PD-8.0 Oracles

https://en.wikipedia.org/wiki/Pythia

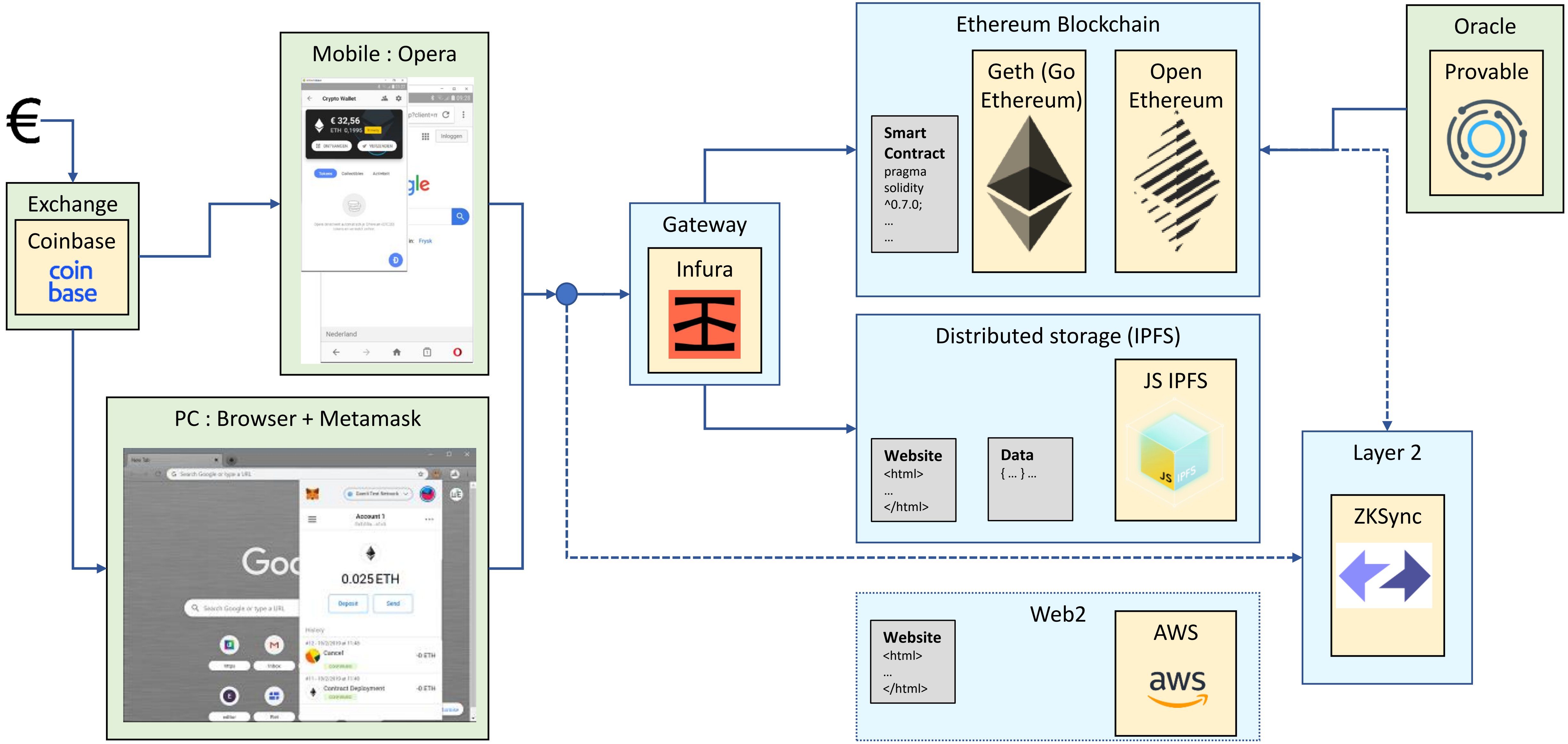
PD-8.1 Provable Oracle

https://docs.provable.xyz

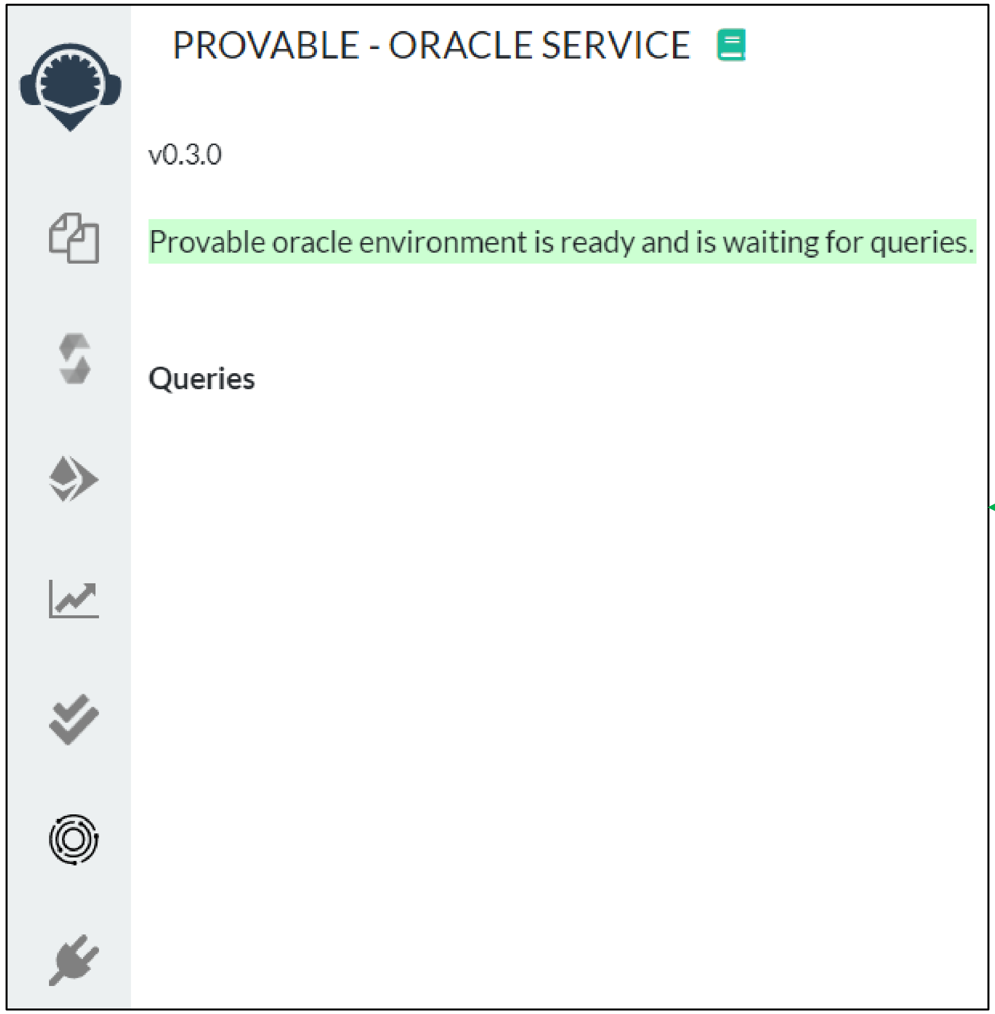
https://docs.provable.xyz/#ethereum

https://www.youtube.com/channel/UCjVjCheDbMel-x-JYeGazcQ/featured

PD-8.1 Provable oracle URL



PD-8.1 Provable oracle service in remix

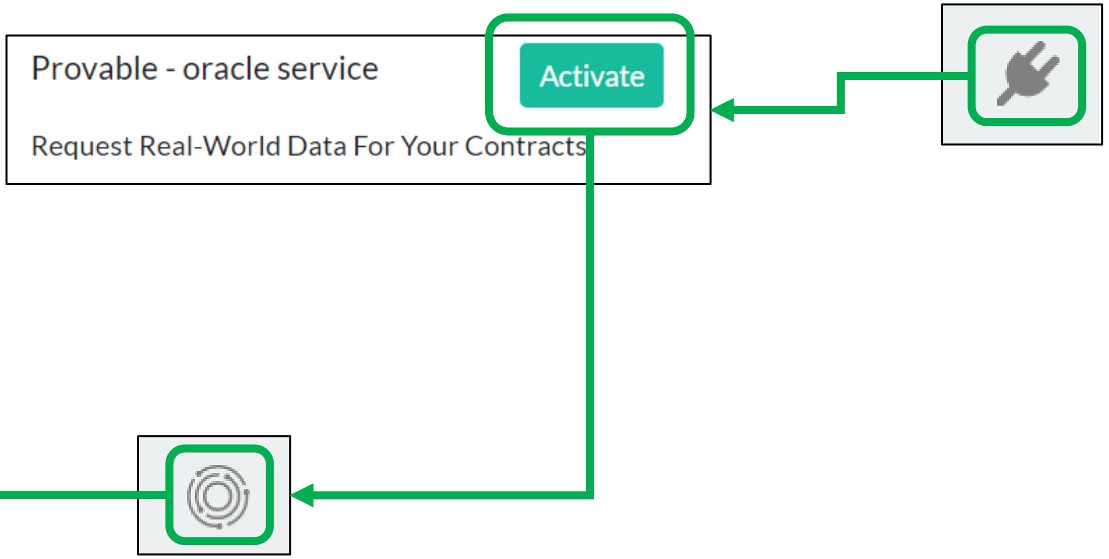


PROVABLE - ORACLE SERVICE

v0.3.0

Provable oracle environment is ready and is waiting for queries.

Queries



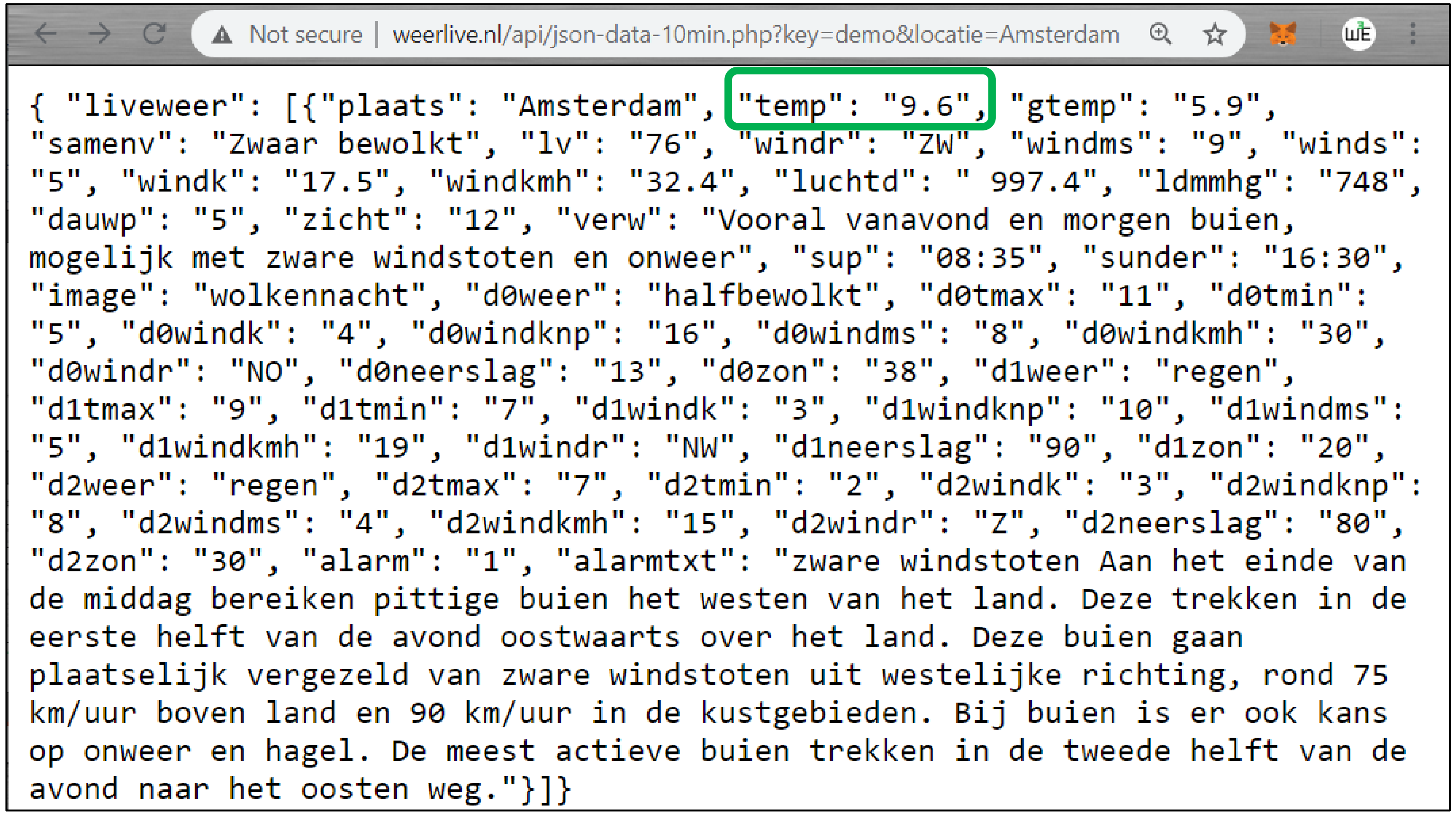
Provable - oracle service

Request Real-World Data For Your Contracts

http://remix.ethereum.org/

https://docs.provable.xyz/#development-tools-remix-ide-provable-plugin

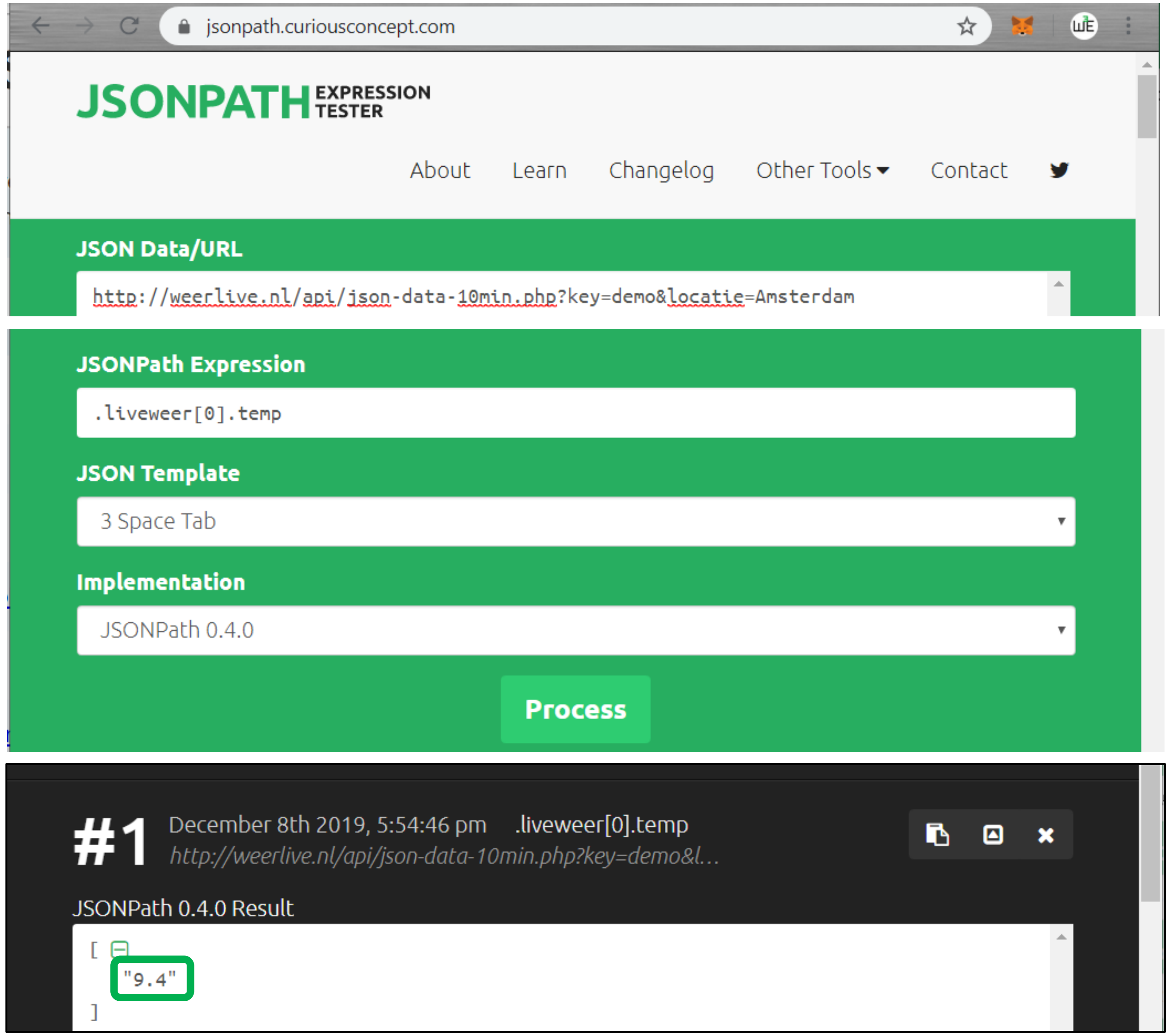
Weather information via API


Not secure | weerlive.nl/api/json-data-10min.php?key=demo&locatie=Amsterdam

```
{ "liveweer": [{"plaats": "Amsterdam", "temp": "9.6", "gtemp": "5.9",
"samenv": "Zwaar bewolkt", "lv": "76", "windr": "ZW", "windms": "9", "winds":
"5", "windk": "17.5", "windkmh": "32.4", "luchtd": " 997.4", "ldmmhg": "748",
"dauwp": "5", "zicht": "12", "verw": "Vooral vanavond en morgen buien,
mogelijk met zware windstoten en onweer", "sup": "08:35", "sunder": "16:30",
"image": "wolkennacht", "d0weer": "halfbewolkt", "d0tmax": "11", "d0tmin":
"5", "d0windk": "4", "d0windknp": "16", "d0windms": "8", "d0windkmh": "30",
"d0windr": "NO", "d0neerslag": "13", "d0zon": "38", "d1weer": "regen",
"d1tmax": "9", "d1tmin": "7", "d1windk": "3", "d1windknp": "10", "d1windms":
"5", "d1windkmh": "19", "d1windr": "NW", "d1neerslag": "90", "d1zon": "20",
"d2weer": "regen", "d2tmax": "7", "d2tmin": "2", "d2windk": "3", "d2windknp":
"8", "d2windms": "4", "d2windkmh": "15", "d2windr": "Z", "d2neerslag": "80",
"d2zon": "30", "alarm": "1", "alarmtxt": "zware windstoten Aan het einde van
de middag bereiken pittige buien het westen van het land. Deze trekken in de
eerste helft van de avond oostwaarts over het land. Deze buien gaan
plaatselijk vergezeld van zware windstoten uit westelijke richting, rond 75
km/uur boven land en 90 km/uur in de kustgebieden. Bij buien is er ook kans
op onweer en hagel. De meest actieve buien trekken in de tweede helft van de
avond naar het oosten weg."}]}
```

http://weerlive.nl/api/json-data-10min.php?key=demo&locatie=Amsterdam

PD-8.1 Jsonpath test



jsonpath.curiousconcept.com

JSONPATH EXPRESSION TESTER

About Learn Changelog Other Tools Contact

JSON Data/URL

```
http://weerlive.nl/api/json-data-10min.php?key=demo&locatie=Amsterdam
```

JSONPath Expression

```
.liveweer[0].temp
```

JSON Template

3 Space Tab

Implementation

JSONPath 0.4.0

Process

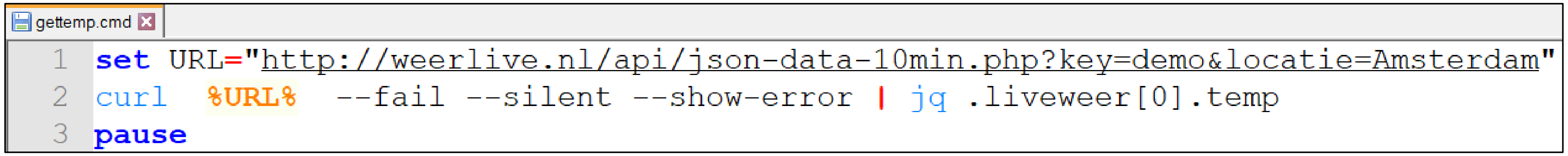
#1 December 8th 2019, 5:54:46 pm .liveweer[0].temp
http://weerlive.nl/api/json-data-10min.php?key=demo&l...

JSONPath 0.4.0 Result

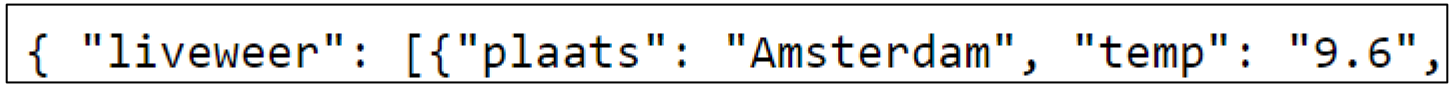
```
[
   "9.4"
]
```

https://jsonpath.curiousconcept.com/

PD-8.1 Parse & query JSON data



```
set URL="http://weerlive.nl/api/json-data-10min.php?key=demo&locatie=Amsterdam"
curl  %URL%  --fail --silent --show-error | jq .liveweer[0].temp
pause
```



```
{ "liveweer": [{"plaats": "Amsterdam", "temp": "9.6",
```

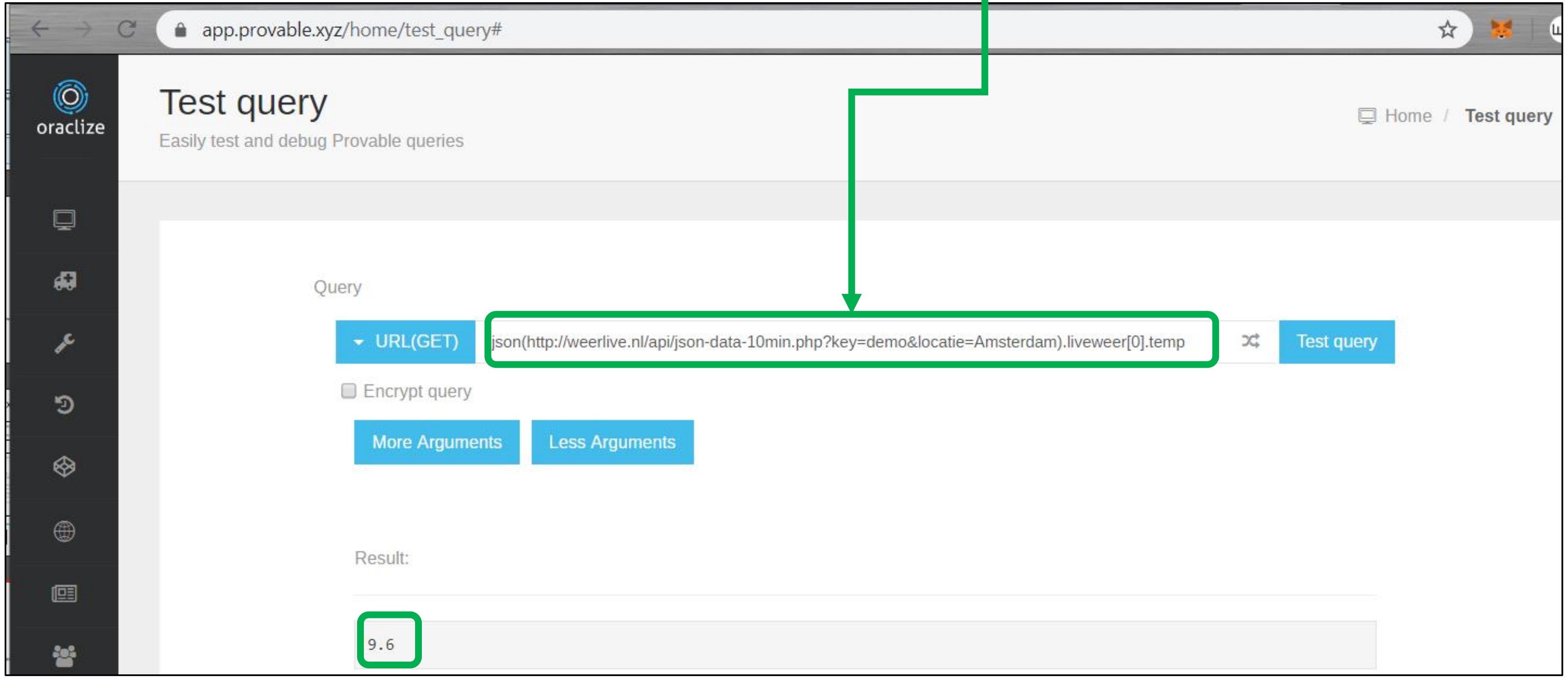
https://github.com/web3examples/ethereum/blob/master/oracle_examples/gettemp.cmd

https://stedolan.github.io/jq/download

http://weerlive.nl/api/json-data-10min.php?key=demo&locatie=Amsterdam

PD-8.1 Test provable query

json(http://weerlive.nl/api/json-data-10min.php?key=demo&locatie=Amsterdam).liveweer[0].temp



Test query

Easily test and debug Provable queries

Query

json(http://weerlive.nl/api/json-data-10min.php?key=demo&locatie=Amsterdam).liveweer[0].temp

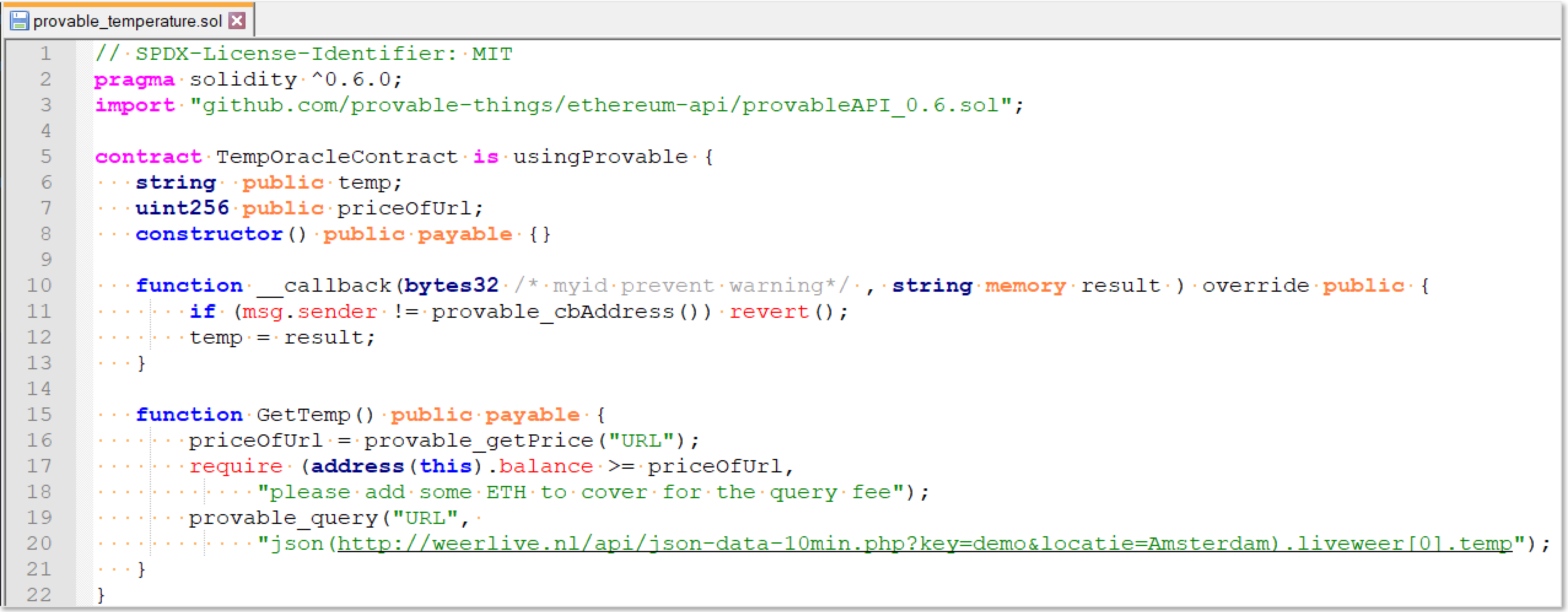
☐ Encrypt query

Result:

```
9.6
```

https://app.provable.xyz/home/test_query

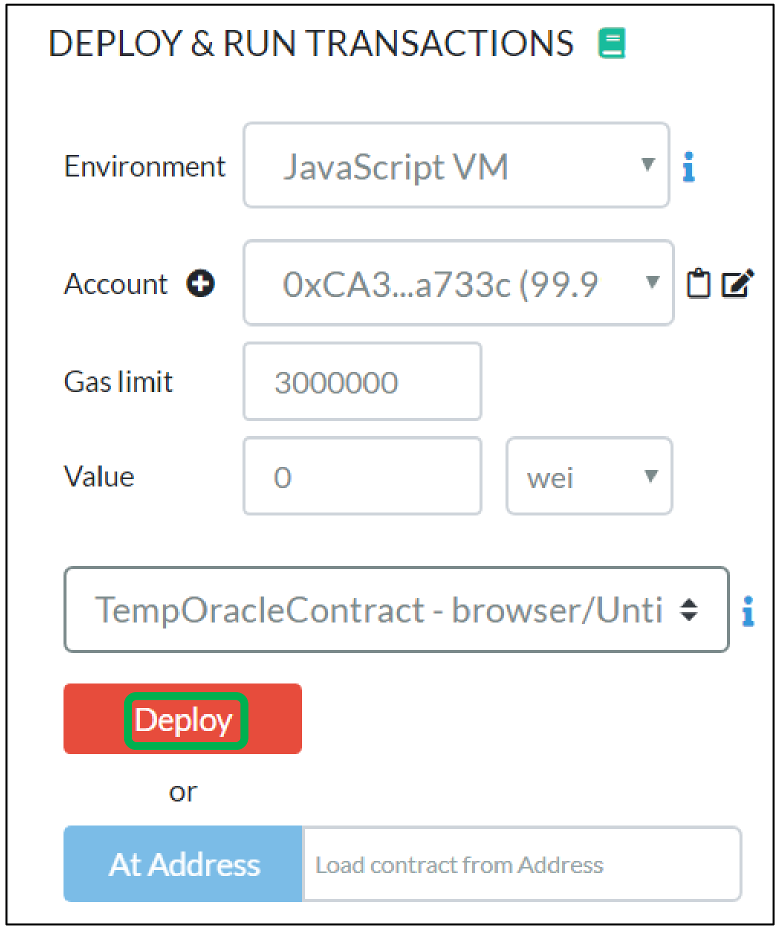
PD-8.1 Temperature (url) oracle with Provable


provable_temperature.sol

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.6.0;
import "github.com/provable-things/ethereum-api/provableAPI_0.6.sol";

contract TempOracleContract is usingProvable {
   string  public temp;
   uint256 public priceOfUrl;
   constructor() public payable {}

   function __callback(bytes32 /* myid prevent warning*/ , string memory result ) override public {
         if (msg.sender != provable_cbAddress()) revert();
         temp = result;
   }

   function GetTemp() public payable {
         priceOfUrl = provable_getPrice("URL");
         require (address(this).balance >= priceOfUrl,
                   "please add some ETH to cover for the query fee");
         provable_query("URL", 
                   "json(http://weerlive.nl/api/json-data-10min.php?key=demo&locatie=Amsterdam).liveweer[0].temp");
   }
}
```

https://github.com/web3examples/ethereum/blob/master/oracle_examples/provable_temperature.sol

PD-8.1 Temperature (url) oracle with Provable



DEPLOY & RUN TRANSACTIONS

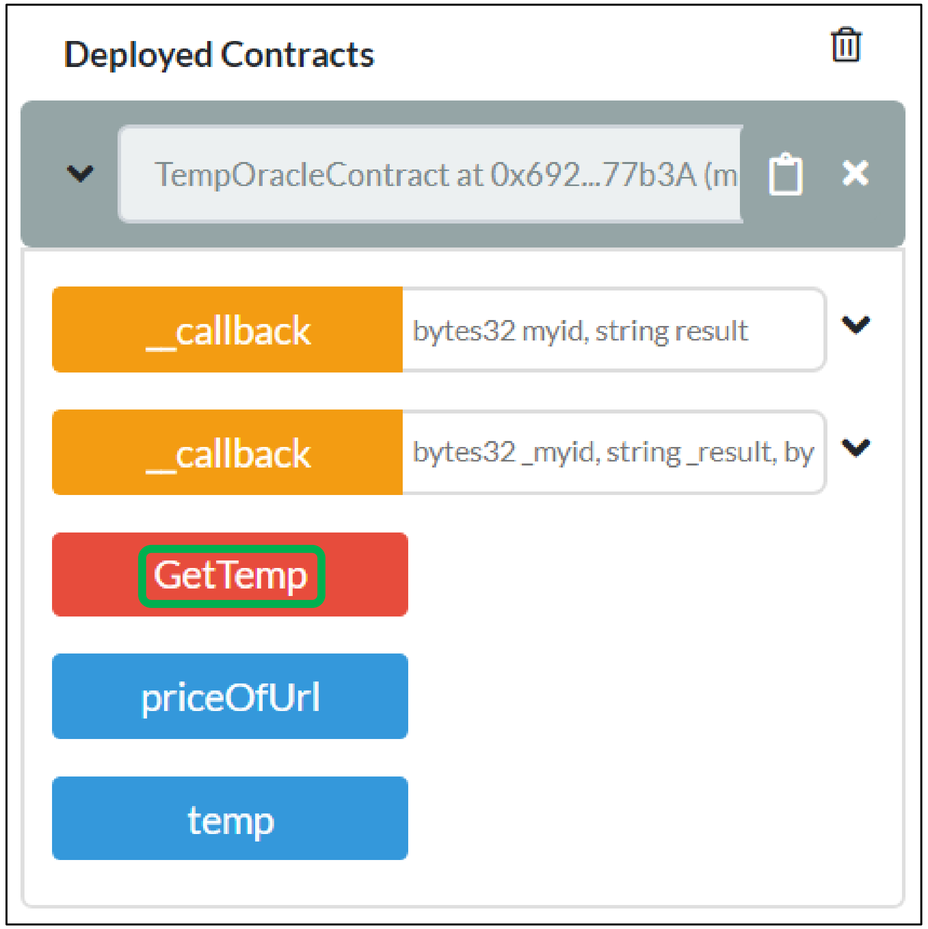
Environment	JavaScript VM
Account	0xCA3...a733c (99.9
Gas limit	3000000
Value	0 wei

TempOracleContract - browser/Unti

Deploy

or

At Address — Load contract from Address



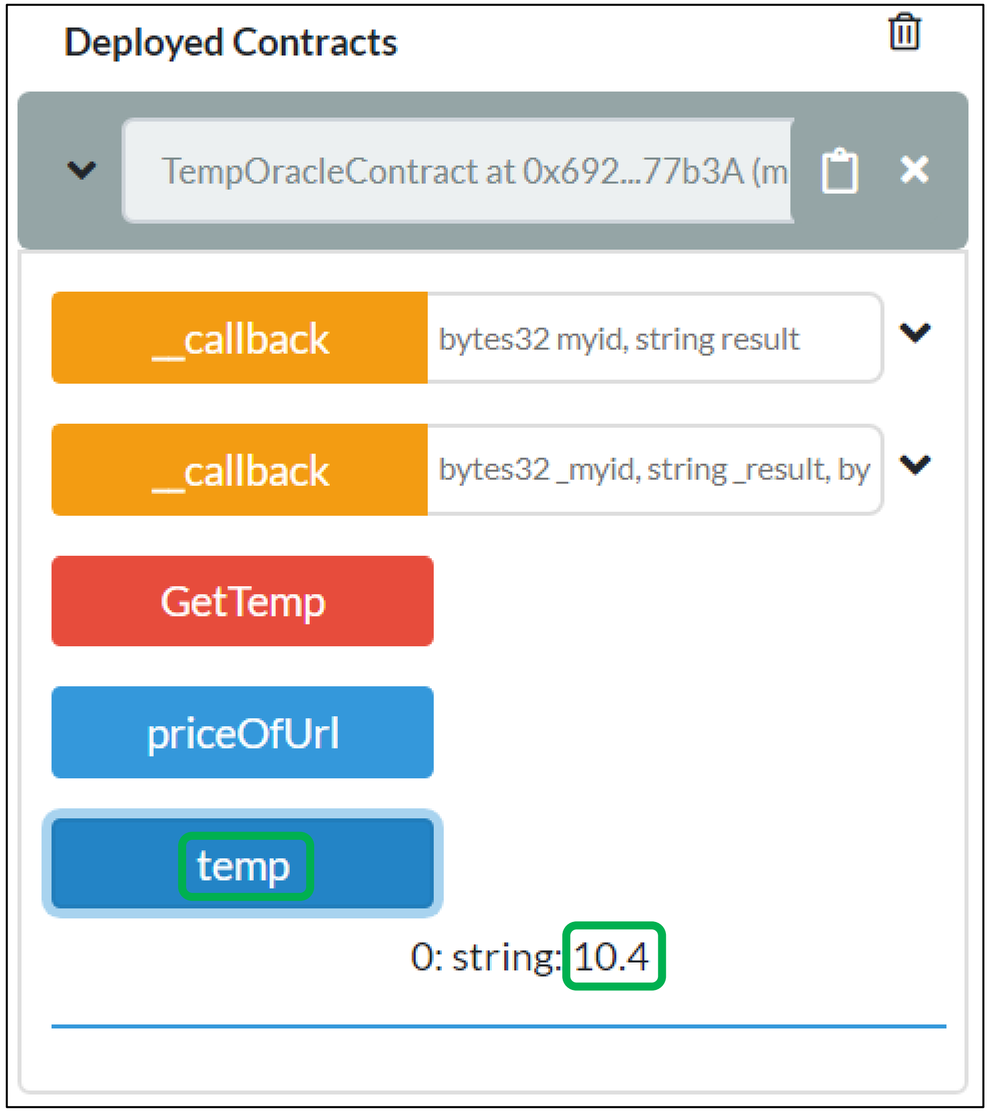
Deployed Contracts

TempOracleContract at 0x692...77b3A (m

_callback	bytes32 myid, string result
_callback	bytes32 _myid, string _result, by
GetTemp	
priceOfUrl	
temp	

https://github.com/web3examples/ethereum/blob/master/oracle_examples/provable_temperature.sol

PD-8.1 Result

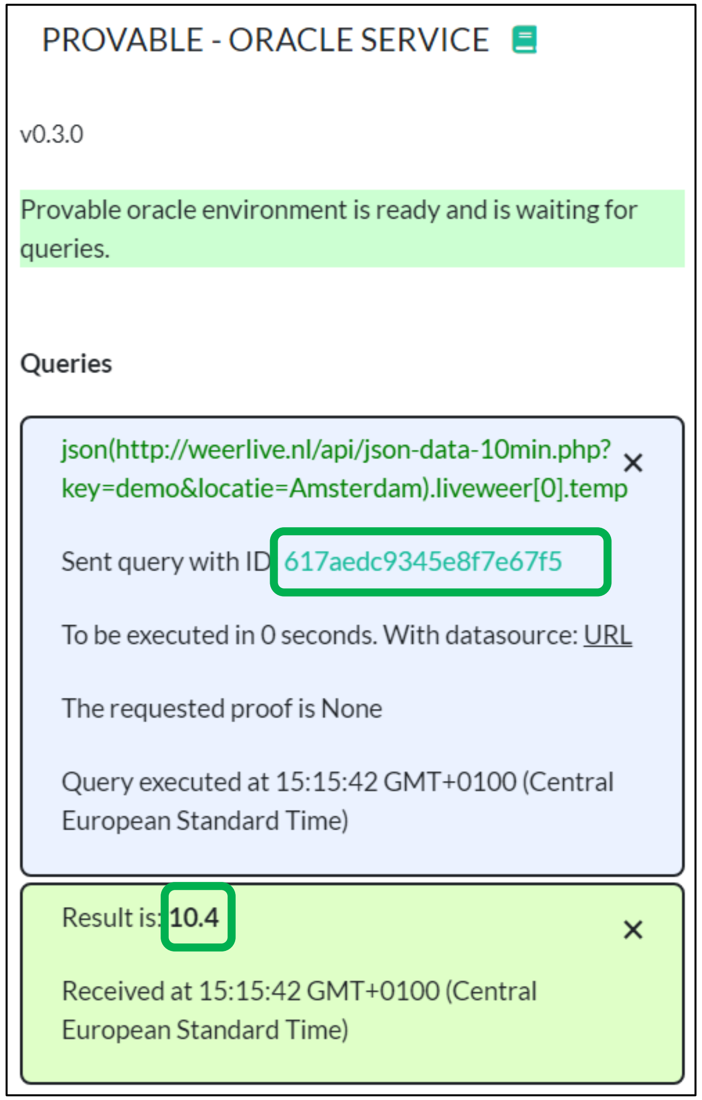

Deployed Contracts

TempOracleContract at 0x692...77b3A (m

_callback
bytes32 myid, string result
_callback
bytes32 _myid, string _result, by
GetTemp
priceOfUrl
temp

```
0: string: 10.4
```

PD-8.1 Provable status in remix



PROVABLE - ORACLE SERVICE

v0.3.0

Provable oracle environment is ready and is waiting for queries.

Queries

json(http://weerlive.nl/api/json-data-10min.php?key=demo&locatie=Amsterdam).liveweer[0].temp

Sent query with ID 617aedc9345e8f7e67f5

To be executed in 0 seconds. With datasource: URL

The requested proof is None

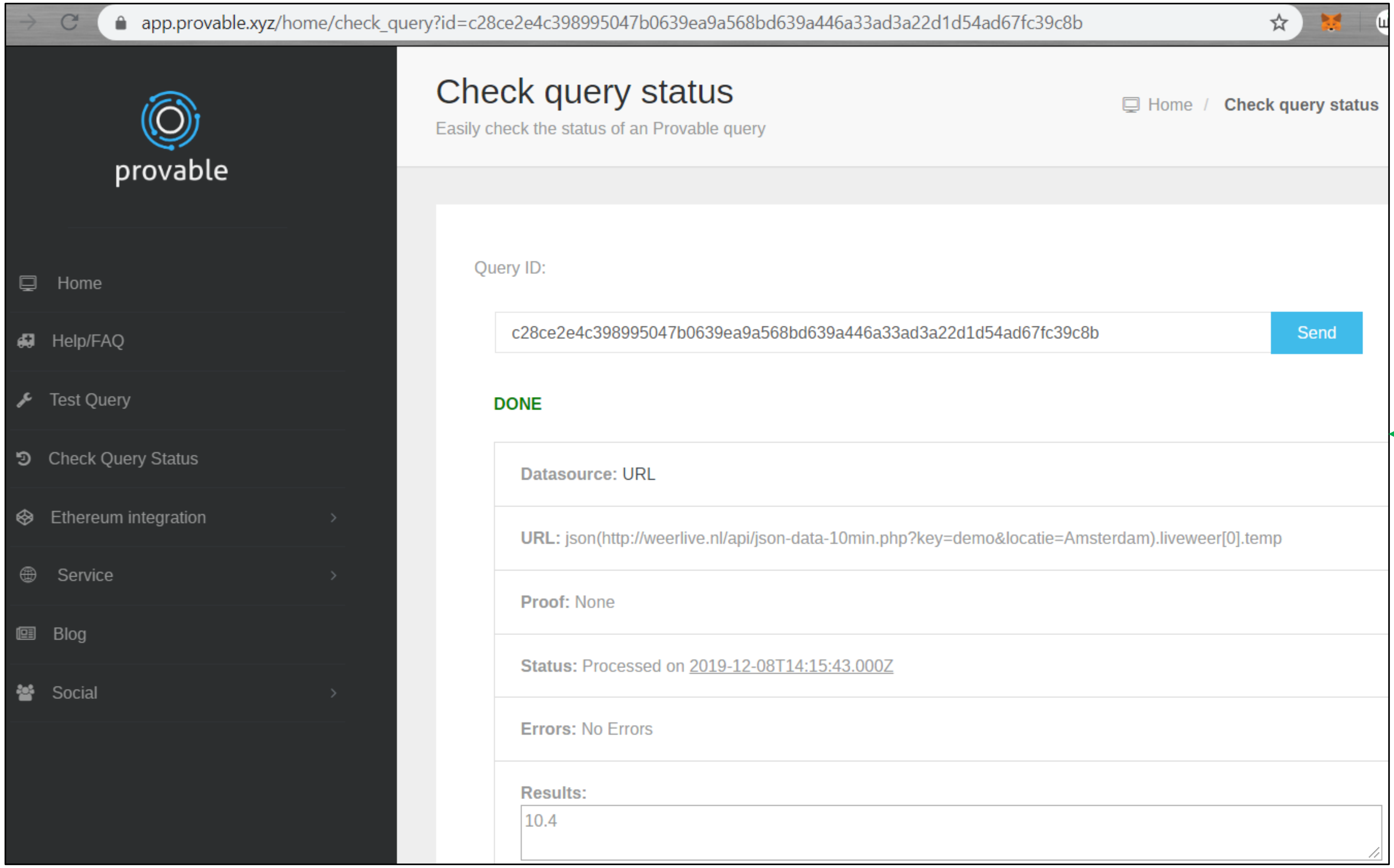
Query executed at 15:15:42 GMT+0100 (Central European Standard Time)

Result is: **10.4**

Received at 15:15:42 GMT+0100 (Central European Standard Time)

PD-8.1 Check status

Go to Provable query with ID: 0x617aedc9345e8f7e67f590fc4abfe60c54cb520e7a41737385e32d283e426398



app.provable.xyz/home/check_query?id=c28ce2e4c398995047b0639ea9a568bd639a446a33ad3a22d1d54ad67fc39c8b

Check query status

Easily check the status of an Provable query

Query ID:

c28ce2e4c398995047b0639ea9a568bd639a446a33ad3a22d1d54ad67fc39c8b

Send

DONE

Datasource: URL

URL: json(http://weerlive.nl/api/json-data-10min.php?key=demo&locatie=Amsterdam).liveweer[0].temp

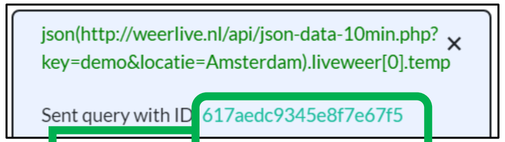
Proof: None

Status: Processed on 2019-12-08T14:15:43.000Z

Errors: No Errors

Results:

10.4

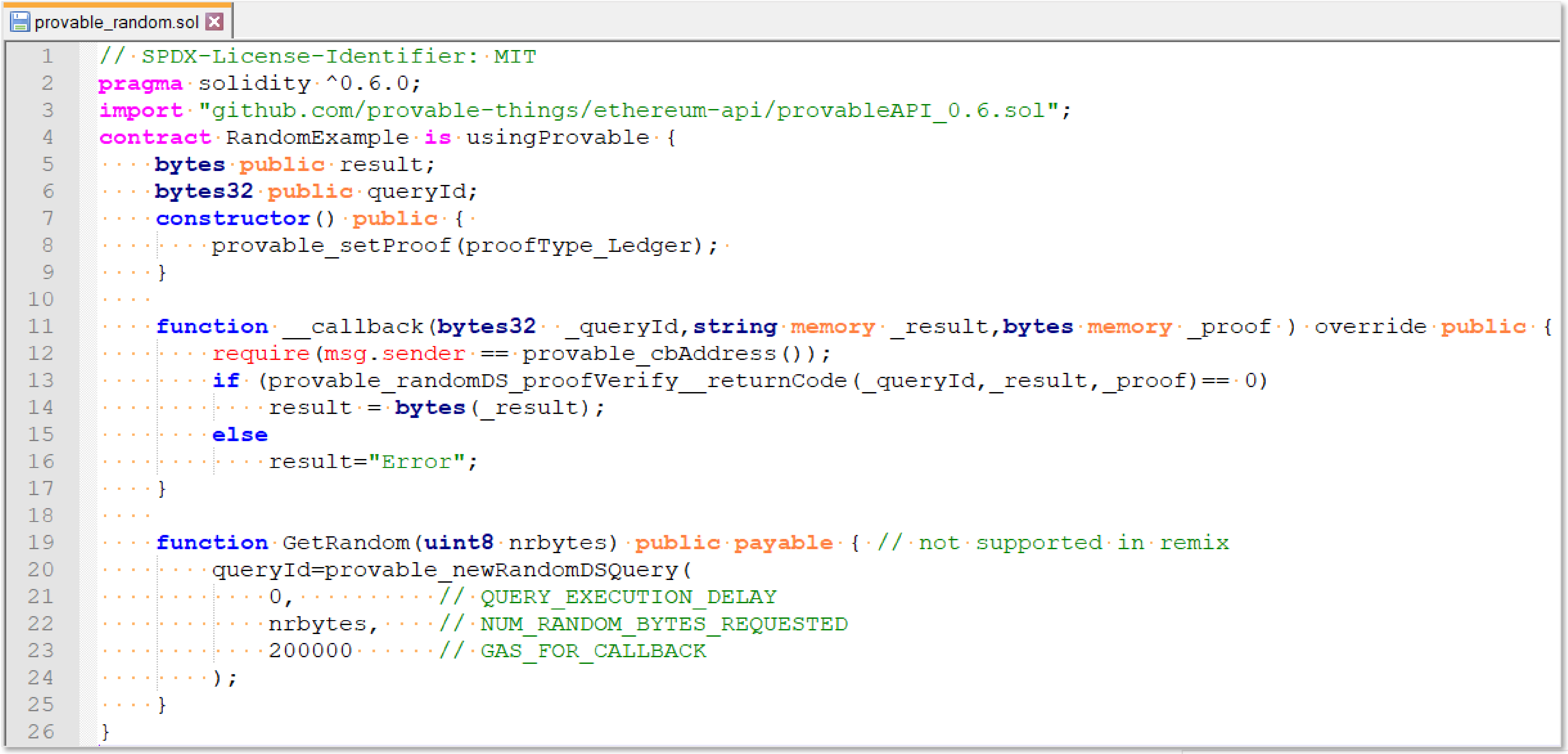


json(http://weerlive.nl/api/json-data-10min.php?key=demo&locatie=Amsterdam).liveweer[0].temp

Sent query with ID 617aedc9345e8f7e67f5

http://app.oraclize.it/home/check_query?id=c28ce2e4c398995047b0639ea9a568bd639a446a33ad3a22d1d54ad67fc39c8b

PD-8.2 Provable oracle random

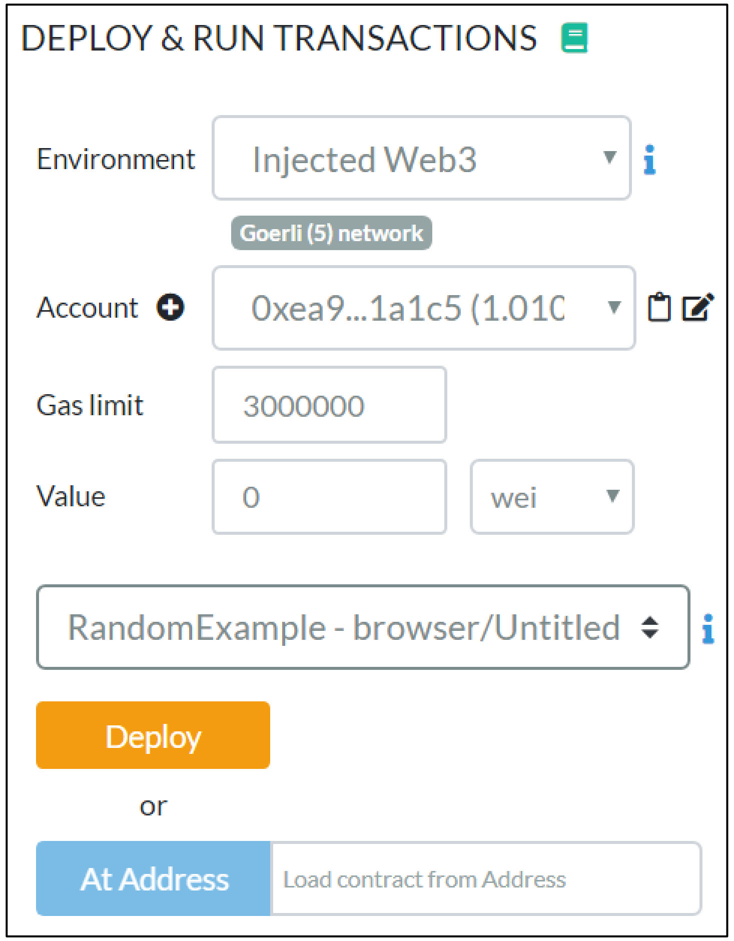


```
// SPDX-License-Identifier: MIT
pragma solidity ^0.6.0;
import "github.com/provable-things/ethereum-api/provableAPI_0.6.sol";
contract RandomExample is usingProvable {
    bytes public result;
    bytes32 public queryId;
    constructor() public {
        provable_setProof(proofType_Ledger);
    }
    
    function __callback(bytes32  _queryId,string memory _result,bytes memory _proof ) override public {
        require(msg.sender == provable_cbAddress());
        if (provable_randomDS_proofVerify__returnCode(_queryId,_result,_proof)== 0)
            result = bytes(_result);
        else
            result="Error";
    }
    
    function GetRandom(uint8 nrbytes) public payable { // not supported in remix
        queryId=provable_newRandomDSQuery(
            0,          // QUERY_EXECUTION_DELAY
            nrbytes,    // NUM_RANDOM_BYTES_REQUESTED
            200000      // GAS_FOR_CALLBACK
        );
    }
}
```

https://docs.provable.xyz/#data-sources-random

https://github.com/web3examples/ethereum/blob/master/oracle_examples/provable_random.sol

PD-8.2 Deploy via remix on Goerli



DEPLOY & RUN TRANSACTIONS

Environment: Injected Web3

Goerli (5) network

Account: 0xea9...1a1c5 (1.010

Gas limit: 3000000

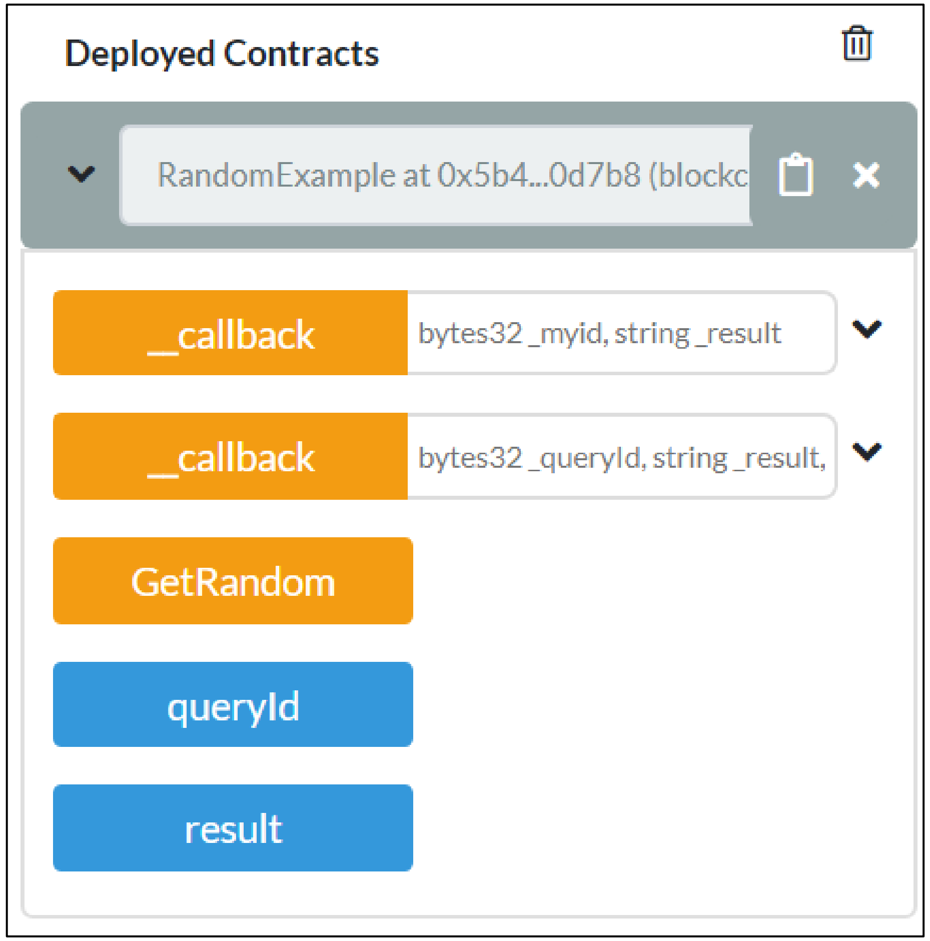
Value: 0 wei

RandomExample - browser/Untitled

Deploy

or

At Address | Load contract from Address



Deployed Contracts

RandomExample at 0x5b4...0d7b8 (blockc

_callback | bytes32 _myid, string _result

_callback | bytes32 _queryId, string _result,

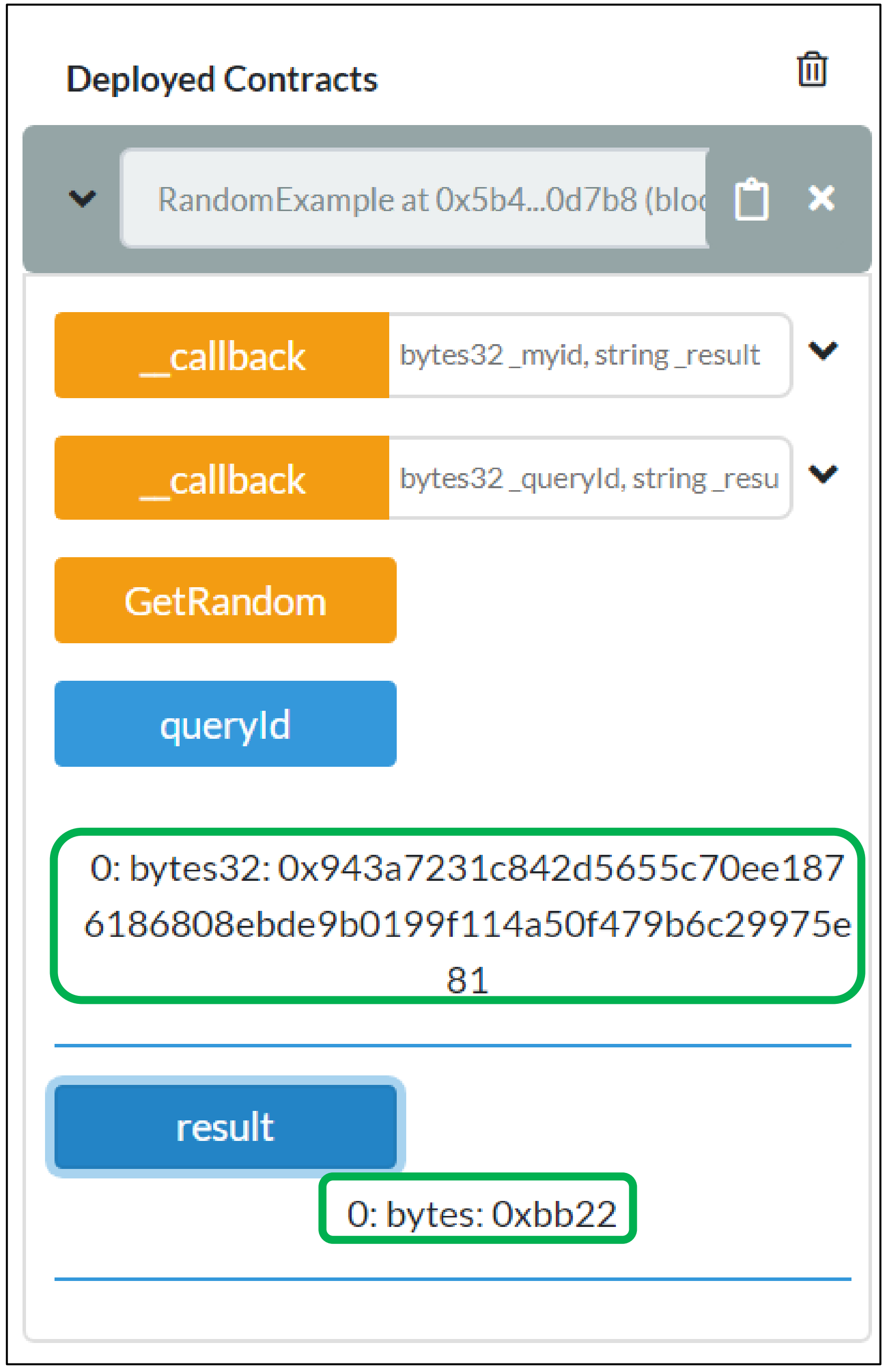
GetRandom

queryId

result

PD-8.2 Result

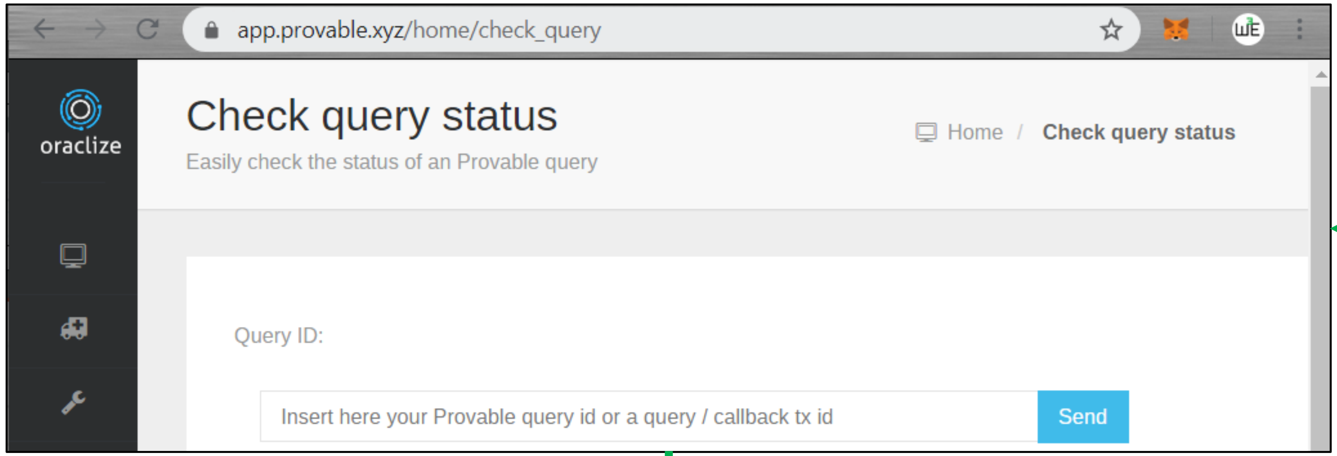
Wait until transaction is ready


Deployed Contracts
RandomExample at 0x5b4...0d7b8 (bloc
__callback
bytes32 _myid, string _result
__callback
bytes32 _queryId, string _resu
GetRandom
queryId
result

```
0: bytes32: 0x943a7231c842d5655c70ee187
6186808ebde9b0199f114a50f479b6c29975e
81
```

```
0: bytes: 0xbb22
```

PD-8.2 Check query status


app.provable.xyz/home/check_query

Check query status

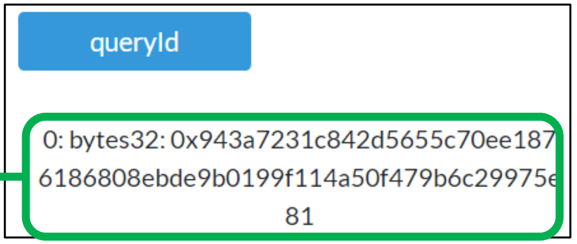
Easily check the status of an Provable query

Home / Check query status

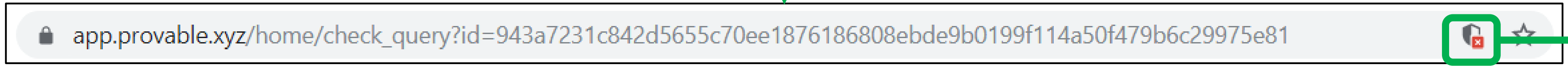
Query ID:

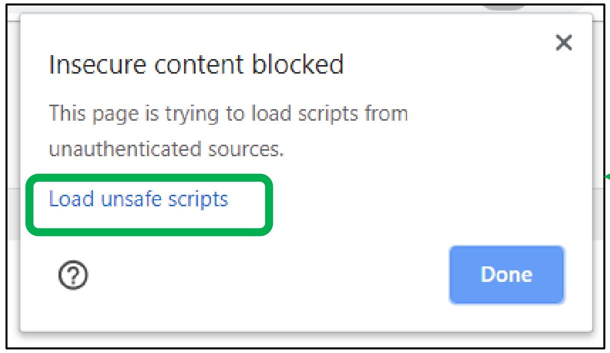
Insert here your Provable query id or a query / callback tx id

Send


queryId

0: bytes32: 0x943a7231c842d5655c70ee1876186808ebde9b0199f114a50f479b6c29975e81


app.provable.xyz/home/check_query?id=943a7231c842d5655c70ee1876186808ebde9b0199f114a50f479b6c29975e81


Insecure content blocked

This page is trying to load scripts from unauthenticated sources.

Load unsafe scripts

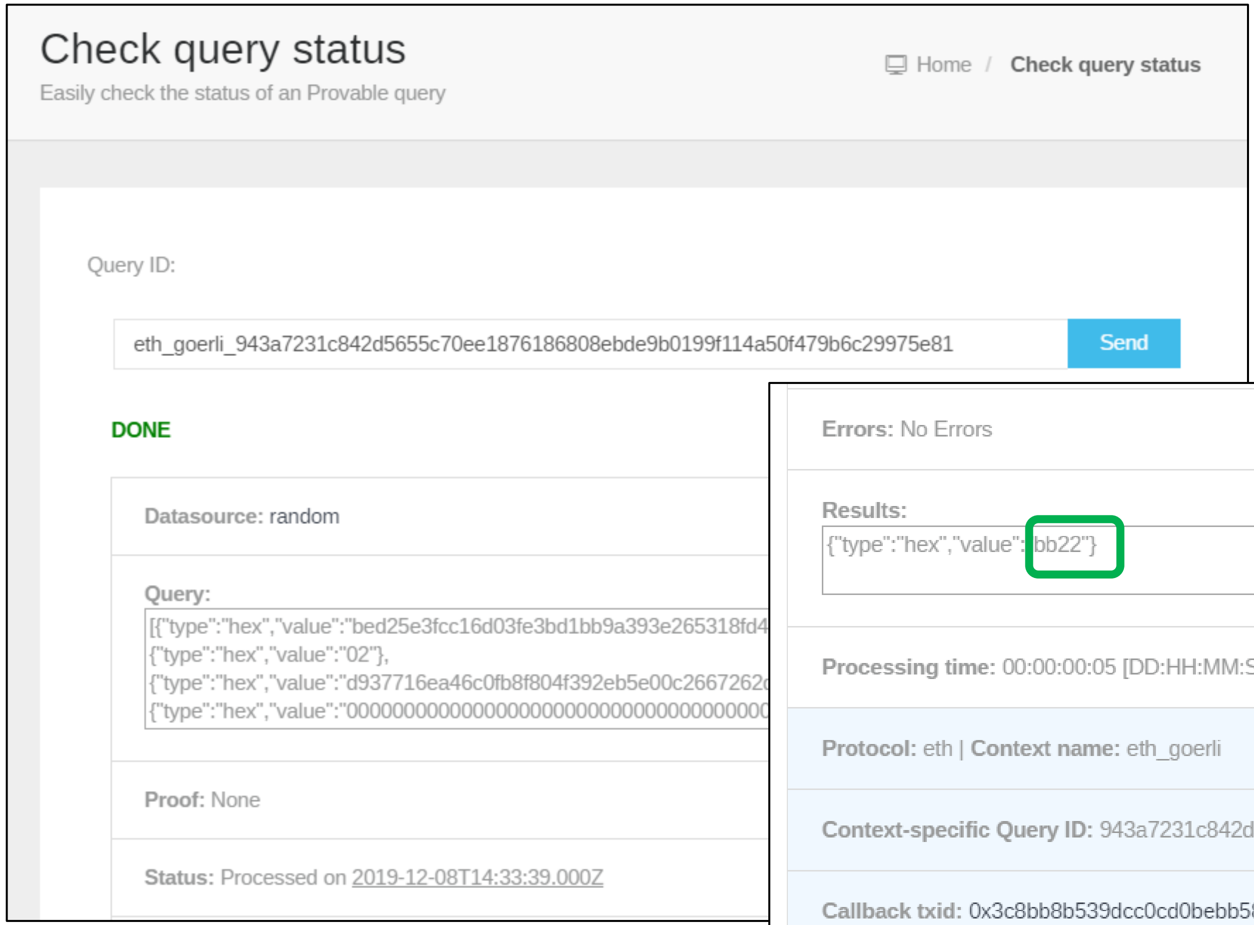
Done

https://app.provable.xyz/home/check_query

0x943a7231c842d5655c70ee1876186808ebde9b0199f114a50f479b6c29975e81

http://app.oraclize.it/home/check_query?id=943a7231c842d5655c70ee1876186808ebde9b0199f114a50f479b6c29975e81

PD-8.2 Check query status result



Check query status

Easily check the status of an Provable query

Home / **Check query status**

Query ID:

eth_goerli_943a7231c842d5655c70ee1876186808ebde9b0199f114a50f479b6c29975e81

Send

DONE

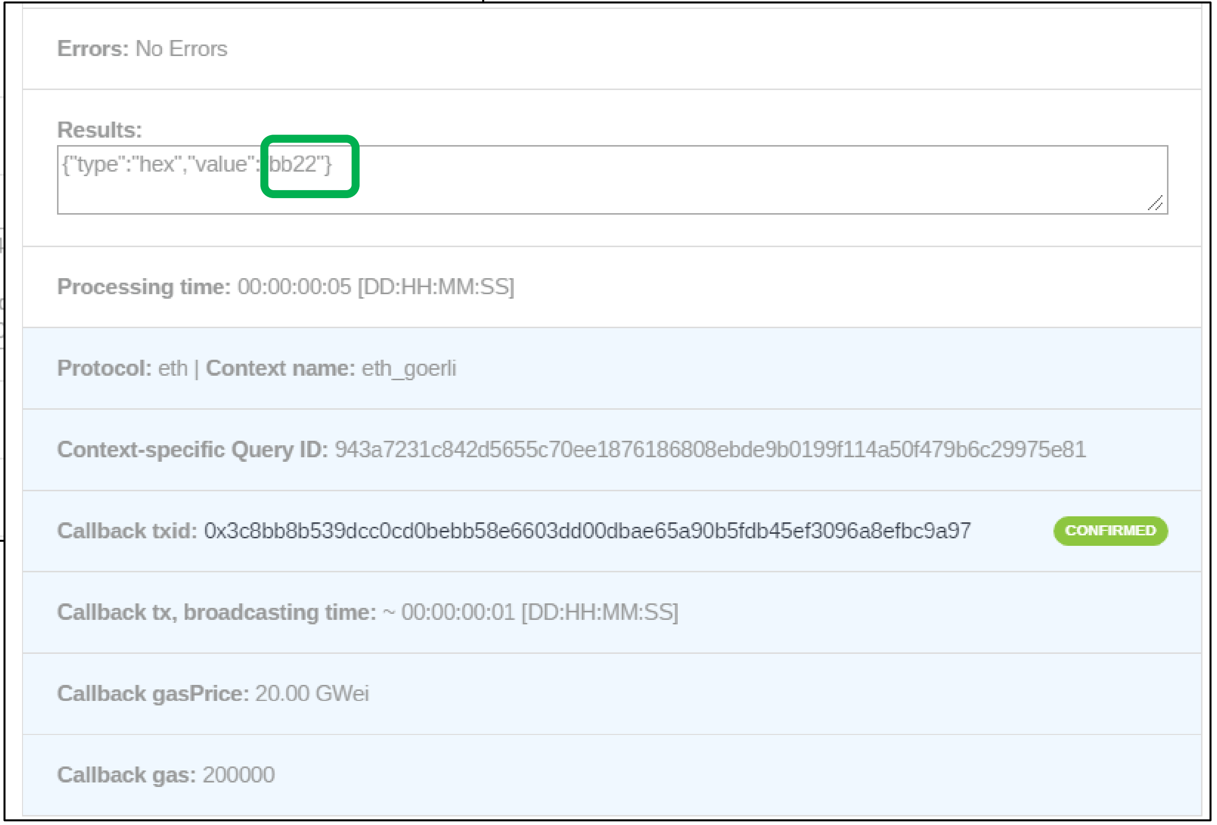
Datasource: random

Query:

```
[{"type":"hex","value":"bed25e3fcc16d03fe3bd1bb9a393e265318fd4
{"type":"hex","value":"02"},
{"type":"hex","value":"d937716ea46c0fb8f804f392eb5e00c2667262
{"type":"hex","value":"000000000000000000000000000000000000
```

Proof: None

Status: Processed on 2019-12-08T14:33:39.000Z



Errors: No Errors

Results:

```
{"type":"hex","value":"bb22"}
```

Processing time: 00:00:00:05 [DD:HH:MM:SS]

Protocol: eth | **Context name:** eth_goerli

Context-specific Query ID: 943a7231c842d5655c70ee1876186808ebde9b0199f114a50f479b6c29975e81

Callback txid: 0x3c8bb8b539dcc0cd0bebb58e6603dd00dbae65a90b5fdb45ef3096a8efbc9a97 CONFIRMED

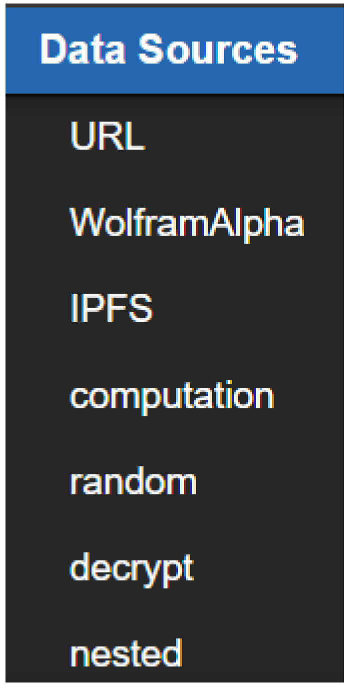
Callback tx, broadcasting time: ~ 00:00:00:01 [DD:HH:MM:SS]

Callback gasPrice: 20.00 GWei

Callback gas: 200000

http://app.oraclize.it/home/check_query?id=943a7231c842d5655c70ee1876186808ebde9b0199f114a50f479b6c29975e81

PD-8.3 Provable other possibilities



1. URL: access to any API or web page on the Internet
2. WolframAlpha: access to the WolframAlpha Knowledge Engine API
3. IPFS: retrieve the content of a file on the IPFS network
4. Computation: execution of an application or a script on a sandboxed Amazon Web Service virtual machine
5. Random: randomness from a secure Ledger device
6. Decryp: decrypt data
7. Nested: comnation of sources

https://docs.provable.xyz/#data-sources

PD-8.3 Parsing Helpers

1. **JSON** Parsing: using JSONPATH standard
   json(https://api.kraken.com/0/public/Ticker?pair=ETHUSD).result.XETHZUSD.c.0.

2. **XML** Parser: using XPATH standard
   xml(https://www.fueleconomy.gov/ws/rest/fuelprices).fuelPrices.diesel.

3. **HTML** Parser: using XPATH standard
   html(https://twitter.com/oraclizeit/status/671316655893561344).xpath(//*[contains(@class, 'tweet-text')]/text()).

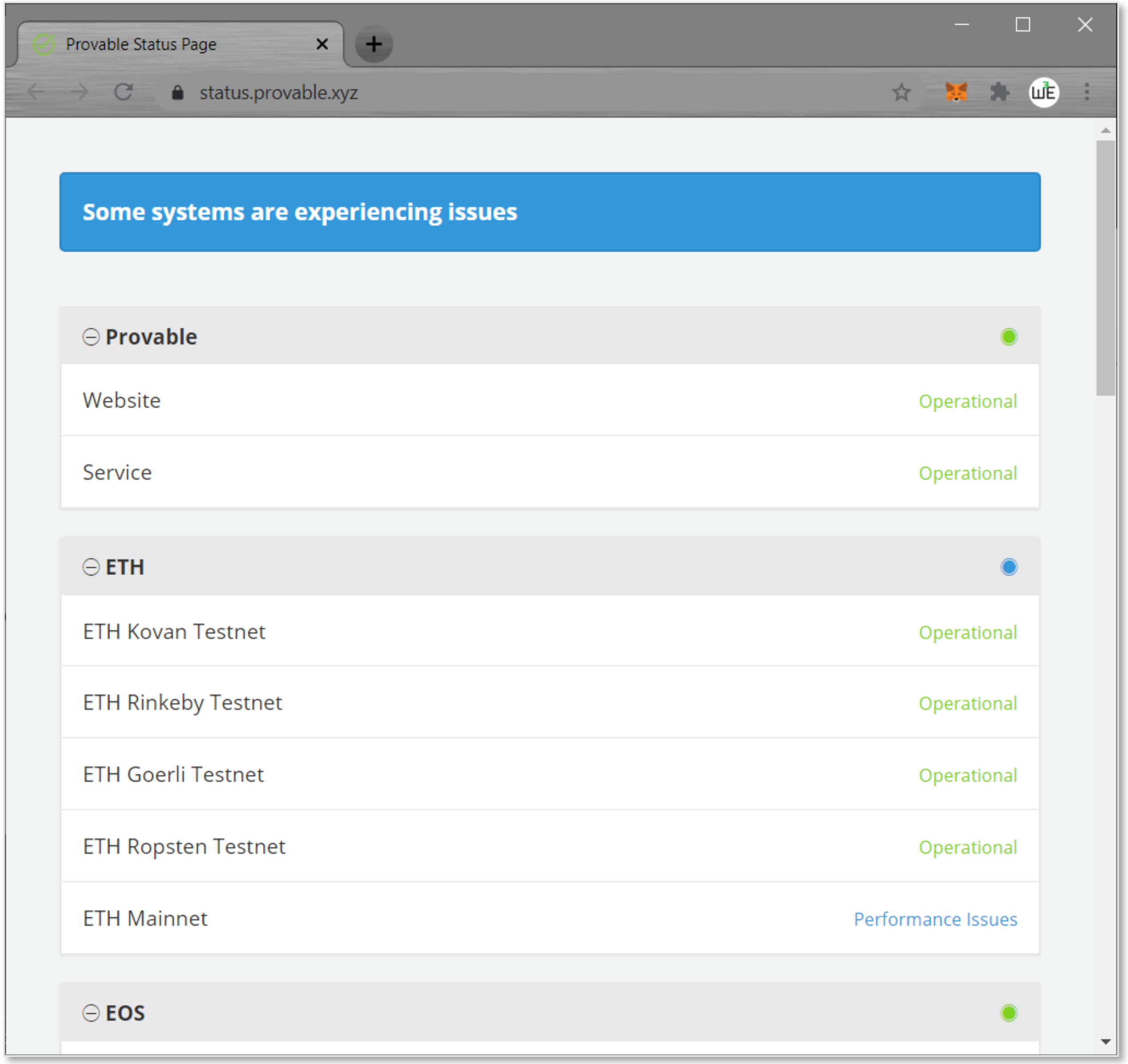
4. **Binary** Helper: Slice binary data
   binary(https://www.sk.ee/crls/esteid/esteid2015.crl).slice(0,300)

https://www.w3.org/TR/xpath/all/

https://github.com/FlowCommunications/JSONPath#expression-syntax

https://docs.provable.xyz/#general-concepts-parsing-helpers

PD-8.3 Status provable



Provable Status Page

status.provable.xyz

Some systems are experiencing issues

Provable	●
Website	Operational
Service	Operational

ETH	●
ETH Kovan Testnet	Operational
ETH Rinkeby Testnet	Operational
ETH Goerli Testnet	Operational
ETH Ropsten Testnet	Operational
ETH Mainnet	Performance Issues

EOS ●

This shows which networks are supported.

https://status.provable.xyz/

PD-8.3 Schedule a Query in the Future

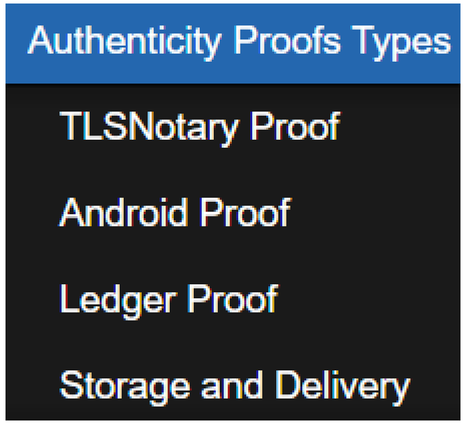
Schedule a Query in the Fut…

The execution of a query can be scheduled in a future date.

No more than 60 days

https://docs.provable.xyz/#ethereum-quick-start-schedule-a-query-in-the-future

PD-8.3 Proofs


Authenticity Proofs Types

TLSNotary Proof

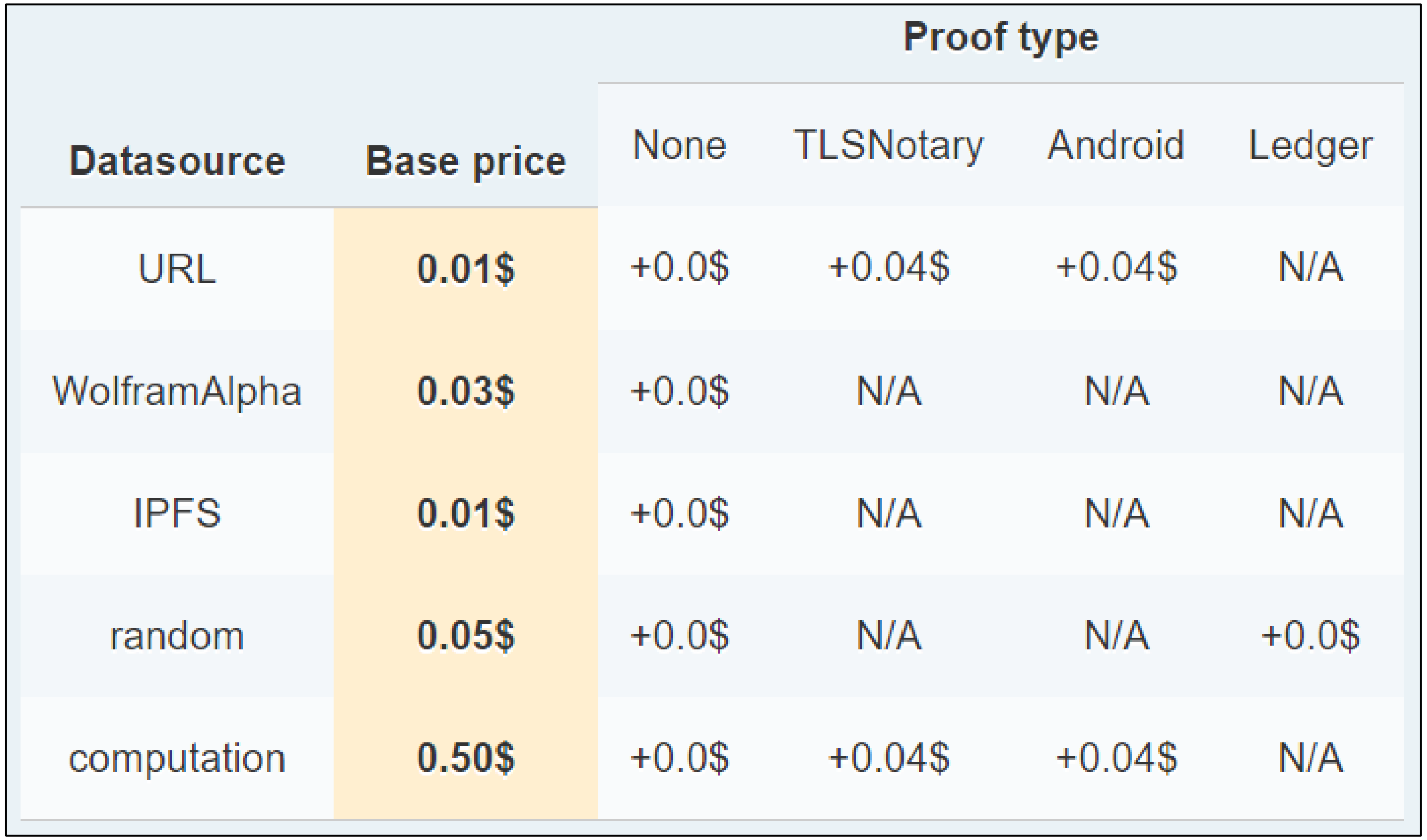
Android Proof

Ledger Proof

Storage and Delivery

https://docs.provable.xyz/#security-deepdive-authenticity-proofs-types

PD-8.3 Pricing



Datasource	Base price	Proof type			
		None	TLSNotary	Android	Ledger
URL	**0.01$**	+0.0$	+0.04$	+0.04$	N/A
WolframAlpha	**0.03$**	+0.0$	N/A	N/A	N/A
IPFS	**0.01$**	+0.0$	N/A	N/A	N/A
random	**0.05$**	+0.0$	N/A	N/A	+0.0$
computation	**0.50$**	+0.0$	+0.04$	+0.04$	N/A

https://docs.provable.xyz/#pricing-advanced-datasources-call-fee

PD-8.3 Further examples

https://github.com/provable-things/ethereum-examples/tree/master/solidity

https://github.com/provable-things/ethereum-examples/tree/master/solidity/truffle-examples

https://www.youtube.com/channel/UCjVjCheDbMel-x-JYeGazcQ/featured

PD-8.4 Chainlink Oracle URL

https://chain.link

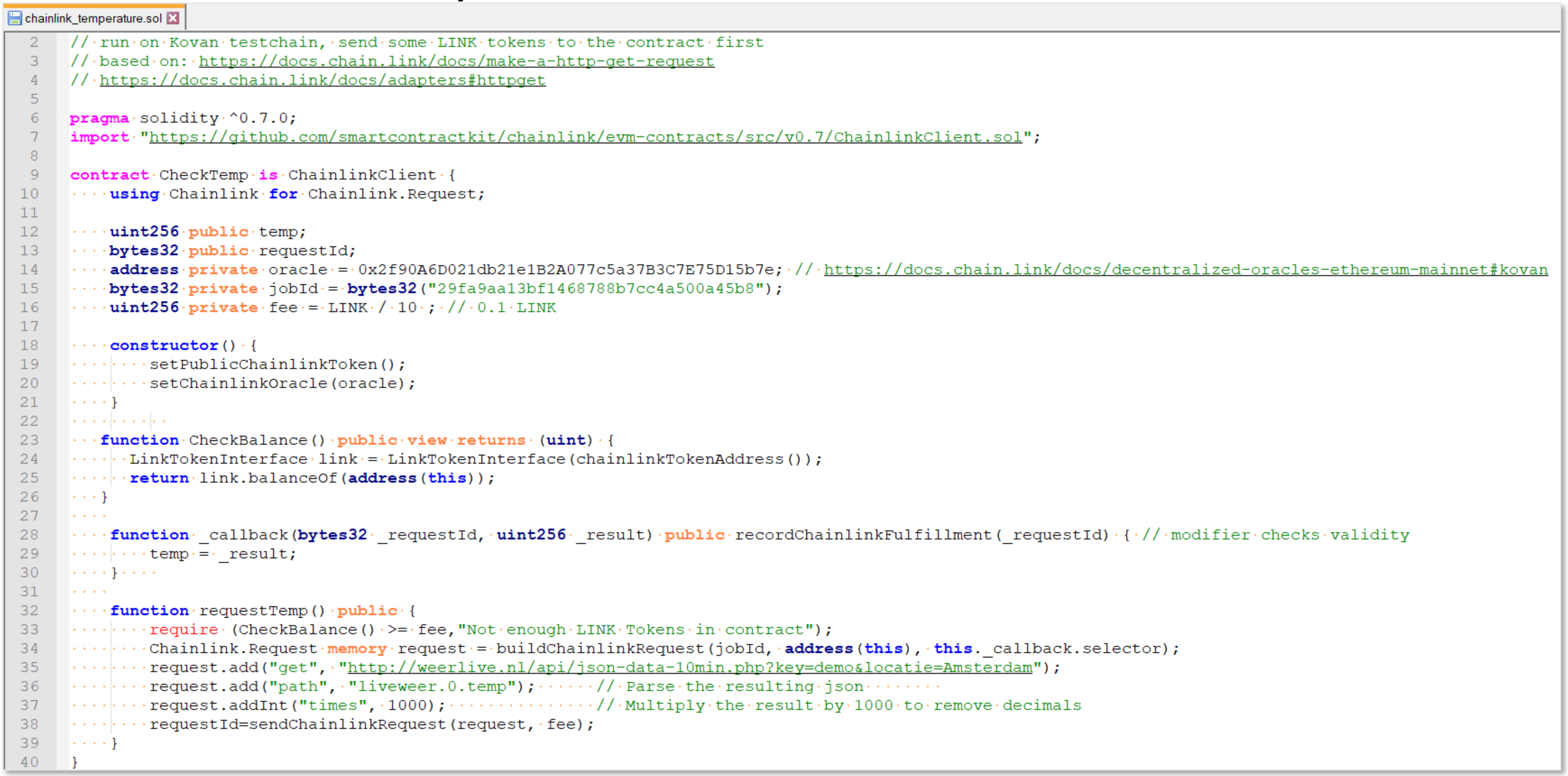
https://kovan.chain.link

https://docs.chain.link/docs/acquire-link

https://docs.chain.link/docs/deploy-your-first-contract

https://docs.chain.link/docs/decentralized-oracles-ethereum-mainnet#kovan

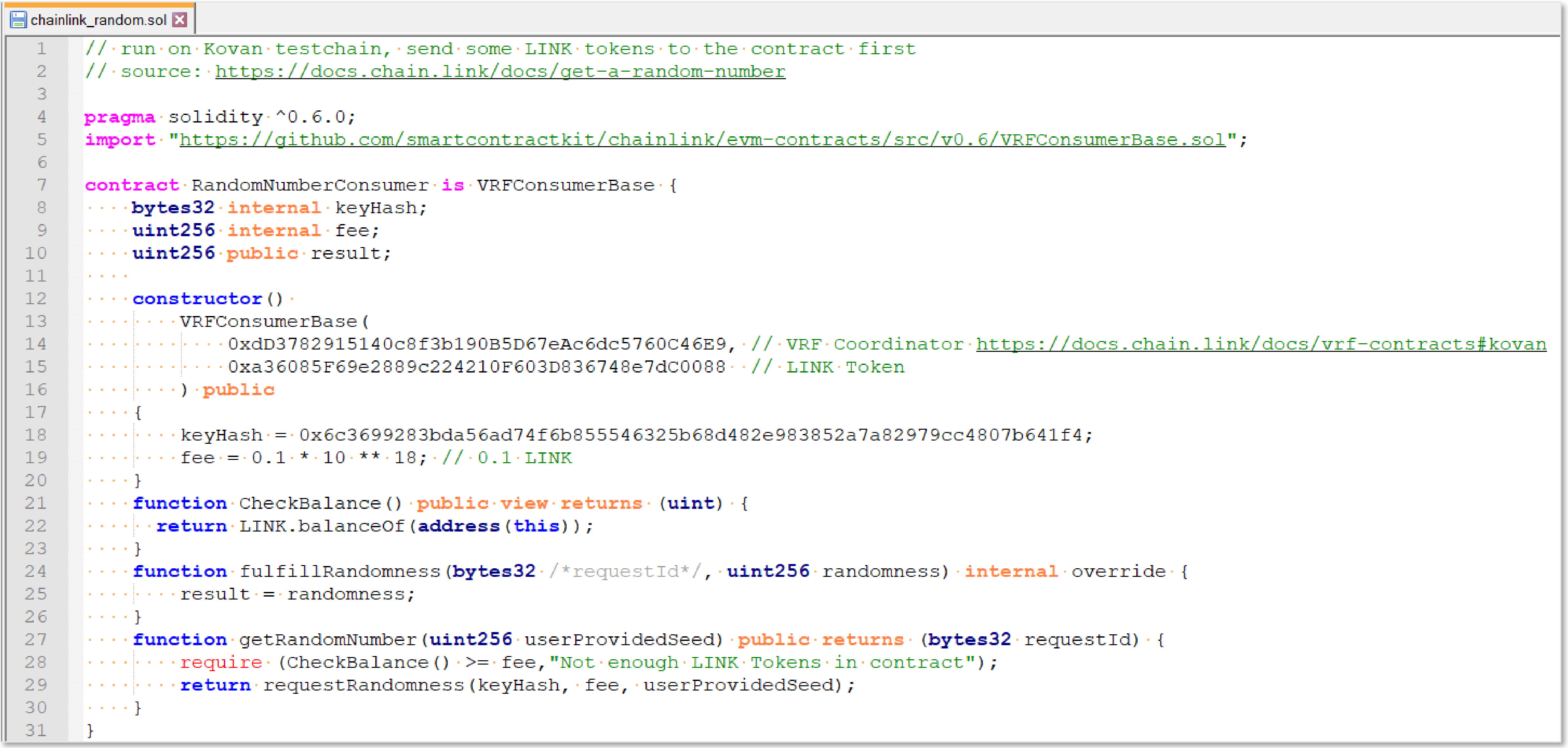
PD-8.4 Solidity code


chainlink_temperature.sol

```
// run on Kovan testchain, send some LINK tokens to the contract first
// based on: https://docs.chain.link/docs/make-a-http-get-request
// https://docs.chain.link/docs/adapters#httpget

pragma solidity ^0.7.0;
import "https://github.com/smartcontractkit/chainlink/evm-contracts/src/v0.7/ChainlinkClient.sol";

contract CheckTemp is ChainlinkClient {
    using Chainlink for Chainlink.Request;

    uint256 public temp;
    bytes32 public requestId;
    address private oracle = 0x2f90A6D021db21e1B2A077c5a37B3C7E75D15b7e; // https://docs.chain.link/docs/decentralized-oracles-ethereum-mainnet#kovan
    bytes32 private jobId = bytes32("29fa9aa13bf1468788b7cc4a500a45b8");
    uint256 private fee = LINK / 10 ; // 0.1 LINK

    constructor() {
        setPublicChainlinkToken();
        setChainlinkOracle(oracle);
    }

   function CheckBalance() public view returns (uint) {
      LinkTokenInterface link = LinkTokenInterface(chainlinkTokenAddress());
      return link.balanceOf(address(this));
   }

    function _callback(bytes32 _requestId, uint256 _result) public recordChainlinkFulfillment(_requestId) { // modifier checks validity
        temp = _result;
    }

    function requestTemp() public {
        require (CheckBalance() >= fee,"Not enough LINK Tokens in contract");
        Chainlink.Request memory request = buildChainlinkRequest(jobId, address(this), this._callback.selector);
        request.add("get", "http://weerlive.nl/api/json-data-10min.php?key=demo&locatie=Amsterdam");
        request.add("path", "liveweer.0.temp");      // Parse the resulting json
        request.addInt("times", 1000);               // Multiply the result by 1000 to remove decimals
        requestId=sendChainlinkRequest(request, fee);
    }
}
```

https://github.com/web3examples/ethereum/blob/master/oracle_examples/chainlink_temperature.sol

PD-8.5 Chainlink Random



```solidity
// run on Kovan testchain, send some LINK tokens to the contract first
// source: https://docs.chain.link/docs/get-a-random-number

pragma solidity ^0.6.0;
import "https://github.com/smartcontractkit/chainlink/evm-contracts/src/v0.6/VRFConsumerBase.sol";

contract RandomNumberConsumer is VRFConsumerBase {
    bytes32 internal keyHash;
    uint256 internal fee;
    uint256 public result;
    
    constructor() 
        VRFConsumerBase(
            0xdD3782915140c8f3b190B5D67eAc6dc5760C46E9, // VRF Coordinator https://docs.chain.link/docs/vrf-contracts#kovan
            0xa36085F69e2889c224210F603D836748e7dC0088  // LINK Token
        ) public
    {
        keyHash = 0x6c3699283bda56ad74f6b855546325b68d482e983852a7a82979cc4807b641f4;
        fee = 0.1 * 10 ** 18; // 0.1 LINK
    }
    function CheckBalance() public view returns (uint) {
      return LINK.balanceOf(address(this));
    }
    function fulfillRandomness(bytes32 /*requestId*/, uint256 randomness) internal override {
        result = randomness;
    }
    function getRandomNumber(uint256 userProvidedSeed) public returns (bytes32 requestId) {
        require (CheckBalance() >= fee,"Not enough LINK Tokens in contract");
        return requestRandomness(keyHash, fee, userProvidedSeed);
    }
}
```

https://docs.chain.link/docs/chainlink-vrf

https://docs.chain.link/docs/get-a-random-number

https://docs.chain.link/docs/vrf-contracts#kovan

https://github.com/web3examples/ethereum/blob/master/oracle_examples/chainlink_random.sol