

# Gensler: an LP analysis tool

C. Lewis (@ctjlewis), for the Levels team

October 31, 2021

## Abstract

Initial notes on Gensler, a tool for collecting balance, trade, and liquidity provision metadata from various liquidity pools.

## Contents

<b>Problem</b>	<b>1</b>
<b>Research</b>	<b>2</b>
Example data . . . . .	2
<b>Reading account actions from transactions</b>	<b>3</b>
Complications . . . . .	3
Recommendations . . . . .	4
Prototypical example . . . . .	4
Reading the user's transactions . . . . .	5
Looking for SushiSwap trades . . . . .	5
Iterating in the future . . . . .	5
<b>Exhibits</b>	<b>6</b>
Internal transactions . . . . .	6
State changes . . . . .	6
Identifying an ERC20 LP token . . . . .	6

## Problem

Given a transaction that interacts with a liquidity pool, how can we deduce the specific account action (swap vs. LP add/remove and token pair) that was taken using only the transaction data?

## Research

We are currently researching transactions that interact with the following protocols:

- Uniswap
- Compound
- SushiSwap
- Olympus

## Example data

A random account which has interacted with various services was picked randomly from a recent block, at address:

0x4929f4d38f2955649b5aa34343da04cf790b9d92

The following transactions it sent will be referenced:

Action	Tx Hash
<b>Uniswap V2:</b> Remove <b>0.41 MKR</b> and <b>0.81 ETH</b> Liquidity	0xa22a00...304775
<b>Uniswap V2:</b> Supply <b>2474.15 WILD</b> and <b>0.985 ETH</b> Liquidity	0x64f98a...e67548
<b>Uniswap V3:</b> Swap <b>0.75 ETH</b> for <b>2050.50 WILD</b> and <b>0.25 ETH</b> for <b>696.65 WILD</b>	0x817f4a...a51a89
<b>SushiSwap:</b> Swap <b>1.63 COMP</b> for <b>0.12 ETH</b>	0x41f977...0ac0b5

## Reading account actions from transactions

We know that account actions can be inferred from raw transactions not only on principle but also because Etherscan does it, as seen below under **Complications** regarding transaction **0xa22a00...304775**, correctly identifying the account action (removing liquidity) and the tokens in question (ETH and MKR). This transaction involves the user interacting with the **Uniswap v2 Router** contract at **0x7a250d...f2488d**.

### Complications

For our problem, we have only an address, and it is necessary to decode the transaction input data and/or logs in order to reason about the underlying trade represented by a given transaction (which coins were swapped, whether wETH was burnt, etc). It is not easy to do this, though Etherscan is able to do it as shown in the **Transaction Action** field.

*Below, in yellow: Etherscan's computed "Transaction Action" field parsing the account action from a transaction (not available via API); in red: ERC20 transfers involved in the transaction.*

Transaction Hash:	0xa22a005bad96842a239f36cc5e3b1fed883812f3d3e1628f175abc9f36304775
Status:	Success
Block:	10214004 3301604 Block Confirmations
Timestamp:	510 days 6 hrs ago (Jun-06-2020 07:18:02 PM +UTC)
From:	0x4929fd38f2955649b5aa34343da04cf790b9d92
Interacted With (To):	Contract 0x7a250d5630b4cf539739df2c5dadb4c659f2488d (Uniswap V2: Router 2) TRANSFER 0.810749426054538127 Ether From Wrapped Ether To → Uniswap V2: Rout... TRANSFER 0.810749426054538127 Ether From Uniswap V2: Rout... To → 0x4929fd38f2955649b5aa343...
Transaction Action:	Remove 0.409162310574503654 MKR And 0.810749426054538127 Ether Liquidity From Uniswap V2
Tokens Transferred:	<div>From 0x4929fd38f2955... To Uniswap V2: MKR 2 For 0.5725772398982556 (\$4,140.23) Uniswap V2 (UNI-V2)</div> <div>From Uniswap V2: MKR 2 To Black Hole: 0x000... For 0.5725772398982556 (\$4,140.23) Uniswap V2 (UNI-V2)</div> <div>From Uniswap V2: MKR 2 To Uniswap V2: Rout... For 0.409162310574503654 (\$1,002.10) 1x1 Maker (MKR)</div> <div>From Uniswap V2: MKR 2 To Uniswap V2: Rout... For 0.810749426054538127 (\$3,539.02) Wrapped Ether (WETH)</div> <div>From Uniswap V2: Rout... To 0x4929fd38f2955... For 0.409162310574503654 (\$1,002.10) 1x1 Maker (MKR)</div>

This is not exposed by the API, and has **been requested many times**, enough that Etherscan has commented on it:

We unfortunately do not have an endpoint that returns the “Transaction Action” information at this point of time. It is a popular one by request, and we will definitely consider to add it in a future update.

—Etherscan Support

Thus, it seems to be nontrivial to decode the account action from a raw transaction, and this will involve reproducing logic similar to that of Etherscan’s **Transaction Action** field.

In addition, because each liquidity pair (at least on Uniswap) has its own dedicated contract, it is also not feasible to simply look at an address's interactions with a single contract to gather liquidity provision trades. It would be necessary to load a list of *all* Uniswap LP contracts for *each* token pair and then cross-reference an account's ERC20 token transfers with each contract address using the following Etherscan API call:

```
`https://api.etherscan.io/api
  ?module=account
  &action=tokenTx
  &address=${address}
  &contractAddress=${pairLPContractAddress}`
```

And this does not appear to be a sustainable approach.

## Recommendations

Despite these complications, in general, it *is* possible to find *all* of a user's account actions on a protocol, without distinguishing the specific action or learning much about it, by loading all of an address's transactions (after a given block height, for a fixed lookback period) and searching the user's transactions for interactions with the protocol's routing contract.

Once logic to decode account actions from raw transactions is established, it will be possible to parse these transactions to understand whether they are swaps or adding/removing liquidity, and the underlying token pair, etc.

For now, it is recommended to simply **count the number of trades on a protocol** and use this as an initial stat until a way to decode transactions is known.

## Prototypical example

To count the number of SushiSwap transactions our test trader, `0x4929f4...0b9d92`, has made, we simply count their interactions with the SushiSwap routing contract, `0xd9e1cE...378B9F`.

## Reading the user's transactions

An address's transactions can be read for free, up to 5 requests per second, from the following endpoint:

```
`https://api.etherscan.io/api
  ?module=account
  &action=txlist
  &address=0x4929f4d38f2955649b5aa34343da04cf790b9d92`
```

## Looking for SushiSwap trades

It is trivial to count the number of trades (liquidity provision and swaps) the user has made like so:

```
const address = `0x4929f4d38f2955649b5aa34343da04cf790b9d92`;
const transactionsRequest = await fetch(
  `https://api.etherscan.io/api?module=account&action=txlist&address=${address}`
);

const transactions = await transactionsRequest.json();
const trades = transactions.result.filter(
  (tx) => tx.to == '0xd9e1ce17f2641f24ae83637ab66a2cca9c378b9f'
);

console.log(`User ${address} made ${trades.length} SushiSwap trades!`);
// User 0x4929f4d38f2955649b5aa34343da04cf790b9d92 made 64 SushiSwap trades!
```

## Iterating in the future

Ultimately it will not be ideal to depend only on lightweight free APIs, and the service will likely call for setting up a complete Ethereum node and querying it for data directly rather than relying on Etherscan and other services.

It will also be necessary to learn more about how to reliably decode raw transactions and read underlying trade data from them, and some writing of particular interest to this research is [this blog post](#) by Diran Li. It is certainly possible and is just a matter of time investment.

## Exhibits

Various block explorers are capable of simulating internal transactions and/or decoding input data.

### Internal transactions

Below: *Ethplorer displaying the internal transactions for tx `0xa22a00...304775`.*

6 internal operations found			
✓	APPROVE from 0x4929f4d38f2955649b5aa34343da04cf790b9d92 to 0x7a250d5630b4cf539739df2c5dacb4c659f2488d	0.5725772398982556	UNI-V2
✓	TRANSFER from 0x4929f4d38f2955649b5aa34343da04cf790b9d92 to 0xc2adda861f89bbb333c90c492cb837741916a225	0.5725772398982556	UNI-V2
✓	TRANSFER from 0xc2adda861f89bbb333c90c492cb837741916a225 to 0x00	0.5725772398982556	UNI-V2
✓	TRANSFER from 0xc2adda861f89bbb333c90c492cb837741916a225 to 0x7a250d5630b4cf539739df2c5dacb4c659f2488d	0.40916231057450364	MKR
✓	TRANSFER from 0xc2adda861f89bbb333c90c492cb837741916a225 to 0x7a250d5630b4cf539739df2c5dacb4c659f2488d	0.8107494260545381	WETH
✓	TRANSFER from 0x7a250d5630b4cf539739df2c5dacb4c659f2488d to 0x4929f4d38f2955649b5aa34343da04cf790b9d92	0.40916231057450364	MKR

### State changes

Below: *The state changes for the same transaction as above.*

Address		Before	After	State Difference
0x04608ec2f57cc15c381...	Mixer (zhizhu.top)	1,926,882,132,929,015,035,197 Eth	1,926,885,736,735,015,035,197 Eth	+ 0.003603806
0x4929f4d38f2955649b5...		5,226,442,485,144,923,518 Eth Nonce: 356	6,033,988,105,199,461,645 Eth Nonce: 357	+ 0.807145620054538127
0x9f8f72aa9304c8b593d...	MKR			
0xc02aaa39b223fe8d0a...	WETH	2,395,724,306,207,652,754,516,798 Eth	2,395,723,497,458,226,699,978,671 Eth	- 0.810749426054538127
0xc2adda861f89bbb333...	UNI-V2			

### Identifying an ERC20 LP token

Below (highlighted): *Etherscan identifying the pair a given SushiSwap LP token represents, here the MCAT/ETH SushiSwap LP token `0xfc9873...8d7c14`.*

Token SushiSwap LP Token

Buy Exchange Earn Gaming

Featured: Curious on Ethereum's hottest trading pairs? View top pairs and details with DEX Trading Pairs!

Overview [ERC-20]

Max Total Supply

231,300,667,580,100,1660 ... SLP

Holders

3

Transfers

5

Profile Summary [Edit]

Contract

0xfc98737f842bfff2f4f8b48589943b52728d7c14

Decimals

18

Social Profiles

Not Available, Update ?

Ad

AA

Crypto savings with 60% APY Earn while you sleep

Earn now

This SushiSwap Liquidity Provider (LP) token represents the MetaCat and ETH pairing.