

Create and submit Orders

Use this endpoint to submit orders to the Flood protocol. Flood requires you to provide the order object along with a valid signature of the permit hash of the order. The permit hash is the keccak256 hash of a [Permit2\(opens in a new tab\)](#) BatchWitnessTransferFrom struct using the Flood Order struct as witness.

Regular signatures, [EIP2098\(opens in a new tab\)](#) signatures, and [EIP1271\(opens in a new tab\)](#) smart contract signature are all supported.

A FloodPlain order has an arbitrary amount of ERC20 offer items and consideration items. Offer items are what the offerer (order maker) wants to give, and the consideration items are what the offerer wants to receive in return.

An order is a single [Permit2\(opens in a new tab\)](#) BatchWitnessTransferFrom struct using the Flood Order struct as witness which includes necessary token approvals and the order data. The signer verification and order cancellation features are both relegated to Permit2.

To create an order, offerer has to sign PermitBatchWitnessTransferFrom struct of Permit2 based on the [EIP712\(opens in a new tab\)](#) standard. The struct is below:

```
struct
PermitBatchWitnessTransferFrom { TokenPermissions[] permitted; address spender; uint256 nonce; uint256 deadline; Order
witness; }

struct
Hook { address target; bytes data; }

struct
Item { address token; uint256 amount; }

struct
Order { address offerer; address zone; Item[] offer; Item consideration; uint256 deadline; uint256 nonce; }

struct
TokenPermissions { address token; uint256 amount; } Note that in this structure, witness.nonce == nonce , witness.deadline
== deadline , and witness.offer == permitted . This redundancy is for simplicity, because it allows the Order struct to contain
all the information related to an order.
```

For convenience, the [FloodPlain\(opens in a new tab\)](#) contract exposes a getPermitHash method that takes an Order struct as parameter and returns the permit hash.

After the order is signed, the order and its signature can be submitted to the Flood API.

API Typescript Solidity curl

-X

POST

```
"https://arbitrum.flood.bid/orders/new" \ -H "Content-Type: application/json" \ -d '{ "signature": "YOUR_SIGNATURE",
"order": { "offerer": "0x...", "zone": "0xf100d0988D6c9A02236A83d2E12e231226c814Ee", "offer": [ { "token":
"0xFd086bC7CD5C481DCC9C85ebE478A1C0b69FCbb9", "amount": "10000000" } ], "preHooks": [], "postHooks": [],
"consideration": [ { "token": "0xFF970A61A04b1cA14834A43f5dE4533eBDDb5CC8", "amount": "0" }, "deadline":
"340282366920938463463374607431768211455", "nonce": "0", "recipient": "0x..." }, } }' Last updated on February 3, 2024
Get Supported tokens Query orders
```