5CFE9D;}

.css-kun0x7{fill:transparent;opacity:0.5;margin:0 0.2rem;}.css-kun0x7:hover{fill:#FAF40A;}

.css-1ix0nx7{fill:transparent;opacity:0.5;}.css-1ix0nx7:hover{fill:#F14544;} On this page

Testing Hooks

Testing hooks

Testing hooks is same as testing contracts. The template includes a test for the Counter hook, which you can find intest/Counter.t.sol .

Here are some key points about the Counter hook test:

1. The hook extends from a couple of utilities that facilitate easier testing of hooks.
2. import
3. "forge-std/Test.sol"
4. ;
5. import
6. {
7. Deployers
8. }
9. from
10. "v4-core/test/utils/Deployers.sol"
11. ;
12. contract
13. CounterTest
14. is
15. Test
16. ,
17. Deployers
18. {
19. }
20. Copy
21. Thesetup
22. function, called before every test, creates a few test tokens, retrieves the hook address, and then initializes the pool with this hook address.
23. function
24. setup
25. (
26.)
27. public
28. {
29. Deployers
30. .
31. deployFreshManagerAndRouters
32. (
33.)
34. ;
35. (
36. currency0
37. ,
38. currency1
39.)
40. =
41. Deployers
42. .
43. deployMintAndApprove2Currencies
44. (
45.)
46. ;
47. // Deploy the hook to an address with the correct flags
48. uint160
49. flags
50. =

```
uint160
(
Hooks
.
BEFORE_SWAP_FLAG
|
Hooks
.
AFTER_SWAP_FLAG
|
Hooks
.
BEFORE_ADD_LIQUIDITY_FLAG
|
Hooks
.
BEFORE_REMOVE_LIQUIDITY_FLAG
)
;
(
address
hookAddress
,
bytes32
salt
)
=
HookMiner
.
find
(
address
(
this
)
,
flags
,
type
(
Counter
)
.
creationCode
,
abi
.
encode
(
address
(
manager
)
)
)
;
counter
=
new
Counter
{
salt
:
salt
}
(
IPoolManager
(
```

address
(
manager
)
)
)
;
}
Copy
Pool is then initialized containing this hook
// Create the pool
key
=
PoolKey
(
currency0
,
currency1
,
3000
,
60
,
IHooks
(
counter
)
)
;
poolId
=
key
.
toId
(
)
;
initializeRouter
.
initialize
(
key
,
SQRT_RATIO_1_1
,
ZERO_BYTES
)
;
Copy
Hook tests utilize a router, namelyPoolModifyPositionTest
, to modify positions. PoolModifyPositionTest implements theILockCallback
interface and adds thelockAcquired
function, which in turn calls themanager.modifyPosition
function.
PoolManager manager
;
PoolModifyPositionTest modifyPositionRouter
;
manager
=
new
PoolManager
(
500000
)
;
// Helpers for interacting with the pool
modifyPositionRouter

=
new
PoolModifyPositionTest
(
IPoolManager
(
address
(
manager
)
)
)
;
modifyPositionRouter
.
modifyPosition
(
poolKey
,
IPoolManager
.
ModifyPositionParams
(
-
120
,
120
,
10
ether
)
,
ZERO_BYTES
)
;
Copy
Similarly, for token swaps, the test usesPoolSwapTest
, which also implements theILockCallback
interface.
Testing the hook closely resembles testing any other smart contract. The functiontestCounterHooks
executes swaps and verifies if the counters are updated correctly.
function
testCounterHooks
(
)
public
{
// positions were created in setup()
assertEq
(
counter
.
beforeAddLiquidityCount
(
poolId
)
,
3
)
;
assertEq
(
counter
.
beforeRemoveLiquidityCount
(
poolId
)

```
,
0
)
;
assertEq
(
counter
.
beforeSwapCount
(
poolId
)
,
0
)
;
assertEq
(
counter
.
afterSwapCount
(
poolId
)
,
0
)
;
// Perform a test swap //
bool
zeroForOne
=
true
;
int256
amountSpecified
=
1e18
;
BalanceDelta swapDelta
=
swap
(
key
,
zeroForOne
,
amountSpecified
,
ZERO_BYTES
)
;
// ---------- //
assertEq
(
int256
(
swapDelta
.
amount0
(
)
)
,
amountSpecified
)
;
assertEq
```

```
(
counter
.
beforeSwapCount
(
poolId
)
,
1
)
;
assertEq
(
counter
.
afterSwapCount
(
poolId
)
,
1
)
;
}
/// Test Helper
function
swap
(
PoolKey
memory
key
,
bool
zeroForOne
,
int256
amountSpecified
,
bytes
memory
hookData
)
internal
returns
(
BalanceDelta delta
)
{
IPoolManager
.
SwapParams
memory
params
=
IPoolManager
.
SwapParams
(
{
zeroForOne
:
zeroForOne
,
amountSpecified
:
amountSpecified
,
sqrtPriceLimitX96
```

391. :
392. zeroForOne
393. ?
394. TickMath
395. .
396. MIN_SQRT_RATIO
397. +
398. 1
399. :
400. TickMath
401. .
402. MAX_SQRT_RATIO
403. -
404. 1
405. // unlimited impact
406. }
407.)
408. ;
409. PoolSwapTest
410. .
411. TestSettings
412. memory
413. testSettings
414. =
415. PoolSwapTest
416. .
417. TestSettings
418. (
419. {
420. withdrawTokens
421. :
422. true
423. ,
424. settleUsingTransfer
425. :
426. true
427. ,
428. currencyAlreadySent
429. :
430. false
431. }
432.)
433. ;
434. delta
435. =
436. swapRouter
437. .
438. swap
439. (
440. key
441. ,
442. params
443. ,
444. testSettings
445. ,
446. hookData
447.)
448. ;
449. }
450. Copy Edit this page .css-1tclyyl{margin-top:1.5rem;} .css-1c3fvx8{display:-webkit-box;display:-webkit-flex;display:-ms-flexbox;display:flex;-webkit-flex-direction:row;-ms-flex-direction:row;flex-direction:row;-webkit-align-items:center;-webkit-box-align:center;-ms-flex-align:center;align-items:center;-webkit-box-pack:center;-ms-flex-pack:center;-webkit-justify-content:center;justify-content:center;} .css-1wsnqg4{font-size:1rem;padding-right:0.5rem;} Helpful? .css-y2jwfw{fill:transparent;opacity:0.5;}.css-y2jwfw:hover{fill:#5CFE9D;}

.css-kun0x7{fill:transparent;opacity:0.5;margin:0 0.2rem;}.css-kun0x7:hover{fill:#FAF40A;}

.css-1ix0nx7{fill:transparent;opacity:0.5;}.css-1ix0nx7:hover{fill:#F14544;} Previous Building your own hook Next Hook Deployment * Testing hooks