

# PD-5.0 Web3.js

- PD-5.1 Javascript
- PD-5.2 Web3.js in Node.js
- PD-5.3 Web3.js in browser
- PD-5.4 Web3.js in browser  
(more advanced examples)



# PD-5.1 Javascript

# PD-5.1.1 Microsoft JavaScript course



# PD-5.1.2 Arrow functions

```
arrow.js x
1  function Test1 (x) {
2      console.log (x) ;
3  }
4
5  const Test2 = y => console.log (y) ;
6
7  Test1 (1) ;
8
9  Test2 (2) ;
10
11 (z => console.log (z)) (3)
```

## Output:

```
1
2
3
```

# PD-5.1.2 Then

```
then.js x
1  // .npm .install .node-fetch
2  const fetch = require('node-fetch');
3
4  function StepTwo (Response) {
5      ... return Response.json();
6  }
7
8  function StepThree (content) {
9      ... console.log (content [0] .name) ;
10     }
11
12  fetch ('https://chainid.network/chains.json') .then (StepTwo) .then (StepThree) ;
```

## Output:

Ethereum Mainnet

# PD-5.1.2 Then and Arrow

```
then_arrow.js x
1 // ·npm·install·node-fetch
2 const·fetch·=·require('node-fetch');
3
4 fetch('https://chainid.network/chains.json')
5   .then(·Response·=>·Response.json())
6   .then(·content·=>·console.log(content[0].name))
```

## Output:

Ethereum Mainnet

# PD-5.1.2 Catch

```
catch.js x
1 // npm install node-fetch
2 const fetch = require('node-fetch');
3
4 fetch('https://chainid.network/chains.json2')
5   .then(response => response.json())
6   .catch(err => console.error('Catch 1: fetch failed', err.message))
7   .then(content => console.log(content[0].name))
8   .catch(err => console.error('Catch 2: fetch failed', err.message))
```

**Output:**

Catch 1: fetch failed invalid json response body at https://chainid.network/chains.json2 reason: Unexpected token < in JSON at position 0  
Catch 2: fetch failed Cannot read property '0' of undefined

# PD-5.1.2 Async

```
async.js x
1 // npm install node-fetch
2 const fetch = require('node-fetch');
3
4 async function asyncFunction() {
5   const Response = await fetch('https://chainid.network/chains.json');
6   const content = await Response.json();
7   console.log(content[0].name);
8 }
9 asyncFunction();
```

**Output:**  
Ethereum Mainnet



# PD-5.1.3 Template Literal

```
template_literal.js x
1  var x=1;
2  var y=4;
3  console.log(`x=${x} · y=${y} · x+y=${x+y}`);
```

**Output:**

x=1 y=4 x+y=5

# PD-5.1.4 VarLetConst

```
var.js x
1 // var declarations are globally scoped or function scoped
2 // var variables can be re-declared (in the same scope)
3 // var variables are initialized with undefined
4
5 var greeter
6 console.log(greeter);
7 var greeter = "hey hi";
8 console.log(greeter);
9 {
10   ... var greeter = "new block now";
11   ... console.log(greeter);
12 }
13 console.log(greeter);
```

**Output:**  
undefined  
hey hi  
new block now  
new block now

```
constjs x
1 // const declarations are block scoped
2 // const cannot be updated or re-declared
3 // const have to be initialized
4
5 const greeter = "hey hi";
6 console.log(greeter)
7 {
8   ... const greeter = "new block now";
9   ... console.log(greeter)
10 }
11 console.log(greeter)
```

**Output:**  
hey hi  
new block now  
hey hi

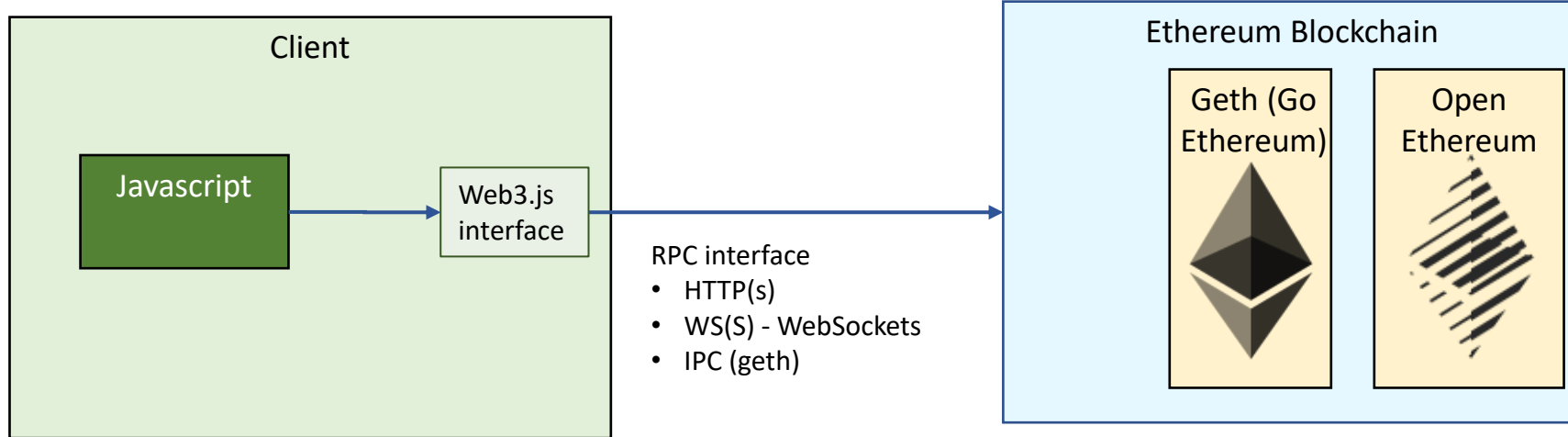
```
let.js x
1 // let is block scoped
2 // let variables are initialized with undefined
3
4 let greeter
5 console.log(greeter)
6 greeter = "hey hi";
7 console.log(greeter)
8 {
9   ... let greeter = "new block now";
10  ... console.log(greeter)
11 }
12 console.log(greeter)
```

**Output:**  
undefined  
hey hi  
new block now  
hey hi

- <https://github.com/web3examples/javascript/blob/master/var.js>
- <https://github.com/web3examples/javascript/blob/master/let.js>
- <https://github.com/web3examples/javascript/blob/master/const.js>

# PD-5.2 Web3.js in Node.js

# PD-5.2.0.0 Interfaces



<https://github.com/ethereum/wiki/wiki/JSON-RPC>

<https://github.com/ethereum/wiki/wiki/JSON-RPC#json-rpc-support>

# PD-5.2.0.0 Web3.js version 1.3.0

```
>npm install -g web3
```

```
..
```

```
+ web3@1.3.0
```

```
added 43 packages from 108 contributors, removed 59 packages
```

```
> npm link web3
```

```
.\node_modules\web3 -> C:\Users\..\AppData\Roaming\npm\node_modules\web3
```

```
> Node
```

```
Welcome to Node.js v12.16.3.
```

```
>Web3 = require('web3');web3 = new Web3();web3.version;
```

```
'1.3.0'
```

<https://web3js.readthedocs.io/en/latest>

[http://web3examples.com/ethereum/install/Install\\_Web3\\_Windows.html](http://web3examples.com/ethereum/install/Install_Web3_Windows.html)

[http://web3examples.com/ethereum/install/Install\\_Web3\\_Macintosh.html](http://web3examples.com/ethereum/install/Install_Web3_Macintosh.html)

[http://web3examples.com/ethereum/install/Install\\_Web3\\_Ubuntu.html](http://web3examples.com/ethereum/install/Install_Web3_Ubuntu.html)

# PD-5.2.0.0 Version

```
version.js x
1 Web3 = require('web3');
2 web3 = new Web3();
3 console.log(web3.version);
4
```

```
> Node version.js
1.3.0
```

# PD-5.2.0.0 getBlockNumber via Cloudflare

```
getBlockNumber.js x
1 Web3 = require('web3');
2 var web3 = new Web3("https://cloudflare-eth.com");
3 console.log(web3.version);
4 web3.eth.getBlockNumber()
5 .....then(console.log);
6
```

> Node getBlockNumber\_cloudflare.js

1.3.0

11196806

# PD-5.2.0.0 getBlockNumber via Infura

```
getBlockNumber_infura.js x
1  const Web3 = require('web3');
2  const fs = require('fs');
3  const infuraKey = fs.readFileSync('.infura').toString().trim(); // infura key
4  const web3 = new Web3(`https://mainnet.infura.io/v3/${infuraKey}`);
5  console.log(web3.version);
6  web3.eth.getBlockNumber()
7  .then(console.log);
```

First put Infura key in the file “.infura”

```
> Npm install fs
```

```
> Node getBlockNumber_infura.js
```

```
1.3.0
```

```
11196806
```



# PD-5.2.0.0 Vanity account

```
vanity_account.js x
1 Web3 = require('web3');
2 web3 = new Web3();
3 var i=0;
4 var find="123";
5 var findlength_plus2=find.length+2;
6 var prefix;
7 do {
8     ...newAddress=web3.eth.accounts.create();
9     ...prefix=newAddress.address.slice(2,findlength_plus2).toLowerCase();
10    ...if(++i%1000==0) console.log(i);
11 } while(prefix!=find);
12 console.log(`Found an address with prefix ${prefix}`);
13 console.log(`address=${newAddress.address}`);
14 console.log(`privatekey=${newAddress.privateKey}`);
15
```

> Node vanity\_account.js

Found an address with prefix 123

address=0x123b3E19fF0EB023DFE0A51cc38EE13f4753335C

privatekey=0xe200905aaa05592c2bc81ca9b263e45e8fe09374fb297bbfa0150d021e101a3b

# PD-5.2.0.0 Sign data

```
sign.js x
1  const Web3 = require('web3');
2  const web3 = new Web3();
3  async function f() {
4      ... const privateKey = web3.utils.keccak256('Test'); // Make a "random" private key (not secure)
5      ... const address = web3.eth.accounts.privateKeyToAccount(privateKey); // Make an account based on the private key
6      ... const toSign = "This is a test";
7      ... const signedData = await web3.eth.accounts.sign(toSign, privateKey);
8      ... console.log(`Signing "${toSign}" by account ${address.address}`);
9      ... console.log(`gives signature ${signedData.signature}`);
10     ...
11     ... const signer = web3.eth.accounts.recover(toSign, signedData.signature);
12     ... console.log(`Knowing the signed text: "${toSign}"`);
13     ... console.log(`we can get the signer: ${signer}`);
14 }
15 f();
```

**>node sign.js**

Signing "This is a test" by account 0x33347BA01fAF9ae8C32f9029a65C7f2Fdc38d865

gives signature 0x9396552f3178d0ec2aba7e51c3ce487b2c49b75d91bac9a1381c6c43da8a34881bf177986c5927a6566f6de239ccd498532e2fabfa9392ac06493b00de38d87b1c

Knowing the signed text: "This is a test"

we can get the signer: 0x33347BA01fAF9ae8C32f9029a65C7f2Fdc38d865

# PD-5.2.0.0 sendTransaction

```
sendtransaction.js x
1  const Web3 = require('web3');
2  const web3 = new Web3('http://localhost:8545');
3  async function f() {
4      ... var acts = await web3.eth.getAccounts();
5      ... var fromadr = acts[0];
6      ... var toadr = acts[1];
7      ... console.log(`fromadr ${fromadr} has ${Web3.utils.fromWei(await web3.eth.getBalance(fromadr), 'ether')} ether`);
8      ... console.log(`toadr ... ${toadr} has ${Web3.utils.fromWei(await web3.eth.getBalance(toadr), 'ether')} ether`);
9      ... console.log(`Transferring 1 ether`);
10     ... obj = await web3.eth.sendTransaction({
11         ... from: fromadr,
12         ... to: toadr,
13         ... value: Web3.utils.toWei('1', 'ether')
14     ... }).catch(console.log);
15     ... console.log(`Stored in block ${obj.blockNumber}`);
16     ... console.log(`fromadr ${fromadr} has ${Web3.utils.fromWei(await web3.eth.getBalance(fromadr), 'ether')} ether`);
17     ... console.log(`toadr ... ${toadr} has ${Web3.utils.fromWei(await web3.eth.getBalance(toadr), 'ether')} ether`);
18 }
19 f();
```

**>node sendtransaction.js**

fromadr 0x6c728716a68499d486cDA1701AB13C7b57f30aA0 has 95.7614514 ether

toadr 0xC30e12E40A95976B74dF45f021f1351847fc44B9 has 102 ether

Transferring 1 ether

Stored in block 369

fromadr 0x6c728716a68499d486cDA1701AB13C7b57f30aA0 has 94.7610314 ether

toadr 0xC30e12E40A95976B74dF45f021f1351847fc44B9 has 103 ether



# PD-5.2.0.0 Transfer contract - ContractBalance

```
contract.sol output.json
1 // https://raw.githubusercontent.com/web3examples/ethereum/master/solidity_examples/Transfer.sol
2 pragma solidity >=0.4.0 <0.7.0;
3 contract TestPay {
4     function ContractBalance() public view returns (uint) { return address(this).balance; }
```

```
contract.sol output.json
[
  {
    "name": "TestPay",
    "abi": [
      {
        "payable": true,
        "stateMutability": "payable",
        "type": "fallback"
      },
      {
        "constant": true,
        "inputs": [],
        "name": "ContractBalance",
        "outputs": [
          {
            "internalType": "uint256",
            "name": "",
            "type": "uint256"
          }
        ],
        "payable": false,
        "stateMutability": "view",
        "type": "function"
      }
    ]
  }
]
```

```
const TestPayABI=[{
  ...."constant": true,
  ...."inputs": [],
  ...."name": "ContractBalance",
  ...."outputs": [{"internalType": "uint256","name": "", "type": "uint256"}],
  ...."payable": false,
  ...."stateMutability": "view",
  ...."type": "function"
  ....}];
```

# PD-5.2.0.0 Smart contract call

```
call_ropsten.js x
1  const Web3 = require('web3');
2  const fs = require('fs');
3  const infuraKey = fs.readFileSync('.infura').toString().trim(); // infura key
4  const web3 = new Web3(`https://ropsten.infura.io/v3/${infuraKey}`);
5  const TestPayABI = [{
6    ... "constant": true,
7    ... "inputs": [],
8    ... "name": "ContractBalance",
9    ... "outputs": [{"internalType": "uint256", "name": "", "type": "uint256"}],
10   ... "payable": false,
11   ... "stateMutability": "view",
12   ... "type": "function"
13   ... }];
14  async function f() {
15   ... const TestPayaddress = "0x01848a2cFc60d5b24EB6630F12048C1825059Afc"
16   ... const ContractTestPay = new web3.eth.Contract(TestPayABI, TestPayaddress);
17   ... var result = await ContractTestPay.methods.ContractBalance().call();
18   ... console.log(`ContractBalance shows ${Web3.utils.fromWei(result)} ether`);
19   ... var x = Web3.utils.fromWei(await web3.eth.getBalance(TestPayaddress), 'ether');
20   ... console.log(`TestPayaddress ${TestPayaddress} has ${x} ether`);
21  }
22  f();
```

> node call\_ropsten.js

ContractBalance shows 0.195 ether

TestPayaddress 0x01848a2cFc60d5b24EB6630F12048C1825059Afc has 0.195 ether

# PD-5.2.0.0 Transfer contract - PayToContract

TestPay 

published  
Tue Nov 26 2019 13:26:02 GMT+0100

contract address (unknown):  
0xFa01Aa6D84Db3EAD322573478132F70718341F59

published by  
0x6B5bB8441DD079F8Da87FF48F74F3A4F08bf417B

```
contract.sol  output.json  
},  
{  
  "constant": false,  
  "inputs": [],  
  "name": "PayToContract",  
  "outputs": [],  
  "payable": true,  
  "stateMutability": "payable",  
  "type": "function"  
},  
}
```

# PD-5.2.0.0 PayToContract

```
send_ganache.js x
1  const Web3 = require('web3');
2  const web3 = new Web3('http://localhost:8545');
3  const TestPayABI=[{
4      "constant": false,
5      "inputs": [],
6      "name": "PayToContract",
7      "outputs": [],
8      "payable": true,
9      "stateMutability": "payable",
10     "type": "function"
11  ]},
12  async function f() {
13     const TestPayaddress="0xFa01Aa6D84Db3EAD322573478132F70718341F59"
14     var acts=await web3.eth.getAccounts(),
15     var x=Web3.utils.fromWei(await web3.eth.getBalance(TestPayaddress), 'ether');
16     console.log(`TestPayaddress ${TestPayaddress} has ${x} ether`);
17     var toTransfer=web3.utils.toWei("1")
18     const ContractTestPay = new web3.eth.Contract (TestPayABI, TestPayaddress);
19     var result = await ContractTestPay.methods.PayToContract().send({from: acts[0], value: toTransfer});
20     console.log(`Stored in block ${result.blockNumber}`);
21     var x=Web3.utils.fromWei(await web3.eth.getBalance(TestPayaddress), 'ether');
22     console.log(`TestPayaddress ${TestPayaddress} has ${x} ether`);
23  }
24  f();
25
```

```
Console
NPP_EXEC: "Run"
CD
Current directory: Z:\blockchain\web3examples\ethereum\web3js
node send_ganache.js
Process started (PID=6968) >>>
TestPayaddress 0xFa01Aa6D84Db3EAD322573478132F70718341F59 has 8.00000007818749352 ether
Stored in block 36
TestPayaddress 0xFa01Aa6D84Db3EAD322573478132F70718341F59 has 9.00000007818749352 ether
<<< Process finished (PID=6968). (Exit code 0)
===== READY =====
```

[https://github.com/web3examples/ethereum/blob/master/web3js/send\\_ganache.js](https://github.com/web3examples/ethereum/blob/master/web3js/send_ganache.js)



# PD-5.2.0.0 Subscribe via WebSockets

```
subscribe.js x
1  const Web3 = require('web3');
2  const fs = require('fs');
3  const infuraKey = fs.readFileSync(".infura").toString().trim(); // infura key
4  const web3 = new Web3(`wss://ropsten.infura.io/ws/v3/${infuraKey}`);
5
6  async function f() {
7    var currentBlockNumber = await web3.eth.getBlockNumber();
8    console.log(currentBlockNumber);
9
10   var subscription = web3.eth.subscribe('logs', {fromBlock: currentBlockNumber})
11     .on("data", console.log)
12     .on("changed", console.log)
13     .on("error", console.log)
14   }
15   f();
```

```
>node subscribe.js
```

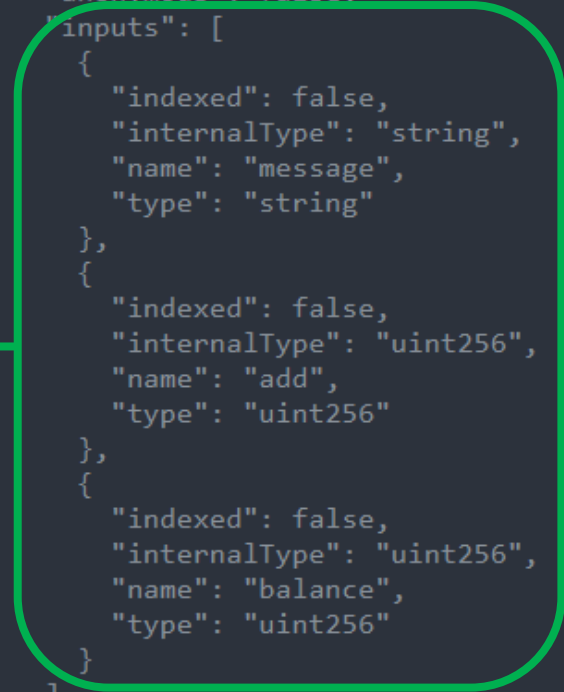
```
9010794
```

```
{
  removed: false,
  logIndex: 0,
  transactionIndex: 4,
  transactionHash: '0x56d5853bf9b930f2e52b1cd3f9312866558eb220a0f8e6ad50d0e7bcff91c1f2',
  blockHash: '0x5dceb80933951970b892674c66a03222bb2b0ef2df66c41d88cbf8a59dcb4e56',
  blockNumber: 9010795,
  ....
```

# PD-5.2.0.0 decodeLog

```
Remix_Functions.sol x
1  pragma solidity ^0.5.12;
2
3  contract Functions {
4
5      ... event Log(string message, uint add, uint balance);
6
7      ... constructor() public payable {
8          ... emit Log("In constructor", msg.value, address(this).balance);
9      }
10
11     ... function abc() public payable {
12         ... emit Log("In function abc", msg.value, address(this).balance);
13     }
14
15     ... function() external payable {
16         ... emit Log("In fallback function", msg.value, address(this).balance);
17     }
18 }
```

```
contract.sol  output.json
[
{
  "name": "Functions",
  "abi": [
    {
      "inputs": [],
      "payable": true,
      "stateMutability": "payable",
      "type": "constructor"
    },
    {
      "anonymous": false,
      "inputs": [
        {
          "indexed": false,
          "internalType": "string",
          "name": "message",
          "type": "string"
        },
        {
          "indexed": false,
          "internalType": "uint256",
          "name": "add",
          "type": "uint256"
        },
        {
          "indexed": false,
          "internalType": "uint256",
          "name": "balance",
          "type": "uint256"
        }
      ],
      "name": "Log",
      "type": "event"
    }
  ]
}
```



```
ABI Log Types=[
... {"indexed": false, "internalType": "string", "name": "message", "type": "string"},
... {"indexed": false, "internalType": "uint256", "name": "add", "type": "uint256"},
... {"indexed": false, "internalType": "uint256", "name": "balance", "type": "uint256"}
]
```

# PD-5.2.0.0 decodeLog

```
logtest.js x
1 //https://github.com/web3examples/ethereum/blob/master/solidity_examples/Remix_Functions.sol
2 const Web3 = require('web3');
3 const web3 = new Web3("ws://localhost:8545");
4 const contract = "0xe48E3405D6A9a172a04507D7fEde3529e89F8494"
5 ABILogTypes = [
6   { "indexed": false, "internalType": "string", "name": "message", "type": "string" },
7   { "indexed": false, "internalType": "uint256", "name": "add", "type": "uint256" },
8   { "indexed": false, "internalType": "uint256", "name": "balance", "type": "uint256" }
9 ]
10
11 async function processevent(object) {
12   var data = object.data
13   data = data.replace(/000+/g, '..00..') // shorten the number of 0's
14   console.log(`Raw data: ${data}`)
15   var decoded = web3.eth.abi.decodeLog(ABILogTypes, object.data);
16   console.log(`Decoded: ${decoded.message} add:${decoded.add} balance:${decoded.balance}`)
17 }
18 var subscription = web3.eth.subscribe('logs', {fromBlock: '0x0', address: contract})
19   .on("data", processevent)
20   .on("changed", console.log)
21   .on("error", console.log);
```

## >node logtest.js

Raw data: 0x..00..6..00..e496e20636f6e7374727563746f72..00..

Decoded: In constructor add:0 balance:0

Raw data: 0x..00..6..00..f496e2066756e6374696f6e20616263..00..

Decoded: In function abc add:0 balance:0

Raw data: 0x..00..6..00..1..00..1..00..f496e2066756e6374696f6e20616263..00..

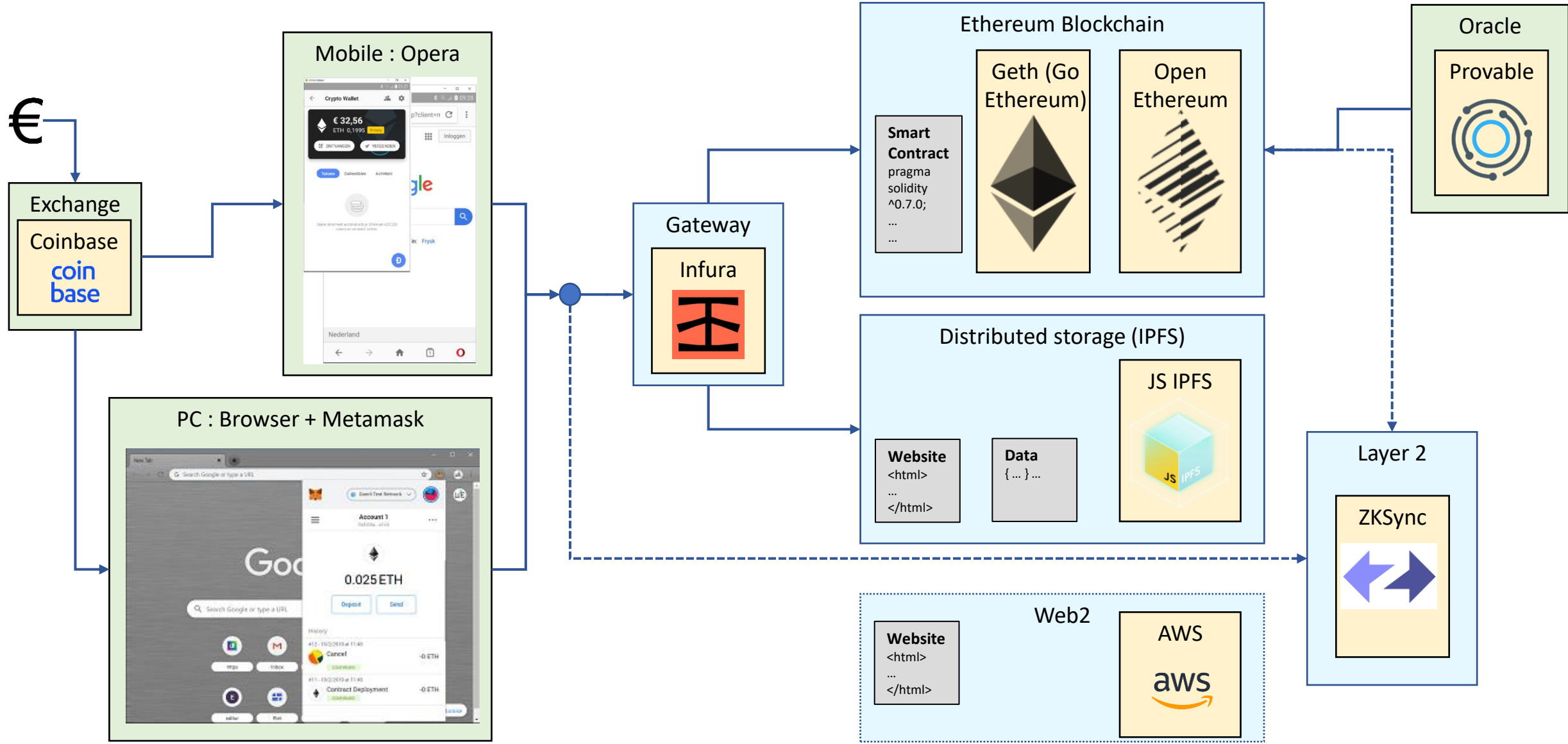
Decoded: In function abc add:1 balance:1

[https://github.com/web3examples/ethereum/blob/master/solidity\\_examples/Remix\\_Functions.sol](https://github.com/web3examples/ethereum/blob/master/solidity_examples/Remix_Functions.sol)

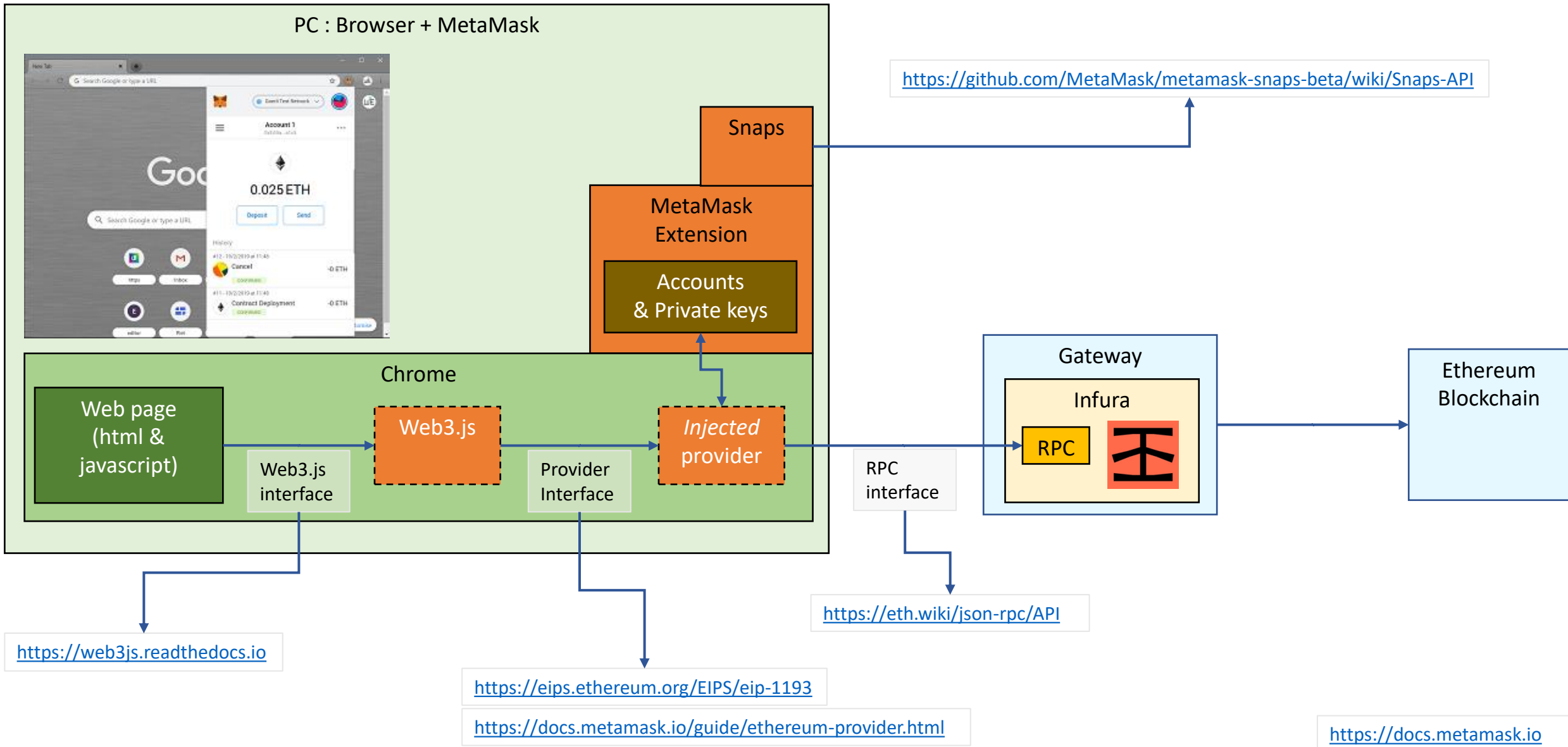
<https://github.com/web3examples/ethereum/blob/master/web3js/logtest.js>

# PD-5.3 Web3.js in browser

# PD-5.3.1 Architecture

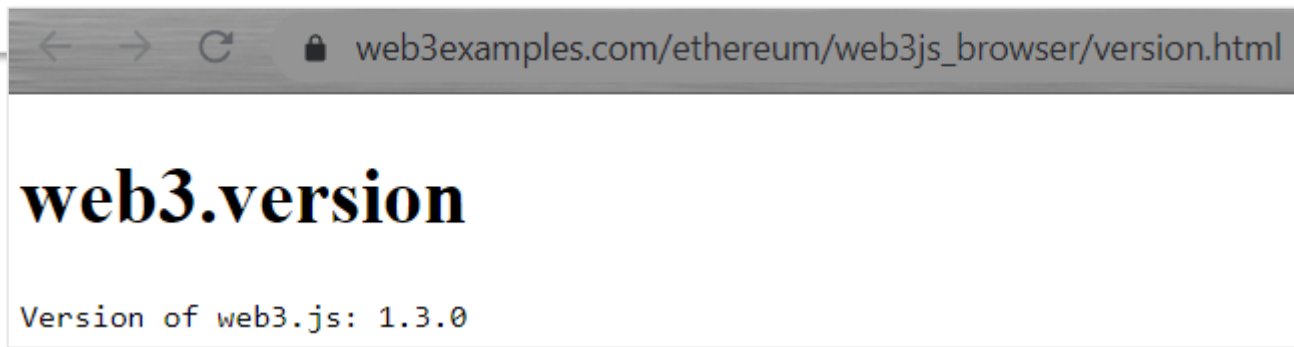


# PD-5.3.1 Architecture MetaMask



# PD-5.3.2 Get Version of web3.js

```
version.html x
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta name="viewport" content="width=device-width, initial-scale=1.0">
5     <script src="https://unpkg.com/web3@latest/dist/web3.min.js"></script>
6   </head>
7   <body>
8     <h1>web3.version</h1>
9     <pre id="log" style="width:100%;height:200px"></pre>
10    <script type="text/javascript">
11      function log(logstr) {
12        document.getElementById("log").innerHTML += logstr + "\n";
13      }
14      async function f () {
15        web3 = new Web3 ();
16        log(`Version of web3.js: ${web3.version}`);
17      }
18      window.addEventListener('DOMContentLoaded', f);
19    </script>
20  </body>
21 </html>
```

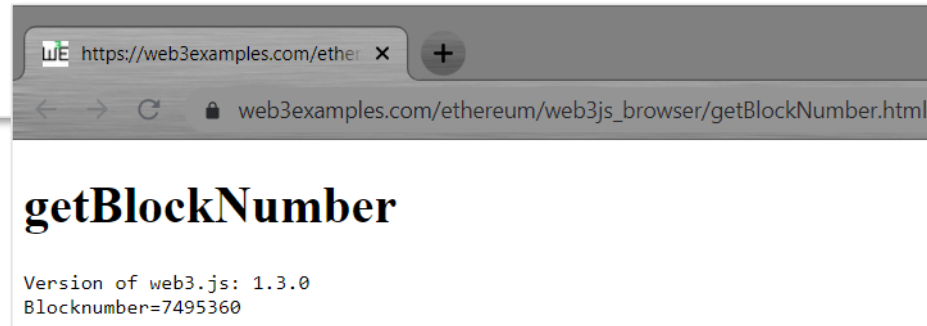


[https://web3examples.com/ethereum/web3js\\_browser/version.html](https://web3examples.com/ethereum/web3js_browser/version.html)

[https://github.com/web3examples/ethereum/blob/master/web3js\\_browser/version.html](https://github.com/web3examples/ethereum/blob/master/web3js_browser/version.html)

# PD-5.3.3 getBlockNumber

```
getBlockNumber.html x
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta name="viewport" content="width=device-width, initial-scale=1.0">
5     <script src="https://unpkg.com/web3@latest/dist/web3.min.js"></script>
6   </head>
7   <body>
8     <h1>getBlockNumber</h1>
9     <pre id="log" style="width:100%;height:200px"></pre>
10    <script type="text/javascript">
11      function log(logstr) {
12        document.getElementById("log").innerHTML += logstr + "\n";
13      }
14      async function f () {
15        web3 = new Web3(Web3.givenProvider); // provider from metamask
16        log(`Version of web3.js: ${web3.version}`);
17        var BlockNumber = await web3.eth.getBlockNumber().catch(log);
18        log(`Blocknumber=${BlockNumber}`);
19      }
20      window.addEventListener('DOMContentLoaded', f);
21    </script>
22  </body>
23 </html>
```

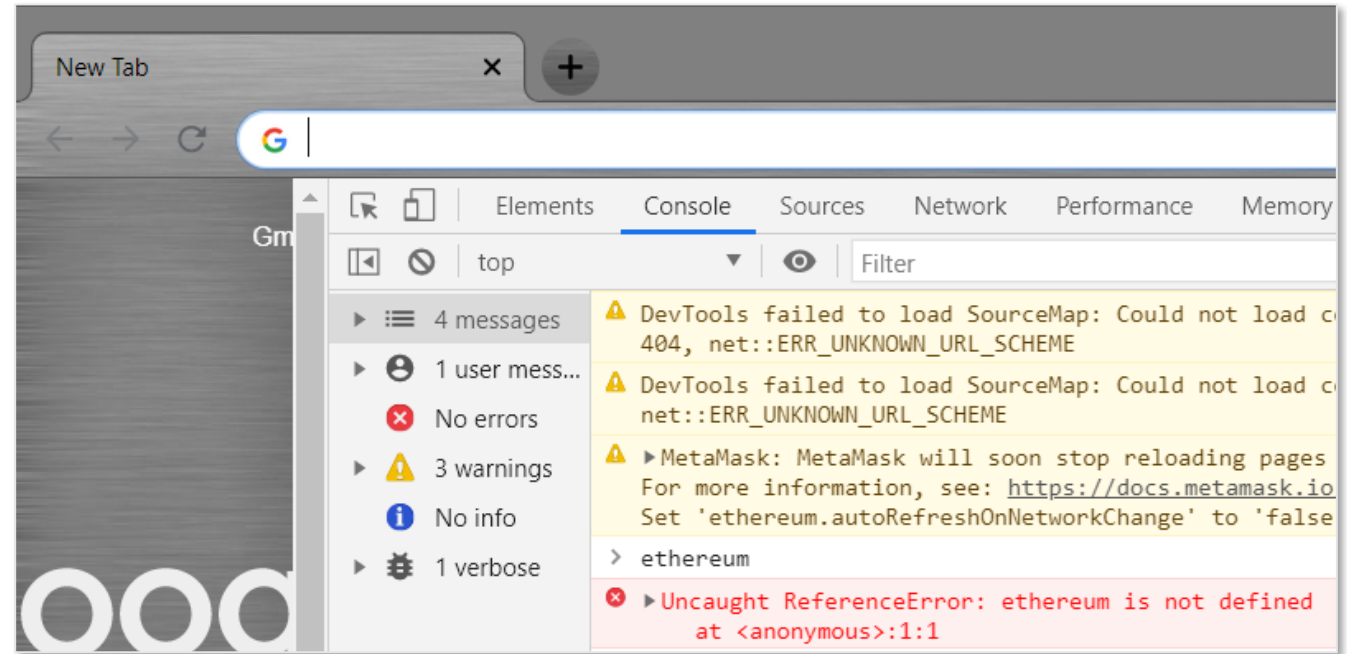
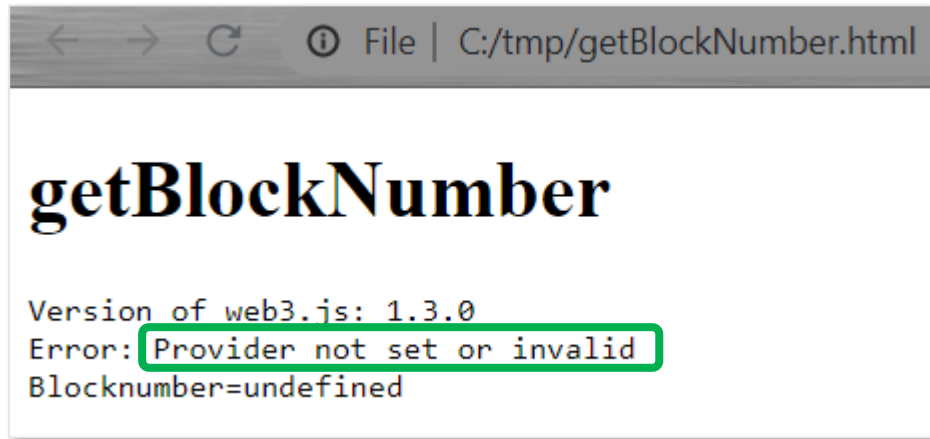


[https://web3examples.com/ethereum/web3js\\_browser/getBlockNumber.html](https://web3examples.com/ethereum/web3js_browser/getBlockNumber.html)

[https://github.com/web3examples/ethereum/blob/master/web3js\\_browser/getBlockNumber.html](https://github.com/web3examples/ethereum/blob/master/web3js_browser/getBlockNumber.html)



# PD-5.3.4 Use local webserver



```
> Npm install -g budo
```

```
C:\tmp>budo
[0003] info  Server running at http://192.168.0.42:9966/ (connect)
[0020] 6ms   115B GET 200 / (generated)
```

<http://localhost:9966>

[https://web3examples.com/ethereum/web3js\\_browser/getBlockNumber.html](https://web3examples.com/ethereum/web3js_browser/getBlockNumber.html)

[https://github.com/web3examples/ethereum/blob/master/web3js\\_browser/getBlockNumber.html](https://github.com/web3examples/ethereum/blob/master/web3js_browser/getBlockNumber.html)

# PD-5.3.5 requestAccounts



```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta name="viewport" content="width=device-width, initial-scale=1.0">
5     <script src="https://unpkg.com/web3@latest/dist/web3.min.js"></script>
6   </head>
7   <body>
8     <h1>requestAccounts</h1>
9     <pre id="log" style="width:100%;height:200px"></pre>
10    <script type="text/javascript">
11      function log(logstr) {
12        document.getElementById("log").innerHTML += logstr + "\n";
13      }
14      async function f() {
15        web3 = new Web3(Web3.givenProvider); // provider from metamask
16        ...
17        var result = await web3.eth.requestAccounts().catch(x => {log(x.message); console.log(x)});
18        log(`Value from requestAccounts: ${JSON.stringify(result)}`);
19        var acts = await web3.eth.getAccounts().catch(log);
20        log(`Here are the accounts: ${JSON.stringify(acts)}`);
21      }
22      window.addEventListener('DOMContentLoaded', f);
23    </script>
24  </body>
25 </html>
```

requestAccounts

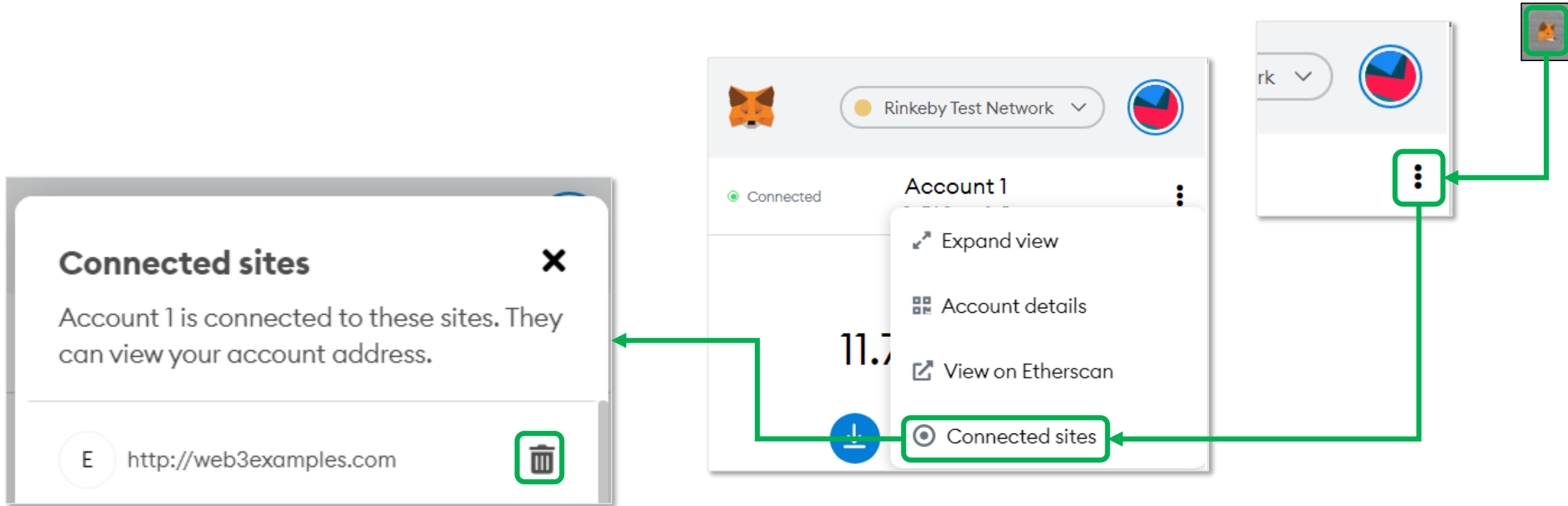
Value from requestAccounts: ["0xD0E65Aec202BA4561EEC3328fCa0C48e4B1D02aC"]  
Here are the accounts: ["0xD0E65Aec202BA4561EEC3328fCa0C48e4B1D02aC"]

<https://web3js.readthedocs.io/en/v1.3.0/web3-eth.html?highlight=requestaccounts#requestaccounts>

[https://web3examples.com/ethereum/web3js\\_browser/RequestAccounts.html](https://web3examples.com/ethereum/web3js_browser/RequestAccounts.html)

[https://github.com/web3examples/ethereum/blob/master/web3js\\_browser/RequestAccounts.html](https://github.com/web3examples/ethereum/blob/master/web3js_browser/RequestAccounts.html)

# PD-5.3.5 MetaMask remove site connection



PD-5.4 Web3.js in browser  
(more advanced examples)

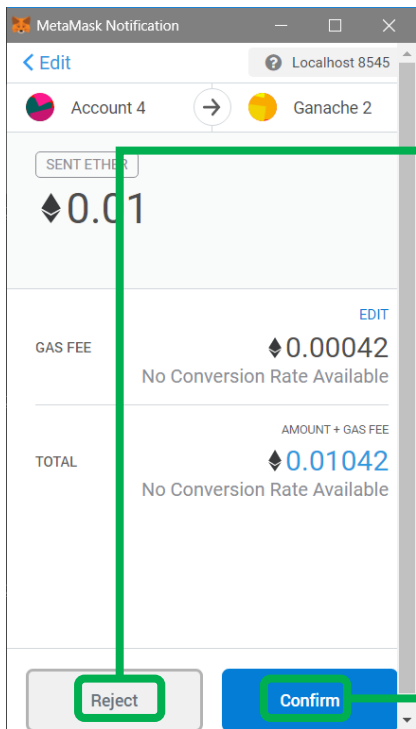
# PD-5.4.1 sendTransaction

```
var fromadr=acts[0];
toadr="0x356D3d2e3dd533E1293087e73c2C36D91337d0c6";
log(`fromadr ${fromadr} has ${Web3.utils.fromWei(
... await web3.eth.getBalance(fromadr), 'ether') } ether`);
log(`toadr ... ${toadr} has ${Web3.utils.fromWei(
... await web3.eth.getBalance(toadr), 'ether') } ether`);
log(`Transferring 0.01 ether`);
obj= await web3.eth.sendTransaction({
... from: fromadr,
... to: toadr,
... value: Web3.utils.toWei('0.01', 'ether')
}).catch(x=>log(x.message));
```

[https://web3examples.com/ethereum/web3js\\_browser/sendtransaction.html](https://web3examples.com/ethereum/web3js_browser/sendtransaction.html)

[https://github.com/web3examples/ethereum/blob/master/web3js\\_browser/sendtransaction.html](https://github.com/web3examples/ethereum/blob/master/web3js_browser/sendtransaction.html)

# PD-5.4.1 sendTransaction - output



```
web3examples.com/ethereum/web3js_browser/sendtransaction.html

sendTransaction

Version of web3.js: 1.3.0
Value from eth_requestAccounts: ["0xEA9a7c7cD8d4Dc3acc6f0AaEc1506C8D6041a1c5"]
Here are the accounts: ["0xEA9a7c7cD8d4Dc3acc6f0AaEc1506C8D6041a1c5"]
fromadr 0xEA9a7c7cD8d4Dc3acc6f0AaEc1506C8D6041a1c5 has 11.740936517493942412 ether
toadr 0x356D3d2e3dd533E1293087e73c2C36D91337d0c6 has 0 ether
Transferring 0.01 ether
MetaMask Tx Signature: User denied transaction signature.
```

```
web3examples.com/ethereum/web3js_browser/sendtransaction.html

sendTransaction

Version of web3.js: 1.3.0
Value from eth_requestAccounts: ["0xEA9a7c7cD8d4Dc3acc6f0AaEc1506C8D6041a1c5"]
Here are the accounts: ["0xEA9a7c7cD8d4Dc3acc6f0AaEc1506C8D6041a1c5"]
fromadr 0xEA9a7c7cD8d4Dc3acc6f0AaEc1506C8D6041a1c5 has 11.740936517493942412 ether
toadr 0x356D3d2e3dd533E1293087e73c2C36D91337d0c6 has 0 ether
Transferring 0.01 ether
Stored in block 7495510
fromadr 0xEA9a7c7cD8d4Dc3acc6f0AaEc1506C8D6041a1c5 has 11.730915517493942412 ether
toadr 0x356D3d2e3dd533E1293087e73c2C36D91337d0c6 has 0.01 ether
```

[https://web3examples.com/ethereum/web3js\\_browser/sendtransaction.html](https://web3examples.com/ethereum/web3js_browser/sendtransaction.html)

# PD-5.4.2 ChainOrAccountsChanged

```
ChainOrAccountsChanged.html x
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta name="viewport" content="width=device-width, initial-scale=1.0">
5     <script src="https://unpkg.com/web3@latest/dist/web3.min.js"></script>
6   </head>
7   <body>
8     <h1>Detect change of account, chain and network</h1>
9     <pre id="log" style="width:100%;height:200px"></pre>
10    <script type="text/javascript">
11      function log(logstr) {
12        document.getElementById("log").innerHTML += logstr + "\n";
13      }
14      async function f() {
15        web3 = new Web3(Web3.givenProvider); // provider from metamask
16        await newChain()
17        await newActs()
18        ethereum.on('accountsChanged', newActs)
19        ethereum.on('chainChanged', newChain);
20        ethereum.on('networkChanged', newChain); // deprecated // still used in metamask mobile
21        ethereum.on('chainIdChanged', newChain); // deprecated // temp workaround
22        ethereum.autoRefreshOnNetworkChange = false; // temp workaround // doesn't work on metamask mobile
23      }
24      var result = await web3.eth.requestAccounts().catch(x => {log(x.message); console.log(x)});
25
26      async function newChain(newchainId) {
27        var chainId = await web3.eth.getChainId();
28        var networkId = await web3.eth.net.getId();
29        log(`We are on chain: ${chainId} and network: ${networkId}`);
30      }
31      async function newActs() {
32        var acts = await web3.eth.getAccounts();
33        log(`We have accounts: ${JSON.stringify(acts)}`);
34      }
35    }
36    window.addEventListener('DOMContentLoaded', f);
37  </script>
38 </body>
39 </html>
```

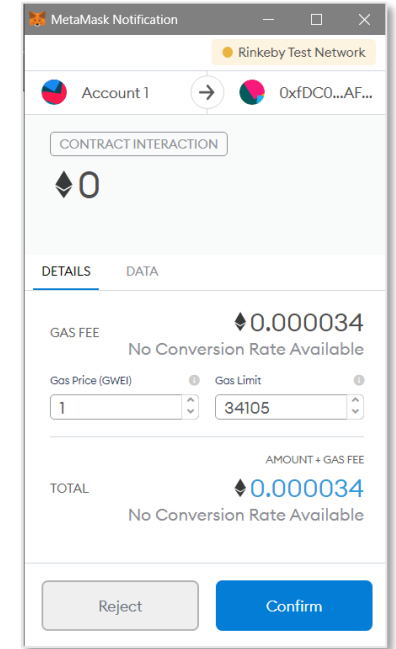
[https://web3examples.com/ethereum/web3js\\_browser/ChainOrAccountsChanged.html](https://web3examples.com/ethereum/web3js_browser/ChainOrAccountsChanged.html)

[https://github.com/web3examples/ethereum/blob/master/web3js\\_browser/ChainOrAccountsChanged.html](https://github.com/web3examples/ethereum/blob/master/web3js_browser/ChainOrAccountsChanged.html)

<https://docs.metamask.io/guide/ethereum-provider.html#events>

# PD-5.4.3 Casino – snippet (Rinkeby)

```
casino_snippet.html
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta name="viewport" content="width=device-width, initial-scale=1.0">
5 <script src="https://unpkg.com/web3@latest/dist/web3.min.js"></script>
6 </head>
7 <body>
8 <h1>Casino (select Rinkeby)</h1>
9 <pre id="log" style="width:100%;height:200px"></pre>
10 <script type="text/javascript">
11 function log(logstr) {
12     document.getElementById("log").innerHTML += logstr + "\n";
13 }
14 async function f() {
15     web3 = new Web3(Web3.givenProvider); // provider from metamask
16     var acts = await web3.eth.requestAccounts().catch(x => log(x.message));
17     const contractCasino = "0xfDC02989Ba838da44aCb16D6d9fB393DfBe3AFC5";
18     const CasinoABI = [
19         { "constant": false,
20           "inputs": [],
21           "name": "betAndWin",
22           "outputs": [],
23           "payable": true,
24           "stateMutability": "payable",
25           "type": "function"
26         }];
27     const CasinoContract = new web3.eth.Contract(CasinoABI, contractCasino);
28     var result = await CasinoContract.methods.betAndWin().send({from: acts[0], value: 1});
29     var win = web3.utils.hexToNumber(result.events[0].raw.data);
30     log(`Win result=${win}`);
31 }
32 window.addEventListener('DOMContentLoaded', f);
33 </script>
34 </body>
35 </html>
```



[https://web3examples.com/ethereum/web3js\\_browser/casino\\_snippet.html](https://web3examples.com/ethereum/web3js_browser/casino_snippet.html)

[https://github.com/web3examples/ethereum/blob/master/web3js\\_browser/casino\\_snippet.html](https://github.com/web3examples/ethereum/blob/master/web3js_browser/casino_snippet.html)



# PD-5.4.3 Casino Solidity

```
Casino.sol x
1 // Load in remix: remix.loadurl("https://github.com/web3examples/ethereum/solidity_examples/Casino.sol")
2 pragma solidity >=0.5.0 <0.7.0;
3
4 /** @author Gerard Persoon
5     @title A simple casino, for educational purposes only
6 */
7 contract Casino {
8
9     event Won(bool win); // declaring event
10
11     // Setup an initial amount for the bank, supplied during the creation of the contract.
12     constructor() public payable {
13     }
14
15     // Perform the bet and pay out if you win
16     // @dev several temporary variables are created to make debugging easier
17     */
18     function betAndWin() public payable { // returning value isn't easy to retrieve
19         address payable betPlacer = address(msg.sender);
20         uint bet = msg.value;
21         uint payout = bet * 2;
22         uint balance = getBankBalance();
23         require(bet > 0, "No money added to bet.");
24         require(payout <= balance, "Not enough money in bank for this bet."); // bet has already been added to bank balance
25         bool win = bool(getRandom() % 2 == 0);
26         if (win)
27             betPlacer.transfer(payout);
28         emit Won(win); // logging event
29     }
30
31     // Check the balance of the bank
32     // @return the balance
33     */
34     function getBankBalance() public view returns (uint256 ret) {
35         return address(this).balance;
36     }
37
38     // Draw a random number
39     // @dev this is not secure but only to demonstrate
40     // @return a pseudo random number
41     */
42     function getRandom() public view returns (uint256) {
43         return uint256(keccak256(abi.encodePacked(block.difficulty, block.coinbase, block.timestamp)));
44     }
45
46     // Deposit more funds for bank
47     // @dev used when the bank runs out of money
48     */
49     function () external payable {
50     }
51 }
```

[https://github.com/web3examples/ethereum/blob/master/solidity\\_examples/Casino.sol](https://github.com/web3examples/ethereum/blob/master/solidity_examples/Casino.sol)

# PD-5.4.3 Casino – full version

← → ↻ 🔒 web3examples.com/ethereum/casino/

## Bet and Win

Balance:  ETH

Bank Balance:  ETH

Bet:  ETH

[Play Now](#)

web3 is present: 1.3.0  
Ethereum network: Goerli  
Ethereum address: [0xEA9a7c7cD8d4Dc3acc6f0AaEc1506C8D6041a1c5](#)  
Contract address: [0x548961aBD98E4C45AD87Fc510A663DEc20007580](#)

<https://web3examples.com/ethereum/casino>

<https://github.com/web3examples/ethereum/tree/master/casino>

# PD-5.4.4 Location (Rinkeby)

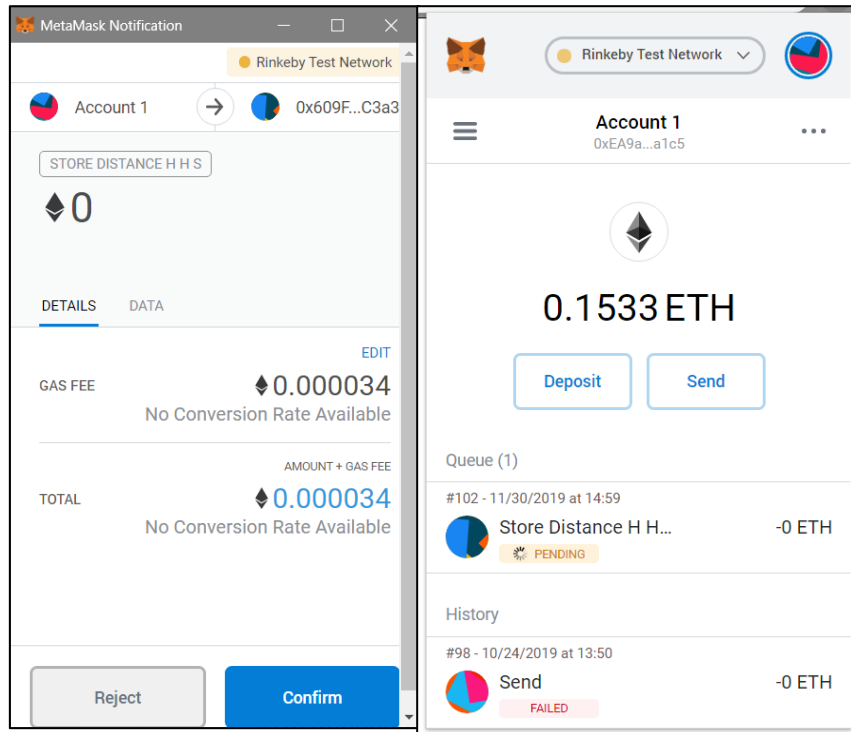
```
DistanceHHS.sol x
1  pragma solidity ^0.5.0;
2
3  contract DistanceHHS {
4      ... event DistanceHHSLog(uint256 distance) ...; ... // declaring event ...
5      ... function StoreDistanceHHS(uint256 distance) public {
6          ... emit DistanceHHSLog(distance); // logging event
7      ... }
8  }
```

<https://rinkeby.etherscan.io/address/0x609F224c0c9405a1e7FD404114ca8A8606edC3a3>

<https://github.com/web3examples/location/blob/master/truffle/contracts/DistanceHHS.sol>

# PD-5.4.4 Location – snippet (Rinkeby)

```
var dist=100;
DistanceHHSaddress="0x609F224c0c9405a1e7FD404114ca8A8606edC3a3"
DistanceHHSABI=[{"constant": false, "inputs": [{"name": "distance",
  "type": "uint256"}], "name": "StoreDistanceHHS", "outputs": [],
  "payable": false, "stateMutability": "nonpayable", "type": "function"}];
const DistanceHHSContract=new web3.eth.Contract (DistanceHHSABI,DistanceHHSaddress);
log(`Storing distance to HHS: distance=${dist} (hex:${web3.utils.toHex(dist)})`);
x=await DistanceHHSContract.methods.StoreDistanceHHS (dist)
  .send({from: acts[0]}) .catch((reason)=>log(`Cannot do transaction ${reason}`));
log(`Transaction hash: ${x.transactionHash}`);
```



[https://web3examples.com/ethereum/web3js\\_browser/location\\_snippet.html](https://web3examples.com/ethereum/web3js_browser/location_snippet.html)

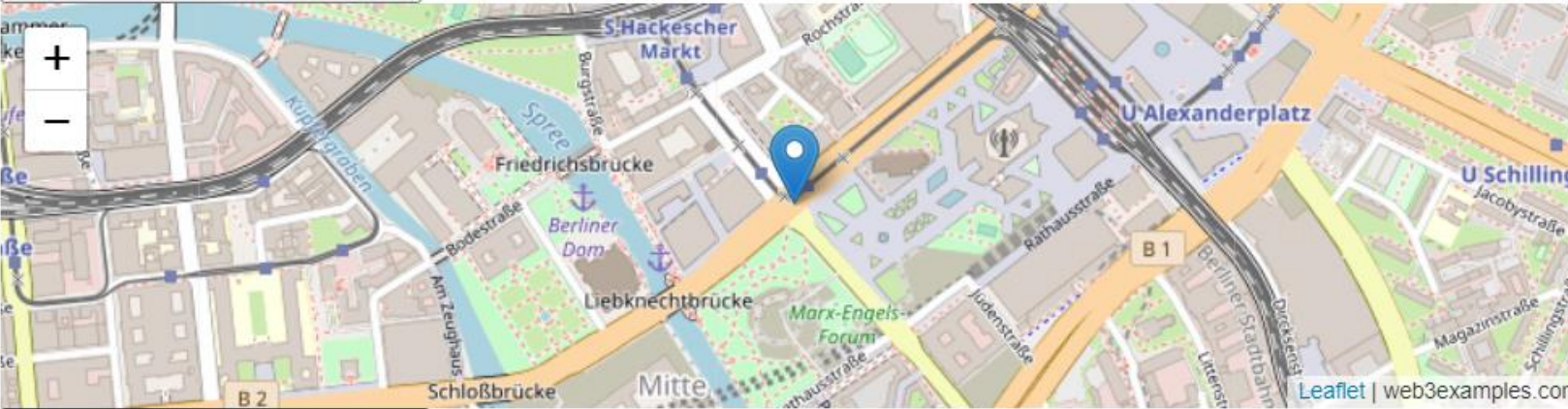
[https://github.com/web3examples/ethereum/blob/master/web3js\\_browser/location\\_snippet.html](https://github.com/web3examples/ethereum/blob/master/web3js_browser/location_snippet.html)

# PD-5.4.4 Location

← → ↻ 🔒 web3examples.com/location/

## Distance to HHS on ethereum blockchain

Press here to determine location



Press here to save on blockchain

web3 is present: 1.3.0  
Ethereum network: Rinkeby  
Ethereum address: [0xEA9a7c7cD8d4Dc3acc6f0AaEc1506C8D6041a1c5](#)  
[See check-ins of address 0xEA9a7c7cD8d4Dc3acc6f0AaEc1506C8D6041a1c5](#)

[Get test ETH](#)

Trying to get location  
Permission granted  
Coordinates: lat=52.520007, lon=13.404954 Distance to HHS=619155  
Storing distance to HHS: distance=619155 (hex:0x97293)  
Transaction hash: [0xce5274fc2bd2fceef0e24e202b5da10917033a9d27c53566db64ad3ed2d20b8b](#)

<https://web3examples.com/location/>

<https://github.com/web3examples/location>

# PD-5.4.4 List Location

```
async function f () { ...  
  ... address="0x609F224c0c9405a1e7FD404114ca8A8606edC3a3";  
  ... var subscription= web3.eth.subscribe ('logs', {fromBlock: '0x0', address: address})  
  ... .on ("data", processevent)  
  ... .on ("changed", console.log)  
  ... .on ("error", console.log)  
}
```

← → ↻ ▲ Not secure | web3examples.com/location/listlocation.html ☆ 🐱 ⚙️ WE

## List of checkins to locations relative to HHS

From (address), From (name), Date-time, Distance (meters)

0x111Eb30919a11DA63423cECC8424969A5cF18099,	Fri	Nov	15	2019	12:34:48	GMT+0100	(Central European Standard Time),	22409
0xfeC65AbE091919645AD2dAFaF28a0607F817056C,	Fri	Nov	06	2020	10:45:49	GMT+0100	(Central European Standard Time),	100
0x111Eb30919a11DA63423cECC8424969A5cF18099,	Fri	Nov	15	2019	12:26:33	GMT+0100	(Central European Standard Time),	22408
0xEA9a7c7cD8d4Dc3acc6f0AaEc1506C8D6041a1c5,	Thu	Nov	05	2020	18:21:10	GMT+0100	(Central European Standard Time),	619155
0xa272F0fD7736125e1a9DbdBFAa69172b488D6ccb,	Fri	Nov	15	2019	14:04:18	GMT+0100	(Central European Standard Time),	8
0x972F6C4485ed39Cc4a10C4C5b9FAFB4593E2C827,	Fri	Nov	15	2019	16:58:03	GMT+0100	(Central European Standard Time),	76
0xEA9a7c7cD8d4Dc3acc6f0AaEc1506C8D6041a1c5,	Sat	Nov	30	2019	14:59:39	GMT+0100	(Central European Standard Time),	100
0xEA9a7c7cD8d4Dc3acc6f0AaEc1506C8D6041a1c5,	Tue	Nov	24	2020	11:33:16	GMT+0100	(Central European Standard Time),	100
0xEA9a7c7cD8d4Dc3acc6f0AaEc1506C8D6041a1c5,	Tue	Nov	24	2020	11:07:54	GMT+0100	(Central European Standard Time),	38730
0xEA9a7c7cD8d4Dc3acc6f0AaEc1506C8D6041a1c5,	Tue	Nov	24	2020	11:34:32	GMT+0100	(Central European Standard Time),	100
0xa272F0fD7736125e1a9DbdBFAa69172b488D6ccb,	Fri	Nov	15	2019	14:02:48	GMT+0100	(Central European Standard Time),	8
0xEA9a7c7cD8d4Dc3acc6f0AaEc1506C8D6041a1c5,	Sat	Nov	30	2019	15:02:39	GMT+0100	(Central European Standard Time),	100
0xEA9a7c7cD8d4Dc3acc6f0AaEc1506C8D6041a1c5,	Tue	Nov	24	2020	11:57:53	GMT+0100	(Central European Standard Time),	38730
0xEA9a7c7cD8d4Dc3acc6f0AaEc1506C8D6041a1c5,	Thu	Nov	05	2020	17:58:10	GMT+0100	(Central European Standard Time),	100
0xEA9a7c7cD8d4Dc3acc6f0AaEc1506C8D6041a1c5,	Mon	Nov	23	2020	11:02:10	GMT+0100	(Central European Standard Time),	100
0x8e2A89fF2F45ed7f8C8506f846200D671e2f176f,	Thu	Oct	22	2020	16:46:38	GMT+0200	(Central European Summer Time),	100
0x8818DcDDCC45C4ccBf67Eb7343e2a45fD8A84546,	Fri	Nov	15	2019	14:00:48	GMT+0100	(Central European Standard Time),	30
0xEA9a7c7cD8d4Dc3acc6f0AaEc1506C8D6041a1c5,	Tue	Nov	24	2020	11:57:08	GMT+0100	(Central European Standard Time),	100
0xEA9a7c7cD8d4Dc3acc6f0AaEc1506C8D6041a1c5,	Tue	Nov	24	2020	11:47:51	GMT+0100	(Central European Standard Time),	100
0xEA9a7c7cD8d4Dc3acc6f0AaEc1506C8D6041a1c5,	Tue	Nov	24	2020	11:54:38	GMT+0100	(Central European Standard Time),	38730
0xEA9a7c7cD8d4Dc3acc6f0AaEc1506C8D6041a1c5,	Tue	Nov	24	2020	12:11:27	GMT+0100	(Central European Standard Time),	100
0x4Ad2eaE4137e11EB3834840f1DC38F5f0fa181c3,	Fri	Nov	15	2019	14:21:48	GMT+0100	(Central European Standard Time),	25

<https://github.com/web3examples/location/blob/master/listlocation.html>

<http://web3examples.com/location/listlocation.html>

# PD-5.4.5 Faucet - snippet

```
const privateKey = '0x0da19552d21de3da01e4a5ff72f6722b9a86c78ee6c6a46e5cdcf0fb5a936110';  
.....  
// note very insecure, but for test ETH this is usable  
var addressFaucet = web3.eth.accounts.privateKeyToAccount(privateKey).address;  
web3.eth.accounts.wallet.add(privateKey);  
var fb = await web3.eth.getBalance(addressFaucet);  
log(`Faucet has ${fb} on address ${addressFaucet}`);  
log(`Wait about 20 seconds for transaction, note: no metamask confirm`);  
result = await web3.eth.sendTransaction({from: addressFaucet, to: acts[0], gas: 200000, value: 1});  
.....  
.catch(x => log(`Error: ${x.code} ${x.message}`));  
log(`Transaction hash: ${result.transactionHash}`);  
fb = await web3.eth.getBalance(addressFaucet);  
log(`Faucet now has ${fb}`);
```

web3examples.com/ethereum/web3js\_browser/faucet\_snippet.html

## Faucet snippet (select Rinkeby)

Faucet has 2096472647999999993 on address 0x111Eb30919a11DA63423cECC8424969A5cF18099  
Wait about 20 seconds for transaction, note: no metamask confirm  
Transaction hash: 0x53f8adefc4b82d188f748e902ad1f3ada6ae76ccbcaea0666e409cffffa82581  
Faucet now has 2096451647999999992

[https://web3examples.com/ethereum/web3js\\_browser/faucet\\_snippet.html](https://web3examples.com/ethereum/web3js_browser/faucet_snippet.html)

[https://github.com/web3examples/ethereum/blob/master/web3js\\_browser/faucet\\_snippet.html](https://github.com/web3examples/ethereum/blob/master/web3js_browser/faucet_snippet.html)

# PD-5.4.5 Faucet

← → ↻ 🔒 web3examples.com/ethereum/faucet/

## Web3Examples Faucet

My own Balance:  ETH

Faucet Balance:  ETH

Get / Give:  ETH

**Give to faucet**

**Get from faucet**

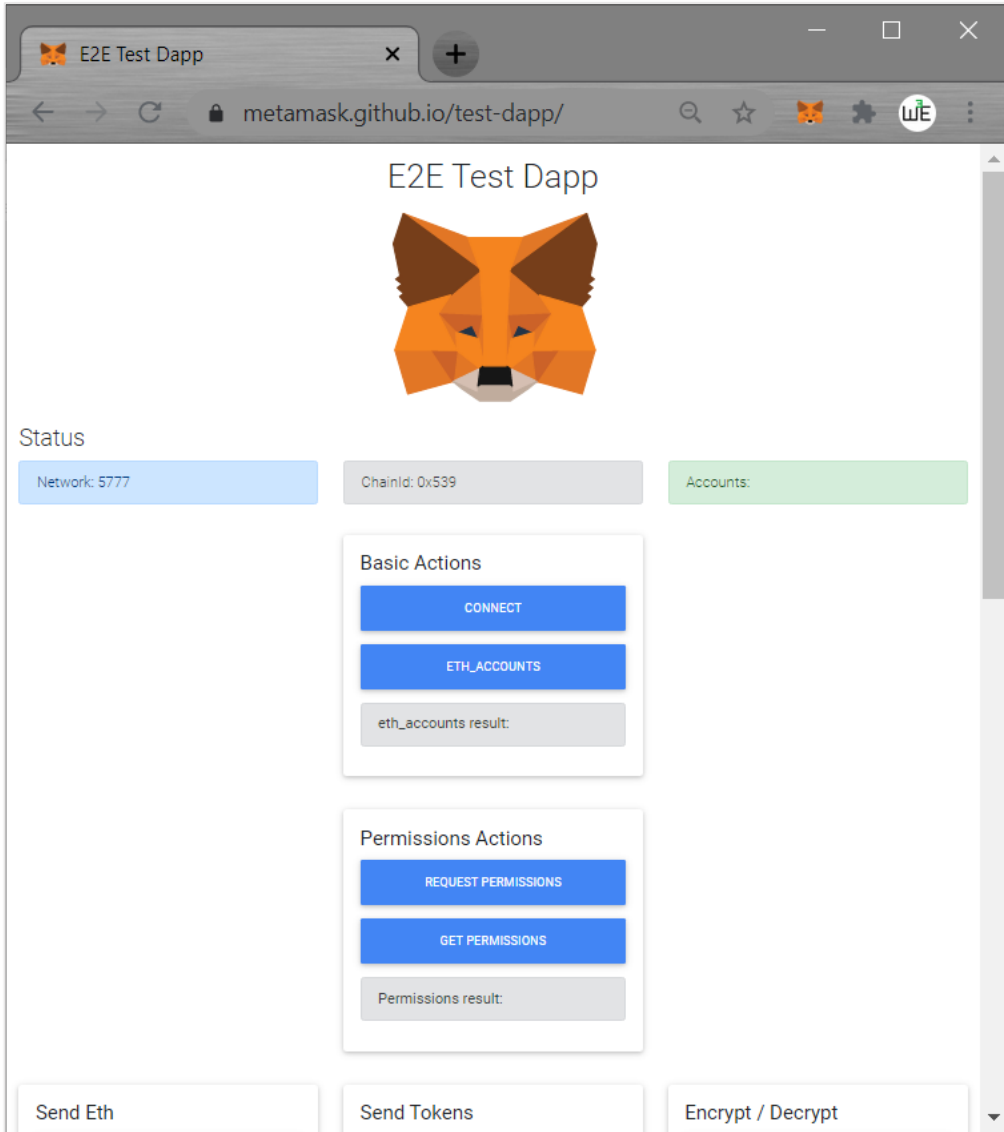
web3 is present: 1.3.0  
Found chainid 4  
Chain id: 4 name: Ethereum Testnet Rinkeby shortname: rin [more info](#)  
Ethereum network: rinkeby  
My own address: [0xEA9a7c7cD8d4Dc3acc6f0AaEc1506C8D6041a1c5](#)  
Faucet address: [0x111Eb30919a11DA63423cECC8424969A5cF18099](#)  
Transaction: [0xd5b88be891ac9bdcb05687947863bedc5d612271ec4069a45d01eefd71882021](#)

<https://web3examples.com/ethereum/faucet>

<https://github.com/web3examples/ethereum/tree/master/faucet>



# PD-5.4.6 E2E Test Dapp



<https://metamask.github.io/test-dapp>

<https://github.com/MetaMask/test-dapp/blob/master/src/index.js>

