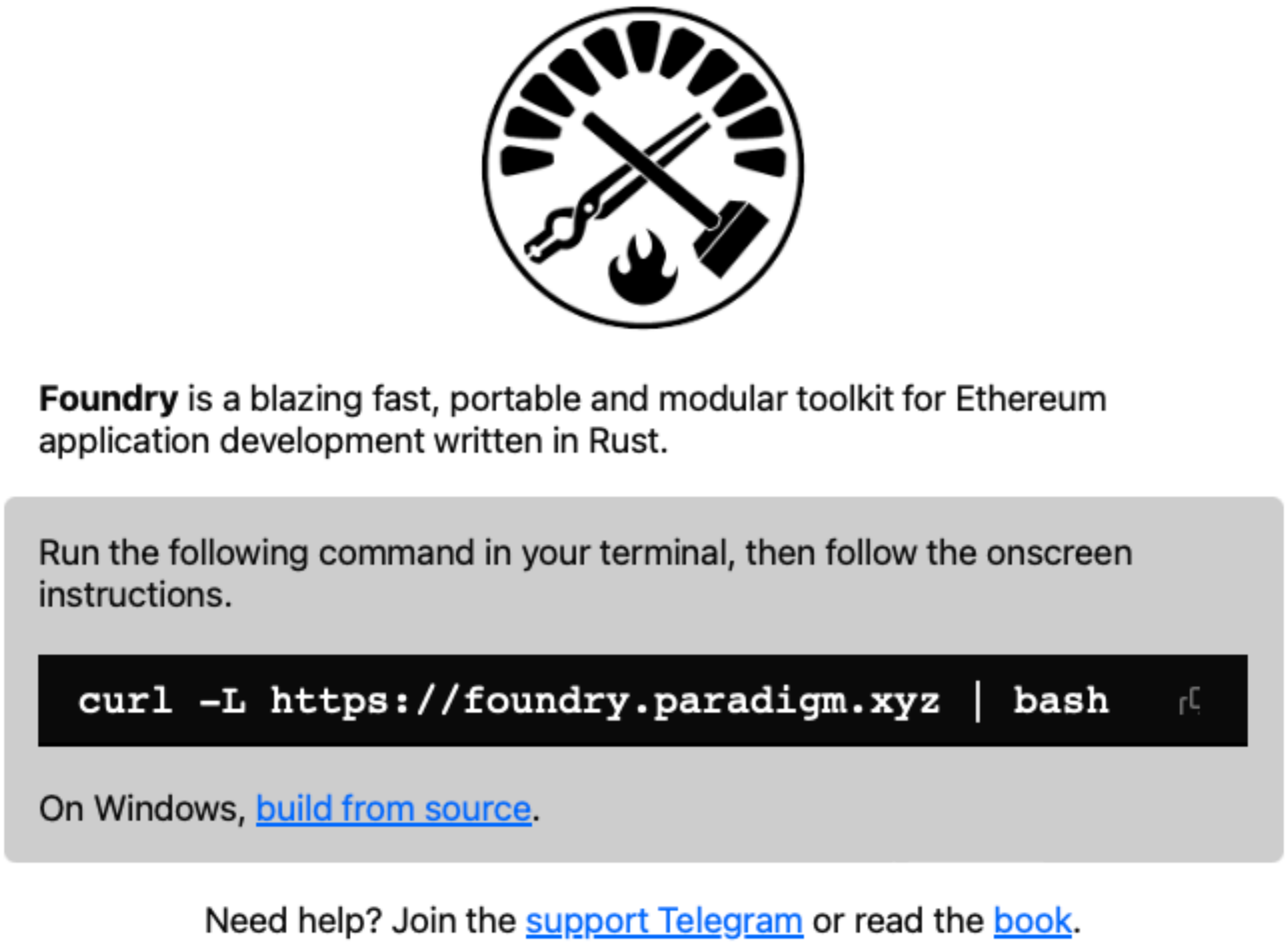
Foundry 101

31.01.2023 Sebastian Ławniczak

About me

- Software Developer @ Monerium
- mostly Go nowadays
- web3 enthusiast



Foundry is a blazing fast, portable and modular toolkit for Ethereum application development written in Rust.

Run the following command in your terminal, then follow the onscreen instructions.

```
curl -L https://foundry.paradigm.xyz | bash
```

On Windows, build from source.

Need help? Join the support Telegram or read the book.

Tools

- Forge – build -> test -> deploy -> repeat
- Anvil – local testnode
- Cast – RPC client
- Chisel – REPL (read–eval–print loop)

Forge

- mkdir counter; cd counter; forge init
- forge init counter

```
❯ tree -L 2
├── foundry.toml
├── lib
│   └── forge-std
├── script
│   └── Counter.s.sol
├── src
│   └── Counter.sol
└── test
    └── Counter.t.sol
```

Forge

lib/

- dependencies (git submodules)
- example:

```
❯ forge install transmissions11/solmate
[…]

❯ forge remappings
ds-test/=lib/forge-std/lib/ds-test/src/
forge-std/=lib/forge-std/src/
solmate/=lib/solmate/src/
```

Forge

src/Counter.sol

```solidity
// SPDX-License-Identifier: UNLICENSED
pragma solidity ^0.8.13;

contract Counter {
    uint256 public number;

    function setNumber(uint256 newNumber) public {
        number = newNumber;
    }

    function increment() public {
        number++;
    }
}
```

Forge

test/Counter.t.sol

```solidity
// SPDX-License-Identifier: UNLICENSED
pragma solidity ^0.8.13;

import "forge-std/Test.sol";
import "../src/Counter.sol";

contract CounterTest is Test {
    Counter public counter;

    function setUp() public {
        counter = new Counter();
        counter.setNumber(0);
    }

    function testIncrement() public {
        counter.increment();
        assertEq(counter.number(), 1);
    }
    […]
}
```

Forge

test/Counter.t.sol

```
// SPDX-License-Identifier: UNLICENSED
pragma solidity ^0.8.13;

import "forge-std/Test.sol";
import "../src/Counter.sol";

contract CounterTest is Test {
    Counter public counter;

    function setUp() public {
        counter = new Counter();
        counter.setNumber(0);
    }
    […]

    function testSetNumber(uint256 x) public {
        counter.setNumber(x);
        assertEq(counter.number(), x);
    }
}
```

Forge

script/Counter.s.sol

```
// SPDX-License-Identifier: UNLICENSED
pragma solidity ^0.8.13;

import "forge-std/Script.sol";

contract CounterScript is Script {
    function setUp() public {}

    function run() public {
        vm.broadcast();
    }
}
```

Running tests

```
❯ forge test -vvvv --match-test testIncrement

Running 1 test for test/
Counter.t.sol:CounterTest
[PASS] testIncrement() (gas: 28356)
Traces:
  [28356] CounterTest::testIncrement()
    ├─ [22340] Counter::increment()
    │   └─ ← ()
    ├─ [283] Counter::number() [staticcall]
    │   └─ ← 1
    └─ ← ()

Test result: ok. 1 passed; 0 failed; finished in
418.29µs
```

Running tests

```
❯ forge test -vvvv --match-test testSetNumber

Running 1 test for test/Counter.t.sol:CounterTest
[PASS] testSetNumber(uint256) (runs: 256, μ: 27564, ~: 28342)
Traces:
  [28342] CounterTest::testSetNumber(115792089……)
    ├─ [22290] Counter::setNumber(115792089……)
    │   └─ ← ()
    ├─ [283] Counter::number() [staticcall]
    │   └─ ← 115792089……
    └─ ← ()

Test result: ok. 1 passed; 0 failed; finished in 9.61ms
```

Anvil

```
❯ anvil -a 2

[…]

Available Accounts
==================

(0) "0xf39Fd6e51aad88F6F4ce6aB8827279cffFb92266" (10000 ETH)
(1) "0x70997970C51812dc3A010C7d01b50e0d17dc79C8" (10000 ETH)

Private Keys
==================

(0) 0xac0974bec39a17e36ba4a6b4d238ff944bacb478cbed5efcae784d7bf4f2ff80
(1) 0x59c6995e998f97a5a0044966f0945389dc9e86dae88c7a8412f4603b6b78690d

[...]

Listening on 127.0.0.1:8545
```

Deploying

forge create

```
❯ forge create \
    --rpc-url http://127.0.0.1:8545 \
    --private-key 0xac0974... \
    src/Counter.sol:Counter

Deployer: 0xf39Fd6e51aad88F6F4ce6aB8827279cffFb92266
Deployed to: 0xa513E6E4b8f2a923D98304ec87F64353C4D5C853
Transaction hash: 0xdced6e543d9a403ee9d57aaa4b2575150258b9d19450fd809071ae4cee42099b
```

Cast

- send ether
  - cast send <to_addr> –from <from_addr> –value 1ether
  - cast balance <to_addr>
- send transaction
  - cast send <addr> "increment()" --private-key <pk>
- view data
  - cast call <addr> "number()"

Chisel

- REPL, think of it as typing “python” in your terminal

```
❯ chisel
Welcome to Chisel! Type `!help` to show available commands.
➜ string memory message;
➜ message = "hello!";
➜ message
Type: string
├ UTF-8: hello!
├ Hex (Memory):
├─ Length ([0x00:0x20]): 0x000000000...
├─ Contents ([0x20:..]): 0x68656c6c6...
├ Hex (Tuple Encoded):
├─ Pointer ([0x00:0x20]): 0x000000000...
├─ Length ([0x20:0x40]): 0x0000000000...
└─ Contents ([0x40:..]): 0x68656c6c6f...
```

References

- Foundry website: https://getfoundry.sh/
- The book: https://book.getfoundry.sh/
- Github: https://github.com/foundry-rs/foundry
- Georgios' twitter: https://twitter.com/gakonst

Thanks!

Questions?