw49N3/yA0HHjpV6Kzb2aQ==
}
    dev: false

  /level-errors@1.0.5:
    resolution: {integrity: sha512-
/cLUpQduF6bNrWuAC4pwtUKA5t669pCsCi2XbmojG2tFeOr9j6ShtdDCtFFQO1DRt+EVZhx9gPzP9G2bUa
G4ig==}
    dependencies:
      errno: 0.1.8
    dev: false

  /level-iterator-stream@1.3.1:
    resolution: {integrity: sha512-
1qua0RHNtr4nrZBgYlpV0qHHeHpcRRWTxEZJ8xsemoHAXNL5tbooh4tPEEqIqsbWCAJBmUmkwYK/sW5O
rFjWWw==}
    dependencies:
      inherits: 2.0.4
      level-errors: 1.0.5
      readable-stream: 1.1.14
      xtend: 4.0.2
    dev: false

  /level-ws@0.0.0:
    resolution: {integrity: sha512-
XUTaO/+Db51Uiyp/t7fCMGVFOTdtLS/NIACxE/GHsij15mKzxksZifKVjlXDF41JMUP/oM1Oc4YNGdKnc3dVL
w==}
    dependencies:
      readable-stream: 1.0.34
      xtend: 2.1.2
    dev: false

  /levelup@1.3.9:
    resolution: {integrity: sha512-
VVGHfKIlmw8w1XqpGOAGwq6sZm2WwWLmlDcULkKWQXEA5EopA8OBNJ2Ck2v6bdk8HeEZSbCSEgzX
adyQFm76sQ==}
```

```
    dependencies:
      deferred-leveldown: 1.2.2
      level-codec: 7.0.1
      level-errors: 1.0.5
      level-iterator-stream: 1.3.1
      prr: 1.0.1
      semver: 5.4.1
      xtend: 4.0.2
    dev: false

  /levn@0.4.1:
    resolution: {integrity: sha512-
+bT2uH4E5LGE7h/n3evcS/sQlJXCpIp6ym8OWJ5eV6+67Dsql/LaaT7qJBAt2rzfoa/5QBGBhxDix1dMt2kQK
Q==}
    engines: {node: '>= 0.8.0'}
    dependencies:
      prelude-ls: 1.2.1
      type-check: 0.4.0
    dev: true

  /light-my-request@5.11.0:
    resolution: {integrity: sha512-
qkFCeloXCOMpmEdZ/MV91P8AT4fjwFXWaAFz3lUeStM8RcoM1ks4J/F8r1b3r6y/H4u3ACEJ1T+Gv5bopj7o
DA==}
    dependencies:
      cookie: 0.5.0
      process-warning: 2.2.0
      set-cookie-parser: 2.6.0
    dev: false

  /lilconfig@2.1.0:
    resolution: {integrity: sha512-
utWOt/GHzuUxnLKxB6dk81RoOeoNeHgbrXiuGk4yyF5qlRz+iIVWu56E2fqGHFrXz0QNUhLB/8nKqvRH66J
KGQ==}
    engines: {node: '>=10'}
    dev: false

  /lines-and-columns@1.2.4:
    resolution: {integrity: sha512-
7ylylesZQ/PV29jhEDl3Ufjo6ZX7gCqJr5F7PKrqc93v7fzSymt1BpwEU8nAUXs8qzzvqhbjhK5QZg6Mt/HkBg==
}

  /lit-element@3.3.3:
    resolution: {integrity: sha512-
XbeRxmTHubXENkV4h8RIPyr8lXc+Ff28rkcQzw3G6up2xg5E8Zu1IgOWIwBLEQsu3cOVFqdYwiVi0hv0Slpq
UA==}
    dependencies:
      '@lit-labs/ssr-dom-shim': 1.1.2
      '@lit/reactive-element': 1.6.3
      lit-html: 2.8.0
    dev: false
  /lit-html@2.8.0:
    resolution: {integrity: sha512-
o9t+MQM3P4y7M7yNzqAyjp7z+mQGa4NS4CxiyLqFPyFWyc4O+nodLrkrxSaCTrla6M5YOLaT3RpbbqjszB5
g3Q==}
    dependencies:
      '@types/trusted-types': 2.0.2
    dev: false

  /lit@2.7.4:
    resolution: {integrity: sha512-
cgD7xrZoYr21mbrkZIuIrj98YTMw/snJPg52deWVV4A8icLyNHI3bF70xsJeAgwTuiq5Kkd+ZR8gybSJDCPB7g
==}
    dependencies:
      '@lit/reactive-element': 1.6.3
      lit-element: 3.3.3
      lit-html: 2.8.0
    dev: false

  /lit@2.8.0:
    resolution: {integrity: sha512-
4Sc3OFX9QHOJaHbmTMk28SYgVxLN3ePDjg7hofEft2zWlehFL3LiAuapWc4U/kYwMYJSh2hTCPZ6/LIC7ii
0MA==}
    dependencies:
      '@lit/reactive-element': 1.6.3
      lit-element: 3.3.3
      lit-html: 2.8.0
    dev: false

  /locate-path@5.0.0:
    resolution: {integrity: sha512-
t7hw9pI+WvuwNJXwk5zVHpyhIqzg2qTlklJOf0mVxGSbe3Fp2VieZcduNYjaLDoy6p9uGpQEGWG87WpMKl
Nq8g==}
    engines: {node: '>=8'}
    dependencies:
      p-locate: 4.1.0

  /locate-path@6.0.0:
    resolution: {integrity: sha512-
iPZK6eYjbxRu3uB4/WZ3EsEIMJFMqAoopl3R+zuq0UjcAm/MO6KCweDgPfP3elTztoKP3KtnVHxTn2NHBS
DVUw==}
    engines: {node: '>=10'}
    dependencies:
      p-locate: 5.0.0
    dev: true

  /lodash.camelcase@4.3.0:
    resolution: {integrity: sha512-
TwuEnCnxbc3rAvhf/LbG7tJUDzhqXyFnv3dtzLOPgCG/hODL7WFnsbwktkD7yUV0RrreP/l1PALq/YSg6VvjlA
==}

  /lodash.curry@4.1.1:
    resolution: {integrity: sha512-
/u14pXGviLaweY5JI0IUzgzF2J6Ne8INyzAZjImcryjgkZ+ebruBxy2/JaOOkTqScddcYtakjhSaeemV8lR0tA==}
    dev: false

  /lodash.debounce@4.0.8:
    resolution: {integrity: sha512-
FT1yDzDYEoYWhnSGnpE/4Kj1fLZkDFyqRb7fNt6FdYOSxlUWAtp42Eh6Wb0rGIv/m9Bgo7x4GhQbm5Ys4S
G5ow==}
    dev: false

  /lodash.flow@3.5.0:
    resolution: {integrity: sha512-
ff3BX/tSioo+XojX4MOsOMhJw0nZoUEF011LX8g8d3gvjVbxd89cCio4BCXronjxcTUIJUoqKEUA+n4CqvvRP
w==}
    dev: false

  /lodash.isequal@4.5.0:
    resolution: {integrity: sha512-
pDo3lu8Jhfjqls6GkMgpahsF9kCyayhgykjyLMNFTKWrpVdAQtYyB4muAMWozBB4ig/dtWAmsMxLEI8wuz+
DYQ==}
    dev: false

  /lodash.merge@4.6.2:
    resolution: {integrity: sha512-
0KpjqXRVvrYyCsX1swR/XTK0va6VQkQM6MNo7PqW77ByjAhoARA8EfrP1N4+KlKj8YS0ZUCtRT/YUuhyY
DujIQ==}
    dev: true

  /lodash@4.17.21:
    resolution: {integrity: sha512-
v2kDEe57lecTulaDIuNTPy3Ry4gLGJ6Z1O3vE1krgXZNrsQ+LFTGHVxVjcXPs17LhbZVGedAJv8XZ1tvj5Fv
Sg==}

  /loose-envify@1.4.0:
    resolution: {integrity: sha512-
lyuxPGr/Wfhrlem2CL/UcnUc1zcqKAImBDzukY7Y5F/yQiNdko6+fRLevlw1HgMySw7f611UIY408EtxRSoK3Q
==}
    hasBin: true
    dependencies:
      js-tokens: 4.0.0

  /lowercase-keys@2.0.0:
    resolution: {integrity: sha512-
tqNXrS78oMOE73NMxK4EMLQsQowWf8jKooH9g7xPavRT706R6bkQJ6DY2Te7QukaZsulxa30wQ7bk0pm
4XiHmA==}
    engines: {node: '>=8'}
    dev: false

  /lowercase-keys@3.0.0:
    resolution: {integrity: sha512-
ozCC6gdQ+glXOQsveKD0YsDy8DSQFjDTz4zyzEHNV5+JP5D62LmfDZ6o1cycFx9ouG940M5dE8C8CTew
dj2YWQ==}
    engines: {node: ^12.20.0 || ^14.13.1 || >=16.0.0}
    dev: false

  /lru-cache@5.1.1:
    resolution: {integrity: sha512-
KpNARQA3Iwv+jTA0utUVVbrh+Jlrr1Fv0e56GGzAFOXN7dk/FviaDW8LHmK52DlcH4WP2n6gI8vN1aesBFg
o9w==}
    dependencies:
      yallist: 3.1.1

  /lru-cache@6.0.0:
    resolution: {integrity: sha512-
Jo6dJ04CmSjuznwJSS3pUeWmd/H0ffTlkXXgwZi+eq1UCmqQwCh+eLsYOYCwY991i2Fah4h1BEMCx4qTh
GbsiA==}
    engines: {node: '>=10'}
    dependencies:
      yallist: 4.0.0

  /ltgt@2.2.1:
    resolution: {integrity: sha512-
AI2r85+4MquTw9ZYqabu4nMwy9Oftlfa/e/52t9IjtfG+mGBbTNdAoZ3RQKLHR6r0wQnwZnPIEh/Ya6XTWAK
NA==}
    dev: false

  /make-error@1.3.6:
    resolution: {integrity: sha512-
s8UhlNe7vPKomQhC1qFelMokr/Sc3AgNbso3n74mVPA5LTZwkB9NlXf4XPamLxJE8h0gh73rM94xvwRT2C
VInw==}

  /md5.js@1.3.5:
    resolution: {integrity: sha512-
xitP+WxNPcTTOgnTJcrhM0xvdPepipPSf3I8EIpGKeFLjt3PlJLIDG3u8EX53ZIubkb+5U2+3rELYpEhHhzdkg==
}
    dependencies:
      hash-base: 3.1.0
      inherits: 2.0.4
      safe-buffer: 5.2.1
    dev: false

  /media-typer@0.3.0:
    resolution: {integrity: sha512-
dq+qelQ9akHpcOl/gUVRTxVIOkAJ1wR3QAvb4RsVjS8oVoFjDGTc679wJYmUmknUF5HwMLOgb5O+a3Kxf
WapPQ==}
    engines: {node: '>= 0.6'}
    dev: false

  /memdown@1.4.1:
    resolution: {integrity: sha512-
iVrGHZB8i4OQfM155xx8akvG9FIj+ht14DX5CQkCTG4EHzZ3d3sgckIf/Lm9ivZalEsFuEVnWv2B2WZvbrro2
w==}
    dependencies:
      abstract-leveldown: 2.7.2
      functional-red-black-tree: 1.0.1
      immediate: 3.3.0
      inherits: 2.0.4
      ltgt: 2.2.1
      safe-buffer: 5.1.2
    dev: false

  /merge-descriptors@1.0.1:
    resolution: {integrity: sha512-cCi6g3/Zr1iqQi6ySbseM1Xvooa98N0w31jzUYrXPX2xqObmFGHJ0tQ5u74H3mVh7wLouTseZyYIq39g8cNp1w==}
    dev: false

  /merge-stream@2.0.0:
    resolution: {integrity: sha512-abv/qOcuPfk3URPfDzmZU1LKmuw8kT+0nIHvKrKgFrwifol/doWcdA4ZqsWQ8ENrFKkd67Mfpo/LovbIUsbt3w==}
    dev: true

  /merge2@1.4.1:
    resolution: {integrity: sha512-8q7VEgMJW4J8tcfVPy8g09NcQwZdbwFEqhe/WZkoIzjn/3TGDwtOCYtXGxA3O8tPzpczCCDgv+P2P5y00ZJOOg==}
    engines: {node: '>= 8'}

  /merkle-patricia-tree@2.3.2:
    resolution: {integrity: sha512-81PW5m8oz/pz3GvsAwbauj7Y00rqm81Tzad77tHBwU7pIAtN+TJnMSOJhxBKflSVYhptMMb9RskhqHqrSm1V+g==}
    dependencies:
      async: 1.5.2
      ethereumjs-util: 5.2.1
      level-ws: 0.0.0
      levelup: 1.3.9
      memdown: 1.4.1
      readable-stream: 2.3.8
      rlp: 2.2.7
      semaphore: 1.1.0
    dev: false

  /methods@1.1.2:
    resolution: {integrity: sha512-iclAHeNqNm68zFtnZ0e+1L2yUIdvzNoauKU4WBA3VvH/vPFieF7qfRlwUZU+DA9P9bPXIS90ulxoUoCH23sV2w==}
    engines: {node: '>= 0.6'}
    dev: false

  /micromatch@4.0.5:
    resolution: {integrity: sha512-DMy+ERcEW2q8Z2Po+WNXuw3c5YaUSFjAO5GsJqfEl7UjvtIuFKO6ZrKvcItdy98dwFI2N1tg3zNIdKaQT+aNdA==}
    engines: {node: '>=8.6'}
    dependencies:
      braces: 3.0.2
      picomatch: 2.3.1

  /mime-db@1.52.0:
    resolution: {integrity: sha512-sPU4uV7dYlvtWJxwwxHD0PuihVNiE7TyAbQ5SWxDCB9mUYvOgroQOwYQQOKPJ8CIbE+1ETVlOoK1UC2nU3gYvg==}
    engines: {node: '>= 0.6'}

  /mime-types@2.1.35:
    resolution: {integrity: sha512-ZDY+bPm5zTTF+YpCrAU9nK0UgICYPT0QtT1NZWFv4s++TNkcgVaT0g6+4R2uI4MjQjzysHB1zxuWL50hzaeXiw==}
    engines: {node: '>= 0.6'}
    dependencies:
      mime-db: 1.52.0

  /mime@1.6.0:
    resolution: {integrity: sha512-x0Vn8spI+wuJ1O6S7gnbaQg8Pxh4NNHb7KSINmEWKiPE4RKOplvijn+NkmYmmRgP68mc70j2EbeTFRsrswaQeg==}
    engines: {node: '>=4'}
    hasBin: true
    dev: false

  /mime@3.0.0:
    resolution: {integrity: sha512-jSCU7/VB1loIWBZe14aEYHU/+1UMEHoaO7qxCOVJOw9GgH72VAWppxNcjU+x9a2k3GSIBXNKxXQFqRvvZ7vr3A==}
    engines: {node: '>=10.0.0'}
    hasBin: true
    dev: false

  /mimic-fn@2.1.0:
    resolution: {integrity: sha512-OqbOk5oEQeAZ8WXWydlu9HJjz9WVdEIvamMCcXmuqUYjTknH/sqsWvhQ3vgwKFRR1HpjvNBKQ37nbJgYzGqGcg==}
    engines: {node: '>=6'}
    dev: true

  /mimic-fn@4.0.0:
    resolution: {integrity: sha512-vqiC06CuhBTUdZH+RYl8sFrL096vA45Ok5ISO6sE/Mr1jRbGH4Csnhi8f3wKVl7x8mO4Au7Ir9D3Oyv1VYMFJw==}
    engines: {node: '>=12'}
    dev: true

  /mimic-response@1.0.1:
    resolution: {integrity: sha512-j5EctnkH7amfV/q5Hgmoal1g2QHFJRraOtmx0JpIqkxhBhI/lJSl1nMpQ45hVarwNETOoWEimndZ4QK0RHxuxQ==}
    engines: {node: '>=4'}
    dev: false

  /mimic-response@3.1.0:
    resolution: {integrity: sha512-
z0yWI+4FDrrweS8Zmt4Ej5HdJmky15+L2e6Wgn3+iK5fWzb6T3fhNFq2+MeTRb064c6Wr4N/wv0DzQTjNzH
NGQ==}
    engines: {node: '>=10'}
    dev: false

  /min-document@2.19.0:
    resolution: {integrity: sha512-
9Wy1B3m3f66bPPmU5hdA4DR4PB2OfDU/+GS3yAB7IQozE3tqXaVv2zOjgla7MEGSRv95+ILmOuvhLkOK
6wJtCQ==}
    dependencies:
      dom-walk: 0.1.2
    dev: false

  /min-indent@1.0.1:
    resolution: {integrity: sha512-
I9jwMn07Sy/IwOj3zVkVik2JTvgpaykDZEigL6Rx6N9LbMywwUSMtxET+7lVoDLLd3O3IXwJwvuuns8UB/HeA
g==}
    engines: {node: '>=4'}
    dev: true

  /mini-svg-data-uri@1.4.4:
    resolution: {integrity: sha512-
r9deDe9p5FJUPZAk3A59wGH7Ii9YrjjWw0jmw/liSbHl2CHiyXj6FcDXDu2K3TjVAXqiJdaw3xxwlZZr9E6nHg==
}
    hasBin: true
    dev: false

  /minimalistic-assert@1.0.1:
    resolution: {integrity: sha512-
UtJcAD4yEaGtjPezWuO9wC4nwUnVH/8/Im3yEHQP4b67cXlD/Qr9hdITCU1xDbSEXg2XKNaP8jsReV7vQd
00/A==}
    dev: false

  /minimalistic-crypto-utils@1.0.1:
    resolution: {integrity: sha512-
JIYlbt6g8i5jKfJ3xz7rF0LXmv2TkDxBLUkiBeZ7bAx4GnnNMr8xFpGnOxn6GhTEHx3SjRrZEoU+j04prX1ktg==
}
    dev: false

  /minimatch@3.1.2:
    resolution: {integrity: sha512-
J7p63hRiAjw1NDEww1W7i37+ByIrOWO5XQQAzZ3VOcL0PNybwpfmV/N05zFAzwQ9USyEcX6t3UO+K5aq
BQOIHw==}
    dependencies:
      brace-expansion: 1.1.11

  /minimatch@5.1.6:
    resolution: {integrity: sha512-
```

```
lKwV/1brpG6mBUFHtb7NUmtABCb2WZZmm2wNiOA5hAb8VdCS4B3dtMWyvcoViccwAW/COERjXLt0zP1z
XUN26g==}
    engines: {node: '>=10'}
    dependencies:
      brace-expansion: 2.0.1
    dev: false

  /minimist@1.2.8:
    resolution: {integrity: sha512-
2yyAR8qBkN3YuheJanUpWC5U3bb5osDywNB8RzDVlDwDHbocAJveqqj1u8+SVD7jkWT4yvsHCpWqqWq
Axb0zCA==}

  /minipass@2.9.0:
    resolution: {integrity: sha512-
wxfUjg9WebH+CUDX/CdbRlh5SmfZiy/hpkxaRI16Y9W56Pa75sWgd/rvFilSgrauD9NyFymP/+JFV3KwzIsJeg
==}
    dependencies:
      safe-buffer: 5.2.1
      yallist: 3.1.1
    dev: false

  /minizlib@1.3.3:
    resolution: {integrity: sha512-
6ZYMOEnmVsdCeTJVE0W9ZD+pVnE8h9Hma/iOwwRDsdQoePpoX56/8B6z3P9VNwppJuBKNRuFDRNRq
RWexT9G9Q==}
    dependencies:
      minipass: 2.9.0
    dev: false

  /mkdirp-promise@5.0.1:
    resolution: {integrity: sha512-
Hepn5kb1lJPtVW84RFT40YG1OddBNTOVUZR2bzQUHc+Z03en8/3uX0+060JDhcEzyO08HmipsN9DcnFM
xhIL9w==}
    engines: {node: '>=4'}
    deprecated: This package is broken and no longer maintained. 'mkdirp' itself supports promises now,
please switch to that.
    dependencies:
      mkdirp: 1.0.4
    dev: false

  /mkdirp@0.5.6:
    resolution: {integrity: sha512-
FP+p8RB8OWpF3YZBCrP5gtADmtXApB5AMLn+vdyA+PyxCjrCs00mjyUozssO33cwDeT3wNGdLxJ5M//Yq
tHAJw==}
    hasBin: true
    dependencies:
      minimist: 1.2.8
    dev: false

  /mkdirp@1.0.4:
    resolution: {integrity: sha512-
vVqVZQyf3WLx2Shd0qJ9xuvqgAyKPLAiqITEtqW0oIUjzo3PePDd6fW9iFz30ef7Ysp/oiWqbhszeGWW2T6G
```

```
zw==}
    engines: {node: '>=10'}
    hasBin: true

  /mnemonist@0.39.5:
    resolution: {integrity: sha512-
FPUtkhtJ0efmEFGpU14x7jGbTB+s18LrzRL2KgoWz9YvcY3cPomz8tih01GbHwnGk/OmkOKfqd/RAQoc8Lm
7DQ==}
    dependencies:
      obliterator: 2.0.4
    dev: false

  /mock-fs@4.14.0:
    resolution: {integrity: sha512-
qYvlv/exQ4+svI3UOvPUpLDF0OMX5euvUH0Ny4N5QyRyhNdgAgUrVH3iUINSzEPLvx0kbo/Bp28GJKIqvE7
URw==}
    dev: false

  /motion@10.15.5:
    resolution: {integrity: sha512-
ejP6KioN4pigTGxL93APzOnvtLklParL59UQB2T3HWXQBxFcIp5/7YXFmkgiA6pNKKzjvnLhnonRBN5iSFMn
Nw==}
    dependencies:
      '@motionone/animation': 10.16.3
      '@motionone/dom': 10.16.4
      '@motionone/svelte': 10.16.4
      '@motionone/types': 10.16.3
      '@motionone/utils': 10.16.3
      '@motionone/vue': 10.16.4
    dev: false

  /motion@10.16.2:
    resolution: {integrity: sha512-
p+PurYqfUdcJZvtnmAqu5fJgV2kR0uLFQuBKtLeFVTrYEVllI99tiOTSefVNYuip9ELTEkepIIDftNdze76NAQ==
}
    dependencies:
      '@motionone/animation': 10.16.3
      '@motionone/dom': 10.16.4
      '@motionone/svelte': 10.16.4
      '@motionone/types': 10.16.3
      '@motionone/utils': 10.16.3
      '@motionone/vue': 10.16.4
    dev: false

  /ms@2.0.0:
    resolution: {integrity: sha512-
Tpp60P6IUJDTuOq/5Z8cdskzJujfwqfOTkrwIwj7IRISpnkJnT6SyJ4PCPnGMoFjC9ddhal5KVIYtAt97ix05A==}
    dev: false

  /ms@2.1.2:
    resolution: {integrity: sha512-
sGkPx+VjMtmA6MX27oA4FBFELFCZZ4S4XqeGOXCv68tT+jb3vk/RyaKWP0PTKyWtmLSM0b+adUTEvbs1
```

```
PEaH2w==}

  /ms@2.1.3:
    resolution: {integrity: sha512-
6FlzubTLZG3J2a/NVCAleEhjzq5oxgHyaCU9yYXvcLsvoVaHJq/s5xXI6/XXP6tz7R9xAOtHnSO/tXtF3WRTlA
==}

  /multibase@0.6.1:
    resolution: {integrity: sha512-
pFfAwyTjbbQgNc3G7D48JkJxWtoJoBMaR4xQUOuB8RnCgRqaYmWNFeJTTvrJ2w51bjLq2zTby6Rqj9TQ9
elSUw==}
    deprecated: This module has been superseded by the multiformats module
    dependencies:
      base-x: 3.0.9
      buffer: 5.7.1
    dev: false

  /multibase@0.7.0:
    resolution: {integrity: sha512-
TW8q03O0f6PNFTQDvh3xxH03c8CjGaaYrjkl9UQPG6rz53TQzzxJVCIWVjzcbN/Q5Y53Zd0IBQBMVktVgNx
4Fg==}
    deprecated: This module has been superseded by the multiformats module
    dependencies:
      base-x: 3.0.9
      buffer: 5.7.1
    dev: false

  /multicodec@0.5.7:
    resolution: {integrity: sha512-
PscoRxm3f+88fAtELwUnZxGDkduE2HD9Q6GHUOywQLjOGT/HAdhjLDYNZ1e7VR0s0TP0EwZ16LNUTFp
oBGivOA==}
    deprecated: This module has been superseded by the multiformats module
    dependencies:
      varint: 5.0.2
    dev: false

  /multicodec@1.0.4:
    resolution: {integrity: sha512-
NDd7FeS3QamVtbgfvu5h7fd1IlbaC4EQ0/pgU4zqE2vdHCmBGsUa0TiM8/TdSeG6BMPC92OOCf8F1ocE/W
krrg==}
    deprecated: This module has been superseded by the multiformats module
    dependencies:
      buffer: 5.7.1
      varint: 5.0.2
    dev: false

  /multiformats@9.9.0:
    resolution: {integrity: sha512-
HoMUjhH9T8DDBNT+6xzkrd9ga/XiBI4xLr58LJACwK6G3HTOPeMz4nB4KJs33L2BelrIJa7P0VuNaVF3hMY
fjg==}
    dev: false
  /multihashes@0.4.21:
    resolution: {integrity: sha512-
uVSvmeCWf36pU2nB4/1kzYZjsXD9vofZKpgudqkceYY5g2aZZXJ5r9lxuzoRLl1OAp28XljXsEJ/X/85ZsKmKw
==}
    dependencies:
      buffer: 5.7.1
      multibase: 0.7.0
      varint: 5.0.2
    dev: false

  /mz@2.7.0:
    resolution: {integrity: sha512-
z81GNO7nnYMEhrGh9LeymoE4+Yr0Wn5McHIZMK5cfQCl+NDX08sCZgUc9/6MHni9IWuFLm1Z3HTCXu2z
9fN62Q==}
    dependencies:
      any-promise: 1.3.0
      object-assign: 4.1.1
      thenify-all: 1.6.0
    dev: false

  /nano-json-stream-parser@0.1.2:
    resolution: {integrity: sha512-
9MqxMH/BSJC7dnLsEMPyfN5Dvoo49IsPFYMcHw3Bcfc2kN0lpHRBSzlMSVx4HGyJ7s9B31CyBTVehWJo
Q8Ctew==}
    dev: false

  /nanoid@3.3.6:
    resolution: {integrity: sha512-
BGcqMMJuToF7i1rt+2PWSNVnWIkGCU78jBG3RxO/bZlnZPK2Cmi2QaffxGO/2RvWi9sL+FAiRiXMgsyxQ1
DIDA==}
    engines: {node: ^10 || ^12 || ^13.7 || ^14 || >=15.0.1}
    hasBin: true

  /natural-compare@1.4.0:
    resolution: {integrity: sha512-
OWND8ei3VtNC9h7V60qff3SVobHr996CTwgxubgyQYEpg290h9J0buyECNNJexkFm5sOajh5G116RYA1c8
ZMSw==}
    dev: true

  /negotiator@0.6.3:
    resolution: {integrity: sha512-
+EUsqGPLsM+j/zdChZjsnX51g4XrHFOIXwfnCVPGlQk/k5giakcKsuxCObBRu6DSm9opw/O6slWbJdghQM4
bBg==}
    engines: {node: '>= 0.6'}
    dev: false

  /next-tick@1.1.0:
    resolution: {integrity: sha512-
CXdUiJembsNjuToQvxayPZF9Vqht7hewsvy2sOWafLvi2awflj9mOC6bHIg50orX8IJvWKY9wYQ/zB2kogPsl
Q==}
    dev: false
```

```
 /node-addon-api@2.0.2:
   resolution: {integrity: sha512-
Ntyt4AIXyaLIuMHF6IOoTakB3K+RWxwtsHNRxllEoA6vPwP9o4866g6YWDLUdnucilZhmkxiHwHr11gAENw
+QA==}
   dev: false

 /node-fetch@2.6.7:
   resolution: {integrity: sha512-
ZjMPFEfVx5j+y2yF35Kzx5sF7kDzxuDj6ziH4FFbOp87zKDZNx8yExJIb05OGF4Nlt9IHFIMBkRl41VdvcNdbQ
==}
   engines: {node: 4.x || >=6.0.0}
   peerDependencies:
     encoding: ^0.1.0
   peerDependenciesMeta:
     encoding:
       optional: true
   dependencies:
     whatwg-url: 5.0.0
   dev: false

 /node-fetch@2.7.0:
   resolution: {integrity: sha512-
c4FRfUm/dbcWZ7U+1Wq0AwCyFL+3nt2bEw05wfxSz+DWpWsitgmSgYmy2dQdWyKC1694ELPqMs/YzUS
NozLt8A==}
   engines: {node: 4.x || >=6.0.0}
   peerDependencies:
     encoding: ^0.1.0
   peerDependenciesMeta:
     encoding:
       optional: true
   dependencies:
     whatwg-url: 5.0.0

 /node-gyp-build@4.5.0:
   resolution: {integrity: sha512-
2iGbaQBV+ITgCz76ZEjmhUKAKVf7xfY1sRl4UiKQspfZMH2h06SyhNsnSVy50cwkFQDGLyif6m/6uFXHkOZ
6rg==}
   hasBin: true
   dev: false

 /node-releases@2.0.13:
   resolution: {integrity: sha512-
uYr7J37ae/ORWdZeQ1xxMJe3NtdmqMC/JZK+geofDrkLUApKRHPd18/TxtBOJ4A0/+uUIliorNrfYV6s1b02e
Q==}

 /nodemon@3.0.1:
   resolution: {integrity: sha512-
g9AZ7HmkhQkqXkRc20w+ZfQ73cHLbE8hnPbtaFbFtCumZsjyMhKk9LajQ07U5Ux28lvFjZ5X7HvWR1xzU8j
HVw==}
   engines: {node: '>=10'}
   hasBin: true
   dependencies:
     chokidar: 3.5.3
     debug: 3.2.7(supports-color@5.5.0)
     ignore-by-default: 1.0.1
     minimatch: 3.1.2
     pstree.remy: 1.1.8
     semver: 7.5.4
     simple-update-notifier: 2.0.0
     supports-color: 5.5.0
     touch: 3.1.0
     undefsafe: 2.0.5
   dev: true

 /nopt@1.0.10:
   resolution: {integrity: sha512-
NWmpvLSqUrgrAC9HCuxEvb+PSloHpqVu+FqcO4eeF2h5qYRhA7ev6KvelyQAKtegUbC6RypJnlEOhd8vlo
NKYg==}
   hasBin: true
   dependencies:
     abbrev: 1.1.1
   dev: true

 /normalize-package-data@2.5.0:
   resolution: {integrity: sha512-
/5CMN3T0R4XTj4DcGaexo+roZSdSFW/0AOOTROrjxzCG1wrWXEsGbRKevjlIL+ZDE4sZlJr5ED4YW0yqmk
K+eA==}
   dependencies:
     hosted-git-info: 2.8.9
     resolve: 1.22.8
     semver: 5.4.1
     validate-npm-package-license: 3.0.4
   dev: true

 /normalize-path@3.0.0:
   resolution: {integrity: sha512-
6eZs5Ls3WtCisHWp9S2GUy8dqkpGi4BVSz3GaqiE6ezub0512ESztXUwUB6C6IKbQkY2Pnb/mD4WYojCR
wcwLA==}
   engines: {node: '>=0.10.0'}

 /normalize-range@0.1.2:
   resolution: {integrity: sha512-
bdok/XvKII3nUpklnV6P2hxtMNrCboOjAcyBuQnWEhO665FwrSNRxU+AqpsyvO6LgGYPspN+lu5CLtw4jPR
KNA==}
   engines: {node: '>=0.10.0'}
   dev: false

 /normalize-url@6.1.0:
   resolution: {integrity: sha512-
DlL+XwOy3NxAQ8xuC0okPgK46iuVNAK01YN7RueYBqqFeGsBjV9XmCAzAdgt+667bCl5kPh9EqKKDwna
PG1I7A==}
   engines: {node: '>=10'}
   dev: false
 /npm-run-path@4.0.1:
   resolution: {integrity: sha512-S48WzZW777zhNIrn7gxOlISNAqi9ZC/uQFnRdbeIHhZhCA6UqpkOT8T1G7BvfdgP4Er8gF4sUbaS0i7QvIfCWw==}
   engines: {node: '>=8'}
   dependencies:
     path-key: 3.1.1
   dev: true

 /npm-run-path@5.1.0:
   resolution: {integrity: sha512-sJOdmRGrY2sjNTRMbSvluQqg+8X7ZK61yvzBEIDhz4f8z1TZFYABsqjjCBd/0PUNE9M6QDgHJXQkGUEm7Q+l9Q==}
   engines: {node: ^12.20.0 || ^14.13.1 || >=16.0.0}
   dependencies:
     path-key: 4.0.0
   dev: true

 /number-to-bn@1.7.0:
   resolution: {integrity: sha512-wsJ9gfSz1/s4ZsJN01lyonwuxA1tml6X1yBDnfpMglypcBRFZZkus26EdPSlqS5GJfYddVZa22p3VNb3z5m5Ig==}
   engines: {node: '>=6.5.0', npm: '>=3'}
   dependencies:
     bn.js: 4.11.6
     strip-hex-prefix: 1.0.0
   dev: false

 /oauth-sign@0.9.0:
   resolution: {integrity: sha512-fexhUFFPTGV8ybAtSIGbV6gOkSv8UtRbDBnAyLQw4QPKkgNlsH2ByPGtMUqdWkos6YCRmAqViwgZrJc/mRDzZQ==}
   dev: false

 /object-assign@4.1.1:
   resolution: {integrity: sha512-rJgTQnkUnH1sFw8yT6VSU3zD3sWmu6sZhIseY8VX+GRu3P6F7Fu+JNDoXfklElbLJSnc3FUQHVe4cU5hj+BcUg==}
   engines: {node: '>=0.10.0'}

 /object-hash@3.0.0:
   resolution: {integrity: sha512-RSn9F68PjH9HqtltsSnqYC1XXoWe9Bju5+213R98cNGttag9q9yAOTzdbsqvIa7aNm5WffBZFpWYr2aWrklWAw==}
   engines: {node: '>= 6'}
   dev: false

 /object-inspect@1.12.2:
   resolution: {integrity: sha512-z+cPxW0QGUp0mcqcsgQyLVRDoXFQbXOwBaqyF7VIgI4TWNQsDHrBpUQslRmIfAoYWdYzs6UlKJtB2XJpTaNSpQ==}
 /object-inspect@1.13.1:
   resolution: {integrity: sha512-
5qoj1RUiKOMsCCNLV1CBiPYE10sziTsnmNxkAI/rZhiD63CF7IqdFGC/XzjWjpSgLf0LxXX3bDFIh0E18f6Uh
Q==}
   dev: true

 /object-keys@0.4.0:
   resolution: {integrity: sha512-
ncrLw+X55z7bkl5PnUvHwFK9FcGuFYo9gtjws2XtSzL+aZ8tm830P60WJ0dSmFVaSalWieW5MD7kEdnXda
9yJw==}
   dev: false

 /object-keys@1.1.1:
   resolution: {integrity: sha512-
NuAESUOUMrlIXOfHKzD6bpPu3tYt3xvjNdRIQ+FeT0lNb4K8WR70CaDxhuNguS2XG+GjkyMwOzsN5ZktIm
fhLA==}
   engines: {node: '>= 0.4'}
   dev: true

 /object.assign@4.1.4:
   resolution: {integrity: sha512-
1mxKf0e58bvyjSCtKYY4sRe9itRk3PJpquJOjeIkz885CczcI4IvJJDLPS72oowuSh+pBxUFROpX+TU++hxhZ
Q==}
   engines: {node: '>= 0.4'}
   dependencies:
     call-bind: 1.0.2
     define-properties: 1.2.1
     has-symbols: 1.0.3
     object-keys: 1.1.1
   dev: true

 /object.entries@1.1.7:
   resolution: {integrity: sha512-
jCBs/0plmPsOnrKAfFQXRG2NFjlhZgjjcBLSmTnEhU8U6vVTsVe8ANeQJCHTl3gSsI4J+0emOoCgoKlmQP
MgmA==}
   engines: {node: '>= 0.4'}
   dependencies:
     call-bind: 1.0.2
     define-properties: 1.2.1
     es-abstract: 1.22.3
   dev: true

 /object.fromentries@2.0.7:
   resolution: {integrity: sha512-
UPbPHML6sL8PI/mOqPwsH4G6iyXcCGzLin8KvEPenOZN5lpCNBZZQ+V62vdjB1mQHrmqGQt5/OJzemUA
+KJmEA==}
   engines: {node: '>= 0.4'}
   dependencies:
     call-bind: 1.0.2
     define-properties: 1.2.1
     es-abstract: 1.22.3
   dev: true
 /object.groupby@1.0.1:
   resolution: {integrity: sha512-HqaQtqLnp/8Bn4GL16cj+CUYbnpe1bh0TtEaWvybszDG4tgxCJuRpV8VGuvNaI1fAnI4lUJzDG55MXcOH4JZcQ==}
   dependencies:
     call-bind: 1.0.2
     define-properties: 1.2.1
     es-abstract: 1.22.3
     get-intrinsic: 1.2.2
   dev: true

 /object.hasown@1.1.3:
   resolution: {integrity: sha512-fFI4VcYpRHvSLXxP7yiZOMAd331cPfd2p7PFDVbgUsYOfCT3tICVqXWngbjr4m49OvsBwUBQ6O2uQoJvy3RexA==}
   dependencies:
     define-properties: 1.2.1
     es-abstract: 1.22.3
   dev: true

 /object.values@1.1.7:
   resolution: {integrity: sha512-aU6xnDFYT3x17e/f0IiiwlGPTy2jzMySGfUB4fq6z7CV8l85CWHDk5ErhyhpfDHhrOMwGFhSQkhMGHaIotA6Ng==}
   engines: {node: '>= 0.4'}
   dependencies:
     call-bind: 1.0.2
     define-properties: 1.2.1
     es-abstract: 1.22.3
   dev: true

 /obliterator@2.0.4:
   resolution: {integrity: sha512-lgHwxlxV1qIg1Eap7LgIeoBWIMFibOjbrYPIPJZcI1mmGAI2m3lNYpK12Y+GBdPQ0U1hRwSord7GIaawz962qQ==}
   dev: false

 /oboe@2.1.5:
   resolution: {integrity: sha512-zRFWiF+FoicxEs3jNI/WYUrVEgA7DeET/InK0XQuudGHRg8iIob3cNPrJTKaz4004uaA9Pbe+Dwa8iluhjLZWA==}
   dependencies:
     http-https: 1.0.0
   dev: false

 /on-exit-leak-free@0.2.0:
   resolution: {integrity: sha512-dqaz3u44QbRXQooZLTUKU41ZrzYrcvLISVgbrzbyCMxpmSLJvZ3ZamIJIZ29P6OhZIkNIQKosdeM6t1LYbA9hg==}
   dev: false
 /on-exit-leak-free@2.1.2:
   resolution: {integrity: sha512-
0eJJY6hXLGf1udHwfNftBqH+g73EU4B504nZeKpz1sYRKafAghwxEJunB2O7rDZkL4PGfsMVnTXZ2EjibbqcsA==}
   engines: {node: '>=14.0.0'}
   dev: false

 /on-finished@2.4.1:
   resolution: {integrity: sha512-
oVlzkg3ENAhCk2zdv7IJwd/QUD4z2RxRwpkcGY8psCVcCYZNq4wYnVWALHM+brtuJjePWiYF/ClmuDr8Ch5+kg==}
   engines: {node: '>= 0.8'}
   dependencies:
     ee-first: 1.1.1
   dev: false

 /once@1.4.0:
   resolution: {integrity: sha512-
lNaJgI+2Q5URQBkccEKHTQOPaXdUxnZZElQTZY0MFUAuaEqe1E+Nyvgdz/aIyNi6Z9MzO5dv1H8n58/GELp3+w==}
   dependencies:
     wrappy: 1.0.2

 /onetime@5.1.2:
   resolution: {integrity: sha512-
kbpaSSGJTWdAY5KPVeMOKXSrPtr8C8C7wodJbcsd51jRnmD+GZu8Y0VoU6Dm5Z4vWr0Ig/1NKuWRKf7j5aaYSg==}
   engines: {node: '>=6'}
   dependencies:
     mimic-fn: 2.1.0
   dev: true

 /onetime@6.0.0:
   resolution: {integrity: sha512-
1FlR+gjXK7X+AsAHso35MnyN5KqGwJRi/31ft6x0M194ht7S+rWAvd7PHss9xSKMzE0asv1pyIHaJYq+BbacAQ==}
   engines: {node: '>=12'}
   dependencies:
     mimic-fn: 4.0.0
   dev: true

 /open@9.1.0:
   resolution: {integrity: sha512-
OS+QTnw1/4vrf+9hh1jc1jnYjzSG4ttTBB8UxOwAnInG3Uo4ssetzC1ihqaIHjLJnA5GGlRl6QlZXOTQhRBUvg==}
   engines: {node: '>=14.16'}
   dependencies:
     default-browser: 4.0.0
     define-lazy-prop: 3.0.0
     is-inside-container: 1.0.0
     is-wsl: 2.2.0
   dev: true
 /optionator@0.9.3:
   resolution: {integrity: sha512-JjCoypp+jKn1ttEFExxhetCKeJt9zhAgAve5FXHixTvFDW/5aEktX9bufBKLRRMdU7bNtpLfcGu94B3cdEJgjg==}
   engines: {node: '>= 0.8.0'}
   dependencies:
     '@aashutoshrathi/word-wrap': 1.2.6
     deep-is: 0.1.4
     fast-levenshtein: 2.0.6
     levn: 0.4.1
     prelude-ls: 1.2.1
     type-check: 0.4.0
   dev: true

 /p-cancelable@2.1.1:
   resolution: {integrity: sha512-BZOr3nRQHOntUjTrH8+Lh54smKHoHyur8We1V8DSMVrl5A2malOOwuJRnKRDjSnkoeBh4at6BwEnb5I7Jl31wg==}
   engines: {node: '>=8'}
   dev: false

 /p-cancelable@3.0.0:
   resolution: {integrity: sha512-mlVgR3PGuzlo0MmTdk4cXqXWlwQDLnONTAg6sm62XkMJEiRxN3GL3SffkYvqwonbkJBcrI7Uvv5Zh9yjvn2iUw==}
   engines: {node: '>=12.20'}
   dev: false

 /p-limit@2.3.0:
   resolution: {integrity: sha512-//88mFWSJx8lxCzwdAABTJL2MyWB12+eIY7MDL2SqLmAkeKU9qxRvWuSyTjm3FUmpBEMuFfckAIqEaVGUDxb6w==}
   engines: {node: '>=6'}
   dependencies:
     p-try: 2.2.0

 /p-limit@3.1.0:
   resolution: {integrity: sha512-TYOanM3wGwNGsZN2cVTYPArw454xnXj5qmWF1bEoAc4+cU/ol7GVh7odevjp1FNHduHc3KZMcFduxU5Xc6uJRQ==}
   engines: {node: '>=10'}
   dependencies:
     yocto-queue: 0.1.0

 /p-locate@4.1.0:
   resolution: {integrity: sha512-R79ZZ/0wAxKGu3oYMlz8jy/kbhsNrS7SKZ7PxEHBgJ5+F2mtFW2fK2cOtBh1cHYkQsbzFV7I+EoRKe6Yt0oK7A==}
   engines: {node: '>=8'}
   dependencies:
     p-limit: 2.3.0
 /p-locate@5.0.0:
   resolution: {integrity: sha512-
LaNjtRWUBY++zB5nE/NwcaoMylSPk+S+ZHNB1TzdbMJMny6dynpAGt7X/tl/QYq3TIeE6nxHppbo2LGymrG
5Pw==}
   engines: {node: '>=10'}
   dependencies:
     p-limit: 3.1.0
   dev: true

 /p-try@2.2.0:
   resolution: {integrity: sha512-
R4nPAVTAU0B9D35/Gk3uJf/7XYbQcyohSKdvAxIRSNghFl4e71hVoGnBNQz9cWaXxO2I10KTC+3jMdvvoK
w6dQ==}
   engines: {node: '>=6'}

 /parent-module@1.0.1:
   resolution: {integrity: sha512-
GQ2EWRpQV8/o+Aw8YqtfZZPfNRWZYkbidE9k5rpl/hC3vtHHBfGm2Ifi6qWV+coDGkrUKZAxE3Lot5kcsRlh
+g==}
   engines: {node: '>=6'}
   dependencies:
     callsites: 3.1.0
   dev: true

 /parse-headers@2.0.5:
   resolution: {integrity: sha512-
ft3iAoLOB/MlwbNXgzy43SWGP6sQki2jQvAyBg/zDFAgr9bfNWZIUj42Kw2eJIl8kEi4PbgE6U1Zau/HwI75HA
==}
   dev: false

 /parse-json@5.2.0:
   resolution: {integrity: sha512-
ayCKvm/phCGxOkYRSCM82iDwct8/EonSEgCSxWxD7ve6jHggsFl4fZVQBPRNgQoKiuV/odhFrGzQXZwbif
C8Rg==}
   engines: {node: '>=8'}
   dependencies:
     '@babel/code-frame': 7.22.13
     error-ex: 1.3.2
     json-parse-even-better-errors: 2.3.1
     lines-and-columns: 1.2.4
   dev: true

 /parseurl@1.3.3:
   resolution: {integrity: sha512-
CiyeOxFT/JZyN5m0z9PfXw4SCBJ6Sygz1Dpl0wqjlhDEGGBP1GnsUVEL0p63hoG1fcj3fHynXi9NYO4nWOL
+qQ==}
   engines: {node: '>= 0.8'}
   dev: false

 /path-exists@4.0.0:
   resolution: {integrity: sha512-
ak9Qy5Q7jYb2Wwcey5Fpvg2KoAc/ZIhLSLOSBmRmygPsGwkVVt0fZa0qrtMz+m6tJTAHfZQ8FnmB4MG4L
Wy7/w==}
   engines: {node: '>=8'}

 /path-is-absolute@1.0.1:
   resolution: {integrity: sha512-
AVbw3UJ2e9bq64vSaS9Am0fje1Pa8pbGqTTsmXfaIiMpnr5DlDhfJOuLj9Sf95ZPVDAUerDfEk88MPmPe7U
CQg==}
   engines: {node: '>=0.10.0'}

 /path-key@3.1.1:
   resolution: {integrity: sha512-
ojmeN0qd+y0jszEtoY48r0Peq5dwMEkIlCOu6Q5f41lfkswXuKtYrhgoTpLnyIcHm24Uhqx+5Tqm2InSwLhE6Q
==}
   engines: {node: '>=8'}
   dev: true

 /path-key@4.0.0:
   resolution: {integrity: sha512-
haREypq7xkM7ErfgIyA0z+Bj4AGKlMSdlQE2jvJo6huWD1EdkKYV+G/T4nq0YEF2vgTT8kqMFKo1uHn950r4
SQ==}
   engines: {node: '>=12'}
   dev: true

 /path-parse@1.0.7:
   resolution: {integrity: sha512-
LDJzPVEEEPR+y48z93A0Ed0yXb8pAByGWo/k5YYdYgpY2/2EsOsksJrq7lOHxryrVOn1ejG6oAp8ahvOIQD
8sw==}

 /path-to-regexp@0.1.7:
   resolution: {integrity: sha512-
5DFkuoqlv1uYQKxy8omFBeJPQcdoE07Kv2sferDCrAq1ohOU+MSDswDIbnx3YAM60qIOnYa53wBhXW0E
bMonrQ==}
   dev: false

 /path-type@4.0.0:
   resolution: {integrity: sha512-
gDKb8aZMDeD/tZWs9P6+q0J9Mwkdl6xMV8TjnGP3qJVJ06bdMgkbBlLU8IdfOsIsFz2BW1rNVT3XuNEl8zP
Avw==}
   engines: {node: '>=8'}
   dev: true

 /pbkdf2@3.1.2:
   resolution: {integrity: sha512-
iuh7L6jA7JEGu2WxDwtQP1ddOpaJNC4KlDEFfdQajSGgGPNi4OyDc2R7QnbY2bR9QjBVGwgvTdNJZoE7
RaxUMA==}
   engines: {node: '>=0.12'}
   dependencies:
     create-hash: 1.2.0
     create-hmac: 1.1.7
     ripemd160: 2.0.2
     safe-buffer: 5.2.1
     sha.js: 2.4.11
   dev: false

 /performance-now@2.1.0:
   resolution: {integrity: sha512-
7EAHlyLHI56VEIdK57uwHdHKIaAGbnXPiw0yWbarQZOKaKpvUIgW0jWRVLiatnM+XXlSwsanIBH/hzGMJul
Mow==}
   dev: false

 /picocolors@1.0.0:
   resolution: {integrity: sha512-
1fygroTLlHu66zi26VoTDv8yRgm0Fccecssto+MhsZ0D/DGW2sm8E8AjW7NU5VVTRt5GxbeZ5qBuJr+HyLY
kjQ==}

 /picomatch@2.3.1:
   resolution: {integrity: sha512-
JU3teHTNjmE2VCGFzuY8EXzCDVwEqB2a8fsIvwaStHhAWJEeVd1o1QD80CU6+ZdEXXSLbSsuLwJjkCB
WqRQUVA==}
   engines: {node: '>=8.6'}

 /pify@2.3.0:
   resolution: {integrity: sha512-
udgsAY+fTnvv7kI7aaxbqwWNb0AHiB0qBO89PZKPkoTmGOgdbrHDKD+0B2X4uTfJ/FT1R09r9gTsjUjNJotu
og==}
   engines: {node: '>=0.10.0'}
   dev: false

 /pify@3.0.0:
   resolution: {integrity: sha512-
C3FsVNH1udSEX48gGX1xfvwTWfsYWj5U+8/uK15BGzIGrKoUpghX8hWZwa/OFnakBiiVNmBvemTJR5mc
y7iPcg==}
   engines: {node: '>=4'}
   dev: false

 /pify@5.0.0:
   resolution: {integrity: sha512-
eW/gHNMlxdSP6dmG6uJip6FXN0EQBwm2clYYd8Wul42Cwu/DK8HEftzsapcNdYe2MfLiIwZqsDk2RDEsTE
79hA==}
   engines: {node: '>=10'}
   dev: false

 /pino-abstract-transport@0.5.0:
   resolution: {integrity: sha512-
+KAgmVeqXYbTtU2FScx1XS3kNyfZ5TrXY07V96QnUSFqo2gAqlvmaxH67Lj7SWazqsMabf+58ctdTcBgnOL
UOQ==}
   dependencies:
     duplexify: 4.1.2
     split2: 4.2.0
   dev: false

 /pino-abstract-transport@1.1.0:
   resolution: {integrity: sha512-
lsleG3/2a/JIWUtf9Q5gUNErBqwIu1tUKTT3dUzaf5DySw9ra1wcqKjJjLX1VTY64Wk1eEOYsVGSaGfCK85ek
A==}
   dependencies:
     readable-stream: 4.4.2
     split2: 4.2.0
   dev: false

 /pino-std-serializers@4.0.0:
   resolution: {integrity: sha512-
cK0pekc1Kjy5w9V2/n+8MkZwusa6EyyxfeQCB799CQRhRt/CqYKiWs5adeu8Shve2ZNffvfC/7J64A2PJo1W/
Q==}
   dev: false

 /pino-std-serializers@6.2.2:
   resolution: {integrity: sha512-
cHjPPsE+vhj/tnhCy/wiMh3M3z3h/j15zHQX+S9GkTBgqJuTuJzYJ4gUyACLhDaJ7kk9ba9iRDmbH2tJU03Oi
A==}
   dev: false

 /pino@7.11.0:
   resolution: {integrity: sha512-
dMACeu63HtRLmCG8VKdy4cShCPKaYDR4youZqoSWLxl5Gu99HUw8bw75thbPv9Nip+H+QYX8o3ZJbTd
VZZ2TVg==}
   hasBin: true
   dependencies:
     atomic-sleep: 1.0.0
     fast-redact: 3.3.0
     on-exit-leak-free: 0.2.0
     pino-abstract-transport: 0.5.0
     pino-std-serializers: 4.0.0
     process-warning: 1.0.0
     quick-format-unescaped: 4.0.4
     real-require: 0.1.0
     safe-stable-stringify: 2.4.3
     sonic-boom: 2.8.0
     thread-stream: 0.15.2
   dev: false

 /pino@8.16.1:
   resolution: {integrity: sha512-
3bKsVhBmgPjGV9pyn4fO/8RtoVDR8ssW1ev819FsRXlRNgW8gR/9Kx+gCK4UPWd4JjrRDLWpzd/pb1AyW
m3MGA==}
   hasBin: true
   dependencies:
     atomic-sleep: 1.0.0
     fast-redact: 3.3.0
     on-exit-leak-free: 2.1.2
     pino-abstract-transport: 1.1.0
     pino-std-serializers: 6.2.2
     process-warning: 2.2.0
     quick-format-unescaped: 4.0.4
     real-require: 0.2.0
     safe-stable-stringify: 2.4.3
     sonic-boom: 3.7.0
     thread-stream: 2.4.1
   dev: false

 /pirates@4.0.6:
   resolution: {integrity: sha512-
saLsH7WeYYPiD25LDuLRRY/i+6HaPYr6G1OUlN39otzkSTxKnubR9RTxS3/Kk50s1g2JTgFwWQDQyplC5/
SHZg==}
   engines: {node: '>= 6'}
   dev: false

 /pluralize@8.0.0:
   resolution: {integrity: sha512-
Nc3IT5yHzflTfbjgqWcCPpo7DaKy4FnpB0l/zCAW0Tc7jxAiuqSxHasntB3D7887LSrA93kDJ9IXovxJYxyLCA==
}
   engines: {node: '>=4'}
   dev: true

 /pngjs@5.0.0:
   resolution: {integrity: sha512-
40QW5YalBNfQo5yRYmiw7Yz6TKKVr3h6970B2YE+3fQpsWcrbj1PzJgxeJ19DRQjhMbKPIuMY8rFaXc8mo
olVw==}
   engines: {node: '>=10.13.0'}
   dev: false

 /postcss-import@15.1.0(postcss@8.4.31):
   resolution: {integrity: sha512-
hpr+J05B2FVYUAXHeK1YyI267J/dDDhMU6B6civm8hSY1jYJnBXxzKDKDswzJmtLHryrjhnDjqqp/49t8FALe
w==}
   engines: {node: '>=14.0.0'}
   peerDependencies:
     postcss: ^8.0.0
   dependencies:
     postcss: 8.4.31
     postcss-value-parser: 4.2.0
     read-cache: 1.0.0
     resolve: 1.22.8
   dev: false

 /postcss-js@4.0.1(postcss@8.4.31):
   resolution: {integrity: sha512-
dDLF8pEO191hJMtlHFPRa8xsizHaM82MLfNkUHdUtVEV3tgTp5oj+8qbEqYM57SLfc74KSbw//4SeJma2LR
VIw==}
   engines: {node: ^12 || ^14 || >= 16}
   peerDependencies:
     postcss: ^8.4.21
   dependencies:
     camelcase-css: 2.0.1
     postcss: 8.4.31
   dev: false
 /postcss-load-config@4.0.1(postcss@8.4.31)(ts-node@10.9.1):
   resolution: {integrity: sha512-vEJIc8RdiBRu3oRAI0ymerOn+7rPuMvRXslTvZUKZonDHFIczxztIyJ1urxM1x9JXEikvpWWTUUqal5j/8QgvA==}
   engines: {node: '>= 14'}
   peerDependencies:
     postcss: '>=8.0.9'
     ts-node: '>=9.0.0'
   peerDependenciesMeta:
     postcss:
       optional: true
     ts-node:
       optional: true
   dependencies:
     lilconfig: 2.1.0
     postcss: 8.4.31
     ts-node: 10.9.1(@types/node@20.8.9)(typescript@5.2.2)
     yaml: 2.3.3
   dev: false

 /postcss-nested@6.0.1(postcss@8.4.31):
   resolution: {integrity: sha512-mEp4xPMi5bSWiMbsgoPfcP74lsWLHkQbZc3sY+jWYd65CUwXrUaTp0fmNpa01ZcETKlIgUdFN/MpS2xZtqL9dQ==}
   engines: {node: '>=12.0'}
   peerDependencies:
     postcss: ^8.2.14
   dependencies:
     postcss: 8.4.31
     postcss-selector-parser: 6.0.13
   dev: false

 /postcss-nesting@12.0.1(postcss@8.4.31):
   resolution: {integrity: sha512-6LCqCWP9pqwXw/njMvNK0hGY44Fxc4B2EsGbn6xDcxbNRzP8GYoxT7yabVVMLrX3quqOJ9hg2jYMsnkedOf8pA==}
   engines: {node: ^14 || ^16 || >=18}
   peerDependencies:
     postcss: ^8.4
   dependencies:
     '@csstools/selector-specificity': 3.0.0(postcss-selector-parser@6.0.13)
     postcss: 8.4.31
     postcss-selector-parser: 6.0.13
   dev: false

 /postcss-selector-parser@6.0.13:
   resolution: {integrity: sha512-EaV1Gl4mUEV4ddhDnv/xtj7sxwrwxdetHdWUGnT4VJQf+4d05v6lHYZr8N573k5Z0BViss7BDhfWtKS3+sfAqQ==}
   engines: {node: '>=4'}
   dependencies:
     cssesc: 3.0.0
     util-deprecate: 1.0.2
   dev: false

 /postcss-value-parser@4.2.0:
   resolution: {integrity: sha512-
1NNCs6uurfkVbeXG4S8JFT9t19m45ICnif8zWLd5oPSZ50QnwMfK+H3jv408d4jw/7Bttv5axS5IiHoLaVNHeQ
==}
   dev: false

 /postcss@8.4.31:
   resolution: {integrity: sha512-
PS08Iboia9mts/2ygV3eLpY5ghnUcfLV/EXTOW1E2qYxJKGGBUtNjN76FYHnMs36RmARn41bC0AZmn+rR
0OVpQ==}
   engines: {node: ^10 || ^12 || >=14}
   dependencies:
     nanoid: 3.3.6
     picocolors: 1.0.0
     source-map-js: 1.0.2

 /preact@10.11.3:
   resolution: {integrity: sha512-
eY93IVpod/zG3uMF22Unl8h9KkrcKIRs2EGar8hwLZZDU1lkjph303V9HZBwufh2s736U6VXuhD109LYqPoff
g==}
   dev: false

 /preact@10.18.1:
   resolution: {integrity: sha512-
mKUD7RRkQQM6s7Rkmi7IFkoEHjuFqRQUaXamO61E6Nn7vqF/bo7EZCmSyrUnp2UWHw0O7XjZ2eeXis+
m7tf4lg==}
   dev: false

 /precond@0.2.3:
   resolution: {integrity: sha512-
QCYG84SgGyGzqJ/vlMsxeXd/pgL/I94ixdNFyh1PusWmTCyVfPJjZ1K1jvHtsbfnXQs2TSkEP2fR7QiMZAnKF
Q==}
   engines: {node: '>= 0.6'}
   dev: false

 /prelude-ls@1.2.1:
   resolution: {integrity: sha512-
vkcDPrRZo1QZLbn5RLGPpg/WmIQ65qoWWhcGKf/b5eplkkarX0m9z8ppCat4mlOqUsWpyNuYgO3VRyrYH
SzX5g==}
   engines: {node: '>= 0.8.0'}
   dev: true

 /prettier-linter-helpers@1.0.0:
   resolution: {integrity: sha512-
GbK2cP9nraSSUF9N2XwUwqfzlAFlMNYYl+ShE/V+H8a9uNl/oUqB1w2EL54Jh0OlyRSd8RfWYJ3coVS4TR
OP2w==}
   engines: {node: '>=6.0.0'}
   dependencies:
     fast-diff: 1.3.0
   dev: true

 /prettier@2.8.8:
   resolution: {integrity: sha512-tdN8qQGvNjw4CHbY+XXk0JgCXn9QiF21a55rBe5LJAU+kDyC4WQn4+awm2Xfk2lQMk5fKup9XgzTZtGkjBdP9Q==}
   engines: {node: '>=10.13.0'}
   hasBin: true

 /prettier@3.0.3:
   resolution: {integrity: sha512-L/4pUDMxcNa8R/EthV08Zt42WBO4h1rarVtK0K+QJG0X187OLo7l699jWw0GKuwzkPQ//jMFA/8Xm6Fh3J/DAg==}
   engines: {node: '>=14'}
   hasBin: true
   dev: true

 /process-nextick-args@2.0.1:
   resolution: {integrity: sha512-3ouUOpQhtgrbOa17J7+uxOTpITYWaGP7/AhoR3+A+/1e9skrzelGi/dXzEYyvbxubEF6Wn2ypscTKiKJFFn1ag==}
   dev: false

 /process-warning@1.0.0:
   resolution: {integrity: sha512-du4wfLyj4yCZq1VupnVSZmRsPJsNuxoDQFdCFHLaYiEbFBD7QE0a+I4D7hOxrVnh78QE/YipFAj9lXHiXocV+Q==}
   dev: false

 /process-warning@2.2.0:
   resolution: {integrity: sha512-/1WZ8+VQjR6avWOgHeEPd7SDQmFQ1B5mC1eRXsCm5TarlNmx/wCsa5GEaxGm05BORRtyG/Ex/3xq3TuRvq57qg==}
   dev: false

 /process@0.11.10:
   resolution: {integrity: sha512-cdGef/drWFoydD1JsMzuFf8100nZl+GT+yacc2bEced5f9Rjk4z+WtFUTBu9PhOi9j/jfmBPu0mMEY4wIdAF8A==}
   engines: {node: '>= 0.6.0'}
   dev: false

 /promise-to-callback@1.0.0:
   resolution: {integrity: sha512-uhMIZmKM5ZteDMfLgJnoSq9GCwsNKrYau73Awf1jIy6/eUcuuZ3P+CD9zUv0kJsIUbU+x6uLNIhXhLHDs1pNPA==}
   engines: {node: '>=0.10.0'}
   dependencies:
     is-fn: 1.0.0
     set-immediate-shim: 1.0.1
   dev: false
```

```
 /promise@7.3.1:
   resolution: {integrity: sha512-
nolQXZ/4L+bP/UGlkfaIujX9BKxGwmQ9OT4mOt5yvy8iK1h3wqTEJCijzGANTCCl9nWjY41juyAn2K3Q1hLLT
g==}
   dependencies:
     asap: 2.0.6
   dev: false

 /prop-types@15.8.1:
   resolution: {integrity: sha512-
oj87CgZICdulUohogVAR7AjlC0327U4el4L6eAvOqCeudMDVU0NThNaV+b9Df4dXgSP1gXMTnPdhfe/2qDH
5cg==}
   dependencies:
     loose-envify: 1.4.0
     object-assign: 4.1.1
     react-is: 16.13.1
   dev: true

 /proxy-addr@2.0.7:
   resolution: {integrity: sha512-
lIQsMLSUDUPT44jdrU/O37qlnifitDP+ZwrmmZcoSKyLKvtZxpyV0n2/bD/N4tBAAZ/gJEdZU7KMraoK1+XYAg
==}
   engines: {node: '>= 0.10'}
   dependencies:
     forwarded: 0.2.0
     ipaddr.js: 1.9.1
   dev: false

 /proxy-compare@2.5.1:
   resolution: {integrity: sha512-
oyfc0Tx87Cpwva5ZXezSp5V9vht1c7dZBhvuV/y3ctkgMVUmiAGDVeeB0dKhGSyT0v1ZTEQYpe/RXlBVBNu
CLA==}
   dev: false

 /proxy-from-env@1.1.0:
   resolution: {integrity: sha512-
D+zkORCbA9f1tdWRK0RaCR3GPv50cMxcrz4X8k5LTSUD1Dkw47mKJEZQNunItRTkWwgtaUSo1RVFRIG
9ZXiFYg==}
   dev: true

 /prr@1.0.1:
   resolution: {integrity: sha512-
yPw4Sng1gWghHQWj0B3ZggWUm4qVbPwPFcRG8KyxiU7J2OHFSoEHKS+EZ3fv5l1t9CyCiop6l/ZYeWbrg
oQejw==}
   dev: false

 /psl@1.9.0:
   resolution: {integrity: sha512-
E/ZsdU4HLs/68gYzgGTkMicWTLPdAftJLfJFlLUAAKZGkStNU72sZjT66SnMDVOfOWY/YAoiD7Jxa9iHvngc
ag==}
   dev: false
```

```
  /pstree.remy@1.1.8:
    resolution: {integrity: sha512-
77DZwxQmxKnu3aR542U+X8FypNzbfJ+C5XQDk3uWjWxn6151aIMGthWYRXTqT1E5oJvg+ljaa2OJi+VfvC
OQ8w==}
    dev: true

  /pump@3.0.0:
    resolution: {integrity: sha512-
LwZy+p3SFs1Pytd/jYct4wpv49HiYCqd9Rlc5ZVdk0V+8Yzv6jR5Blk3TRmPL1ft69TxP0IMZGJ+WPFU2BFhw
w==}
    dependencies:
      end-of-stream: 1.4.4
      once: 1.4.0
    dev: false

  /punycode@2.1.0:
    resolution: {integrity: sha512-
Yxz2kRwT90aPiWEMHVYnEf4+rhwF1tBmmZ4KepCP+Wkium9JxtWnUm1nqGwpiAHr/tnTSeHqr3wb++jgSk
XjhA==}
    engines: {node: '>=6'}
    dev: false

  /punycode@2.3.0:
    resolution: {integrity: sha512-
rRV+zQD8tVFys26lAGR9WUuS4iUAngJScM+ZRSKtvl5tKeZ2t5bvdNFdNHBW9FWR4guGHlgmsZ1G7BSm
2wTbuA==}
    engines: {node: '>=6'}

  /pure-color@1.3.0:
    resolution: {integrity: sha512-
QFADYnsVoBMw1srW7OVKEYjG+MbIa49s54w1MA1EDY6r2r/sTcKKYqRX1f4GYvnXP7eN/Pe9HFcX+hwz
mrXRHA==}
    dev: false

  /qrcode@1.5.1:
    resolution: {integrity: sha512-
nS8NJ1Z3md8uTjKtP+SGGhfqmTCs5flU/xR623oI0JX+Wepz9R8UrRVCTBTJm3qGw3rH6jJ6MUHjkDx15cx
SSg==}
    engines: {node: '>=10.13.0'}
    hasBin: true
    dependencies:
      dijkstrajs: 1.0.2
      encode-utf8: 1.0.3
      pngjs: 5.0.0
      yargs: 15.4.1
    dev: false

  /qrcode@1.5.3:
    resolution: {integrity: sha512-
puyri6ApkEHYiVl4CFzo1tDkAZ+ATcnbJrJ6RiBM1Fhctdn/ix9MTE3hRph33omisEbC/2fcfemsseiKgBPKZg==
}
    engines: {node: '>=10.13.0'}
    hasBin: true
    dependencies:
      dijkstrajs: 1.0.2
      encode-utf8: 1.0.3
      pngjs: 5.0.0
      yargs: 15.4.1
    dev: false

  /qs@6.11.0:
    resolution: {integrity: sha512-
MvjoMCJwEarSbUYk5O+nmoSzSutSsTwF85zcHPQ9OrlFoZOYIjaqBAJIqIXjptyD5vThxGq52Xu/MaJzRkIk4
Q==}
    engines: {node: '>=0.6'}
    dependencies:
      side-channel: 1.0.4
    dev: false

  /qs@6.5.3:
    resolution: {integrity: sha512-
qxXIEh4pCGfHICj1mAJQ2/2XVZkjCDTcEgfoSQxc/fYivUZxTkk7L3bDBJSoNrEzXI17oUO5Dp07ktqE5KzczA
==}
    engines: {node: '>=0.6'}
    dev: false

  /query-string@5.1.1:
    resolution: {integrity: sha512-
gjWOsm2SoGlgLEdAGt7a6slVOk9mGiXmPFMqrEhLQ68rhQuBnpfs3+EmlvqKyxnCo9/PPlF+9MtY02S1aFg
+Jw==}
    engines: {node: '>=0.10.0'}
    dependencies:
      decode-uri-component: 0.2.2
      object-assign: 4.1.1
      strict-uri-encode: 1.1.0
    dev: false

  /query-string@6.13.5:
    resolution: {integrity: sha512-
svk3xg9qHR39P3JlHuD7g3nRnyay5mHbrPctEBDUxUkHRifPHXJDhBUycdCC0NBjXoDf44Gb+IsOZL1Uwn
8M/Q==}
    engines: {node: '>=6'}
    dependencies:
      decode-uri-component: 0.2.0
      split-on-first: 1.1.0
      strict-uri-encode: 2.0.0
    dev: false

  /query-string@7.1.3:
    resolution: {integrity: sha512-
hh2WYhq4fi8+b+/2Kg9CEge4fDPvHS534aOOvOZeQ3+Vf2mCFsaFBYj0i+iXcAq6I9Vzp5fjMFBlONvayDC1
qg==}
    engines: {node: '>=6'}
    dependencies:
      decode-uri-component: 0.2.2
      filter-obj: 1.1.0
      split-on-first: 1.1.0
      strict-uri-encode: 2.0.0
    dev: false

  /queue-microtask@1.2.3:
    resolution: {integrity: sha512-
NuaNSa6flKT5JaSYQzJok04JzTL1CA6aGhv5rfLW3PgqA+M2ChpZQnAC8h8i4ZFkBS8X5RqkDBHA7r4hej
3K9A==}

  /quick-format-unescaped@4.0.4:
    resolution: {integrity: sha512-
tYC1Q1hgyRuHgloV/YXs2w15unPVh8qfu/qCTfhTYamaw7fyhumKa2yGpdSo87vY32rIclj+4fWYQXUMs9EH
vg==}
    dev: false

  /quick-lru@5.1.1:
    resolution: {integrity: sha512-
WuyALRjWPDGtt/wzJiadO5AXY+8hZ80hVpe6MyivgraREW751X3SbhRvG3eLKOYN+8VEvqLcf3wdnt44Z4
S4SA==}
    engines: {node: '>=10'}
    dev: false

  /randombytes@2.1.0:
    resolution: {integrity: sha512-
vYl3iOX+4CKUWuxGi9Ukhie6fsqXqS9FE2Zaic4tNFD2N2QQaXOMFbuKK4QmDHC0JO6B1Zp41J0LpT0o
R68amQ==}
    dependencies:
      safe-buffer: 5.2.1
    dev: false

  /range-parser@1.2.1:
    resolution: {integrity: sha512-
Hrgsx+orqoygnmhFbKaHE6c296J+HTAQXoxEF6gNupROmmGJRoyzfG3ccAveqCBrwr/2yxQ5BVd/GTl5ag
OwSg==}
    engines: {node: '>= 0.6'}
    dev: false

  /raw-body@2.5.1:
    resolution: {integrity: sha512-
qqJBtEyVgS0ZmPGdCFPWJ3FreoqvG4MVQln/kCgF7Olq95lbOp0/BWyMwbdtn4VTvkM8Y7khCQ2Xgk/tcr
CXig==}
    engines: {node: '>= 0.8'}
    dependencies:
      bytes: 3.1.2
      http-errors: 2.0.0
      iconv-lite: 0.4.24
      unpipe: 1.0.0
    dev: false

  /raw-body@2.5.2:
    resolution: {integrity: sha512-
8zGqypfENjCIqGhgXToC8aB2r7YrBX+AQAfIPs/Mlk+BtPTztOvTS01NRW/3Eh60J+a48lt8qsCzirQ6loCVfA==
}
    engines: {node: '>= 0.8'}
    dependencies:
      bytes: 3.1.2
      http-errors: 2.0.0
      iconv-lite: 0.4.24
      unpipe: 1.0.0
    dev: false

  /react-base16-styling@0.6.0:
    resolution: {integrity: sha512-
yvh/7CArceR/jNATXOKDlvTnPKPmGZz7zsenQ3jUwLzHkNUR0CvY3yGYJbWJ/nnxsL8Sgmt5cO3/SILVuP
O6TQ==}
    dependencies:
      base16: 1.0.0
      lodash.curry: 4.1.1
      lodash.flow: 3.5.0
      pure-color: 1.3.0
    dev: false

  /react-dom@18.2.0(react@18.2.0):
    resolution: {integrity: sha512-
6IMTriUmvsjHUjNtEDudZfuDQUoWXVxKHhlEGSk81n4YFS+r/Kl99wXiwlVXtPBtJenozv2P+hxDsw9eA7Xo6
g==}
    peerDependencies:
      react: ^18.2.0
    dependencies:
      loose-envify: 1.4.0
      react: 18.2.0
      scheduler: 0.23.0
    dev: false

  /react-is@16.13.1:
    resolution: {integrity: sha512-
24e6ynE2H+OKt4kqsOvNd8kBpV65zoxbA4BVsEOB3ARVWQki/DHzaUoC5KuON/BiccDaCCTZBuOcfZs70
kR8bQ==}
    dev: true

  /react-json-view@1.21.3(@types/react@18.0.25)(react-dom@18.2.0)(react@18.2.0):
    resolution: {integrity: sha512-
13p8IREj9/x/Ye4WI/JpjhoIwuzEgUAtgJZNBJckfzJt1qyh24BdTm6UQNGnyTq9dapQdrqvquZTo3dz1X6Cjw==
}
    peerDependencies:
      react: ^17.0.0 || ^16.3.0 || ^15.5.4
      react-dom: ^17.0.0 || ^16.3.0 || ^15.5.4
    dependencies:
      flux: 4.0.4(react@18.2.0)
      react: 18.2.0
      react-base16-styling: 0.6.0
      react-dom: 18.2.0(react@18.2.0)
      react-lifecycles-compat: 3.0.4
      react-textarea-autosize: 8.5.3(@types/react@18.0.25)(react@18.2.0)
    transitivePeerDependencies:
      - '@types/react'
      - encoding
    dev: false

  /react-lifecycles-compat@3.0.4:
    resolution: {integrity: sha512-
fBASbA6LnOU9dOU2eW7aQ8xmYBSXUIWr+UmF9b1efZBazGNO+rcXT/icdKnYm2pTwcRylVUYwW7H1P
HfLekVzA==}
    dev: false

  /react-refresh@0.14.0:
    resolution: {integrity: sha512-
wViHqhAd8OHeLS/IRMJjTSDHF3U9eWi62F/MledQGPdJGDhodXJ9PBLNGr6WWL7qlH12Mt3TyTpbS+hG
XMjCzQ==}
    engines: {node: '>=0.10.0'}
    dev: true

  /react-textarea-autosize@8.5.3(@types/react@18.0.25)(react@18.2.0):
    resolution: {integrity: sha512-
XT1024o2pqCuZSuBt9FwHlaDeNtVrtCXu0Rnz88t1jUGheCLa3PhjE1GH8Ctm2axEtvdCl5SUHYschyQ0L5Q
HQ==}
    engines: {node: '>=10'}
    peerDependencies:
      react: ^16.8.0 || ^17.0.0 || ^18.0.0
    dependencies:
      '@babel/runtime': 7.23.2
      react: 18.2.0
      use-composed-ref: 1.3.0(react@18.2.0)
      use-latest: 1.2.1(@types/react@18.0.25)(react@18.2.0)
    transitivePeerDependencies:
      - '@types/react'
    dev: false

  /react@18.2.0:
    resolution: {integrity: sha512-
/3IjMdb2L9QbBdWiW5e3P2/npwMBaU9mHCSCUzNln0ZCYbcfTsGbTJrU/kGemdH2IWmB2ioZ+zkxtmq6g0
9fGQ==}
    engines: {node: '>=0.10.0'}
    dependencies:
      loose-envify: 1.4.0
    dev: false

  /read-cache@1.0.0:
    resolution: {integrity: sha512-
Owdv/Ft7IjOgm/i0xvNDZ1LrRANRfew4b2prF3OWMQLxLfu3bS8FVhCsrSCMK4lR56Y9ya+AThoTpDCTxC
mpRA==}
    dependencies:
      pify: 2.3.0
    dev: false
  /read-pkg-up@7.0.1:
    resolution: {integrity: sha512-
zK0TB7Xd6JpCLmlLmufqykGE+/TlOePD6qKClNW7hHDKFh/J7/7gCWGR7joEQEW1bKq3a3yUZSObOoW
LFQ4ohg==}
    engines: {node: '>=8'}
    dependencies:
      find-up: 4.1.0
      read-pkg: 5.2.0
      type-fest: 0.8.1
    dev: true

  /read-pkg@5.2.0:
    resolution: {integrity: sha512-
Ug69mNOpfvKDAc2Q8DRpMjjzdtrnv9HcSMX+4VsZxD1aZ6ZzrIE7rlzXBtWTyhULSMKg076AW6WR5iZpD0
JiOg==}
    engines: {node: '>=8'}
    dependencies:
      '@types/normalize-package-data': 2.4.3
      normalize-package-data: 2.5.0
      parse-json: 5.2.0
      type-fest: 0.6.0
    dev: true

  /readable-stream@1.0.34:
    resolution: {integrity: sha512-
ok1qVCJuRkNmvebYikljxJA/UEsKwLl2nI1OmaqAu4/UE+h0wKCHok4XkL/gvi39OacXvw59RJUOFUkDib2r
Hg==}
    dependencies:
      core-util-is: 1.0.3
      inherits: 2.0.4
      isarray: 0.0.1
      string_decoder: 0.10.31
    dev: false

  /readable-stream@1.1.14:
    resolution: {integrity: sha512-
+MeVjFf4L44XUkhM1eYbD8fyEsxcV81pqMSR5gblfcLCHfZvbrqy4/qYHE+/R5HoBUT11WV5O08Cr1n3YXk
WVQ==}
    dependencies:
      core-util-is: 1.0.3
      inherits: 2.0.4
      isarray: 0.0.1
      string_decoder: 0.10.31
    dev: false

  /readable-stream@2.3.8:
    resolution: {integrity: sha512-
8p0AUk4XODgIewSi0l8Epjs+EVnWiK7NoDIEGU0HhE7+ZyY8D1IMY7odu5lRrFXGg71L15KG8QrPmum45
RTtdA==}
    dependencies:
      core-util-is: 1.0.3
      inherits: 2.0.4
      isarray: 1.0.0
      process-nextick-args: 2.0.1
      safe-buffer: 5.1.2
      string_decoder: 1.1.1
      util-deprecate: 1.0.2
    dev: false

  /readable-stream@3.6.0:
    resolution: {integrity: sha512-BViHy7LKeTz4oNnkcLJ+lVSL6vpiFeX6/d3oSH8zCW7UxP2onchk+vTGB143xuFjHS3deTgkKoXXymXqymiIdA==}
    engines: {node: '>= 6'}
    dependencies:
      inherits: 2.0.4
      string_decoder: 1.3.0
      util-deprecate: 1.0.2
    dev: false

  /readable-stream@4.4.2:
    resolution: {integrity: sha512-Lk/fICSyIhodxy1IDK2HazkeGjSmezAWX2egdtJnYhtzKEsBPJowlI6F6LPb5tqIQILrMbx22S5o3GuJavPusA==}
    engines: {node: ^12.22.0 || ^14.17.0 || >=16.0.0}
    dependencies:
      abort-controller: 3.0.0
      buffer: 6.0.3
      events: 3.3.0
      process: 0.11.10
      string_decoder: 1.3.0
    dev: false

  /readdirp@3.6.0:
    resolution: {integrity: sha512-hOS089on8RduqdbhvQ5Z37A0ESjsqz6qnRcffsMU3495FuTdqSm+7bhJ29JvIOsBDEEnan5DPu9t3To9VRlMzA==}
    engines: {node: '>=8.10.0'}
    dependencies:
      picomatch: 2.3.1

  /real-require@0.1.0:
    resolution: {integrity: sha512-r/H9MzAWtrv8aSVjPCMFpDMl5q66GqtmmRkRjpHTsp4zBAa+snZyiQNlMONiUmEJcsnaw0wCauJ2GWODr/aFkg==}
    engines: {node: '>= 12.13.0'}
    dev: false

  /real-require@0.2.0:
    resolution: {integrity: sha512-57frrGM/OCTLqLOAh0mhVA9VBMHd+9U7Zb2THMGdBUoZVOtGbJzjxsYGDJ3A9AYYCP4hn6y1TVbaOfzWtm5GFg==}
    engines: {node: '>= 12.13.0'}
    dev: false

  /reduce-flatten@2.0.0:
    resolution: {integrity: sha512-
EJ4UNY/U1t2P/2k6oqotuX2Cc3T6nxJwsM0N0asT7dhrtH1ltUxDn4NalSYmPE2rCkVpcf/X6R0wDwcFpzhd4
w==}
    engines: {node: '>=6'}

  /reflect.getprototypeof@1.0.4:
    resolution: {integrity: sha512-
ECkTw8TmJwW60lOTR+ZkODISW6RQ8+2CL3COqtiJKLd6MmB45hN51HprHFziKLGkAuTGQhBb91V8cy
+KHlaCjw==}
    engines: {node: '>= 0.4'}
    dependencies:
      call-bind: 1.0.2
      define-properties: 1.2.1
      es-abstract: 1.22.3
      get-intrinsic: 1.2.2
      globalthis: 1.0.3
      which-builtin-type: 1.1.3
    dev: true

  /regenerator-runtime@0.14.0:
    resolution: {integrity: sha512-
srw17NI0TUWHuGa5CFGGmhfNIeja30WMBfbslPNhf6JrqQlLN5gcrvig1oqPxiVaXb0oW0XRKtH6Nngs5lKCI
A==}

  /regexp-tree@0.1.27:
    resolution: {integrity: sha512-
iETxpjK6YoRWJG5o6hXLwvjYAoW+FEZn9os0PD/b6AP6xQwsa/Y7lCVgIixBbUPMfhu+i2LtdeAqVTgGlQarf
A==}
    hasBin: true
    dev: true

  /regexp.prototype.flags@1.5.1:
    resolution: {integrity: sha512-
sy6TXMN+hnP/wMy+ISxg3krXx7BAtWVO4UouuCN/ziM9UEne0euamVNafDfvC83bRNr95y0V5iijeDQFUNp
vrg==}
    engines: {node: '>= 0.4'}
    dependencies:
      call-bind: 1.0.2
      define-properties: 1.2.1
      set-function-name: 2.0.1
    dev: true

  /regjsparser@0.10.0:
    resolution: {integrity: sha512-
qx+xQGZVsy55CH0a1hiVwHmqjLryfh7wQyF5HO07XJ9f7dQMY/gPQHhlyDkIzJKC+x2fUCpCcUODUUUFr
m7SHA==}
    hasBin: true
    dependencies:
      jsesc: 0.5.0
    dev: true

  /request@2.88.2:
    resolution: {integrity: sha512-
MsvtOrfG9ZcrOwAW+Qi+F6HbD0CWXEh9ou77uOb7FM2WPhwT7smM833PzanhJLsgXjN89Ir6V2PczXNn
MpwKhw==}
    engines: {node: '>= 6'}
    deprecated: request has been deprecated, see https://github.com/request/request/issues/3142
    dependencies:
      aws-sign2: 0.7.0
      aws4: 1.12.0
      caseless: 0.12.0
      combined-stream: 1.0.8
      extend: 3.0.2
      forever-agent: 0.6.1
      form-data: 2.3.3
      har-validator: 5.1.5
      http-signature: 1.2.0
      is-typedarray: 1.0.0
      isstream: 0.1.2
      json-stringify-safe: 5.0.1
      mime-types: 2.1.35
      oauth-sign: 0.9.0
      performance-now: 2.1.0
      qs: 6.5.3
      safe-buffer: 5.2.1
      tough-cookie: 2.5.0
      tunnel-agent: 0.6.0
      uuid: 3.4.0
    dev: false

  /require-directory@2.1.1:
    resolution: {integrity: sha512-
fGxEI7+wsG9xrvdjsrlmL22OMTTiHRwAMroiEeMgq8gzoLC/PQr7RsRDSTLUg/bZAZtF+TVIkHc6/4RIKrui+Q
==}
    engines: {node: '>=0.10.0'}

  /require-from-string@2.0.2:
    resolution: {integrity: sha512-
Xf0nWe6RseziFMu+Ap9biiUbmplq6S9/p+7w7YXP/JBHhrUDDUhwa+vANyubuqfZWTveU//DYVGsDG7RKL/
vEw==}
    engines: {node: '>=0.10.0'}
    dev: false

  /require-main-filename@2.0.0:
    resolution: {integrity: sha512-
NKN5kMDylKuldxYLSUfrbo5Tuzh4hd+2E8NPPX02mZtn1VuREQToYe/ZdlJy+J3uCpfaiGF05e7B8W0iXbQ
Hmg==}
    dev: false

  /resolve-alpn@1.2.1:
    resolution: {integrity: sha512-
0a1F4l73/ZFZOakJnQ3FvkJ2+gSTQWz/r2KE5OdDY0TxPm5h4GkqkWWfM47T7HsbnOtcJVEF4epCVy6u7Q3K+g==}
    dev: false

  /resolve-from@4.0.0:
    resolution: {integrity: sha512-pb/MYmXstAkysRFx8piNI1tGFNQIFA3vkE3Gq4EuA1dF6gHp/+vgZqsCGJapvy8N3Q+4o7FwvquPJcnZ7RYy4g==}
    engines: {node: '>=4'}
    dev: true

  /resolve-pkg-maps@1.0.0:
    resolution: {integrity: sha512-seS2Tj26TBVOC2NIc2rOe2y2ZO7efxITtLZcGSOnHHNOQ7CkiUBfw0Iw2ck6xkIhPwLhKNLS8BO+hEpngQlqzw==}
    dev: true

  /resolve@1.22.8:
    resolution: {integrity: sha512-oKWePCxqpd6FlLvGV1VU0x7bkPmmCNolxzjMf4NczoDnQcIWrAF+cPtZn5i6n+RfD2d9i0tzpKnG6Yk168yIyw==}
    hasBin: true
    dependencies:
      is-core-module: 2.13.1
      path-parse: 1.0.7
      supports-preserve-symlinks-flag: 1.0.0

  /resolve@2.0.0-next.5:
    resolution: {integrity: sha512-U7WjGVG9sH8tvjW5SmGbQuui75FiyjAX72HX15DwBBwF9dNiQZRQAg9nnPhYy+TUnE0+VcrttuvNI8oSxZcocA==}
    hasBin: true
    dependencies:
      is-core-module: 2.13.1
      path-parse: 1.0.7
      supports-preserve-symlinks-flag: 1.0.0
    dev: true

  /responselike@2.0.1:
    resolution: {integrity: sha512-4gl03wn3hj1HP3yzgdI7d3lCkF95F21Pz4BPGvKHinyQzALR5CapwC8yIi0Rh58DEMQ/SguC03wFj2k0M/mHhw==}
    dependencies:
      lowercase-keys: 2.0.0
    dev: false

  /ret@0.2.2:
    resolution: {integrity: sha512-M0b3YWQs7R3Z917WRQy1HHA7Ba7D8hvZg6UE5mLykJxQVE2ju0IXbGlaHPPlkY+WN7wFP+wUMXmBFA0aV6vYGQ==}
    engines: {node: '>=4'}
    dev: false
  /reusify@1.0.4:
    resolution: {integrity: sha512-
U9nH88a3fc/ekCF1l0/UP1IosiuIjyTh7hBvXVMHYgVcfGvt897Xguj2UOLDeI5BG2m7/uwyaLVT6fbtCwTyzw==
}
    engines: {iojs: '>=1.0.0', node: '>=0.10.0'}

  /rfdc@1.3.0:
    resolution: {integrity: sha512-
V2hovdzFbOi77/WajaSMXk2OLm+xNIeQdMMuB7icj7bk6zi2F8GGAxigcnDFpJHbNyNcgyJDiP+8nOrY5cZG
rA==}
    dev: false

  /rimraf@3.0.2:
    resolution: {integrity: sha512-
JZkJMZkAGFFPP2YqXZXPbMlMBgsxzE8ILs4lMIX/2o0L9UBw9O/Y3o6wFw/i9YLapcUJWwqbi3kdxIPdC62
TIA==}
    hasBin: true
    dependencies:
      glob: 7.1.7
    dev: true

  /ripemd160@2.0.2:
    resolution: {integrity: sha512-
ii4iagi25WusVoiC4B4lq7pbXfAp3D9v5CwfkY33vffw2+pkDjY1D8GaN7spsxvCSx8dkPqOZCEZyfxcmJG2IA==
}
    dependencies:
      hash-base: 3.1.0
      inherits: 2.0.4
    dev: false

  /rlp@2.2.7:
    resolution: {integrity: sha512-
d5gdPmgQ0Z+AklL2NVXr/IoSjNZFfTVvQWzL/AM2AOcSzYP2xjlb0AC8YyCLc41MSNf6P6QVtjgPdmVtzb+4
lQ==}
    hasBin: true
    dependencies:
      bn.js: 5.2.1
    dev: false

  /rollup@3.29.4:
    resolution: {integrity: sha512-
oWzmBZwvYrU0iJHtDmhsm662rC15FRXmcjCk1xD771dFDx5jJ02ufAQQTn0etB2emNk4J9EZg/yWKpsn9B
WGRw==}
    engines: {node: '>=14.18.0', npm: '>=8.0.0'}
    hasBin: true
    optionalDependencies:
      fsevents: 2.3.2
    dev: true

  /rpc-websockets@7.6.1:
    resolution: {integrity: sha512-
MmRGaJJvxTHSRxYPjJJqcj2zWnCetw7YbYbKlD0Yc7qVw6PsZhRJg1MI3mpWlpBs+4zO+urlNfLl9zLsdOD/
gA==}
    dependencies:
      '@babel/runtime': 7.23.2
      eventemitter3: 4.0.7
      uuid: 8.3.2
      ws: 8.13.0(bufferutil@4.0.7)(utf-8-validate@5.0.10)
    optionalDependencies:
      bufferutil: 4.0.7
      utf-8-validate: 5.0.10
    dev: false

  /run-applescript@5.0.0:
    resolution: {integrity: sha512-
XcT5rBksx1QdIhlFOCtgZkB99ZEouFZ1E2Kc2LHqNW13U3/74YGdkQRmThTwxy4QIyookibDKYZOPqX//6B
lAg==}
    engines: {node: '>=12'}
    dependencies:
      execa: 5.1.1
    dev: true

  /run-parallel@1.2.0:
    resolution: {integrity: sha512-
5l4VyZR86LZ/lDxZTR6jqL8AFE2S0IFLMP26AbjsLVADxHdhB/c0GUsH+y39UfCi3dzz8OlQuPmnaJOMoDH
QBA==}
    dependencies:
      queue-microtask: 1.2.3

  /rustbn.js@0.2.0:
    resolution: {integrity: sha512-
4VlvkRUuCJvr2J6Y0ImW7NvTCriMi7ErOAqWk1y69vAdoNIzCF3yPmgeNzx+RQTLEDFq5sHfscn1MwHxP9
hNfA==}
    dev: false

  /rxjs@6.6.7:
    resolution: {integrity: sha512-
hTdwr+7yYNIT5n4AMYp85KA6yw2Va0FLa3Rguvbpa4W3I5xynaBZo41cM3XM+4Q6fRMj3sBYIR1VAmZM
XYJvRQ==}
    engines: {npm: '>=2.0.0'}
    dependencies:
      tslib: 1.14.1
    dev: false

  /rxjs@7.8.1:
    resolution: {integrity: sha512-
AA3TVj+0A2iuIoQkWEK/tqFjBq2j+6PO6Y0zJcvzLAFhEFIO3HL0vls9hWLncZbAAbK0mar7oZ4V079I/qPMx
g==}
    dependencies:
      tslib: 2.4.1
    dev: true

  /safe-array-concat@1.0.1:
    resolution: {integrity: sha512-
6XbUAseYE2KtOuGueyeobCySj9L4+66Tn6KQMOPQJrAJEowYKW/YR/MGJZl7FdydUdaFu4LYyDZjxf4/N
mo23Q==}
    engines: {node: '>=0.4'}
    dependencies:
      call-bind: 1.0.2
      get-intrinsic: 1.2.2
      has-symbols: 1.0.3
      isarray: 2.0.5
    dev: true

  /safe-buffer@5.1.2:
    resolution: {integrity: sha512-
Gd2UZBJDkXlY7GbJxfsE8/nvKkUEU1G38c1siN6QP6a9PT9MmHB8GnpscSmMJSoF8LOIrt8ud/wPtojys4G
6+g==}
    dev: false

  /safe-buffer@5.2.1:
    resolution: {integrity: sha512-
rp3So07KcdmmKbGvgaNxQSJr7bGVSVk5S9Eq1F+ppbRo70+YeaDxkw5Dd8NPN+GD6bjnYm2VuPuCXm
puYvmCXQ==}
    dev: false

  /safe-event-emitter@1.0.1:
    resolution: {integrity: sha512-
e1wFe99A91XYYxoQbcq2ZJUWurxEyP8vfz7A7vuUe1s95q8r5ebraVaA1BukYJcpM6V16ugWoD9vngi8Ccu
5fg==}
    deprecated: Renamed to @metamask/safe-event-emitter
    dependencies:
      events: 3.3.0
    dev: false

  /safe-json-utils@1.1.1:
    resolution: {integrity: sha512-
SAJWGKDs50tAbiDXLf89PDwt9XYkWyANFWVzn4dTXI5QyI8t2o/bW5/OJl3lvc2WVU4MEpTo9Yz5NVFNsp
+OJQ==}
    dev: false

  /safe-regex-test@1.0.0:
    resolution: {integrity: sha512-
JBUUzyOgEwXQY1NuPtvcj/qcBDbDmEvWufhlnXZIm75DEHp+afM1r1ujJpJsV/gSM4t59tpDyPi1sd6ZaPFfs
A==}
    dependencies:
      call-bind: 1.0.5
      get-intrinsic: 1.2.2
      is-regex: 1.1.4
    dev: true

  /safe-regex2@2.0.0:
    resolution: {integrity: sha512-
PaUSFsUaNNuKwkBijoAPHAK6/eM6VirvyPWlZ7BAQy4D+hCvh4B6lIG+nPdhbFfIbP+gTGBcrdsOaUs0F+Z
BOQ==}
```

```
    dependencies:
      ret: 0.2.2
    dev: false

  /safe-stable-stringify@2.4.3:
    resolution: {integrity: sha512-e2bDA2WJT0wxseVd4lsDP4+3ONX6HpMXQa1ZhFQ7SU+GjvORCmShbCMltrtIDfkYhVHrOcPtj+KhmDBdPdZD1g==}
    engines: {node: '>=10'}
    dev: false

  /safer-buffer@2.1.2:
    resolution: {integrity: sha512-YZo3K82SD7Riyi0E1EQPojLz7kpepnSQI9IyPbHHg1XXXevb5dJI7tpyN2ADxGcQbHG7vcyRHk0cbwqcQriUtg==}
    dev: false

  /scheduler@0.23.0:
    resolution: {integrity: sha512-CtuThmgHNg7zIZWAXi3AsyIzA3n4xx7aNyjwC2VJldO2LMVDhFK+63xGqq6CsJH4rTAt6/M+N4GhZiDYPx9eUw==}
    dependencies:
      loose-envify: 1.4.0
    dev: false

  /scrypt-js@3.0.1:
    resolution: {integrity: sha512-cdwTTnqPu0Hyvf5in5asVdZocVDTNRmR7XEcJuIzMjJeSHybHl7vpB66AzwTaIg6CLSbtjcxc8fqcySfnTkccA==}
    dev: false

  /secp256k1@4.0.3:
    resolution: {integrity: sha512-NLZVf+ROMxwtEj3Xa562qgv2BK5e2WNmXPiOdVIPLgs6lyTzMvBq0aWTYMI5XCP9jZMVKOcqZLw/Wc4vDkuxhA==}
    engines: {node: '>=10.0.0'}
    requiresBuild: true
    dependencies:
      elliptic: 6.5.4
      node-addon-api: 2.0.2
      node-gyp-build: 4.5.0
    dev: false

  /secure-json-parse@2.7.0:
    resolution: {integrity: sha512-6aU+Rwsezw7VR8/nyvKTx8QpWH9FrcYiXXlqC4z5d5XQBDRqtbfsRjnwGyqbi3gddNtWHuEk9OANUotL26qKUw==}
    dev: false

  /semaphore@1.1.0:
    resolution: {integrity: sha512-O4OZEaNtkMd/K0i6js9SL+gqy0ZCBMgUvlSqHKi4IBdjhe7wB8pwztUk1BbZ1fmrvpwFrPbHzqd2w5pTcJH6L
```

```
A==}
    engines: {node: '>=0.8.0'}
    dev: false

  /semver@5.4.1:
    resolution: {integrity: sha512-
WfG/X9+oATh81XtllIo/I8gOiY9EXRdv1cQdyykeXK17YcUW3EXUAi2To4pcH6nZtJPr7ZOpM5OMyWJZm+8
Rsg==}
    hasBin: true

  /semver@6.3.1:
    resolution: {integrity: sha512-
BR7VvDCVHO+q2xBEWskxS6DJE1qRnb7DxzUrogb71CWoSficBxYsiAGd+Kl0mmq/MprG9yArRkyrQxTO6
XjMzA==}
    hasBin: true

  /semver@7.5.4:
    resolution: {integrity: sha512-
1bCSESV6Pv+i21Hvpxp3Dx+pSD8lIPt8uVjRrxAUt/nbswYc+tK6Y2btiULjd4+fnq15PX+nqQDC7Oft7WkwcA==
}
    engines: {node: '>=10'}
    hasBin: true
    dependencies:
      lru-cache: 6.0.0

  /send@0.18.0:
    resolution: {integrity: sha512-
qqWzuOjSFOuqPjFe4NOsMLafToQQwBSOEpS+FwEt3A2V3vKubTquT3vmLTQpFgMXp8AIFWFuP1qKaJZ
OtPpVXg==}
    engines: {node: '>= 0.8.0'}
    dependencies:
      debug: 2.6.9
      depd: 2.0.0
      destroy: 1.2.0
      encodeurl: 1.0.2
      escape-html: 1.0.3
      etag: 1.8.1
      fresh: 0.5.2
      http-errors: 2.0.0
      mime: 1.6.0
      ms: 2.1.3
      on-finished: 2.4.1
      range-parser: 1.2.1
      statuses: 2.0.1
    transitivePeerDependencies:
      - supports-color
    dev: false

  /serve-static@1.15.0:
    resolution: {integrity: sha512-
XGuRDNjXUijsUL0vl6nSD7cwURuzEgglbOaFuZM9g3kwDXOWVTck0jLzjPzGD+TazWbboZYu52/9/XPdUg
ne9g==}
```

```
    engines: {node: '>= 0.8.0'}
    dependencies:
      encodeurl: 1.0.2
      escape-html: 1.0.3
      parseurl: 1.3.3
      send: 0.18.0
    transitivePeerDependencies:
      - supports-color
    dev: false

  /servify@0.1.12:
    resolution: {integrity: sha512-
/xE6GvsKKqyo1BAY+KxOWXcLpPsUUyji7Qg3bVD7hh1eRze5bR1uYiuDA/k3Gof1s9BTzQZEJK8sNcNGFI
zeWw==}
    engines: {node: '>=6'}
    dependencies:
      body-parser: 1.20.2
      cors: 2.8.5
      express: 4.18.2
      request: 2.88.2
      xhr: 2.6.0
    transitivePeerDependencies:
      - supports-color
    dev: false

  /set-blocking@2.0.0:
    resolution: {integrity: sha512-
KiKBS8AnWGEyLzofFfmvKwpdPzqiy16LvQfK3yv/fVH7Bj13/wl3JSR1J+rfgRE9q7xUJK4qvgS8raSOeLUehw
==}
    dev: false

  /set-cookie-parser@2.6.0:
    resolution: {integrity: sha512-
RVnVQxTXuerk653XfuliOxBP81Sf0+qfQE73LIYKcyMYHG94AuH0kgrQpRDuTZnSmjpysHmzxJXKNfa6PjF
hyQ==}
    dev: false

  /set-function-length@1.1.1:
    resolution: {integrity: sha512-
VoaqjbBJKiWtg4yRcKBQ7g7wnGnLV3M8oLvVWwOk2PdYY6PEFegR1vezXR0tw6fZGF9csVakIRjrJiy2veS
BFQ==}
    engines: {node: '>= 0.4'}
    dependencies:
      define-data-property: 1.1.1
      get-intrinsic: 1.2.2
      gopd: 1.0.1
      has-property-descriptors: 1.0.1
    dev: true

  /set-function-name@2.0.1:
    resolution: {integrity: sha512-
tMNCiqYVkXIZgc2Hnoy2IvC/f8ezc5koaRFkCjrpWzGpCd3qbZXPzVy9MAZzK1ch/X0jvSkojys3oqJN0qCmdA
==}
    engines: {node: '>= 0.4'}
    dependencies:
      define-data-property: 1.1.1
      functions-have-names: 1.2.3
      has-property-descriptors: 1.0.1
    dev: true

  /set-immediate-shim@1.0.1:
    resolution: {integrity: sha512-
Li5AOqrZWCVA2n5kryzEmqai6bKSIvpz5oUJHPVj6+dsbD3X1ixtsY5tEnsaNpH3pFAHmG8eIHUrtEtohrg+U
Q==}
    engines: {node: '>=0.10.0'}
    dev: false

  /setimmediate@1.0.5:
    resolution: {integrity: sha512-
MATJdZp8sLqDl/68LfQmbP8zKPLQNV6BIZoIgrscFDQ+RsvK/BxeDQOgyxKKoh0y/8h3BqVFnCqQ/gd+reiIX
A==}
    dev: false

  /setprototypeof@1.2.0:
    resolution: {integrity: sha512-
E5LDX7Wrp85Kil5bhZv46j8jOeboKq5JMmYM3gVGdGH8xFpPWXUMsNrlODCrkoxMEeNi/XZIwuRvY4XNw
YMJpw==}
    dev: false

  /sha.js@2.4.11:
    resolution: {integrity: sha512-
QMEp5B7cftE7APOjk5Y6xgrbWu+WkLVQwk8JNjZ8nKRciZaByEW6MubieAiToS7+dwvrjGhH8jRXz3MVd0
AYqQ==}
    hasBin: true
    dependencies:
      inherits: 2.0.4
      safe-buffer: 5.2.1
    dev: false

  /shebang-command@2.0.0:
    resolution: {integrity: sha512-
kHxr2zZpYtdmrN1qDjrrX/Z1rR1kG8Dx+gkpK1G4eXmvXswmcE1hTWBWYUzlraYw1/yZp6YuDY77YtvbN0d
mDA==}
    engines: {node: '>=8'}
    dependencies:
      shebang-regex: 3.0.0
    dev: true

  /shebang-regex@3.0.0:
    resolution: {integrity: sha512-
7++dFhtcx3353uBaq8DDR4NuxBetBzC7ZQOhmTQInHEd6bSrXdiEyzCvG07Z44UYdLShWUyXt5M/yhz8ek
cb1A==}
    engines: {node: '>=8'}
    dev: true
  /shell-quote@1.8.1:
    resolution: {integrity: sha512-6j1W9l1iAs/4xYBI1SYOVZyFcCis9b4KCLQ8fgAGG07QvzaRLVVRQvAy85yNmmZSjYjg4MWh4gNvlPujU/5LpA==}
    dev: true

  /side-channel@1.0.4:
    resolution: {integrity: sha512-q5XPytqFEIKHkGdiMIrY10mvLRvnQh42/+GoBlFW3b2LXLE2xxJpZFdm94we0BaoV3RwJyGqg5wS7epxTv0Zvw==}
    dependencies:
      call-bind: 1.0.2
      get-intrinsic: 1.1.3
      object-inspect: 1.12.2

  /signal-exit@3.0.7:
    resolution: {integrity: sha512-wnD2ZE+l+SPC/uoS0vXeE9L1+0wuaMqKlfz9AMUo38JsyLSBWSFcHR1Rri62LZc12vLr1gb3jl7iwQhgwpAbGQ==}
    dev: true

  /simple-concat@1.0.1:
    resolution: {integrity: sha512-cSFtAPtRhljv69IK0hTVZQ+OfE9nePi/rtJmw5UjHeVyVroEqJXP1sFztKUy1qU+xvz3u/sfYJLa947b7nAN2Q==}
    dev: false

  /simple-get@2.8.2:
    resolution: {integrity: sha512-Ijd/rV5o+mSBBs4F/x9oDPtTx9Zb6X9brmnXvMW4J7IR15ngi9q5xxqWBKU744jTZiaXtxaPL7uHG6vtN8kUkw==}
    dependencies:
      decompress-response: 3.3.0
      once: 1.4.0
      simple-concat: 1.0.1
    dev: false

  /simple-update-notifier@2.0.0:
    resolution: {integrity: sha512-a2B9Y0KlNXl9u/vsW6sTIu9vGEpfKu2wRV6l1H3XEas/0gUIzGzBoP/IouTcUQbm9JWZLH3COxyn03TYlFax6w==}
    engines: {node: '>=10'}
    dependencies:
      semver: 7.5.4
    dev: true

  /siwe@2.1.4(ethers@6.8.0):
    resolution: {integrity: sha512-Dke1Qqa3mgiLm3vjqw/+SQ7dl8WV/Pfk3AlQBF94cBFydTYhztngqYrikzE3X5UTsJ6565dfVbQptszsuYZNYg==}
    peerDependencies:
```

```
    ethers: ^5.6.8 || ^6.0.8
  dependencies:
    '@spruceid/siwe-parser': 2.0.2
    '@stablelib/random': 1.0.2
    ethers: 6.8.0
    uri-js: 4.4.1
    valid-url: 1.0.9
  dev: false

 /slash@3.0.0:
  resolution: {integrity: sha512-
g9Q1haeby36OSStwb4ntCGGGaKsaVSjQ68fBxoQcutl5fS1vuY18H3wSt3jFyFtrkx+Kz0V1G85A4MyAdDMi2
Q==}
  engines: {node: '>=8'}
  dev: true

 /solidity-bytes-utils@0.8.0(@babel/core@7.23.2):
  resolution: {integrity: sha512-
r109ZHEf7zTMm1ENW6/IJFDWilFR/v0BZnGuFgDHJUV80ByobnV2k3txvwQaJ9ApL+6XAfwqsw5VFzjALbQ
PCw==}
  dependencies:
    '@truffle/hdwallet-provider': 2.1.15(@babel/core@7.23.2)
  transitivePeerDependencies:
    - '@babel/core'
    - bufferutil
    - encoding
    - supports-color
    - utf-8-validate
  dev: false

 /sonic-boom@2.8.0:
  resolution: {integrity: sha512-
kuonw1YOYYNOve5iHdSahXPOK49GqwA+LZhI6Wz/l0rP57iKyXXIHaRagOBHAPmGwJC6od2Z9zgvZ5loS
gMlVg==}
  dependencies:
    atomic-sleep: 1.0.0
  dev: false

 /sonic-boom@3.7.0:
  resolution: {integrity: sha512-
IudtNvSqA/ObjN97tfgNmOKyDOs4dNcg4cUUsHDebqsgb8wGBBwb31LIgShNO8fye0dFI52X1+tFoKKI6Rq1
Gg==}
  dependencies:
    atomic-sleep: 1.0.0
  dev: false

 /source-map-js@1.0.2:
  resolution: {integrity: sha512-
R0XvVJ9WusLiqTCEiGCmICCMplcCkIwwR11mOSD9CR5u+IXYdiseeEuXCVAjS54zqwkLcPNnmU4OeJ6t
UrWhDw==}
  engines: {node: '>=0.10.0'}
 /spawn-command@0.0.2:
   resolution: {integrity: sha512-
zC8zGoGkmc8J9ndvml8Xksr1Amk9qBujgbF0JAIWO7kXr43w0h/0GJNM/Vustixu+YE8N/MTrQ7N31FvHUA
CxQ==}
   dev: true

 /spdx-correct@3.2.0:
   resolution: {integrity: sha512-
kN9dJbvnySHULIluDHy32WHRUu3Og7B9sbY7tsFLctQkIqnMh3hErYgdMjTYuqmcXX+lK5T1lnUt3G7zNsw
mZA==}
   dependencies:
     spdx-expression-parse: 3.0.1
     spdx-license-ids: 3.0.16
   dev: true

 /spdx-exceptions@2.3.0:
   resolution: {integrity: sha512-
/tTrYOC7PPI1nUAgx34hUpqXuyJG+DTHJTnIULG4rDygi4xu/tfgmq1e1cIRwRzwZgo4NLySi+ricLkZkw4i5A==
}
   dev: true

 /spdx-expression-parse@3.0.1:
   resolution: {integrity: sha512-
cbqHunsQWnJNE6KhVSMsMeH5H/L9EpymbzqTQ3uLwNCLZ1Q481oWaofqH7nO6V07xlXwY6PhQdQ2Ied
Wx/ZK4Q==}
   dependencies:
     spdx-exceptions: 2.3.0
     spdx-license-ids: 3.0.16
   dev: true

 /spdx-license-ids@3.0.16:
   resolution: {integrity: sha512-
eWN+LnM3GR6gPu35WxNgbGl8rmY1AEmoMDvL/QD6zYmPWgywxWqJWNdLGT+ke8dKNWrcYgYjPpG5
gbTfghP8rw==}
   dev: true

 /split-on-first@1.1.0:
   resolution: {integrity: sha512-
43ZssAJaMusuKWL8sKUBQXHWOpq8d6CfN/u1p4gUzfJkM05C8rxTmYrkIPTXapZpORA6LkkzcUulJ8FqA7
Uudw==}
   engines: {node: '>=6'}
   dev: false

 /split2@4.2.0:
   resolution: {integrity: sha512-
UcjcJOWknrNkF6PLX83qcHM6KHgVKNkV62Y8a5uYDVv9ydGQVwAHMKqHdJje1VTWpljG0WYpCDhrCdA
OYH4TWg==}
   engines: {node: '>= 10.x'}
   dev: false

 /sshpk@1.18.0:
   resolution: {integrity: sha512-
2p2KJZTSqQ/I3+HX42EpYOa2l3f8Erv8MWKsy2I9uf4wA7yFIkXRffYdsx86y6z4vHtV8u7g+pPlr8/4ouAxsQ==
}
   engines: {node: '>=0.10.0'}
   hasBin: true
   dependencies:
     asn1: 0.2.6
     assert-plus: 1.0.0
     bcrypt-pbkdf: 1.0.2
     dashdash: 1.14.1
     ecc-jsbn: 0.1.2
     getpass: 0.1.7
     jsbn: 0.1.1
     safer-buffer: 2.1.2
     tweetnacl: 0.14.5
   dev: false

 /statuses@2.0.1:
   resolution: {integrity: sha512-
RwNA9Z/7PrK06rYLIzFMlaF+l73iwpzsqRIFgbMLbTcLD6cOao82TaWefPXQvB2fOC4AjuYSEndS7N/mTCb
kdQ==}
   engines: {node: '>= 0.8'}
   dev: false

 /stream-browserify@3.0.0:
   resolution: {integrity: sha512-
H73RAHsVBapbim0tU2JwwOiXUj+fikfiaoYAKHF3VJfA0pe2BCzkhAHBlLG6REzE+2WNZcxOXjK7lkso+9eu
LA==}
   dependencies:
     inherits: 2.0.4
     readable-stream: 3.6.0
   dev: false

 /stream-shift@1.0.1:
   resolution: {integrity: sha512-
AiisoFqQ0vbGcZgQPY1cdP2I76glaVA/RauYR4G4thNFgkTqr90yXTo4LYX60Jl+sIlPNHHdGSwo01AvbKUS
VQ==}
   dev: false

 /strict-uri-encode@1.1.0:
   resolution: {integrity: sha512-
R3f198pcvnB+5IpnBlRkphuE9n46WyVl8I39W/ZUTZLz4nqSP/oLYUrcnJrw462Ds8he4YKMov2efsTIw1BDG
Q==}
   engines: {node: '>=0.10.0'}
   dev: false

 /strict-uri-encode@2.0.0:
   resolution: {integrity: sha512-
QwiXZgpRcKkhTj2Scnn++4PKtWsH0kpzZ62L2R6c/LUVYv7hVnZqcg2+sMuT6R7Jusu1vviK/MFsu6kNJfWl
EQ==}
   engines: {node: '>=4'}
   dev: false
 /string-format@2.0.0:
   resolution: {integrity: sha512-
bbEs3scLeYNXLecRRuk6uJxdXUSj6le/8rNPHChIJTn2V79aXVTR1EH2OH5zLKKoz0V02fOUKZZcw01pLU
ShZA==}

 /string-width@4.2.3:
   resolution: {integrity: sha512-
wKyQRQpjJ0sIp62ErSZdGsjMJWsap5oRNihHhu6G7JVO/9jIB6UyevL+tXuOqrng8j/cxKTWyWUwvSTriiZz/g
==}
   engines: {node: '>=8'}
   dependencies:
     emoji-regex: 8.0.0
     is-fullwidth-code-point: 3.0.0
     strip-ansi: 6.0.1

 /string.prototype.matchall@4.0.10:
   resolution: {integrity: sha512-
rGXbGmOEosIQi6Qva94HUjgPs9vKW+dkG7Y8Q5O2OYkWL6wFaTRZO8zM4mhP94uX55wgyrXzfS2aGtG
zUL7EJQ==}
   dependencies:
     call-bind: 1.0.2
     define-properties: 1.2.1
     es-abstract: 1.22.3
     get-intrinsic: 1.2.2
     has-symbols: 1.0.3
     internal-slot: 1.0.6
     regexp.prototype.flags: 1.5.1
     set-function-name: 2.0.1
     side-channel: 1.0.4
   dev: true

 /string.prototype.trim@1.2.8:
   resolution: {integrity: sha512-
lfjY4HcixfQXOfaqCvcBuOIapyaroTXhbkfJN3gcB1OtyupngWK4sEET9Knd0cXd28kTUqu/kHoV4HKSJdnjiQ==
}
   engines: {node: '>= 0.4'}
   dependencies:
     call-bind: 1.0.5
     define-properties: 1.2.1
     es-abstract: 1.22.3
   dev: true

 /string.prototype.trimend@1.0.7:
   resolution: {integrity: sha512-
Ni79DqeB72ZFq1uH/L6zJ+DKZTkOtPIHovb3YZHQViE+HDouuU4mBrLOLDn5Dde3RF8qw5qVETEjhu9loc
MLvA==}
   dependencies:
     call-bind: 1.0.5
     define-properties: 1.2.1
     es-abstract: 1.22.3
   dev: true
```

```
 /string.prototype.trimstart@1.0.7:
   resolution: {integrity: sha512-
NGhtDFu3jCEm7B4Fy0DpLewdJQOZcQ0rGbwQ/+stjnrp2i+rlKeCvos9hOIeCmqwratM47OBxY7uFZzjxHXm
rg==}
   dependencies:
     call-bind: 1.0.5
     define-properties: 1.2.1
     es-abstract: 1.22.3
   dev: true

 /string_decoder@0.10.31:
   resolution: {integrity: sha512-
ev2QzSzWPYmy9GuqfIVildA4OdcGLeFZQrq5ys6RtiuF+RQQiZWr8TZNyAcuVXyQRYfEO+MsoB/1BuQVh
OJuoQ==}
   dev: false

 /string_decoder@1.1.1:
   resolution: {integrity: sha512-
n/ShnvDi6FHbbVfviro+WojiFzv+s8MPMHBczVePfUpDJLwoLT0ht1l4YwBCbi8pJAveEEdnkHyPyTP/mzRfwg
==}
   dependencies:
     safe-buffer: 5.1.2
   dev: false

 /string_decoder@1.3.0:
   resolution: {integrity: sha512-
hkRX8U1WjJFd8LsDJ2yQ/wWWxaopEsABU1XfkM8A+j0+85JAGppt16cr1Whg6KIbb4okU6Mql6BOj+uup/w
KeA==}
   dependencies:
     safe-buffer: 5.2.1
   dev: false

 /strip-ansi@6.0.1:
   resolution: {integrity: sha512-
Y38VPSHcqkFrCpFnQ9vuSXmquuv5oXOKpGeT6aGrr3o3Gc9AlVa6JBfUSOCnbxGGZF+/0ooI7KrPuUSztU
dU5A==}
   engines: {node: '>=8'}
   dependencies:
     ansi-regex: 5.0.1

 /strip-bom@3.0.0:
   resolution: {integrity: sha512-
vavAMRXOgBVNF6nyEEmL3DBK19iRpDcoIwW+swQ+CbGiu7lju6t+JklA1MHweoWtadgt4ISVUsXLyDq34d
dcwA==}
   engines: {node: '>=4'}
   dev: true

 /strip-final-newline@2.0.0:
   resolution: {integrity: sha512-
BrpvfNAE3dcvq7ll3xVumzjKjZQ5tI1sEUIKr3Uoks0XUl45St3FlatVqef9prk4jRDzhW6WZg+3bk93y6pLjA==}
   engines: {node: '>=6'}
   dev: true
```

```
  /strip-final-newline@3.0.0:
    resolution: {integrity: sha512-
dOESqjYr96iWYylGObzd39EuNTa5VJxyvVAEm5Jnh7KGo75V43Hk1odPQkNDyXNmUR6k+gEiDVXnjB8H
J3crXw==}
    engines: {node: '>=12'}
    dev: true

  /strip-hex-prefix@1.0.0:
    resolution: {integrity: sha512-
q8d4ue7JGEiVcypji1bALTos+0pWtyGlivAWyPuTkHzuTCJqrK9sWxYQZUq6Nq3cuyv3bm734IhHvHtGGUR
U6A==}
    engines: {node: '>=6.5.0', npm: '>=3'}
    dependencies:
      is-hex-prefixed: 1.0.0
    dev: false

  /strip-indent@3.0.0:
    resolution: {integrity: sha512-
laJTa3Jb+VQpaC6DseHhF7dXVqHTfJPCRDaEbid/drOhgitgYku/letMUqOXFoWV0zIIUbjpdH2t+tYj4bQMRQ
==}
    engines: {node: '>=8'}
    dependencies:
      min-indent: 1.0.1
    dev: true

  /strip-json-comments@3.1.1:
    resolution: {integrity: sha512-
6fPc+R4ihwqP6N/aIv2f1gMH8lOVtWQHoqC4yK6oSDVVocumAsfCqjkXnqiYMhmMwS/mEHLp7Vehlt3ql6lEi
g==}
    engines: {node: '>=8'}
    dev: true

  /sucrase@3.34.0:
    resolution: {integrity: sha512-
70/LQEZ07TEcxiU2dz51FKaE6hCTWC6vr7FOk3Gr0U60C3shtAN+H+BFr9XlYe5xqf3RA8nrc+VIwzCfnxuX
Jw==}
    engines: {node: '>=8'}
    hasBin: true
    dependencies:
      '@jridgewell/gen-mapping': 0.3.2
      commander: 4.1.1
      glob: 7.1.6
      lines-and-columns: 1.2.4
      mz: 2.7.0
      pirates: 4.0.6
      ts-interface-checker: 0.1.13
    dev: false

  /superstruct@0.14.2:
    resolution: {integrity: sha512-
nPewA6m9mR3d6k7WkZ8N8zpTWfenFH3q9pA2PkuiZxINr9DKB2+40wEQf0ixn8VaGuJ78AB6iWOtStI+/4F
KZQ==}
    dev: false

  /superstruct@1.0.3:
    resolution: {integrity: sha512-
8iTn3oSS8nRGn+C2pgXSKPI3jmpm6FExNazNpjvqS6ZUJQCej3PUXEKM8NjHBOs54ExM+LPW/FBRhymr
dcCiSg==}
    engines: {node: '>=14.0.0'}
    dev: false

  /supports-color@5.5.0:
    resolution: {integrity: sha512-
QjVjwdXIt408MIiAqCX4oUKsgU2EqAGzs2Ppkm4aQYbjm+ZEWEcW4SfFNTr4uMNZma0ey4f5lgLrkB0aX0
QMow==}
    engines: {node: '>=4'}
    dependencies:
      has-flag: 3.0.0

  /supports-color@7.2.0:
    resolution: {integrity: sha512-
qpCAvRl9stuOHveKsn7HncJRvv501qIacKzQlO/+Lwxc9+0q2wLyv4Dfvt80/DPn2pqOBsJdDiogXGR9+OvwR
w==}
    engines: {node: '>=8'}
    dependencies:
      has-flag: 4.0.0

  /supports-color@8.1.1:
    resolution: {integrity: sha512-
MpUEN2OodtUzxvKQl72cUF7RQ5EiHsGvSsVG0ia9c5RbWGL2CI4C7EpPS8UTBIplnlzZiNuV56w+FuNxy3t
y2Q==}
    engines: {node: '>=10'}
    dependencies:
      has-flag: 4.0.0
    dev: true

  /supports-preserve-symlinks-flag@1.0.0:
    resolution: {integrity: sha512-
ot0WnXS9fgdkgIcePe6RHNk1WA8+muPa6cSjeR3V8K27q9BB1rTE3R1p7Hv0z1ZyAc8s6Vvv8DIyWf681M
At0w==}
    engines: {node: '>= 0.4'}

  /swarm-js@0.1.42:
    resolution: {integrity: sha512-
BV7c/dVlA3R6ya1lMlSSNPLYrntt0LUq4YMgy3iwpCIc6rZnS5W2wUoctarZ5pXlpKtxDDf9hNziEkcfrxdhqQ==
}
    dependencies:
      bluebird: 3.7.2
      buffer: 5.7.1
      eth-lib: 0.1.29
      fs-extra: 4.0.3
      got: 11.8.6
      mime-types: 2.1.35
```

```yaml
      mkdirp-promise: 5.0.1
      mock-fs: 4.14.0
      setimmediate: 1.0.5
      tar: 4.4.19
      xhr-request: 1.1.0
    transitivePeerDependencies:
      - bufferutil
      - supports-color
      - utf-8-validate
    dev: false

  /synckit@0.8.5:
    resolution: {integrity: sha512-
L1dapNV6vu2s/4Sputv8xGsCdAVlb5nRDMFU/E27D44l5U6cw1g0dGd45uLc+OXjNMmF4ntiMdCimzcjFKQI
8Q==}
    engines: {node: ^14.18.0 || >=16.0.0}
    dependencies:
      '@pkgr/utils': 2.4.2
      tslib: 2.6.2
    dev: true

  /table-layout@1.0.2:
    resolution: {integrity: sha512-
qd/R7n5rQTRFi+Zf2sk5XVVd9UQl6ZkduPFC3S7WEGJAmetDTjY3qPN50eSKzwuzEyQKy5TN2TiZdkIjos2
L6A==}
    engines: {node: '>=8.0.0'}
    dependencies:
      array-back: 4.0.2
      deep-extend: 0.6.0
      typical: 5.2.0
      wordwrapjs: 4.0.1

  /tailwindcss@3.3.5(ts-node@10.9.1):
    resolution: {integrity: sha512-
5SEZU4J7pxZgSkv7FP1zY8i2TIAOooNZ1e/OGtxIEv6GltpoiXUqWvLy89+a10qYTB1N5Ifkuw9lqQkN9sscvA
==}
    engines: {node: '>=14.0.0'}
    hasBin: true
    dependencies:
      '@alloc/quick-lru': 5.2.0
      arg: 5.0.2
      chokidar: 3.5.3
      didyoumean: 1.2.2
      dlv: 1.1.3
      fast-glob: 3.3.1
      glob-parent: 6.0.2
      is-glob: 4.0.3
      jiti: 1.20.0
      lilconfig: 2.1.0
      micromatch: 4.0.5
      normalize-path: 3.0.0
      object-hash: 3.0.0
      picocolors: 1.0.0
      postcss: 8.4.31
      postcss-import: 15.1.0(postcss@8.4.31)
      postcss-js: 4.0.1(postcss@8.4.31)
      postcss-load-config: 4.0.1(postcss@8.4.31)(ts-node@10.9.1)
      postcss-nested: 6.0.1(postcss@8.4.31)
      postcss-selector-parser: 6.0.13
      resolve: 1.22.8
      sucrase: 3.34.0
    transitivePeerDependencies:
      - ts-node
    dev: false

  /tapable@2.2.1:
    resolution: {integrity: sha512-GNzQvQTOIP6RyTfE2Qxb8ZVlNmw0n88vp1szwWRimP02mnTsx3Wtn5qRdqY9w2XduFNUgvOwhNnQsjwCp+kqaQ==}
    engines: {node: '>=6'}
    dev: true

  /tar@4.4.19:
    resolution: {integrity: sha512-a20gEsvHnWe0ygBY8JbxoM4w3SJdhc7ZAuxkLqh+nvNQN2IOt0B5lLgM490X5Hl8FF0dl0tOf2ewFYAlIFgzVA==}
    engines: {node: '>=4.5'}
    dependencies:
      chownr: 1.1.4
      fs-minipass: 1.2.7
      minipass: 2.9.0
      minizlib: 1.3.3
      mkdirp: 0.5.6
      safe-buffer: 5.2.1
      yallist: 3.1.1
    dev: false

  /text-encoding-utf-8@1.0.2:
    resolution: {integrity: sha512-8bw4MY9WjdsD2aMtO0OzOCY3pXGYNx2d2FfHRVUKkiCPDWjKuOlhLVASS+pD7VkLTVjW268LYJHwsnPFlBpbAg==}
    dev: false

  /text-table@0.2.0:
    resolution: {integrity: sha512-N+8UisAXDGk8PFXP4HAzVR9nbfmVJ3zYLAWiTIoqC5v5isinhr+r5uaO8+7r3BMfuNIufIsA7RdpVgacC2cSpw==}
    dev: true

  /thenify-all@1.6.0:
    resolution: {integrity: sha512-RNxQH/qI8/t3thXJDwcstUO4zeqo64+Uy/+sNVRBx4Xn2OX+OZ9oP+iJnNFqplFra2ZUVeKCSa2oVWi3T4uVmA==}
    engines: {node: '>=0.8'}
    dependencies:
      thenify: 3.3.1
    dev: false

  /thenify@3.3.1:
    resolution: {integrity: sha512-RVZSIV5IG10Hk3enotrhvz0T9em6cyHBLkH/YAZuKqd8hRkKhSfCGIcP2KUY0EPxndzANBmNllzWPwak+bheSw==}
    dependencies:
      any-promise: 1.3.0
    dev: false

  /thread-stream@0.15.2:
    resolution: {integrity: sha512-UkEhKIg2pD+fjkHQKyJO3yoIvAP3N6RlNFt2dUhcS1FGvCD1cQa1M/PGknCLFIyZdtJOWQjejp7bdNqmN7zwdA==}
    dependencies:
      real-require: 0.1.0
    dev: false

  /thread-stream@2.4.1:
    resolution: {integrity: sha512-d/Ex2iWd1whipbT681JmTINKw0ZwOUBZm7+Gjs64DHuX34mmw8vJL2bFAaNacaW72zYiTJxSHi5abUuOi5nsfg==}
    dependencies:
      real-require: 0.2.0
    dev: false

  /through@2.3.8:
    resolution: {integrity: sha512-w89qg7PI8wAdvX60bMDP+bFoD5Dvhm9oLheFp5O4a2QF0cSBGsBX4qZmadPMvVqlLJBBci+WqGGOAPvcDeNSVg==}
    dev: false

  /timed-out@4.0.1:
    resolution: {integrity: sha512-G7r3AhovYtr5YKOWQkta8RKAPb+J9IsO4uVmzjl8AZwfhs8UcUwTiD6gcJYSgOtzyjvQKrKYn41syHbUWMkafA==}
    engines: {node: '>=0.10.0'}
    dev: false

  /titleize@3.0.0:
    resolution: {integrity: sha512-KxVu8EYHDPBdUYdKZdKtU2aj2XfEx9AfjXxE/Aj0vT06w2icA09Vus1rh6eSu1y01akYg6BjIK/hxyLJINoMLQ==}
    engines: {node: '>=12'}
    dev: true

  /to-fast-properties@2.0.0:
    resolution: {integrity: sha512-/OaKK0xYrs3DmxRYqL/yDc+FxFUVYhDlXMhRmv3z915w2HF1tnN1omB354j8VUGO/hbRzyD6Y3sA7v7GS/ceog==}
    engines: {node: '>=4'}

  /to-regex-range@5.0.1:
    resolution: {integrity: sha512-
65P7iz6X5yEr1cwcgvQxbbIw7Uk3gOy5dIdtZ4rDveLqhrdJP+Li/Hx6tyK0NEb+2GCyneCMJiGqrADCSNk8sQ
==}
    engines: {node: '>=8.0'}
    dependencies:
      is-number: 7.0.0

  /toad-cache@3.3.0:
    resolution: {integrity: sha512-
3oDzcogWGHZdkwrHyvJVpPjA7oNzY6ENOV3PsWJY9XYPZ6INo94Yd47s5may1U+nleBPwDhrRiTPMIvKa
a3MQg==}
    engines: {node: '>=12'}
    dev: false

  /toggle-selection@1.0.6:
    resolution: {integrity: sha512-
BiZS+C1OS8g/q2RRbJmy59xpyghNBqrr6k5L/uKBGRsTfxmu3ffiRnd8mlGPUVayg8pvfi5urfnu8TU7DVOkLQ
==}
    dev: false

  /toidentifier@1.0.1:
    resolution: {integrity: sha512-
o5sSPKEkg/DIQNmH43V0/uerLrpzVedkUh8tGNvaeXpfpuwjKenlSox/2O/BTlZUtEe+JG7s5YhEz608PlAHRA
==}
    engines: {node: '>=0.6'}
    dev: false

  /touch@3.1.0:
    resolution: {integrity: sha512-
WBx8Uy5TLtOSRtIq+M03/sKDrXCLHxwDcquSP2c43Le03/9serjQBIztjRz6FkJez9D/hleyAXTBGLwwZUw9l
A==}
    hasBin: true
    dependencies:
      nopt: 1.0.10
    dev: true

  /tough-cookie@2.5.0:
    resolution: {integrity: sha512-
nlLsUzgm1kfLXSXfRZMc1KLAugd4hqJHDTvc2hDIwS3mZAfMEuMbc03SujMF+GEcpaX/qboeycw6iO8JwV
v2+g==}
    engines: {node: '>=0.8'}
    dependencies:
      psl: 1.9.0
      punycode: 2.3.0
    dev: false

  /tr46@0.0.3:
    resolution: {integrity: sha512-
N3WMsuqV66lT30CrXNbEjx4GEwlow3v6rr4mCcv6prnfwhS01rkgyFdjPNBYd9br7LpXV1+Emh01fHnq2Gdgr
w==}

  /tree-kill@1.2.2:
    resolution: {integrity: sha512-L0Orpi8qGpRG//Nd+H90vFB+3iHnue1zSSGmNOOCh1GLJ7rUKVwV2HvijphGQS2UmhUZewS9VgvxYIdgr+fG1A==}
    hasBin: true
    dev: true

  /ts-api-utils@1.0.3(typescript@5.2.2):
    resolution: {integrity: sha512-wNMeqtMz5NtwpT/UZGY5alT+VoKdSsOOP/kqHFcUW1P/VRhH2wJ48+DN2WwUliNbQ976ETwDL0Ifd2VVvgonvg==}
    engines: {node: '>=16.13.0'}
    peerDependencies:
      typescript: '>=4.2.0'
    dependencies:
      typescript: 5.2.2
    dev: true

  /ts-command-line-args@2.5.1:
    resolution: {integrity: sha512-H69ZwTw3rFHb5WYpQya40YAX2/w7Ut75uUECbgBIsLmM+BNuYnxsltfyyLMxy6sEeKxgijLTnQtLd0nKd6+IYw==}
    hasBin: true
    dependencies:
      chalk: 4.1.2
      command-line-args: 5.2.1
      command-line-usage: 6.1.3
      string-format: 2.0.0

  /ts-essentials@7.0.3(typescript@5.2.2):
    resolution: {integrity: sha512-8+gr5+lqO3G84KdiTSMRLtuyJ+nTBVRKuCrK4lidMPdVeEp0uqC875uE5NMcaA7YYMN7XsNiFQuMvasF8HT/xQ==}
    peerDependencies:
      typescript: '>=3.7.0'
    dependencies:
      typescript: 5.2.2

  /ts-interface-checker@0.1.13:
    resolution: {integrity: sha512-Y/arvbn+rrz3JCKl9C4kVNfTfSm2/mEp5FSz5EsZSANGPSlQrpRI5M4PKF+mJnE52jOO90PnPSc3Ur3bTQw0gA==}
    dev: false

  /ts-node@10.9.1(@types/node@20.8.9)(typescript@5.2.2):
    resolution: {integrity: sha512-NtVysVPkxxrwFGUUxGYhfux8k78pQB3JqYBXlLRZgdGUqTO5wU/UyHop5p70iEbGhB7q5KmiZiU0Y3KlJrScEw==}
    hasBin: true
    peerDependencies:
      '@swc/core': '>=1.2.50'
      '@swc/wasm': '>=1.2.50'
      '@types/node': '*'
      typescript: '>=2.7'
    peerDependenciesMeta:
      '@swc/core':
        optional: true
      '@swc/wasm':
        optional: true
    dependencies:
      '@cspotcode/source-map-support': 0.8.1
      '@tsconfig/node10': 1.0.9
      '@tsconfig/node12': 1.0.11
      '@tsconfig/node14': 1.0.3
      '@tsconfig/node16': 1.0.4
      '@types/node': 20.8.9
      acorn: 8.11.2
      acorn-walk: 8.3.0
      arg: 4.1.3
      create-require: 1.1.1
      diff: 4.0.2
      make-error: 1.3.6
      typescript: 5.2.2
      v8-compile-cache-lib: 3.0.1
      yn: 3.1.1

  /tsconfig-paths@3.14.2:
    resolution: {integrity: sha512-o/9iXgCYc5L/JxCHPe3Hvh8Q/2xm5Z+p18PESBU6Ff33695QnCHBEjcytY2q19ua7Mbl/DavtBOLq+oG0RCL+g==}
    dependencies:
      '@types/json5': 0.0.29
      json5: 1.0.2
      minimist: 1.2.8
      strip-bom: 3.0.0
    dev: true

  /tslib@1.14.1:
    resolution: {integrity: sha512-Xni35NKzjgMrwevysHTCArtLDpPvye8zV/0E4EyYn43P7/7qvQwPh9BGkHewbMulVntbigmcT7rdX3BNo9wRJg==}
    dev: false

  /tslib@2.4.0:
    resolution: {integrity: sha512-d6xOpEDfsi2CZVlPQzGeux8XMwLT9hssAsaPYExaQMuYskwb+x1x7J371tWlbBdWHroy99KnVB6qIkUbs5X3UQ==}
    dev: false

  /tslib@2.4.1:
    resolution: {integrity: sha512-tGyy4dAjRIEwI7BzsB0lynWgOpfqjUdq91XXAlIWD2OwKBH7oCl/GZG/HT4BOHrTlPMOASlMQ7veyTqpmRc
```

```
rNA==}

  /tslib@2.6.2:
    resolution: {integrity: sha512-
AEYxH93jGFPn/a2iVAwW87VuUIkR1FVUKB77NwMF7nBTDkDrrT/Hpt/IrCJ0QXhW27jTBDcf5ZY7w6RiqT
Mw2Q==}
    dev: true

  /tunnel-agent@0.6.0:
    resolution: {integrity: sha512-
McnNiV1l8RYeY8tBgEpuodCC1mLUdbSN+CYBL7kJsJNInOP8UjDDEwdk6Mw60vdLLrr5NHKZhMAOSrR2
NZuQ+w==}
    dependencies:
      safe-buffer: 5.2.1
    dev: false

  /tweetnacl-util@0.15.1:
    resolution: {integrity: sha512-
RKJBIj8lySrShN4w6i/BonWp2Z/uxwC3h4y7xsRrpP59ZboCd0GpEVsOnMDYLMmKBpYhb5TgHzZXy7wTfY
FBRw==}
    dev: false

  /tweetnacl@0.14.5:
    resolution: {integrity: sha512-
KXXFFdAbFXY4geFIwoyNK+f5Z1b7swfXABfL7HXCmoIWMKU3dmS26672A4EeQtDzLKy7SXmfBu51JolvE
KwtGA==}
    dev: false

  /tweetnacl@1.0.3:
    resolution: {integrity: sha512-
6rt+RN7aOi1nGMyC4Xa5DdYiukl2UWCbcJft7YhxReBGQD7OAM8Pbxw6YMo4r2diNEA8FEmu32YOn9rhai
E5yw==}
    dev: false

  /type-check@0.4.0:
    resolution: {integrity: sha512-
XleUoc9uwGXqjWwXaUTZAmzMcFZ5858QA2vvx1Ur5xIcixXIP+8LnFDgRplU30us6teqdlskFfu+ae4K79Ooe
w==}
    engines: {node: '>= 0.8.0'}
    dependencies:
      prelude-ls: 1.2.1
    dev: true

  /type-fest@0.20.2:
    resolution: {integrity: sha512-
Ne+eE4r0/iWnpAxD852z3A+N0Bt5RN//NjJwRd2VFHEmrywxf5vsZlh4R6lixl6B+wz/8d+maTSAkN1FIkI3LQ==
}
    engines: {node: '>=10'}
    dev: true

  /type-fest@0.6.0:
    resolution: {integrity: sha512-
```

```
q+MB8nYR1KDLrgr4G5yemftpMC7/QLqVndBmEEdqzmNj5dcFOO4Oo8qlwZE3ULT3+Zim1F8Kq4cBnikNhl
CMlg==}
    engines: {node: '>=8'}
    dev: true

  /type-fest@0.8.1:
    resolution: {integrity: sha512-
4dbzIzqvjtgiM5rw1k5rEHtBANKmdudhGyBEajN01fEyhaAIhsoKNy6y7+IN93IfpFtwY9iqi7kD+xwKhQsNJA==
}
    engines: {node: '>=8'}
    dev: true

  /type-is@1.6.18:
    resolution: {integrity: sha512-
TkRKr9sUTxEH8MdfuCSP7VizJyzRNMjj2J2do2Jr3Kym598JVdEksuzPQCnlFPW4ky9Q+iA+ma9BGm06XQ
By8g==}
    engines: {node: '>= 0.6'}
    dependencies:
      media-typer: 0.3.0
      mime-types: 2.1.35
    dev: false

  /type@1.2.0:
    resolution: {integrity: sha512-
+5nt5AAniqsCnu2cEQQdpzCAh33kVx8n0VoFidKpB1dVVLAN/F+bgVOqOJqOnEnrhp222clB5p3vUlD+1QA
nfg==}
    dev: false

  /type@2.7.2:
    resolution: {integrity: sha512-
dzlvlNlt6AXU7EBSfpAscydQ7gXB+pPGsPnfJnZpiNJBDj7IaJzQlBZYGdEi4R9HmPdBv2XmWJ6YUtoTa7lm
Cw==}
    dev: false

  /typechain@8.3.2(typescript@5.2.2):
    resolution: {integrity: sha512-
x/sQYr5w9K7yv3es7jo4KTX05CLxOf7TRWwoHlrjRh8H82G64g+k7VuWPJlgMo6qrjfCulOdfBjiaDtmhFYD/Q
==}
    hasBin: true
    peerDependencies:
      typescript: '>=4.3.0'
    dependencies:
      '@types/prettier': 2.7.3
      debug: 4.3.4
      fs-extra: 7.0.1
      glob: 7.1.7
      js-sha3: 0.8.0
      lodash: 4.17.21
      mkdirp: 1.0.4
      prettier: 2.8.8
      ts-command-line-args: 2.5.1
      ts-essentials: 7.0.3(typescript@5.2.2)
      typescript: 5.2.2
    transitivePeerDependencies:
      - supports-color

  /typed-array-buffer@1.0.0:
    resolution: {integrity: sha512-
Y8KTSIglk9OZEr8zywiIHG/kmQ7KWyjseXs1CbSo8vC42w7hg2HgYTxSWwP0+is7bWDc1H+Fo026CpHFw
m8tkw==}
    engines: {node: '>= 0.4'}
    dependencies:
      call-bind: 1.0.5
      get-intrinsic: 1.2.2
      is-typed-array: 1.1.12
    dev: true

  /typed-array-byte-length@1.0.0:
    resolution: {integrity: sha512-
Or/+kvLxNpeQ9DtSydonMxCx+9ZXOswtwJn17SNLvhptaXYDJvkFFP5zbfU/uLmvnBJlI4yrnXRxpdWH/M5tN
A==}
    engines: {node: '>= 0.4'}
    dependencies:
      call-bind: 1.0.5
      for-each: 0.3.3
      has-proto: 1.0.1
      is-typed-array: 1.1.12
    dev: true

  /typed-array-byte-offset@1.0.0:
    resolution: {integrity: sha512-
RD97prjEt9EL8YgAgpOkf3O4IF9lhJFr9g0htQkm0rchFp/Vx7LW5Q8fSXXub7BXAODyUQohRMyOc3faCPd0
hg==}
    engines: {node: '>= 0.4'}
    dependencies:
      available-typed-arrays: 1.0.5
      call-bind: 1.0.5
      for-each: 0.3.3
      has-proto: 1.0.1
      is-typed-array: 1.1.12
    dev: true

  /typed-array-length@1.0.4:
    resolution: {integrity: sha512-
KjZypGq+I/H7HI5HlOoGHkWUUGq+Q0TPhQurLbyrVrvnKTBgzLhIJ7j6J/XTQOi0d1RjyZ0wdas8bKs2p0x3N
g==}
    dependencies:
      call-bind: 1.0.5
      for-each: 0.3.3
      is-typed-array: 1.1.12
    dev: true

  /typedarray-to-buffer@3.1.5:
    resolution: {integrity: sha512-
zdu8XMNEDepKKR+XYOXAVPtWui0ly0NtohUscw+UmaHiAWT8hrV1rr//H6V+0DvJ3OQ19S979M0laLfX8r
m82Q==}
    dependencies:
      is-typedarray: 1.0.0
    dev: false

  /typescript@5.2.2:
    resolution: {integrity: sha512-
mI4WrpHsbCIcwT9cF4FZvr80QUeKvsUsUvKDoR+X/7XHQH98xYD8YHZg7ANtz2GtZt/CBq2QJ0thkGJMHf
qc1w==}
    engines: {node: '>=14.17'}
    hasBin: true

  /typical@4.0.0:
    resolution: {integrity: sha512-
VAH4IvQ7BDFYglMd7BPRDfLgxZZX4O4TFcRDA6EN5X7erNJJq+McIEp8np9aVtxrCJ6qx4GTYVfOWNjcq
wZgRw==}
    engines: {node: '>=8'}

  /typical@5.2.0:
    resolution: {integrity: sha512-
dvdQgNDNJo+8B2uBQoqdb11eUCE1JQXhvjC/CZtgvZseVd5TYMXnq0+vuUemXbd/Se29cTaUuPX3YIc2xg
bvIg==}
    engines: {node: '>=8'}

  /ua-parser-js@1.0.37:
    resolution: {integrity: sha512-
bhTyI94tZofjo+Dn8SN6Zv8nBDvyXTymAdM3LDI/0IboIUwTu1rEhW7v2TfiVsoYWgkQ4kOVqnI8APUFbIQIF
Q==}
    dev: false

  /uint8arrays@3.1.1:
    resolution: {integrity: sha512-
+QJa8QRnbdXVpHYjLoTpJIdCTiw9Ir62nocClWuXIq2JIh4Uta0cQsTSpFL678p2CN8B+XSApwcU+pQEqVp
KWg==}
    dependencies:
      multiformats: 9.9.0
    dev: false

  /ultron@1.1.1:
    resolution: {integrity: sha512-
UIEXBNeYmKptWH6z8ZnqTeS8fV74zG0/eRU9VGkpzz+LIJNs8W/zM/L+7ctCkRrgbNnnR0xxw4bKOr0cW0
N0Og==}
    dev: false

  /unbox-primitive@1.0.2:
    resolution: {integrity: sha512-
61pPlCD9h51VoreyJ0BReideM3MDKMKnh6+V9L08331ipq6Q8OFXZYiqP6n/tbHx4s5I9uRhcye6BrbkizkBD
w==}
    dependencies:
      call-bind: 1.0.5
      has-bigints: 1.0.2
      has-symbols: 1.0.3
      which-boxed-primitive: 1.0.2
    dev: true

  /undefsafe@2.0.5:
    resolution: {integrity: sha512-
WxONCrssBM8TSPRqN5EmsjVrsv4A8X12J4ArBiiayv3DyyG3ZlIg6yysuuSYdZsVz3TKcTg2fd//Ujd4CHV1iA
==}
    dev: true

  /undici-types@5.26.5:
    resolution: {integrity: sha512-
JlCMO+ehdEIKqlFxk6IfVoAUVmgz7cU7zD/h9XZ0qzeosSHmUJVOzSQvvYSYWXkFXC+IfLKSIffhv0sVZup6
pA==}

  /universal-user-agent@6.0.0:
    resolution: {integrity: sha512-
isyNax3wXoKaulPDZWHQqbmIx1k2tb9fb3GGDBRxCscfYV2Ch7WxPArBsFEG8s/safwXTT7H4QGhaIkTp9
447w==}
    dev: true

  /universalify@0.1.2:
    resolution: {integrity: sha512-
rBJeI5CXAlmy1pV+617WB9J63U6XcazHHF2f2dbJix4XzpUF0RS3Zbj0FGIOCAva5P/d/GBOYaACQ1w+0az
Ukg==}
    engines: {node: '>= 4.0.0'}

  /unpipe@1.0.0:
    resolution: {integrity: sha512-
pjy2bYhSsufwWlKwPc+l3cN7+wuJlK6uz0YdJEOlQDbl6jo/YlPi4mb8agUkVC8BF7V8NuzeyPNqRksA3hztK
Q==}
    engines: {node: '>= 0.8'}
    dev: false

  /untildify@4.0.0:
    resolution: {integrity: sha512-
KK8xQ1mkzZeg9inewmFVDNkg3l5LUhoq9kN6iWYB/CC9YMG8HA+c1Q8HwDe6dEX7kErrEVNVBO3fWsV
q5iDgtw==}
    engines: {node: '>=8'}
    dev: true

  /update-browserslist-db@1.0.13(browserslist@4.22.1):
    resolution: {integrity: sha512-
xebP81SNcPuNpPP3uzeW1NYXxI3rxyJzF3pD6sH4jE7o/IX+WtSpwnVU+qIsDPyk0d3hmFQ7mjqc6AtV604
hbg==}
    hasBin: true
    peerDependencies:
      browserslist: '>= 4.21.0'
    dependencies:
      browserslist: 4.22.1
      escalade: 3.1.1
      picocolors: 1.0.0
  /uri-js@4.4.1:
    resolution: {integrity: sha512-
7rKUyy33Q1yc98pQ1DAmLtwX109F7TIfWlW1Ydo8Wl1ii1SeHieeh0HHfPeL2fMXK6z0s8ecKs9frCuLJvndB
g==}
    dependencies:
      punycode: 2.3.0

  /url-set-query@1.0.0:
    resolution: {integrity: sha512-
3AChu4NiXquPfeckE5R5cGdiHCMWJx1dwCWOmWIL4KHAziJNOFIYJlpGFeKDvwLPHovZRCxK3cYlwzqI
9Vp+Gg==}
    dev: false

  /use-composed-ref@1.3.0(react@18.2.0):
    resolution: {integrity: sha512-
GLMG0Jc/jiKov/3Ulid1wbv3r54K9HlMW29IWcDFPEqFkSO2nS0MuefWgMJpeHQ9YJeXDL3ZUF+P3jdXlZX
/cQ==}
    peerDependencies:
      react: ^16.8.0 || ^17.0.0 || ^18.0.0
    dependencies:
      react: 18.2.0
    dev: false

  /use-isomorphic-layout-effect@1.1.2(@types/react@18.0.25)(react@18.2.0):
    resolution: {integrity: sha512-
49L8yCO3iGT/ZF9QttjwLF/ZD9Iwto5LnH5LmEdk/6cFmXddqi2ulF0edxTwjj+7mqvpVVGQWvbXZdn32wRSH
A==}
    peerDependencies:
      '@types/react': '*'
      react: ^16.8.0 || ^17.0.0 || ^18.0.0
    peerDependenciesMeta:
      '@types/react':
        optional: true
    dependencies:
      '@types/react': 18.0.25
      react: 18.2.0
    dev: false

  /use-latest@1.2.1(@types/react@18.0.25)(react@18.2.0):
    resolution: {integrity: sha512-
xA+AVm/Wlg3e2P/JiItTziwS7FK92LWrDB0p+hgXloIMuVCeJJ8v6f0eeHyPZaJrM+usM1FkFfbNCrJGs8A/zw
==}
    peerDependencies:
      '@types/react': '*'
      react: ^16.8.0 || ^17.0.0 || ^18.0.0
    peerDependenciesMeta:
      '@types/react':
        optional: true
    dependencies:
      '@types/react': 18.0.25
      react: 18.2.0
      use-isomorphic-layout-effect: 1.1.2(@types/react@18.0.25)(react@18.2.0)
    dev: false

  /use-sync-external-store@1.2.0(react@18.2.0):
    resolution: {integrity: sha512-eEgnFxGQ1Ife9bzYs6VLi8/4X6CObHMw9Qr9tPY43iKwsPw8xE8+EFsf/2cFZ5S3esXgpWgtSCtLNS41F+sKPA==}
    peerDependencies:
      react: ^16.8.0 || ^17.0.0 || ^18.0.0
    dependencies:
      react: 18.2.0
    dev: false

  /utf-8-validate@5.0.10:
    resolution: {integrity: sha512-Z6czzLq4u8fPOyx7TU6X3dvUZVvoJmxSQ+IcrlmagKhilxlhZgxPK6C5Jqbkw1IDUmFTM+cz9QDnnLTwDz/2gQ==}
    engines: {node: '>=6.14.2'}
    requiresBuild: true
    dependencies:
      node-gyp-build: 4.5.0
    dev: false

  /utf8@3.0.0:
    resolution: {integrity: sha512-E8VjFIQ/TyQgp+TZfS6l8yp/xWppSAHzidGiRrqe4bK4XP9pTRyKFgGJpO3SN7zdX4DeomTrwaseCHovfpFcqQ==}
    dev: false

  /util-deprecate@1.0.2:
    resolution: {integrity: sha512-EPD5q1uXyFxJpCrLnCc1nHnq3gOa6DZBocAIiI2TaSCA7VCJ1UJDMagCzIkXNsUYfD1daK//LTEQ8xiIbrHtcw==}
    dev: false

  /util@0.12.5:
    resolution: {integrity: sha512-kZf/K6hEIrWHI6XqOFUiiMa+79wE/D8Q+NCNAWclkyg3b4d2k7s0QGepNjiABc+aR3N1PAyHL7p6UcLY6LmrnA==}
    dependencies:
      inherits: 2.0.4
      is-arguments: 1.1.1
      is-generator-function: 1.0.10
      is-typed-array: 1.1.10
      which-typed-array: 1.1.9
    dev: false

  /utils-merge@1.0.1:
    resolution: {integrity: sha512-pMZTvIkT1d+TFGvDOqodOclx0QWkkgi6Tdoa8gC8ffGAAqz9pzPTZWAybbsHHoED/ztMtkv/VoYTYyShUn81hA==}
    engines: {node: '>= 0.4.0'}
    dev: false

  /uuid@3.4.0:
    resolution: {integrity: sha512-
HjSDRw6gZE5JMggctHBcjVak08+KEVhSIiDzFnT9S9aegmp85S/bReBVTb4QTFaRNptJ9kuYaNhnbNEOkb
Kb/A==}
    deprecated: Please upgrade  to version 7 or higher.  Older versions may use Math.random() in certain
circumstances, which is known to be problematic.  See https://v8.dev/blog/math-random for details.
    hasBin: true
    dev: false

  /uuid@8.3.2:
    resolution: {integrity: sha512-
+NYs2QeMWy+GWFOEm9xnn6HCDp0l7QBD7ml8zLUmJ+93Q5NF0NocErnwkTkXVFNiX3/fpC6afS8Dhb/g
z7R7eg==}
    hasBin: true
    dev: false

  /uuid@9.0.1:
    resolution: {integrity: sha512-
b+1eJOlsR9K8HJpow9Ok3fiWOWSIcIzXodvv0rQjVoOVNpWMpxf1wZNpt4y9h10odCNrqnYp1OBzRktckBe
3sA==}
    hasBin: true
    dev: false

  /v8-compile-cache-lib@3.0.1:
    resolution: {integrity: sha512-
wa7YjyUGfNZngI/vtK0UHAN+lgDCxBPCylVXGp0zu59Fz5aiGtNXaq3DhIov063MorB+VfufLh3JlF2KdTK3xg
==}

  /valid-url@1.0.9:
    resolution: {integrity: sha512-
QQDsV8OnSf5Uc30CKSwG9lnhMPe6exHtTXLRYX8uMwKENy640pU+2BgBL0LRbDh/eYRahNCS7aewCx
0wf3NYVA==}
    dev: false

  /validate-npm-package-license@3.0.4:
    resolution: {integrity: sha512-
DpKm2Ui/xN7/HQKCtpZxoRWBhZ9Z0kqtygG8XCgNQ8ZlDnxuQmWhj566j8fN4Cu3/JmbhsDo7fcAJq4s9h2
7Ew==}
    dependencies:
      spdx-correct: 3.2.0
      spdx-expression-parse: 3.0.1
    dev: true

  /valtio@1.10.5(react@18.2.0):
    resolution: {integrity: sha512-
jTp0k63VXf4r5hPoaC6a6LCG4POkVSh629WLi1+d5PlajLsbynTMd7qAgEiOSPxzoX5iNvbN7iZ/k/g29wrNiQ
==}
    engines: {node: '>=12.20.0'}
    peerDependencies:
      react: '>=16.8'
    peerDependenciesMeta:
      react:
        optional: true
    dependencies:
      proxy-compare: 2.5.1
      react: 18.2.0
      use-sync-external-store: 1.2.0(react@18.2.0)
    dev: false

  /valtio@1.11.2(@types/react@18.0.25)(react@18.2.0):
    resolution: {integrity: sha512-
1XfIxnUXzyswPAPXo1P3Pdx2mq/pIqZICkWN60Hby0d9Iqb+MEIpqgYVlbflvHdrp2YR/q3jyKWRPJJ100yxaw
==}
    engines: {node: '>=12.20.0'}
    peerDependencies:
      '@types/react': '>=16.8'
      react: '>=16.8'
    peerDependenciesMeta:
      '@types/react':
        optional: true
      react:
        optional: true
    dependencies:
      '@types/react': 18.0.25
      proxy-compare: 2.5.1
      react: 18.2.0
      use-sync-external-store: 1.2.0(react@18.2.0)
    dev: false

  /varint@5.0.2:
    resolution: {integrity: sha512-
lKxKYG6H03yCZUpAGOPOsMcGxd1RHCu1iKvEHYDPmTyq2HueGhD73ssNBqqQWfvYs04G9iUFRvmAV
LW20Jw6ow==}
    dev: false

  /vary@1.1.2:
    resolution: {integrity: sha512-
BNGbWLfd0eUPabhkXUVm0j8uuvREyTh5ovRa/dyow/BqAbZJyC+5fU+IzQOzmAKzYqYRAISoRhdQr3eIZ/
PXqg==}
    engines: {node: '>= 0.8'}
    dev: false

  /verror@1.10.0:
    resolution: {integrity: sha512-
ZZKSmDAEFOijERBLkmYfJ+vmk3w+7hOLYDNkRCuRuMJGEmqYNCNLyBBFwWKVMhfwaEF3WOd0Zlw
86U/WC/+nYw==}
    engines: {'0': node >=0.6.0}
    dependencies:
      assert-plus: 1.0.0
      core-util-is: 1.0.2
      extsprintf: 1.3.0
    dev: false
```

```
 /viem@0.3.50(typescript@5.2.2):
   resolution: {integrity: sha512-
s+LxCYZTR9F/qPk1/n1YDVAX9vSeVz7GraqBZWGrDuenCJxo9ArCoIceJ6ksI0WwSeNzcZ0VVbD/kWRzTx
kipw==}
   dependencies:
     '@adraffy/ens-normalize': 1.9.0
     '@noble/curves': 1.0.0
     '@noble/hashes': 1.3.0
     '@scure/bip32': 1.3.0
     '@scure/bip39': 1.2.0
     '@wagmi/chains': 1.0.0(typescript@5.2.2)
     abitype: 0.8.7(typescript@5.2.2)
     isomorphic-ws: 5.0.0(ws@8.12.0)
     ws: 8.12.0
   transitivePeerDependencies:
     - bufferutil
     - typescript
     - utf-8-validate
     - zod
   dev: false

 /viem@1.17.2(typescript@5.2.2):
   resolution: {integrity: sha512-
lsF4qqo5cu9lIE9g3736MIY2Qzeb/V/BebB4f0DaVuujQVebA/myHaplZGLjAqjPqCf9iui1VRZ6YCQsQ1FIBw==
}
   peerDependencies:
     typescript: '>=5.0.4'
   peerDependenciesMeta:
     typescript:
       optional: true
   dependencies:
     '@adraffy/ens-normalize': 1.9.4
     '@noble/curves': 1.2.0
     '@noble/hashes': 1.3.2
     '@scure/bip32': 1.3.2
     '@scure/bip39': 1.2.1
     abitype: 0.9.8(typescript@5.2.2)
     isows: 1.0.3(ws@8.13.0)
     typescript: 5.2.2
     ws: 8.13.0(bufferutil@4.0.7)(utf-8-validate@5.0.10)
   transitivePeerDependencies:
     - bufferutil
     - utf-8-validate
     - zod
   dev: false

 /vite-plugin-mkcert@1.16.0(vite@4.5.0):
   resolution: {integrity: sha512-
5r+g8SB9wZzLNUFekGwZo3e0P6QlS6rbxK5p9z/itxNAimsYohgjK/YfVPVxM9EuglP9hjridq0lUejo9v1nVg==}
   engines: {node: '>=v16.7.0'}
   peerDependencies:
     vite: '>=3'
   dependencies:
     '@octokit/rest': 19.0.13
     axios: 1.6.0(debug@4.3.4)
     debug: 4.3.4
     picocolors: 1.0.0
     vite: 4.5.0(@types/node@20.8.9)
   transitivePeerDependencies:
     - encoding
     - supports-color
   dev: true

 /vite@4.5.0(@types/node@20.8.9):
   resolution: {integrity: sha512-ulr8rNLA6rkyFAlVWw2q5YJ91v098AFQ2R0PRFwPzREXOUJQPtFUG0t+/ZikhaOCDqFoDhN6/v8Sq0o4araFAw==}
   engines: {node: ^14.18.0 || >=16.0.0}
   hasBin: true
   peerDependencies:
     '@types/node': '>= 14'
     less: '*'
     lightningcss: ^1.21.0
     sass: '*'
     stylus: '*'
     sugarss: '*'
     terser: ^5.4.0
   peerDependenciesMeta:
     '@types/node':
       optional: true
     less:
       optional: true
     lightningcss:
       optional: true
     sass:
       optional: true
     stylus:
       optional: true
     sugarss:
       optional: true
     terser:
       optional: true
   dependencies:
     '@types/node': 20.8.9
     esbuild: 0.18.20
     postcss: 8.4.31
     rollup: 3.29.4
   optionalDependencies:
     fsevents: 2.3.2
   dev: true

 /wagmi@1.4.5(@types/react@18.0.25)(react-dom@18.2.0)(react@18.2.0)(typescript@5.2.2)(viem@0.3.50):
   resolution: {integrity: sha512-
Ph62E6cO5n2Z8Z5LTyZrkaNprxTsbC4w0qZJT4OJdXrEELziI8z/b4FO6amVFXdu2rDp/wpvF56e4mhKC8/K
dw==}
   peerDependencies:
     react: '>=17.0.0'
     typescript: '>=5.0.4'
     viem: '>=0.3.35'
   peerDependenciesMeta:
     typescript:
       optional: true
   dependencies:
     '@tanstack/query-sync-storage-persister': 4.36.1
     '@tanstack/react-query': 4.36.1(react-dom@18.2.0)(react@18.2.0)
     '@tanstack/react-query-persist-client': 4.36.1(@tanstack/react-query@4.36.1)
     '@wagmi/core': 1.4.5(@types/react@18.0.25)(react@18.2.0)(typescript@5.2.2)(viem@0.3.50)
     abitype: 0.8.7(typescript@5.2.2)
     react: 18.2.0
     typescript: 5.2.2
     use-sync-external-store: 1.2.0(react@18.2.0)
     viem: 0.3.50(typescript@5.2.2)
   transitivePeerDependencies:
     - '@react-native-async-storage/async-storage'
     - '@types/react'
     - bufferutil
     - encoding
     - immer
     - lokijs
     - react-dom
     - react-native
     - supports-color
     - utf-8-validate
     - zod
   dev: false

 /web3-bzz@1.10.0:
   resolution: {integrity: sha512-
o9IR59io3pDUsXTsps5pO5hW1D5zBmg46iNc2t4j2DkaYHNdDLwk2IP9ukoM2wg47QILfPEJYzhTfkS/CcX0
KA==}
   engines: {node: '>=8.0.0'}
   requiresBuild: true
   dependencies:
     '@types/node': 12.20.55
     got: 12.1.0
     swarm-js: 0.1.42
   transitivePeerDependencies:
     - bufferutil
     - supports-color
     - utf-8-validate
   dev: false

 /web3-core-helpers@1.10.0:
   resolution: {integrity: sha512-
pIxAzFDS5vnbXvfvLSpaA1tfRykAe9adw43YCKsEYQwH0gCLL0kMLkaCX3q+Q8EVmAh+e1jWL/nl9U0de1
+++g==}
   engines: {node: '>=8.0.0'}
   dependencies:
     web3-eth-iban: 1.10.0
     web3-utils: 1.10.0
   dev: false

 /web3-core-method@1.10.0:
   resolution: {integrity: sha512-
4R700jTLAMKDMhQ+nsVfIXvH6IGJlJzGisIfMKWAIswH31h5AZz7uDUW2YctI+HrYd+5uOAlS4OJeeT9bIpv
kA==}
   engines: {node: '>=8.0.0'}
   dependencies:
     '@ethersproject/transactions': 5.7.0
     web3-core-helpers: 1.10.0
     web3-core-promievent: 1.10.0
     web3-core-subscriptions: 1.10.0
     web3-utils: 1.10.0
   dev: false

 /web3-core-promievent@1.10.0:
   resolution: {integrity: sha512-
68N7k5LWL5R38xRaKFrTFT2pm2jBNFaM4GioS00YjAKXRQ3KjmhijOMG3TICz6Aa5+6GDWYeIDNx21YA
eZ4YTg==}
   engines: {node: '>=8.0.0'}
   dependencies:
     eventemitter3: 4.0.4
   dev: false

 /web3-core-requestmanager@1.10.0:
   resolution: {integrity: sha512-
3z/JKE++Os62APml4dvBM+GAuId4h3L9ckUrj7ebEtS2AR0ixyQPbrBodgL91Sv7j7cQ3Y+hllaluqjguxvSaQ==
}
   engines: {node: '>=8.0.0'}
   dependencies:
     util: 0.12.5
     web3-core-helpers: 1.10.0
     web3-providers-http: 1.10.0
     web3-providers-ipc: 1.10.0
     web3-providers-ws: 1.10.0
   transitivePeerDependencies:
     - encoding
     - supports-color
   dev: false

 /web3-core-subscriptions@1.10.0:
   resolution: {integrity: sha512-
HGm1PbDqsxejI075gxBc5OSkwymilRWZufIy9zEpnWKNmfbuv5FfHgW1/chtJP6aP3Uq2vHkvTDl3smQBb8l
+g==}
   engines: {node: '>=8.0.0'}
   dependencies:
     eventemitter3: 4.0.4
     web3-core-helpers: 1.10.0
   dev: false

 /web3-core@1.10.0:
   resolution: {integrity: sha512-
fWySwqy2hn3TL89w5TM8wXF1Z2Q6frQTKHWmP0ppRQorEK8NcHJRfeMiv/mQlSKoTS1F6n/nv2uyZsixFy
cjYQ==}
   engines: {node: '>=8.0.0'}
   dependencies:
     '@types/bn.js': 5.1.3
     '@types/node': 12.20.55
     bignumber.js: 9.1.2
     web3-core-helpers: 1.10.0
     web3-core-method: 1.10.0
     web3-core-requestmanager: 1.10.0
     web3-utils: 1.10.0
   transitivePeerDependencies:
     - encoding
     - supports-color
   dev: false

 /web3-eth-abi@1.10.0:
   resolution: {integrity: sha512-
cwS+qRBWpJ43aI9L3JS88QYPfFcSJJ3XapxOQ4j40v6mk7ATpA8CVK1vGTzpihNlOfMVRBkR95oAj7oL6ai
DOg==}
   engines: {node: '>=8.0.0'}
   dependencies:
     '@ethersproject/abi': 5.7.0
     web3-utils: 1.10.0
   dev: false

 /web3-eth-accounts@1.10.0:
   resolution: {integrity: sha512-
wiq39Uc3mOI8rw24wE2n15hboLE0E9BsQLdlmsL4Zua9diDS6B5abXG0XhFcoNsXIGMWXVZz4TOq3u4Ed
pXF/Q==}
   engines: {node: '>=8.0.0'}
   dependencies:
     '@ethereumjs/common': 2.5.0
     '@ethereumjs/tx': 3.3.2
     eth-lib: 0.2.8
     ethereumjs-util: 7.1.5
     scrypt-js: 3.0.1
     uuid: 9.0.1
     web3-core: 1.10.0
     web3-core-helpers: 1.10.0
     web3-core-method: 1.10.0
     web3-utils: 1.10.0
   transitivePeerDependencies:
     - encoding
     - supports-color
   dev: false
```

```
 /web3-eth-contract@1.10.0:
   resolution: {integrity: sha512-
MIC5FOzP/+2evDksQQ/dpcXhSqa/2hFNytdl/x61IeWxhh6vlFeSjq0YVTAyIzdjwnL7nEmZpjfI6y6/Ufhy7w==}
   engines: {node: '>=8.0.0'}
   dependencies:
     '@types/bn.js': 5.1.3
     web3-core: 1.10.0
     web3-core-helpers: 1.10.0
     web3-core-method: 1.10.0
     web3-core-promievent: 1.10.0
     web3-core-subscriptions: 1.10.0
     web3-eth-abi: 1.10.0
     web3-utils: 1.10.0
   transitivePeerDependencies:
     - encoding
     - supports-color
   dev: false

 /web3-eth-ens@1.10.0:
   resolution: {integrity: sha512-
3hpGgzX3qjgxNAmqdrC2YUQMTfnZbs4GeLEmy8aCWziVwogbuqQZ+Gzdfrym45eOZodk+lmXyLuAdqkNlv
kc1g==}
   engines: {node: '>=8.0.0'}
   dependencies:
     content-hash: 2.5.2
     eth-ens-namehash: 2.0.8
     web3-core: 1.10.0
     web3-core-helpers: 1.10.0
     web3-core-promievent: 1.10.0
     web3-eth-abi: 1.10.0
     web3-eth-contract: 1.10.0
     web3-utils: 1.10.0
   transitivePeerDependencies:
     - encoding
     - supports-color
   dev: false

 /web3-eth-iban@1.10.0:
   resolution: {integrity: sha512-
0l+SP3IGhInw7Q20LY3IVafYEuufo4Dn75jAHT7c2aDJsIolvf2Lc6ugHkBajlwUneGfbRQs/ccYPQ9JeMUbrg==
}
   engines: {node: '>=8.0.0'}
   dependencies:
     bn.js: 5.2.1
     web3-utils: 1.10.0
   dev: false

 /web3-eth-personal@1.10.0:
   resolution: {integrity: sha512-
anseKn98w/d703eWq52uNuZi7GhQeVjTC5/svrBWEKob0WZ5kPdo+EZoFN0sp5a5ubbrk/E0xSl1/M5yORMt
pg==}
```

```
    engines: {node: '>=8.0.0'}
    dependencies:
      '@types/node': 12.20.55
      web3-core: 1.10.0
      web3-core-helpers: 1.10.0
      web3-core-method: 1.10.0
      web3-net: 1.10.0
      web3-utils: 1.10.0
    transitivePeerDependencies:
      - encoding
      - supports-color
    dev: false

  /web3-eth@1.10.0:
    resolution: {integrity: sha512-
Z5vT6slNMLPKuwRyKGbqeGYC87OAy8bOblaqRTgg94CXcn/mmqU7iPIlG4506YdcdK3x6cfEDG7B6w+jRx
ypKA==}
    engines: {node: '>=8.0.0'}
    dependencies:
      web3-core: 1.10.0
      web3-core-helpers: 1.10.0
      web3-core-method: 1.10.0
      web3-core-subscriptions: 1.10.0
      web3-eth-abi: 1.10.0
      web3-eth-accounts: 1.10.0
      web3-eth-contract: 1.10.0
      web3-eth-ens: 1.10.0
      web3-eth-iban: 1.10.0
      web3-eth-personal: 1.10.0
      web3-net: 1.10.0
      web3-utils: 1.10.0
    transitivePeerDependencies:
      - encoding
      - supports-color
    dev: false

  /web3-net@1.10.0:
    resolution: {integrity: sha512-
NLH/N3IshYWASpxk4/18Ge6n60GEvWBVeM8inx2dmZJVmRI6SJIlUxbL8jySgiTn3MMZlhbdvrGo8fpUW7a1
GA==}
    engines: {node: '>=8.0.0'}
    dependencies:
      web3-core: 1.10.0
      web3-core-method: 1.10.0
      web3-utils: 1.10.0
    transitivePeerDependencies:
      - encoding
      - supports-color
    dev: false

  /web3-provider-engine@16.0.3(@babel/core@7.23.2):
    resolution: {integrity: sha512-
Q3bKhGqLfMTdLvkd4TtkGYJHcoVQ82D1l8jTIwwuJp/sAp7VHnRYb9YJ14SW/69VMWoOhSpPLZV2tWb9V
0WJoA==}
    engines: {node: '>=12.0.0'}
    dependencies:
      '@ethereumjs/tx': 3.5.2
      async: 2.6.4
      backoff: 2.5.0
      clone: 2.1.2
      cross-fetch: 2.2.6
      eth-block-tracker: 4.4.3(@babel/core@7.23.2)
      eth-json-rpc-filters: 4.2.2
      eth-json-rpc-infura: 5.1.0
      eth-json-rpc-middleware: 6.0.0
      eth-rpc-errors: 3.0.0
      eth-sig-util: 1.4.2
      ethereumjs-block: 1.7.1
      ethereumjs-util: 5.2.1
      ethereumjs-vm: 2.6.0
      json-stable-stringify: 1.0.2
      promise-to-callback: 1.0.0
      readable-stream: 2.3.8
      request: 2.88.2
      semaphore: 1.1.0
      ws: 5.2.3
      xhr: 2.6.0
      xtend: 4.0.2
    transitivePeerDependencies:
      - '@babel/core'
      - bufferutil
      - encoding
      - supports-color
      - utf-8-validate
    dev: false

  /web3-providers-http@1.10.0:
    resolution: {integrity: sha512-
eNr965YB8a9mLiNrkjAWNAPXgmQWfpBfkkn7tpEFlghfww0u3I0tktMZiaToJVcL2+Xq+81cxbkpeWJ5XQDw
OA==}
    engines: {node: '>=8.0.0'}
    dependencies:
      abortcontroller-polyfill: 1.7.5
      cross-fetch: 3.1.5
      es6-promise: 4.2.8
      web3-core-helpers: 1.10.0
    transitivePeerDependencies:
      - encoding
    dev: false

  /web3-providers-ipc@1.10.0:
    resolution: {integrity: sha512-
OfXG1aWN8L1OUqppshzq8YISkWrYHaATW9H8eh0p89TlWMc1KZOL9vttBuaBEi96D/n0eYDn2trzt22bqH
WfXA==}
```

```
    engines: {node: '>=8.0.0'}
    dependencies:
      oboe: 2.1.5
      web3-core-helpers: 1.10.0
    dev: false

  /web3-providers-ws@1.10.0:
    resolution: {integrity: sha512-
sK0fNcglW36yD5xjnjtSGBnEtf59cbw4vZzJ+CmOWIKGIR96mP5l684g0WD0Eo+f4NQc2anWWXG74lRc9O
VMCQ==}
    engines: {node: '>=8.0.0'}
    dependencies:
      eventemitter3: 4.0.4
      web3-core-helpers: 1.10.0
      websocket: 1.0.34
    transitivePeerDependencies:
      - supports-color
    dev: false

  /web3-shh@1.10.0:
    resolution: {integrity: sha512-
uNUUuNsO2AjX41GJARV9zJibs11eq6HtOe6Wr0FtRUcj8SN6nHeYIzwstAvJ4fXA53gRqFMTxdntHEt9aXVj
pg==}
    engines: {node: '>=8.0.0'}
    requiresBuild: true
    dependencies:
      web3-core: 1.10.0
      web3-core-method: 1.10.0
      web3-core-subscriptions: 1.10.0
      web3-net: 1.10.0
    transitivePeerDependencies:
      - encoding
      - supports-color
    dev: false

  /web3-utils@1.10.0:
    resolution: {integrity: sha512-
kSaCM0uMcZTNUSmn5vMEhlo02RObGNRRCkdX0V9UTAU0+lrvn0HSaudyCo6CQzuXUsnuY2ERJGCGP
feWmv19Rg==}
    engines: {node: '>=8.0.0'}
    dependencies:
      bn.js: 5.2.1
      ethereum-bloom-filters: 1.0.10
      ethereumjs-util: 7.1.5
      ethjs-unit: 0.1.6
      number-to-bn: 1.7.0
      randombytes: 2.1.0
      utf8: 3.0.0
    dev: false

  /web3@1.10.0:
    resolution: {integrity: sha512-
YfKY9wSkGcM8seO+daR89oVTcbu18NsVfvOngzqMYGUU0pPSQmE57qQDvQzUeoIOHAnXEBNzrhjQJm
m8ER0rng==}
    engines: {node: '>=8.0.0'}
    requiresBuild: true
    dependencies:
      web3-bzz: 1.10.0
      web3-core: 1.10.0
      web3-eth: 1.10.0
      web3-eth-personal: 1.10.0
      web3-net: 1.10.0
      web3-shh: 1.10.0
      web3-utils: 1.10.0
    transitivePeerDependencies:
      - bufferutil
      - encoding
      - supports-color
      - utf-8-validate
    dev: false

  /webidl-conversions@3.0.1:
    resolution: {integrity: sha512-
2JAn3z8AR6rjK8Sm8orRC0h/bcl/DqL7tRPdGZ4I1CjdF+EaMLmYxBHyXuKL849eucPFhvBoxMsflfOb8kxae
Q==}

  /websocket@1.0.34:
    resolution: {integrity: sha512-
PRDso2sGwF6kM75QykIesBijKSVceR6jL2G8NGYyq2XrItNC2P5/qL5XeR056GhA+Ly7JMFvJb9I312mJfmq
nQ==}
    engines: {node: '>=4.0.0'}
    dependencies:
      bufferutil: 4.0.7
      debug: 2.6.9
      es5-ext: 0.10.62
      typedarray-to-buffer: 3.1.5
      utf-8-validate: 5.0.10
      yaeti: 0.0.6
    transitivePeerDependencies:
      - supports-color
    dev: false

  /whatwg-fetch@2.0.4:
    resolution: {integrity: sha512-
dcQ1GWpOD/eEQ97k66aiEVpNnapVj90/+R+SXTPYGHpYBBypfKJEQjLrvMZ7YXbKm21gXd4NcuxUTjiv1Y
tLng==}
    dev: false

  /whatwg-url@5.0.0:
    resolution: {integrity: sha512-
saE57nupxk6v3HY35+jzBwYa0rKSy0XR8JSxZPwgLr7ys0IBzhGviA1/TUGJLmSVqs8pb9AnvICXEuOHLprY
Tw==}
    dependencies:
      tr46: 0.0.3
```

```
   webidl-conversions: 3.0.1

 /which-boxed-primitive@1.0.2:
  resolution: {integrity: sha512-
bwZdv0AKLpplFY2KZRX6TvyuN7ojjr7lwkg6ml0roIy9YeuSr7JS372qlNW18UQYzgYK9ziGcerWqZOmEn9V
Ng==}
  dependencies:
   is-bigint: 1.0.4
   is-boolean-object: 1.1.2
   is-number-object: 1.0.7
   is-string: 1.0.7
   is-symbol: 1.0.4
  dev: true

 /which-builtin-type@1.1.3:
  resolution: {integrity: sha512-
YmjsSMDBYsM1CaFiayOVT06+KJeXf0o5M/CAd4o1lTadFAtacTUM49zoYxr/oroopFDfhvN6iEcBxUyc3gvK
mw==}
  engines: {node: '>= 0.4'}
  dependencies:
   function.prototype.name: 1.1.6
   has-tostringtag: 1.0.0
   is-async-function: 2.0.0
   is-date-object: 1.0.5
   is-finalizationregistry: 1.0.2
   is-generator-function: 1.0.10
   is-regex: 1.1.4
   is-weakref: 1.0.2
   isarray: 2.0.5
   which-boxed-primitive: 1.0.2
   which-collection: 1.0.1
   which-typed-array: 1.1.9
  dev: true

 /which-collection@1.0.1:
  resolution: {integrity: sha512-
W8xeTUwaln8i3K/cY1nGXzdnVZlidBcagyNFtBdD5kxnb4TvGKR7FfSIS3mYpwWS1QUCutfKz8IY8RjftB0+1
A==}
  dependencies:
   is-map: 2.0.2
   is-set: 2.0.2
   is-weakmap: 2.0.1
   is-weakset: 2.0.2
  dev: true

 /which-module@2.0.0:
  resolution: {integrity: sha512-
B+enWhmw6cjfVC7kS8Pj9pCrKSc5txArRyaYGe088shv/FGWH+0Rjx/xPgtsWfsUtS27FkP697E4DDhgrgoc0
Q==}
  dev: false

 /which-typed-array@1.1.13:
    resolution: {integrity: sha512-
P5Nra0qjSncduVPEAr7xhoF5guty49ArDTwzJ/yNuPIbZppyRxFQsRCWrocxIY+CnMVG+qfbU2FmDKyvSGC
low==}
    engines: {node: '>= 0.4'}
    dependencies:
      available-typed-arrays: 1.0.5
      call-bind: 1.0.5
      for-each: 0.3.3
      gopd: 1.0.1
      has-tostringtag: 1.0.0
    dev: true

  /which-typed-array@1.1.9:
    resolution: {integrity: sha512-
w9c4xkx6mPidwp7180ckYWfMmvxpjlZuIudNtDf4N/tTAUB8VJbX25qZoAsrtGuYNnGw3pa0AXgbGKRB8/Ec
eA==}
    engines: {node: '>= 0.4'}
    dependencies:
      available-typed-arrays: 1.0.5
      call-bind: 1.0.2
      for-each: 0.3.3
      gopd: 1.0.1
      has-tostringtag: 1.0.0
      is-typed-array: 1.1.10

  /which@2.0.2:
    resolution: {integrity: sha512-
BLI3Tl1TW3Pvl70l3yq3Y64i+awpwXqsGBYWkkqMtnbXgrMD+yj7rhW0kuEDxzJaYXGjEW5ogapKNMEKNM
jibA==}
    engines: {node: '>= 8'}
    hasBin: true
    dependencies:
      isexe: 2.0.0
    dev: true

  /wordwrapjs@4.0.1:
    resolution: {integrity: sha512-
kKlNACbvHrkpIw6oPeYDSmdCTu2hdMHoyXLTcUKala++lx5Y+wjJ/e474Jqv5abnVmwxw08DiTuHmw69lJG
ksA==}
    engines: {node: '>=8.0.0'}
    dependencies:
      reduce-flatten: 2.0.0
      typical: 5.2.0

  /wrap-ansi@6.2.0:
    resolution: {integrity: sha512-
r6lPcBGxZXlIcymEu7InxDMhdW0KDxpLgoFLcguasxCaJ/SOIZwINatK9KY/tf+ZrlywOKU0UDj3ATXUBfxJXA
==}
    engines: {node: '>=8'}
    dependencies:
      ansi-styles: 4.3.0
      string-width: 4.2.3
      strip-ansi: 6.0.1
    dev: false

  /wrap-ansi@7.0.0:
    resolution: {integrity: sha512-
YVGIj2kamLSTxw6NsZjoBxfSwsn0ycdesmc4p+Q21c5zPuZ1pl+NfxVdxPtdHvmNVOQ6XSYG4AUtyt/Fi7D1
6Q==}
    engines: {node: '>=10'}
    dependencies:
      ansi-styles: 4.3.0
      string-width: 4.2.3
      strip-ansi: 6.0.1
    dev: true

  /wrappy@1.0.2:
    resolution: {integrity: sha512-
l4Sp/DRseor9wL6EvV2+TuQn63dMkPjZ/sp9XkghTEbV9KlPS1xUsZ3u7/IQO4wxtcFB4bgpQPRcR3QCvezP
cQ==}

  /ws@3.3.3:
    resolution: {integrity: sha512-
nnWLa/NwZSt4KQJu51MYlCcSQ5g7INpOrOMt4XV8j4dqTXdmlUmSHQ8/oLC069ckre0fRsgfvsKwbTdtKLC
DkA==}
    peerDependencies:
      bufferutil: ^4.0.1
      utf-8-validate: ^5.0.2
    peerDependenciesMeta:
      bufferutil:
        optional: true
      utf-8-validate:
        optional: true
    dependencies:
      async-limiter: 1.0.1
      safe-buffer: 5.1.2
      ultron: 1.1.1
    dev: false

  /ws@5.2.3:
    resolution: {integrity: sha512-
jZArVERrMsKUatIdnLzqvcfydI85dvd/Fp1u/VOpfdDWQ4c9qWXe+VIeAbQ5FrDwciAkr+lzofXLz3Kuf26AOA==
}
    peerDependencies:
      bufferutil: ^4.0.1
      utf-8-validate: ^5.0.2
    peerDependenciesMeta:
      bufferutil:
        optional: true
      utf-8-validate:
        optional: true
    dependencies:
      async-limiter: 1.0.1
    dev: false
  /ws@7.5.9:
    resolution: {integrity: sha512-
F+P9Jil7UiSKSkppIiD94dN07AwvFixvLIj1Og1Rl9GGMuNipJnV9JzjD6XuqmAeiswGvUmNLjr5cFuXwNS77
Q==}
    engines: {node: '>=8.3.0'}
    peerDependencies:
      bufferutil: ^4.0.1
      utf-8-validate: ^5.0.2
    peerDependenciesMeta:
      bufferutil:
        optional: true
      utf-8-validate:
        optional: true
    dev: false

  /ws@8.12.0:
    resolution: {integrity: sha512-
kU62emKIdKVeEIOIKVegvqpXMSTAMLJozpHZaJNDYqBjzlSYXQGviYwN1osDLJ9av68qHd4a2oSjd7yD4p
acig==}
    engines: {node: '>=10.0.0'}
    peerDependencies:
      bufferutil: ^4.0.1
      utf-8-validate: '>=5.0.2'
    peerDependenciesMeta:
      bufferutil:
        optional: true
      utf-8-validate:
        optional: true
    dev: false

  /ws@8.13.0(bufferutil@4.0.7)(utf-8-validate@5.0.10):
    resolution: {integrity: sha512-
x9vcZYTrFPC7aSIbj7sRCYo7L/Xb8Iy+pW0ng0wt2vCJv7M9HOMy0UoN3rr+IFC7hb7vXoqS+P9ktyLLLhO+
LA==}
    engines: {node: '>=10.0.0'}
    peerDependencies:
      bufferutil: ^4.0.1
      utf-8-validate: '>=5.0.2'
    peerDependenciesMeta:
      bufferutil:
        optional: true
      utf-8-validate:
        optional: true
    dependencies:
      bufferutil: 4.0.7
      utf-8-validate: 5.0.10
    dev: false

  /ws@8.5.0:
    resolution: {integrity: sha512-
BWX0SWVgLPzYwF8lTzEy1egjhS4S4OEAHfsO8o65WOVsrnSRGaSiUaa9e0ggGlkMTtBlmOpEXiie9RUcB
O86qg==}
    engines: {node: '>=10.0.0'}
    peerDependencies:
      bufferutil: ^4.0.1
      utf-8-validate: ^5.0.2
    peerDependenciesMeta:
      bufferutil:
        optional: true
      utf-8-validate:
        optional: true
    dev: false

  /xhr-request-promise@0.1.3:
    resolution: {integrity: sha512-
YUBytBsuwgitWtdRzXDDkWAXzhdGB8bYm0sSzMPZT7Z2MBjMSTHFsyCT1yCRATY+XC69DUrQraRAEg
coCRaIPg==}
    dependencies:
      xhr-request: 1.1.0
    dev: false

  /xhr-request@1.1.0:
    resolution: {integrity: sha512-
Y7qzEaR3FDtL3fP30k9wO/e+FBnBByZeybKOhASsGP30NIkRAAkKD/sCnLvgEfAIEC1rcmK7YG8f4oEnIrr
WzA==}
    dependencies:
      buffer-to-arraybuffer: 0.0.5
      object-assign: 4.1.1
      query-string: 5.1.1
      simple-get: 2.8.2
      timed-out: 4.0.1
      url-set-query: 1.0.0
      xhr: 2.6.0
    dev: false

  /xhr@2.6.0:
    resolution: {integrity: sha512-
/eCGLb5rxjx5e3mF1A7s+pLlR6CGyqWN91fv1JgER5mVWg1MZmlhBvy9kjcsOdRk8RrIujotWyJamfyrp+WIc
A==}
    dependencies:
      global: 4.4.0
      is-function: 1.0.2
      parse-headers: 2.0.5
      xtend: 4.0.2
    dev: false

  /xtend@2.1.2:
    resolution: {integrity: sha512-
vMNKzr2rHP9Dp/e1NQFnLQlwlhp9L/LfvnsVdHxN1f+uggyVI3i08uD14GPvCToPkdsRfyPqIyYGmIk58V98Z
Q==}
    engines: {node: '>=0.4'}
    dependencies:
      object-keys: 0.4.0
    dev: false

  /xtend@4.0.2:
    resolution: {integrity: sha512-LKYU1iAXJXUgAXn9URjiu+MWhyUXHsvfp7mcuYm9dSUKK0/CjtrUwFAxD82/mCWbtLsGjFIad0wIsod4zrTAEQ==}
    engines: {node: '>=0.4'}
    dev: false

  /y18n@4.0.3:
    resolution: {integrity: sha512-JKhqTOwSrqNA1NY5lSztJ1GrBiUodLMmIZuLiDaMRJ+itFd+ABVE8XBjOvIWL+rSqNDC74LCSFmlb/U4UZ4hJQ==}
    dev: false

  /y18n@5.0.8:
    resolution: {integrity: sha512-0pfFzegeDWJHJIAmTLRP2DwHjdF5s7jo9tuztdQxAhINCdvS+3nGINqPd00AphqJR/0LhANUS6/+7SCb98YOfA==}
    engines: {node: '>=10'}
    dev: true

  /yaeti@0.0.6:
    resolution: {integrity: sha512-MvQa//+KcZCUkBTIC9blM+CU9J2GzuTytsOUwf2lidtvkx/6gnEp1QvJv34t9vdjhFmha/mUiNDbN0D0mJWdug==}
    engines: {node: '>=0.10.32'}
    dev: false

  /yallist@3.1.1:
    resolution: {integrity: sha512-a4UGQaWPH59mOXUYnAG2ewncQS4i4F43Tv3JoAM+s2VDAmS9NsK8GpDMLrCHPksFT7h3K6TOoUNn2pb7RoXx4g==}

  /yallist@4.0.0:
    resolution: {integrity: sha512-3wdGidZyq5PB084XLES5TpOSRA3wjXAlIWMhum2kRcv/41Sn2emQ0dycQW4uZXLejwKvg6EsvbdlVL+FYEct7A==}

  /yaml@2.3.3:
    resolution: {integrity: sha512-zw0VAJxgeZ6+++/su5AFoqBbZbrEakwu+X0M5HmcwUiBL7AzcuPKjj5we4xfQLp78LkEMpD0cOnUhmgOVy3KdQ==}
    engines: {node: '>= 14'}
    dev: false

  /yargs-parser@18.1.3:
    resolution: {integrity: sha512-o50j0JeToy/4K6OZcaQmW6lyXXKhq7csREXcDwk2omFPJEwUNOVtJKvmDr9EI1fAJZUyZcRF7kxGBWmRXudrCQ==}
    engines: {node: '>=6'}
    dependencies:
      camelcase: 5.3.1
      decamelize: 1.2.0
    dev: false

  /yargs-parser@21.1.1:
    resolution: {integrity: sha512-
tVpsJW7DdjecAiFpbIB1e3qxIQsE6NoPc5/eTdrbbIC4h0LVsWhnoa3g+m2HclBIujHzsxZ4VJVA+GUuc2/LBw
==}
    engines: {node: '>=12'}
    dev: true

  /yargs@15.4.1:
    resolution: {integrity: sha512-
aePbxDmcYW++PaqBsJ+HYUFwCdv4LVvdnhBy78E57PIor8/OVvhMrADFFEDh8DHDFRv/O9i3lPhsENjO7
QX0+A==}
    engines: {node: '>=8'}
    dependencies:
      cliui: 6.0.0
      decamelize: 1.2.0
      find-up: 4.1.0
      get-caller-file: 2.0.5
      require-directory: 2.1.1
      require-main-filename: 2.0.0
      set-blocking: 2.0.0
      string-width: 4.2.3
      which-module: 2.0.0
      y18n: 4.0.3
      yargs-parser: 18.1.3
    dev: false

  /yargs@17.7.2:
    resolution: {integrity: sha512-
7dSzzRQ++CKnNI/krKnYRV7JKKPUXMEh61soaHKg9mrWEhzFWhFnxPxGl+69cD1Ou63C13NUPCnmIcrv
qCuM6w==}
    engines: {node: '>=12'}
    dependencies:
      cliui: 8.0.1
      escalade: 3.1.1
      get-caller-file: 2.0.5
      require-directory: 2.1.1
      string-width: 4.2.3
      y18n: 5.0.8
      yargs-parser: 21.1.1
    dev: true

  /yn@3.1.1:
    resolution: {integrity: sha512-
Ux4ygGWsu2c7isFWe8Yu1YluJmqVhxqK2cLXNQA5AcC3QfbGNpM7fu0Y8b/z16pXLnFxZYvWhd3fhBY9D
LmC6Q==}
    engines: {node: '>=6'}

  /yocto-queue@0.1.0:
    resolution: {integrity: sha512-rVksvsnNCdJ/ohGc6xgPwyN8eheCxsiLM8mxuE/t/mOVqJewPuO1miLpTHQiRgTKCLexL4MeAFVagts7HmNZ2Q==}
    engines: {node: '>=10'}

  /zustand@4.4.4(@types/react@18.0.25)(react@18.2.0):
    resolution: {integrity: sha512-5UTUIAiHMNf5+mFp7/AnzJXS7+XxktULFN0+D1sCiZWyX7ZG+AQpqs2qpYrynRij4QvoDdCD+U+bmg/cG3Ucxw==}
    engines: {node: '>=12.7.0'}
    peerDependencies:
      '@types/react': '>=16.8'
      immer: '>=9.0'
      react: '>=16.8'
    peerDependenciesMeta:
      '@types/react':
        optional: true
      immer:
        optional: true
      react:
        optional: true
    dependencies:
      '@types/react': 18.0.25
      react: 18.2.0
      use-sync-external-store: 1.2.0(react@18.2.0)
    dev: false

  github.com/ethereumjs/ethereumjs-abi/ee3994657fa7a427238e6ba92a84d0b529bbcde0:
    resolution: {tarball: https://codeload.github.com/ethereumjs/ethereumjs-abi/tar.gz/ee3994657fa7a427238e6ba92a84d0b529bbcde0}
    name: ethereumjs-abi
    version: 0.6.8
    dependencies:
      bn.js: 4.12.0
      ethereumjs-util: 6.2.1
    dev: false
```

</file>

<file>

path: /polyfills.ts

url: https://github.com/richtera/lukso-siwe-demo/blob/main/polyfills.ts

```
import { Buffer } from 'buffer'
import process from 'process'

window.global = window
window.process = process
window.Buffer = Buffer
```

</file>

<file>

path: /postcss.config.js

url: https://github.com/richtera/lukso-siwe-demo/blob/main/postcss.config.js

```
const data = {
  plugins: {
    tailwindcss: {},
    autoprefixer: {},
  },
};

export default data;
```

</file>

<file>

path: /public/browserconfig.xml

url: https://github.com/richtera/lukso-siwe-demo/blob/main/public/browserconfig.xml

```
<?xml version="1.0" encoding="utf-8"?>
<browserconfig>
    <msapplication>
        <tile>
            <square150x150logo src="/mstile-150x150.png"/>
            <TileColor>#da532c</TileColor>
        </tile>
    </msapplication>
</browserconfig>
```

</file>

<file>

path: /public/site.webmanifest

url: https://github.com/richtera/lukso-siwe-demo/blob/main/public/site.webmanifest

```
{
  "name": "",
  "short_name": "",
  "icons": [
    {
      "src": "/android-chrome-192x192.png",
      "sizes": "192x192",
      "type": "image/png"
    },
    {
      "src": "/android-chrome-512x512.png",
      "sizes": "512x512",
      "type": "image/png"
    }
  ],
  "theme_color": "#ffffff",
  "background_color": "#ffffff",
  "display": "standalone"
}
```

</file>

<file>

path: /server/index.ts

url: https://github.com/richtera/lukso-siwe-demo/blob/main/server/index.ts

```
/**
 * Welcome to Cloudflare Workers! This is your first worker.
 *
 * - Run "npm run dev" in your terminal to start a development server
 * - Open a browser tab at http://localhost:8787/ to see your worker in action
 * - Run "npm run deploy" to publish your worker
 *
 * Learn more at https://developers.cloudflare.com/workers/
 */
import fastifyCors from '@fastify/cors';
import formBody from '@fastify/formbody';
import fastifyStatic from '@fastify/static';
import base64url from 'base64url';
import { config } from 'dotenv';
import { JsonRpcProvider, getAddress, hexlify, HDNodeWallet } from 'ethers';
import Fastify from 'fastify';
import { exportJWK, importSPKI } from 'jose';
import KeyEncoder from 'key-encoder';
import crypto from 'node:crypto';
import path from 'node:path';
import { SiweMessage } from 'siwe';
import { hashMessage, verifyMessage } from 'viem';
import { LSP0ERC725AccountInit__factory } from
'../libs/contracts/factories/LSP0ERC725AccountInit__factory.js';

config();

const fastify = Fastify({
  logger: true,
  ...(process.env.ENABLE_HTTP2 === 'true' ? { http2: true } : {}),
});
await fastify.register(formBody);
await fastify.register(fastifyCors, {
  // put your options here
});
// eslint-disable-next-line unicorn/no-null
fastify.decorateRequest('user', null);
// eslint-disable-next-line unicorn/no-null
fastify.decorateRequest('tags', null);
// eslint-disable-next-line unicorn/no-null
fastify.decorateRequest('scope', null);

// now let's create a router (note the lack of "new")
const provider = new JsonRpcProvider('https://rpc.testnet.lukso.network');

if (!process.env.VITE_MNEMONIC) {
  const wallet = HDNodeWallet.createRandom();
  const phrase = wallet.mnemonic?.phrase;
  fastify.log.info(phrase);
  fastify.log.info(
    &nbsp;Please configure your .env with\nVITE_MNEMONIC=${phrase}\nor any other mnemonic you want
to use.\nStarting up with random new phrase above.\nThis private/public key pair is used to sign and verify
server tokens.&nbsp;,
  );
  process.env.VITE_MNEMONIC = phrase;
}

const wallet = HDNodeWallet.fromPhrase(process.env.VITE_MNEMONIC || '');

async function getJwks() {
  const KE = (KeyEncoder as any).default;
  const keyEncoder = new KE('secp256k1');
  const pem = keyEncoder.encodePublic(wallet.publicKey.slice(2), 'raw', 'pem');
  const pk = await importSPKI(pem as string, 'ES256K');
  const jwk = await exportJWK(pk);
  return [{ ...jwk, kid: '1' }];
}

fastify.get('/.well-known/walletconnect.txt', async (_request, reply) => {
  reply
    .code(200)
    .header('Content-Type', 'text/plain; charset=utf-8')
    .send(process.env.VITE_WALLET_CONNECT_VERIFY || '');
});
```

```
fastify.get('/.well-known/jwks', async (_request, reply) => {
  reply
    .code(200)
    .header('Content-Type', 'application/json; charset=utf-8')
    .send(await getJwks());
});

fastify.get('/.well-known/public-key', async (_request, reply) => {
  reply
    .code(200)
    .header('Content-Type', 'application/json; charset=utf-8')
    .send({
      address: wallet.address,
      publicKey: wallet.publicKey,
    });
});

/**
 * Exchange user message for a server message and jwt
 *
 * @param siwe user siwe message
 * @param uri
 * @param nonce
 * @param domain
 * @param address
 * @returns
 */
async function exchangeToken(
  siwe: SiweMessage,
  uri: string,
  nonce: string,
  domain: string,
  address: string,
) {
  const set = new Set([
    ...(siwe.resources || []),
    siwe.uri,
    'did:web:' + siwe.domain,
    'did:account:' + siwe.address,
  ]);
  set.delete('did:account:' + getAddress(wallet.address));
  const newMessage = new SiweMessage({
    ...siwe,
    uri,
    domain,
    address,
    resources: [...set.values()],
    issuedAt: new Date().toISOString(),
    expirationTime: new Date(Date.now() + 30 * 24 * 60 * 60_000).toISOString(),
    nonce,
  });
  const message = newMessage.prepareMessage();
  const signature = await wallet.signMessage(message);
  const header = {
    alg: 'ES256K',
    typ: 'JWT',
    kid: '1',
  };
  return {
    jwt: [
      Buffer.from(JSON.stringify(header)),
      Buffer.from(message),
      Buffer.from(signature.slice(2), 'hex'),
    ]
      .map((d) => base64url(d, 'base64'))
      .join('.'),
    message: newMessage,
    signature: hexlify(signature),
  };
}

/**
 *
 * @param token jwt token to decode
 * @returns decoded siwe message
 */
async function decodeToken(token: string) {
  if (!token) {
    throw new Error('No token');
  }
  const [_header, message, signature] = token.split('.').map((segment) => {
    return base64url.toBuffer(segment);
  });
  if (!signature || !message || !_header) {
    throw new Error('No token');
  }
  const header = JSON.parse(_header.toString());
  if (header.alg !== 'ES256K' || header.typ !== 'JWT') {
    throw new Error('Invalid token');
  }

  const messageString = message.toString();

  const message_ = new SiweMessage(messageString);
  if (message_.address === wallet.address) {
    // This is a server token
    const signature_ = hexlify(signature) as &nbsp;0x${string}&nbsp;;
    if (
      !(await verifyMessage({
        message: messageString,
        signature: signature_,
        address: message_.address as &nbsp;0x${string}&nbsp;,
      }))
    ) {
      throw new Error('Invalid token');
    }
    const account = (
      message_.resources?.find((resource) =>
        resource.startsWith('did:account:'),
      ) || ''
    ).replace(/^did:account:/, '');
    return { message: message_, account, valid: true, signature: signature_ };
  }
  const account = getAddress(message_.address);
  const contract = LSP0ERC725AccountInit__factory.connect(account, provider);
  const value = await contract.isValidSignature(
    hashMessage(messageString),
    hexlify(signature),
  );
  if (value !== '0x1626ba7e') {
    throw new Error('Invalid token');
  }
  return {
    account,
    message: message_,
    valid: true,
    signature: hexlify(signature),
  };
}

/**
 * Verify user or server token
 *
 * GET with bearer token in authorization header
 *
 * Returns { message: SiweMessage, valid: boolean } as json
 */
fastify.get('/verify', async (request, reply) => {
  const authorization = request.headers.authorization?.replace(/^bearer /i, '');
  if (!authorization) {
    reply.code(401).send({
      error: 'invalid_request',
      error_description: 'Missing authorization header',
    });
    return;
  }
  try {
    return await decodeToken(authorization);
  } catch (error) {
    reply.code(401).send({
      error: (error as any).message,
      valid: false,
      error_description: 'Invalid authorization header',
    });
  }
```

```
});

/**
 * Exchange a user token for a server token
 *
 * GET with bearer token in authorization header
 *
 * Returns { message: SiweMessage, originalMessage: SiweMessage, jwt: string } as JSON
 */
fastify.get('/exchange', async (request, reply) => {
  const authorization = request.headers.authorization?.replace(/^bearer /i, '');
  if (!authorization) {
    reply.code(401).send({
      error: 'invalid_request',
      valid: false,
      error_description: 'Missing authorization header',
    });
    return;
  }
  try {
    const {
      message,
      account,
      signature: originalSignature,
    } = await decodeToken(authorization);
    const {
      jwt,
      message: newMessage,
      signature,
    } = await exchangeToken(
      message,
      &nbsp;https://${process.env.VITE_APP_DOMAIN}&nbsp;,
      Buffer.from(crypto.getRandomValues(new Uint8Array(32))).toString('hex'),
      process.env.VITE_APP_DOMAIN || '',
      wallet.address,
    );
    return {
      message: newMessage,
      originalMessage: message,
      originalSignature,
      jwt: jwt,
      signature,
      account,
    };
  } catch (error) {
    fastify.log.error(error);
    reply.code(401).send({
      error: (error as any).message,
      valid: false,
      error_description: 'Invalid authorization header',
    });
  }
```

```
});

const root = path.resolve(process.env.BASE_DIR || './dist');

fastify.register(
  (childContext, _, done) => {
    childContext.register(fastifyStatic, {
      root,
      wildcard: true,
      index: 'index.html',
      prefix: '/',
      redirect: true,
    });
    childContext.setNotFoundHandler((_, reply) => {
      return reply.code(200).type('text/html').sendFile('index.html', root);
    });
    done();
  },
  { prefix: '/' },
);

async function run() {
  const port = process.env.PORT ? Number.parseInt(process.env.PORT, 10) : 3000;
  await fastify.listen({
    host: '::',
    port,
    listenTextResolver: (address) => &nbsp;Listening on ${address}&nbsp;,
  });
}

// eslint-disable-next-line unicorn/prefer-top-level-await
run().catch((error) => {
  fastify.log.error(error);
  // eslint-disable-next-line unicorn/no-process-exit
  process.exit(1);
});
```

</file>

<file>

path: /src/App.tsx

url: https://github.com/richtera/lukso-siwe-demo/blob/main/src/App.tsx

```
import { Web3Button } from '@web3modal/react';
import { useAccount } from 'wagmi';

import { SignMessage } from './components/SignMessage';

export function App() {
  const { isConnected } = useAccount();

  return (
    <div className="container mx-auto p-4 bg-slate-300">
      <nav className="flex justify-between items-center">
        <Web3Button />
      </nav>
      <h1 className="text-lg">
        Web3 Auth: Using Universal Profile and SIWE for Login
      </h1>
      <h2 className="text-md">
        This webpage shows several basic functions needed to use Universal
        Profiles and SIWE for authentication, authorization and verification.
        Create signatures as JWT tokens, validate them from client to server and
        server to client.{' '}
        {isConnected ? (
          ''
        ) : (
          <span className="text-red bg-gray-500">
            Please connect the Universal Profile Extension.
          </span>
        )}
      </h2>

      {isConnected && <SignMessage />}
    </div>
  );
}
```

</file>

<file>

path: /src/components/ClientVerifyJwt.tsx

url: https://github.com/richtera/lukso-siwe-demo/blob/main/src/components/ClientVerifyJwt.tsx

```
import { useCallback, useState } from 'react';
import JSONView from 'react-json-view';
import { useAccount } from 'wagmi';

import { verifyTokenOnClient } from '../utils/signing';

export function ClientVerifyJwt({
  jwt,
  title,
}: {
  jwt: string;
  title: string;
}) {
  const [output, setOutput] = useState<any>();
  const account = useAccount();
  const onClick = useCallback(async () => {
    if (!account.address) return;
    const output = await verifyTokenOnClient(account.address, jwt);
    setOutput(output);
  }, [jwt, account]);
  return (
    <div className="shadow-lg rounded-2xl m-3 p-5 bg-white">
      <h3 className="text-md">{title}</h3>
      <button
        className="rounded border-solid border-2 border-sky-500 disabled:text-gray-300 disabled:border-gray-200 disabled:bg-gray-100 px-2"
        disabled={!jwt}
        onClick={onClick}
      >
        Verify
      </button>
      <hr />
      <div className="w-full overflow-auto">
        <JSONView name="result" src={output} />
      </div>
    </div>
  );
}
```

</file>

<file>

path: /src/components/CreateJwt.tsx

url: https://github.com/richtera/lukso-siwe-demo/blob/main/src/components/CreateJwt.tsx

```
import { FormEvent, useCallback, useContext, useState } from 'react';
import JSONView from 'react-json-view';
import { useAccount, useChainId } from 'wagmi';
import { SignMessageContext } from './SignMessage';
import { createJwt } from '../utils/signing';

export function CreateJwt({ title }: { title: string }) {
  const account = useAccount();
  const chainId = useChainId();
  const [output, setOutput] = useState<any>();
  const { setData } = useContext(SignMessageContext);
  const onSubmit = useCallback(
    async (event: FormEvent<HTMLFormElement>) => {
      event.preventDefault();
      if (!account.address) {
        return;
      }
      const element = event.target as HTMLFormElement;
      const formData = new FormData(element);
      const statement = formData.get('message') as string;
      const { message, jwt, signature } = await createJwt(
        account.address as &nbsp;0x${string}&nbsp;,
        chainId,
        statement,
      );
      setData((value: any) => ({
        ...value,
        client: { ...value?.client, jwt },
      }));
      setOutput({ message, jwt, signature, statement });
    },
    [account.address, chainId, setData],
  );
  return (
    <div className="shadow-lg rounded-2xl m-3 p-5 bg-white">
      <h3 className="text-md">{title}</h3>
      <form onSubmit={(event) => onSubmit(event)}>
        <input
          className="rounded border-solid border-2 border-sky-500 w-1/2"
          name="message"
          type="text"
          placeholder="By logging in, you confirm the terms and conditions"
          required
        />
        <button
          className="rounded border-solid border-2 border-sky-500 disabled:text-gray-300 disabled:border-gray-200 disabled:bg-gray-100 px-2"
          disabled={!account.address}
          type="submit"
        >
          {account.address ? 'Login' : 'Check Wallet'}
        </button>
      </form>
      <hr />
      <div className="w-full overflow-auto">
```

```
        <JSONView name="result" src={output} />
      </div>
    </div>
  );
}
```

</file>

<file>

path: /src/components/ExchangeToken.tsx

url: https://github.com/richtera/lukso-siwe-demo/blob/main/src/components/ExchangeToken.tsx

```
import { useCallback, useState } from 'react';
import JSONView from 'react-json-view';
import { useAccount } from 'wagmi';

import { exchangeToken } from '../utils/signing';

export function ExchangeToken({
  jwt,
  title,
  onToken,
}: {
  jwt: string;
  title: string;
  onToken: (token: string) => void;
}) {
  const [output, setOutput] = useState<any>();
  const account = useAccount();
  const onClick = useCallback(async () => {
    if (!account.address) return;
    const output = await exchangeToken(jwt);
    if (output.jwt) {
      onToken?.(output.jwt);
    }
    setOutput(output);
  }, [account.address, jwt, onToken]);
  return (
    <div className="shadow-lg rounded-2xl m-3 p-5 bg-white">
      <h3 className="text-md">{title}</h3>
      <button
        className="rounded border-solid border-2 border-sky-500 disabled:text-gray-300 disabled:border-gray-200 disabled:bg-gray-100 px-2"
        disabled={!jwt}
        onClick={onClick}
      >
        Exchange
      </button>
      <hr />
      <div className="w-full overflow-auto">
        <JSONView name="result" src={output} />
      </div>
    </div>
  );
}
```

</file>

<file>

path: /src/components/ServerVerifyJwt.tsx

url: https://github.com/richtera/lukso-siwe-demo/blob/main/src/components/ServerVerifyJwt.tsx

```
import { useCallback, useState } from 'react';
import JSONView from 'react-json-view';
import { useAccount } from 'wagmi';

import { verifyTokenOnServer } from '../utils/signing';

export function ServerVerifyJwt({
  jwt,
  title,
}: {
  jwt: string;
  title: string;
}) {
  const [output, setOutput] = useState<any>();
  const account = useAccount();
  const onClick = useCallback(async () => {
    if (!account.address) return;
    const output = await verifyTokenOnServer(jwt);
    setOutput(output);
  }, [jwt, account]);
  return (
    <div className="shadow-lg rounded-2xl m-3 p-5 bg-white">
      <h3 className="text-md">{title}</h3>
      <button
        className="rounded border-solid border-2 border-sky-500 disabled:text-gray-300 disabled:border-gray-200 disabled:bg-gray-100 px-2"
        disabled={!jwt}
        onClick={onClick}
      >
        Verify
      </button>
      <hr />
      <div className="w-full overflow-auto">
        <JSONView name="result" src={output} />
      </div>
    </div>
  );
}
```

</file>

<file>

path: /src/components/ShowSession.tsx

url: https://github.com/richtera/lukso-siwe-demo/blob/main/src/components/ShowSession.tsx

```
import { useCallback, useContext } from 'react';
import JSONView from 'react-json-view';

import { SignMessageContext } from './SignMessage';

export function ShowSession() {
  const { data, setData } = useContext(SignMessageContext);
  const onClick = useCallback(() => {
    setData({});
  }, [setData]);
  return (
    <div className="shadow-lg rounded-2xl m-3 p-5 bg-white">
      <h3 className="text-md">Session (localStorage)</h3>
      <button
        className="rounded border-solid border-2 border-sky-500 disabled:text-gray-300 disabled:border-gray-200 disabled:bg-gray-100 px-2"
        disabled={Object.keys(data).length === 0}
        onClick={onClick}
      >
        Clear
      </button>
      <div className="w-full overflow-auto">
        <JSONView name="localStorage.session" src={data} />
      </div>
    </div>
  );
}
```

</file>

<file>

path: /src/components/SignMessage.tsx

url: https://github.com/richtera/lukso-siwe-demo/blob/main/src/components/SignMessage.tsx

```
import { createContext, useCallback, useState } from 'react';

import { ClientVerifyJwt } from './ClientVerifyJwt';
import { CreateJwt } from './CreateJwt';
import { ExchangeToken } from './ExchangeToken';
import { ServerVerifyJwt } from './ServerVerifyJwt';
import { ShowSession } from './ShowSession';

export const SignMessageContext = createContext<{
  data: any;
  setData: ReturnType<typeof useState<any>>[1];
}>({
  data: {},
  setData: () => {},
});
```

```
export function SignMessage() {
  const [data, setData_] = useState<any>(() => {
    return JSON.parse(localStorage.getItem('sessionData') || '{}');
  });
  const setData = useCallback((value: any) => {
    setData_((data: any) => {
      const newData = typeof value === 'function' ? value(data) : value;
      localStorage.setItem('sessionData', JSON.stringify(newData));
      return newData;
    });
  }, []);
  return (
    <SignMessageContext.Provider value={{ data, setData }}>
      <CreateJwt title="Sign Client Token" /> <hr />
      <ClientVerifyJwt
        jwt={data?.client?.jwt}
        title="Verify Client token on Client"
      />
      <ServerVerifyJwt
        jwt={data?.client?.jwt}
        title="Verify Client token on Server"
      />
      <ExchangeToken
        jwt={data?.client?.jwt}
        onToken={(jwt) =>
          setData((value: any) => ({ ...value, server: { jwt } }))
        }
        title="Retrieve Server token from Client token"
      />
      <ClientVerifyJwt
        jwt={data?.server?.jwt}
        title="Verify Server token on Client"
      />
      <ExchangeToken
        jwt={data?.server?.jwt}
        onToken={(jwt) =>
          setData((value: any) => ({ ...value, server: { jwt } }))
        }
        title="Refresh Server token"
      />
      <ServerVerifyJwt
        jwt={data?.server?.jwt}
        title="Verify Server token on Server"
      />
      <ShowSession />
    </SignMessageContext.Provider>
  );
}
```

</file>

<file>

path: /src/hooks/useDebounce.ts

url: https://github.com/richtera/lukso-siwe-demo/blob/main/src/hooks/useDebounce.ts

```
import { useEffect, useState } from 'react'

export function useDebounce<T>(value: T, delay?: number): T {
  const [debouncedValue, setDebouncedValue] = useState<T>(value)

  useEffect(() => {
    const timer = setTimeout(() => setDebouncedValue(value), delay ?? 500)
    return () => clearTimeout(timer)
  }, [value, delay])

  return debouncedValue
}
```

</file>

<file>

path: /src/main.tsx

url: https://github.com/richtera/lukso-siwe-demo/blob/main/src/main.tsx

```
import { EthereumClient } from '@web3modal/ethereum';
import { Web3Modal } from '@web3modal/react';
import * as React from 'react';
import * as ReactDOM from 'react-dom/client';
import { WagmiConfig } from 'wagmi';
import './style.css';

import { App } from './App';
import { config, chains, walletConnectProjectId } from './wagmi';

const ethereumClient = new EthereumClient(config, chains);

ReactDOM.createRoot(document.querySelector('#root')!).render(
  <React.StrictMode>
    <WagmiConfig config={config}>
      <App />
      <Web3Modal
        projectId={walletConnectProjectId}
        ethereumClient={ethereumClient}
      />
    </WagmiConfig>
  </React.StrictMode>,
);
```

</file>

<file>

path: /src/style.css

url: https://github.com/richtera/lukso-siwe-demo/blob/main/src/style.css

```
@tailwind base;
@tailwind components;
@tailwind utilities;
```

</file>

<file>

path: /src/utils/signing.ts

url: https://github.com/richtera/lukso-siwe-demo/blob/main/src/utils/signing.ts

```
import base64url from 'base64url';
import { SiweMessage } from 'siwe';
import {
  getAddress,
  getContract,
  hashMessage,
  recoverMessageAddress,
  toHex,
} from 'viem';
import { Address } from 'wagmi';
import { signMessage } from 'wagmi/actions';

import { config } from '../wagmi';

/**
 * Convert the SiweMessage to a JWT
 * Unfortunately the SIWE standard only signs the message part instead of
 * allowing for the header to be included, but an improper header will also
 * cause the signature not to validate so that should not be a problem.
 *
 * @param message
 * @param signature
 * @returns jwt token string
 */
export async function createJwt(
  account: Address,
  chainId: number,
  statement: string,
): Promise<{ message: SiweMessage; jwt: string; signature: &nbsp;0x${string}&nbsp; }> {
  const message_ = new SiweMessage({
    domain: import.meta.env.VITE_APP_DOMAIN,
    uri: &nbsp;https://${import.meta.env.VITE_APP_DOMAIN}&nbsp;,
    statement,
    address: account,
    issuedAt: new Date().toISOString(),
    version: '1',
    expirationTime: new Date(Date.now() + 10_000).toISOString(),
    chainId,
    resources: [
      &nbsp;did:account:${account}&nbsp;,
      &nbsp;did:web:${import.meta.env.VITE_APP_DOMAIN}&nbsp;,
    ],
  });
  const message = message_.prepareMessage();
  const signature = await signMessage({
    message,
  });
  const header = {
    alg: 'ES256K',
    typ: 'JWT',
  };
  return {
    jwt: [
      Buffer.from(JSON.stringify(header)),
      Buffer.from(message),
      Buffer.from(signature.slice(2), 'hex'),
    ]
      .map((d) => base64url(d, 'base64'))
      .join('.'),
    message: message_,
    signature,
  };
}

/**
 * Exchange a client side JWT to a server side JWT
 *
 * @param clientJwt
 * @returns {
 *   message: SiweMessage // new server side message
 *   originalMessage: SiweMessage // original client side message
 *   jwt: string // new server side JWT
 *   valid: boolean // is the client side JWT was valid (always true)
 *   account: &nbsp;0x${string}&nbsp; // user account address
 *   error?: string // error message
 *   error_description?: string // error description
 * }
 */
export async function exchangeToken(clientJwt: string): Promise<{
  message?: SiweMessage;
  originalMessage?: SiweMessage;
  jwt?: string;
  valid: boolean;
  account?: &nbsp;0x${string}&nbsp;;
  error?: string;
  error_description?: string;
}> {
  return await fetch('/exchange', {
    headers: {
      Authorization: &nbsp;Bearer ${clientJwt}&nbsp;,
    },
  })
    .then((response) => response.json())
    .then(({ message, originalMessage, account, ...rest }) => {
      return {
        message: new SiweMessage(message),
        originalMessage: new SiweMessage(originalMessage),
        account: getAddress(account),
        ...rest,
      };
    });
}

/**
 * Simulate API call using the token as a bearer token
 *
 * @param jwt
 * @returns {
 *   message?: SiweMessage // the decoded and reencoded siwe message
 *   valid: boolean // if it was valid
 *   account?: &nbsp;0x${string}&nbsp; // the account address
 *   error?: string // error message
 *   error_description?: string // error description
 * }
 */
export async function verifyTokenOnServer(jwt: string): Promise<{
  message: SiweMessage;
  valid: boolean;
  account: &nbsp;0x${string}&nbsp;;
  error?: string;
  error_description?: string;
  signature?: &nbsp;0x${string}&nbsp;;
}> {
  return await fetch('/verify', {
    headers: {
      Authorization: &nbsp;Bearer ${jwt}&nbsp;,
    },
  })
    .then((response) => response.json())
    .then(({ message, account, ...rest }) => ({
      message: new SiweMessage(message),
      account: getAddress(account),
      ...rest,
    }));
}
```

```
/**
 * Verify JWT on client side
 *
 * @param account
 * @param jwt
 * @returns
 */
export async function verifyTokenOnClient(
  account: Address,
  jwt: string,
): Promise<{
  valid: boolean;
  account?: Address;
  message?: SiweMessage;
  signature?: &nbsp;0x${string}&nbsp;;
}> {
  const [header, message, signature] = (jwt || '')
    .split('.')
    .map((d: string) => base64url.toBuffer(d));
  const message_ = new SiweMessage(message.toString());
  const { typ, alg } = JSON.parse(header.toString());
  if (typ !== 'JWT' || alg !== 'ES256K') {
    return { valid: false };
  }
  if (message_.address === account) {
    if (!config.chains || !config.connector) {
      throw new Error('No chains or connector configured');
    }
    const contract = getContract({
      abi: [
        {
          inputs: [
            {
              internalType: 'bytes32',
              name: 'dataHash',
              type: 'bytes32',
            },
            {
              internalType: 'bytes',
              name: 'signature',
              type: 'bytes',
            },
          ],
          name: 'isValidSignature',
          outputs: [
            {
              internalType: 'bytes4',
              name: 'returnedStatus',
              type: 'bytes4',
            },
          ],
          stateMutability: 'view',
          type: 'function',
        },
      ],
      address: account,
      publicClient: config.getPublicClient(),
      walletClient: await config.connector.getWalletClient(),
    });
    return await contract.read
      .isValidSignature([
        hashMessage(message.toString()) as &nbsp;0x${string}&nbsp;,
        toHex(signature) as &nbsp;0x${string}&nbsp;,
      ])
      .then((output) => {
        if (
          (output as unknown as &nbsp;0x${string}&nbsp;) ===
          ('0x1626ba7e' as &nbsp;0x${string}&nbsp;)
        ) {
          return {
            valid: true,
            message: message_,
            account,
            signature: toHex(signature),
          };
        }
        return { valid: false };
      })
      .catch((error) => {
        console.error(error);
        return { valid: false, message: message_, error: error.message };
      });
  } else if (
    message_.resources?.find(
      (d) =>
        d.startsWith('did:account:') &&
        getAddress(d.replace(/^did:account:/, '')) === account,
    )
  ) {
    const serverAddress = await fetch('/.well-known/public-key')
      .then((response) => response.json())
      .then(({ address }) => address)
      .catch((error) => {
        console.error(error);
        return '0x';
      });

    return await recoverMessageAddress({
      message: message.toString(),
      signature: toHex(signature),
    })
      .then((address) => {
        const valid = address === serverAddress;
```

```
        return { valid, message: message_, account };
      })
      .catch((error) => {
        console.error(error);
        return {
          valid: false,
          message: message_,
          error: error.message,
          signature: toHex(signature),
        };
      });
  }
  return { valid: false };
}
```

</file>

<file>

path: /src/utils/stringify.ts

url: https://github.com/richtera/lukso-siwe-demo/blob/main/src/utils/stringify.ts

```
export const stringify: typeof JSON.stringify = (value, replacer, space) =>
  JSON.stringify(
    value,
    (key, value_) => {
      const value = typeof value_ === 'bigint' ? value_.toString() : value_
      return typeof replacer === 'function' ? replacer(key, value) : value
    },
    space,
  )
```

</file>

<file>

path: /src/vite-env.d.ts

url: https://github.com/richtera/lukso-siwe-demo/blob/main/src/vite-env.d.ts

```
/// <reference types="vite/client" />
```

</file>

<file>

path: /src/wagmi.ts

url: https://github.com/richtera/lukso-siwe-demo/blob/main/src/wagmi.ts

```
import { w3mConnectors, w3mProvider } from '@web3modal/ethereum';
import { configureChains, createConfig } from 'wagmi';

export const walletConnectProjectId =
  import.meta.env.VITE_WALLET_CONNECT_PROJECT_ID || '';

const { chains, publicClient, webSocketPublicClient } = configureChains(
  [
    {
      id: 42,
      name: 'LUKSO Mainnet',
      network: '42',
      blockExplorers: {
        default: {
          name: 'Execution Exporer',
          url: 'https://explorer.execution.mainnet.lukso.network',
        },
      },
      nativeCurrency: { name: 'LYX', symbol: 'LYX', decimals: 18 },
      rpcUrls: {
        '42': {
          http: ['https://rpc.mainnet.lukso.network'],
          webSocket: ['wss://ws-rpc.mainnet.lukso.network'],
        },
        default: {
          http: ['https://rpc.mainnet.lukso.network'],
          webSocket: ['wss://ws-rpc.mainnet.lukso.network'],
        },
        public: {
          http: ['https://rpc.mainnet.lukso.network'],
          webSocket: ['wss://ws-rpc.mainnet.lukso.network'],
        },
      },
    },
    {
      id: 4201,
      name: 'LUKSO Testnet',
      network: '4201',
      blockExplorers: {
        default: {
          name: 'Execution Exporer',
          url: 'https://explorer.execution.testnet.lukso.network',
        },
      },
      nativeCurrency: { name: 'TEST LYX', symbol: 'LYXt', decimals: 18 },
      rpcUrls: {
        '4201': {
          http: ['https://rpc.testnet.lukso.network'],
          webSocket: ['wss://ws-rpc.testnet.lukso.network'],
        },
        default: {
          http: ['https://rpc.testnet.lukso.network'],
          webSocket: ['wss://ws-rpc.testnet.lukso.network'],
        },
        public: {
          http: ['https://rpc.testnet.lukso.network'],
          webSocket: ['wss://ws-rpc.testnet.lukso.network'],
        },
      },
      testnet: true,
    },
  ],
  [w3mProvider({ projectId: walletConnectProjectId })],
);

export const config = createConfig({
  autoConnect: true,
  connectors: w3mConnectors({
    chains,
    projectId: walletConnectProjectId,
    version: 2,
  }),
  publicClient,
  webSocketPublicClient,
});

export { chains, publicClient };
```

</file>

<file>

path: /tailwind.config.js

url: https://github.com/richtera/lukso-siwe-demo/blob/main/tailwind.config.js

```
/** @type {import('tailwindcss').Config} */

const data = {
  content: ['./index.html', './src/**/*.tsx'],
  theme: {
    extend: {},
  },
  // eslint-disable-next-line unicorn/prefer-module
  plugins: [require('@tailwindcss/forms')],
};
export default data;
```

</file>

<file>

path: /tsconfig.json

url: https://github.com/richtera/lukso-siwe-demo/blob/main/tsconfig.json

```json
{
  "compilerOptions": {
    "allowJs": false,
    "allowSyntheticDefaultImports": true,
    "esModuleInterop": false,
    "forceConsistentCasingInFileNames": true,
    "isolatedModules": true,
    "jsx": "react-jsx",
    "lib": ["DOM", "DOM.Iterable", "ESNext"],
    "module": "ESNext",
    "moduleResolution": "Node",
    "noEmit": true,
    "resolveJsonModule": true,
    "skipLibCheck": true,
    "strict": true,
    "target": "ESNext",
    "useDefineForClassFields": true
  },
  "include": ["./src"],
  "exclude": ["./node_modules", "./dist", "./dist.server"]
}
```

</file>

<file>

path: /tsconfig.server.json

url: https://github.com/richtera/lukso-siwe-demo/blob/main/tsconfig.server.json

```
{
  "compilerOptions": {
    "target": "ES2018",
    "module": "ES2022",
    "strict": true,
    "outDir": "dist.server",
    "jsx": "preserve",
    "importHelpers": true,
    "moduleResolution": "node",
    "noImplicitAny": true,
    "experimentalDecorators": true,
    "esModuleInterop": true,
    "allowSyntheticDefaultImports": true,
    "sourceMap": true,
    "skipLibCheck": true,
    "allowJs": true,
    "baseUrl": ".",
    "types": ["node"],
    "paths": {
      "@/*": ["src/*"]
    },
    "lib": [
      "es2018",
      "es2021",
      "dom",
      "dom.iterable",
      "scripthost",
      "ES2020.BigInt"
    ],
    "resolveJsonModule": true
  },
  "include": ["./server/index.ts"],
  "exclude": ["./node_modules", "./dist.server", "./dist"]
}
```

</file>

<file>

path: /vite.config.ts

url: https://github.com/richtera/lukso-siwe-demo/blob/main/vite.config.ts

```
import react from '@vitejs/plugin-react';
import autoprefixer from 'autoprefixer';
import postcssImport from 'postcss-import';
import postcssNesting from 'postcss-nesting';
import tailwindcss from 'tailwindcss';
import tailwindNesting from 'tailwindcss/nesting';
import { defineConfig } from 'vite';
import mkcert from 'vite-plugin-mkcert';

// https://vitejs.dev/config/
export default defineConfig({
  server: {
    host: '::',
    proxy: {
      '/.well-known': 'http://localhost:3000',
      '/verify': 'http://localhost:3000',
      '/exchange': 'http://localhost:3000',
    },
    https: true,
  },
  define: {
    global: 'globalThis',
  },
  resolve: {
    alias: {
      process: 'process/browser',
      util: 'util',
    },
  },
  plugins: [mkcert(), react()],
  css: {
    postcss: {
      plugins: [
        postcssImport(),
        tailwindNesting(postcssNesting),
        autoprefixer,
        tailwindcss,
      ],
    },
  },
});
```

</file>

</repository>