

1.

2. EcoLab (NEO Ecosystem Laboratory)

```
<!-- ===== -->
<!-- ===== -->
<!-- ===== -->
<!-- BEGIN DIV MAIN CONTENTS - MENU AND CONTENT -->
  <div class="col-sm-2 sidenav">
    <ul class="nav nav-pills nav-stacked" id="pillstab">
      <li class="active"><a href="#!/ecolab/compilers" data-target="#compilers" c
      <li><a href="#!/ecolab/network" data-target="#network" data-toggle="tab" >E
      <li><a href="#!/ecolab/transaction" data-target="#transaction" data-toggle
      <li><a href="#!/ecolab/rawRPC" data-target="#rawRPC" data-toggle="tab">Raw
      <li><a href="#!/ecolab/activity" data-target="#activity" data-toggle="tab">
      <li><a href="#!/ecolab/utills" data-target="#utills" data-toggle="tab">Stora
      <li><a href="#!/ecolab/wallet" data-target="#wallet" data-toggle="tab" onc
      <li><a href="#!/ecolab/conversor" data-target="#conversor" data-toggle="tak
      <li><a href="#!/ecolab/incrementalstorage" data-target="#incrementalstorag
      <li><a href="#!/ecolab/cnnodesinfo" data-target="#cnnodesinfo" data-toggle
      <li><a href="#!/ecolab/statistics" data-target="#statistics" data-toggle="t
      <li><a href="#!/ecolab/configurations" data-target="#configurations" data-
    </ul>
  </div>

<!-- ===== -->
<!-- BEGIN DIV MAIN CONTENTS - EVERYTHING IS HERE - I -->
<div class="col-sm-10">
  <!-- ===== -->
  <!-- BEGIN FIXED DIV OF NETWORKS -->
  <div class="row">
    <div class="col-md-2">
      <!--<p><strong>Current Url</strong>: {{ vm.currentUrl }}</p>-->
      <span class="label label-success"><i class="fas fa-network-wired"></i>
      <i class="fas fa-chart-network"></i>
      <select id="ecolabnetworkurlselection" class="custom-select form-cont
        <option title="NeoCompiler Eco services">NeoCompiler Eco</option>
        <option title="Localhost services">Localhost</option>
        <option title="MainNet services">NEO MainNet (unsafe)</option>
        <option title="Testnet services">NEO TestNet</option>
        <!-- <option title="CozTestnet services" value="http://188.68.34.29
      </select>
    </div>
  </div>
</div>
```

```

<!-- END FIXED DIV OF NETWORKS -->
<!-- ===== -->

<!-- ===== -->
<!-- ===== -->
<!-- BEGIN DIV MAIN CONTENTS - EVERYTHING IS HERE - II -->
<div class="tab-content">
  <!-- ===== -->
  <!-- BEGIN DIV CONVERSIONS -->
  <div class="tab-pane fade" id="conversor">
    <div class="panel panel-primary">
      <div class="panel-heading">String <-> Hex</div>
      <div class="panel-body">
        <form id="str2hash">
          <div class="row">
            <div class="col-md-5">
              <div class="form-group">
                <label for="strhex_str">String</label>
                <input id="strhex_str" class="form-control" placeholder="String" type="text">
              </div>
            </div>
            <div class="col-md-5">
              <div class="form-group">
                <label for="strhex_hex">Hex</label>
                <input id="strhex_hex" class="form-control" placeholder="Hex" type="text">
              </div>
            </div>
            <div class="col-md-2">
              <div class="form-group">
                <label for="hex_length">Length (in bytes)</label>
                <input id="hex_length" class="form-control" placeholder="Length" type="text">
              </div>
            </div>
          </div>
        </form>
      </div>
    </div>
    <div class="panel panel-primary">
      <div class="panel-heading">Hex <-> xeH</div>
      <div class="panel-body">
        <form id="frmreverse">
          <div class="row">
            <div class="col-md-6">
              <div class="form-group">
                <label for="hexxeh_hex">Hex</label>
                <input id="hexxeh_hex" class="form-control" placeholder="Hex" type="text">
              </div>
            </div>
            <div class="col-md-6">
              <div class="form-group">
                <label for="hexxeh_xeh">xeH</label>
                <input id="hexxeh_xeh" class="form-control" placeholder="xeH" type="text">
              </div>
            </div>
          </div>
        </form>
      </div>
    </div>
  </div>

```

```

        </form>
    </div>
</div>
<div class="panel panel-primary">
    <div class="panel-heading">Number <-> Fixed8 Hex</div>
    <div class="panel-body">
        <form id="num2fixed8">
            <div class="row">
                <div class="col-md-6">
                    <div class="form-group">
                        <label for="fx8_number">Number</label>
                        <input type="number" id="fx8_number" class="form-cont
                    </div>
                </div>
                <div class="col-md-6">
                    <div class="form-group">
                        <label for="fx8_hex">Fixed8</label>
                        <input id="fx8_hex" class="form-control" placeholder=
                    </div>
                </div>
            </div>
        </form>
    </div>
</div>

<div class="panel panel-primary">
    <div class="panel-heading">Address <-> ScriptHash (big endian)</div>
    <div class="panel-body">
        <form id="addressscripthash">
            <div class="row">
                <div class="col-md-6">
                    <div class="form-group">
                        <label for="address">Address</label>
                        <input id="address" class="form-control" placeholder=
                    </div>
                </div>
                <div class="col-md-6">
                    <div class="form-group">
                        <label for="scripthash">ScriptHash (big endian, shoul
                        <input id="scripthash" class="form-control" placeholc
                    </div>
                </div>
            </div>
        </form>
    </div>
</div>

<div class="panel panel-primary">
    <div class="panel-heading">BigInteger <-> Little-Endian ByteArray
    <div class="panel-body">
        <form id="convert_bigint_to_bytearray">
            <div class="row">
                <div class="col-md-6">
                    <div class="form-group">
                        <label for="convert_bigint">BigInteger</label>
                        <input id="convert_bigint" class="form-control" place

```

```

        </div>
    </div>
    <div class="col-md-6">
        <div class="form-group">
            <label for="convert_bytearray">Little-Endian ByteArra
            <input id="convert_bytearray" class="form-control" pl
        </div>
    </div>
</div>
</form>
</div>
</div>

```

```

<div class="panel panel-primary">
    <div class="panel-heading">Hex String -> Hash</div>
    <div class="panel-body">
        <form id="convert_string_to_hash">
            <div class="row">
                <div class="col-md-3">
                    <div class="form-group">
                        <label for="convert_string_hash">Hex String</label>
                        <input id="convert_string_hash" class="form-control"
                    </div>
                </div>
                <div class="col-md-3">
                    <div class="form-group">
                        <label for="convert_string_hash160">Hash160 (scriptha
                        <input id="convert_string_hash160" class="form-contr
                    </div>
                </div>
                <div class="col-md-3">
                    <div class="form-group">
                        <label for="convert_string_sha256">Sha256</label>
                        <input id="convert_string_sha256" class="form-control
                    </div>
                </div>
                <div class="col-md-3">
                    <div class="form-group">
                        <label for="convert_string_hash256">Hash256</label>
                        <input id="convert_string_hash256" class="form-contr
                    </div>
                </div>
            </div>
        </form>
    </div>
</div>

```

```

<div class="panel panel-primary">
    <div class="panel-heading">Disassembly opcodes</div>
    <div class="panel-body">
        <form id="frm_disassembly">
            <div class="row">
                <div class="col-md-12">
                    <div class="form-group">
                        <label for="avmcode">AVM hex</label>

```

```

        <textarea id="avmcode" class="form-control" placeholder="AVM Code"
    </div>
</div>
<div class="col-md-6">
    <label for="sc_scripthash">ScriptHash (big endian, star
    <input type="text" id="sc_scripthash" class="form-contr
</div>
<div class="col-md-6">
    <label for="sc_address58">Address</label>
    <input type="text" id="sc_address58" class="form-contr
</div>
<div class="col-md-12">
    <div class="form-group">
        <label for="opcodes">OpCodes</label>
        <textarea id="opcodes" class="form-control" rows="5"
    </div>
</div>
</div>
</form>
</div>
</div>
<!-- END DIV CONVERSIONS -->
<!-- ===== -->

<!-- ===== -->
<!-- BEGIN UTILS TRANSACTION -->
<div class="tab-pane fade" id="utils">

    <div class="panel panel-primary">
        <div class="panel-heading">Read storage from a Smart Contract</div>
        <div class="panel-body">
            <form id="formgetstorage" method="post">
                <label for="gsf_contracthash">ScriptHash</label>
                <input id="gsf_contracthash" class="form-control" placeholder="ScriptHash" type="text">
                <label for="getStorageFormat">Input Key</label>
                <div class="row">
                    <div class="col-md-6">
                        <input id="gsf_contractkey" class="form-control" placeholder="Contract Key" type="text">
                    </div>
                    <div class="col-md-6">
                        <select name="getStorageFormat" id="getStorageFormat" class="form-control">
                            <option title="gsf_string" selected>String (no quotes)
                            <option title="gsf_hex">Hex</option>
                        </select>
                    </div>
                </div>
                <div class="row">
                    <div class="col-md-12">
                        <button type="submit" id="gsfsubmit" style="margin-bottom: 10px;">Submit
                    </div>
                </div>
                <label for="gsf_contractvalue">Output Value</label>
                <div class="row">
                    <div class="col-md-6">
                        <input id="gsf_contractvaluehex" class="form-control" type="text">
                    </div>
                </div>
            </form>
        </div>
    </div>

```

```

        </div>
        <div class="col-md-6">
            <input id="gsf_contractvaluestr" class="form-control"
        </div>
    </div>
</form>
</div>
</div>

<div class="panel panel-primary">
    <div class="panel-heading">NEP5 Balance</div>
    <div class="panel-body">
        <form id="getnep5balance" action="">
            <div class="row">
                <div class="col-md-6">
                    <div class="form-group">
                        <label for="getnep5_contract">Contract (scripthash)</label>
                        <input id="getnep5_contract" class="form-control" placeh
                    </div>
                </div>
                <div class="col-md-6">
                    <div class="form-group">
                        <label for="getnep5_address">Address (base58)</label>
                        <input id="getnep5_address" class="form-control" placeh
                    </div>
                </div>
            </div>
            <div class="row">
                <div class="col-md-12">
                    <button type="submit" style="margin-bottom: 10px;" id="t
                </div>
                <div class="col-md-6">
                    <div class="form-group">
                        <label for="output_getnep5">Output</label>
                        <input id="output_getnep5" class="form-control" placehc
                    </div>
                </div>
                <div class="col-md-6">
                    <div class="form-group">
                        <label for="output_getnep5_extra">Address (base58) + He
                        <input id="output_getnep5_extra" class="form-control" p
                    </div>
                </div>
            </div>
        </form>
    </div>
</div>

</div>
<!-- END UTILS TRANSACTION -->
<!-- ===== -->

<!-- ===== -->
<!-- BEGIN DIV TRANSACTION -->
<div class="tab-pane fade" id="transaction">

```

```

<div class="panel panel-primary">
  <div class="panel-heading">Create TX: Transfer UTXO Assets<br>
    <button type="button" onClick="goToWalletCentral()" id="btn_regis
  </div>
  <div class="panel-body">
    <form id="createtx" action="">
      <div class="row">
        <div class="col-md-4">
          <div class="form-group">
            <label for="createtx_from">From(Address)</label>
            <button type="button" onClick="createGasTxForm()" id="t
            <select name="createtx_from" id="createtx_from" class=
          </div>
        </div>
        <div class="col-md-4">
          <div class="form-group">
            <label for="createtx_to">To(Address)</label>
            <select name="createtx_to" id="createtx_to" class="cl
            <!-- <input name="createtx_to" id="createtx_to" place
          </div>
        </div>
      </div>
      <div class="row">
        <div class="col-md-16">
          <div class="col-md-4">
            <div class="form-group">
              <button type="button" onClick="fillAllNeo()" id="txcr
              <input type="number" id="createtx_NEO" value="0" cl
            </div>
          </div>
          <div class="col-md-4">
            <div class="form-group">
              <button type="button" onClick="fillAllGas()" id="txcr
              <input type="number" id="createtx_GAS" value="0" cl
            </div>
          </div>
        </div>
      </div>
      <div class="row">
        <div class="col-md-6">
          <div class="form-group">
            <button type="button" onClick="createSendTxForm()" ic
          </div>
        </div>
      </div>
    </form>
  </div>
</div>

<!-- BEGIN Create TX Withdraw -->
<div class="panel panel-primary">
  <div class="panel-heading">Create TX: Withdraw UTXO Assets from C
  <div class="panel-body">
    <form id="frm_createtx_withdraw" action="">
      <div class="row">

```

```

<div class="col-md-6">
  <div class="form-group">
    <label for="createtx_from_contract">From: Verificatio
    <textarea id="createtx_from_contract" class="form-cor
    <div class="row">
      <div class="col-md-6">
        <label for="createtx_from_scripthash">Scripthash
        <input id="createtx_from_scripthash" class="form-
      </div>
      <div class="col-md-6">
        <label for="createtx_from_address">Address (readc
        <input id="createtx_from_address" class="form-cor
      </div>
    </div>
    <div class="row">
      <div class="col-md-6">
        <button type="button" onClick="addContractToWalle
      </div>
    </div>
    <label for="createtx_from_verification">Invocation ir
    <textarea id="createtx_from_verification" class="form
    <label for="createtx_from_verification_opcodes">Invoc
    <textarea id="createtx_from_verification_opcodes" cla
    <label for="createtx_script_attributes">Script Attrib
    <textarea id="createtx_script_attributes" class="form
    <label for="createtx_attributes">Attributes: 20 Scrip
    <textarea id="createtx_attributes" class="form-contrc
  </div>
</div>
<div class="col-md-6">
  <div class="form-group">
    <select id="createtx_input_assetid" onChange="verifyF
      <option>Select an asset</option>
      <option value="c56f33fc6ecfcd0c225c4ab356fee59390af
      <option value="602c79718b16e442de58778e148d0b1084e3
    </select><br/>
    <label for="createtx_input_utxo2">Input UTXO</label><
    <select id="createtx_input_utxo2" name="createtx_inpu
  </div>
</div>
<div class="col-md-6">
  <div class="form-group">
    <label for="createtx_output_addresses">Output/To Addr
    <textarea id="createtx_output_addresses" class="form-
    <label for="createtx_output_addresses">Output/To Valu
    <textarea id="createtx_output_values" class="form-cor
  </div>
</div>
<div class="col-md-6">
  <div class="form-group">
    <input type="checkbox" id="createtx_cbx_change" check
    <label id="createtx_output_total">Total: 0 NEO (expec
    <label for="createtx_signature">Generate Signature (C
    <input id="createtx_signature" class="form-control" p
    <!-- document.getElementById("createtx_signature").ch
    <button type="button" id="createtx_create_sign" class

```



```

        <!-- Neon.wallet.generateSignature -->
        <!-- https://github.com/CityOfZion/neon-js/blob/cf5f9
    </div>
</div>
<div class="col-md-12">
    <label for="createtx_txhex">Transaction Header/Message
    <textarea id="createtx_txhex" class="form-control" plac
</div>
<div class="col-md-12">
    <button id="txcreate_withdraw" type="button" class="btr
    <button type="submit" id="txcreatesubmit_withdraw" clas
</div>
</div>
</form>
</div>
</div>
<!-- END Create TX Withdraw -->
</div>
<!-- END DIV TRANSACTION -->
<!-- ===== -->

<!-- ===== -->
<!-- BEGIN DIV WALLETS INFO -->
<div class="tab-pane fade" id="wallet">
    <div class="panel panel-info">
        <div class="panel-heading">
            Wallet
            <button type="submit" onClick="populateAllWalletData();" id="bt
        </div>
        <div class="panel-body">
            <div id="divWalletsStatus" class="table-responsive"></div>
        </div>
    </div>
    <div class="panel panel-info">
        <div class="panel-heading">Register new account</div>
        <div class="panel-body">
            <div class="row">
                <div class="col-md-6">
                    Address (17 bytes): <input style="color: green;" type="te
                </div>
            </div>
            <div class="row">
                <div class="col-md-6">
                    ScriptHash (20 bytes): <input style="color: green;" type=
                </div>
            </div>
            <div class="row">
                <div class="col-md-6">
                    PublicKey (33 bytes): <input style="color: green;" type="
                </div>
            </div>
            <div class="row">
                <div class="col-md-6">
                    WIF (26 bytes): <input style="color: green;" type="text"
                </div>
            </div>
        </div>
    </div>

```

```

<div class="row">
  <div class="col-md-6">
    PrivateKey (32 bytes): <input style="color: green;" type="text">
  </div>
</div>
<div class="row">
  <div class="col-md-6">
    EncryptedKey (29 bytes): <input style="color: green;" type="text">
  </div>
</div>
<input data-toggle="collapse" data-target="#div_addMultiSig_byVerificationScript" id="div_addMultiSig_byVerificationScript" type="checkbox"/>
<div class="panel-body collapse" id="div_addMultiSig_byVerificationScript">
  VerificationScript (? bytes): <input style="color: green;" type="text">
</div>
<div class="row">
  <div class="col-md-6">
    <button type="button" onClick="addWalletFromForm();" id="addWalletFromForm">Add Wallet From Form
  </div>
</div>
</div>
<div class="panel panel-info">
  <div class="panel-heading">Accounts detailed info</div>
  <div class="panel-body">
    <div class="row">
      <div class="col-md-4">
        <span class="label label-primary">Pick an address:</span>
        <select name="wallet_info" onChange="changeWalletInfo();" id="wallet_info">
          <option value="1">1
          <option value="2">2
          <option value="3">3
          <option value="4">4
          <option value="5">5
          <option value="6">6
          <option value="7">7
          <option value="8">8
          <option value="9">9
          <option value="10">10
          <option value="11">11
          <option value="12">12
          <option value="13">13
          <option value="14">14
          <option value="15">15
          <option value="16">16
          <option value="17">17
          <option value="18">18
          <option value="19">19
          <option value="20">20
          <option value="21">21
          <option value="22">22
          <option value="23">23
          <option value="24">24
          <option value="25">25
          <option value="26">26
          <option value="27">27
          <option value="28">28
          <option value="29">29
          <option value="30">30
          <option value="31">31
          <option value="32">32
          <option value="33">33
          <option value="34">34
          <option value="35">35
          <option value="36">36
          <option value="37">37
          <option value="38">38
          <option value="39">39
          <option value="40">40
          <option value="41">41
          <option value="42">42
          <option value="43">43
          <option value="44">44
          <option value="45">45
          <option value="46">46
          <option value="47">47
          <option value="48">48
          <option value="49">49
          <option value="50">50
          <option value="51">51
          <option value="52">52
          <option value="53">53
          <option value="54">54
          <option value="55">55
          <option value="56">56
          <option value="57">57
          <option value="58">58
          <option value="59">59
          <option value="60">60
          <option value="61">61
          <option value="62">62
          <option value="63">63
          <option value="64">64
          <option value="65">65
          <option value="66">66
          <option value="67">67
          <option value="68">68
          <option value="69">69
          <option value="70">70
          <option value="71">71
          <option value="72">72
          <option value="73">73
          <option value="74">74
          <option value="75">75
          <option value="76">76
          <option value="77">77
          <option value="78">78
          <option value="79">79
          <option value="80">80
          <option value="81">81
          <option value="82">82
          <option value="83">83
          <option value="84">84
          <option value="85">85
          <option value="86">86
          <option value="87">87
          <option value="88">88
          <option value="89">89
          <option value="90">90
          <option value="91">91
          <option value="92">92
          <option value="93">93
          <option value="94">94
          <option value="95">95
          <option value="96">96
          <option value="97">97
          <option value="98">98
          <option value="99">99
          <option value="100">100
        </select>
      </div>
      <div class="col-md-2">
        <span class="label label-success">AddressBase58:</span>
        <textarea id="walletInfoAddressBase58" readonly="readonly" name="walletInfoAddressBase58">
      </div>
      <div class="col-md-2">
        <span class="label label-success">ScriptHash:</span>
        <textarea id="walletInfoScripthash" readonly="readonly" name="walletInfoScripthash">
      </div>
      <div class="col-md-2">
        <span class="label label-success">PublicKey:</span>
        <textarea id="walletInfoPubKey" readonly="readonly" name="walletInfoPubKey">
      </div>
      <div class="col-md-2">
        <span class="label label-success">VerificationScript:</span>
        <textarea id="addressVerificationScript" readonly="readonly" name="addressVerificationScript">
      </div>
    </div>
    <div class="row">
      <div class="col-md-2">
        <span class="label label-success">WIF:</span>
        <textarea id="walletInfoWIF" readonly="readonly" name="walletInfoWIF">
      </div>
    </div>
  </div>
</div>

```

```

        <div class="col-md-2">
            <span class="label label-success">PrivateKey:</span>
            <textarea id="walletInfoPrivateKey" readonly="readonly" r
        </div>
        <div class="col-md-2">
            <span class="label label-success">Print:</span>
            <textarea id="addressPrintInfo" readonly="readonly" name=
        </div>
        <div class="col-md-2">
            <span class="label label-success">Owners:</span>
            <textarea id="addressOwners" readonly="readonly" name="ov
        </div>
    </div>

    <div class="row">
        <div class="col-md-2">
            <br /><button type="" onClick="removeAccountFromEcoWallet
                <i class="fas fa-user-minus"></i> Remove account
            </button>
        </div>
        <div class="col-md-2">
            <span class="label label-success">EncryptedKey:</span>
            <textarea id="walletInfoEncrypted" readonly="readonly" na
        </div>
        <div class="col-md-2">
            <form class="form-group" onSubmit="decrypt()">
                <div id="dialog" title="Basic dialog" hidden>
                    <input autocomplete="eco" id="passwordNep2" class="
                    <button type="submit" class="form-control btn btn-i
                </div>
            </form>
        </div>
    </div>

    </div> <!-- END PANEL BODY -->
</div>
<!-- END DIV WALLETS INFO -->
<!-- ===== -->

<!-- ===== -->
<!-- BEGIN DIV NODEINFO -->
    <div class="tab-pane fade" id="cnnodesinfo">
        <div class="panel panel-primary">
            <div class="panel-heading">
                Node1
                <button type="submit" onClick="getNodeInfo()" class="btn btn-info
            </div>
            <div class="panel-body">
                <div class="row">
                    <div class="col-md-12">
                        <textarea id="node1data" class="form-control" style="font-size:
                    </div>
                </div>
            </div>
        </div>
    </div> <!-- node1 end -->

```

```

<div class="panel panel-primary">
  <div class="panel-heading">
    Node2
    <button type="submit" onClick="getNodeInfo()" class="btn btn-info">
  </div>
  <div class="panel-body">
    <div class="row">
      <div class="col-md-12">
        <textarea id="node2data" class="form-control" style="font-size: 1.2em;">
      </div>
    </div>
  </div>
</div> <!-- node2 end -->
<div class="panel panel-primary">
  <div class="panel-heading">
    Node3
    <button type="submit" onClick="getNodeInfo()" class="btn btn-info">
  </div>
  <div class="panel-body">
    <div class="row">
      <div class="col-md-12">
        <textarea id="node3data" class="form-control" style="font-size: 1.2em;">
      </div>
    </div>
  </div>
</div> <!-- node3 end -->
<div class="panel panel-primary">
  <div class="panel-heading">
    Node4
    <button type="submit" onClick="getNodeInfo()" class="btn btn-info">
  </div>
  <div class="panel-body">
    <div class="row">
      <div class="col-md-12">
        <textarea id="node4data" class="form-control" style="font-size: 1.2em;">
      </div>
    </div>
  </div>
</div> <!-- noderpc end -->
<div class="panel panel-primary">
  <div class="panel-heading">
    NodeRPC(WatchOnly)
    <button type="submit" onClick="getNodeInfo()" class="btn btn-info">
  </div>
  <div class="panel-body">
    <div class="row">
      <div class="col-md-12">
        <textarea id="noderpcdata" class="form-control" style="font-size: 1.2em;">
      </div>
    </div>
  </div>
</div> <!-- noderpc end -->
</div>
<!-- END DIV NODES INFO -->
<!-- ===== -->

```

```

<!-- ===== -->
<!-- BEGIN STATISTICS NODEINFO -->
  <div class="tab-pane fade" id="statistics">
    <div class="panel panel-primary">
      <div class="panel-heading">
        Blocks timestamps
        <button type="submit" onClick="callCoreMetricsFromForm()" class="btn btn-primary">Submit
      </div>

      <div class="panel-body">
        <div class="row">
          <div class="col-md-2">
            <span class="label label-success">Max nBlocks to query</span>
            <br/><input type="text" onChange="callCoreMetricsFromForm()" id="cbx_max_nBlocks">
          </div>
          <div class="col-md-2">
            <span class="label label-success">Height to start</span>
            <br/><input type="text" onChange="callCoreMetricsFromForm()" id="cbx_height">
          </div>
          <div class="col-md-2">
            <span class="label label-success">Min/Max yTick timeseries</span>
            <br/><input type="text" onChange="reloadTSGraph()" id="cbx_min_yTick">
            <input type="text" onChange="reloadTSGraph()" id="cbx_max_yTick">
          </div>
        </div>

        <div class="row">
          <div class="col-md-8">
            <textarea id="txt_area_timestampsStats" placeholder="Timestamps" style="height: 100px;">
          </div>
        </div>

        <div class="row">
          <div class="col-md-8">
            <canvas id="canvas_blocktime_chart"></canvas>
          </div>
        </div>
      </div>

    </div> <!-- Statis end -->
  </div>
<!-- END DIV STATISTICS INFO -->
<!-- ===== -->

<!-- ===== -->
<!-- BEGIN CONFIGURATIONS NODEINFO -->
  <div class="tab-pane fade" id="configurations">
    <div class="panel panel-primary">
      <div class="panel-heading">
        Personalized Base Paths
      </div>
      <div class="panel-body">
        <div class="col-md-2">
          <!-- <input type="text" onChange="updatePATHS()" id="neonodeurl" class="form-control">
          <span class="label label-success">Neo-cli RPCs</span>
          <select id="selectionBoxRPCNodes" class="custom-select form-control">

```

```

        <span class="label label-info">Selected service</span>
        <input type="text" onchange="updateBASEPATHS()" id="rpc_nodes_path" />
    </div>
    <div class="col-md-2">
        <span class="label label-success">Compilers</span>
        <select id="selectionBoxCompilers" class="custom-select form-control">
            <option value="0">None</option>
            <option value="1">C#</option>
            <option value="2">C++</option>
            <option value="3">C</option>
            <option value="4">C#</option>
            <option value="5">C++</option>
            <option value="6">C</option>
        </select>
        <span class="label label-info">Selected service</span>
        <input type="text" onchange="updateBASEPATHS()" id="compilers_path" />
    </div>
    <div class="col-md-2">
        <span class="label label-success">EcoServices</span>
        <select id="selectionBoxEcoServices" class="custom-select form-control">
            <option value="0">None</option>
            <option value="1">C#</option>
            <option value="2">C++</option>
            <option value="3">C</option>
            <option value="4">C#</option>
            <option value="5">C++</option>
            <option value="6">C</option>
        </select>
        <span class="label label-info">Selected service</span>
        <input type="text" onchange="updateBASEPATHS()" id="eco_services_path" />
    </div>
    <div class="col-md-2">
        <span class="label label-success">NEOSCAN</span>
        <select id="selectionBoxNeoScan" class="custom-select form-control">
            <option value="0">None</option>
            <option value="1">C#</option>
            <option value="2">C++</option>
            <option value="3">C</option>
            <option value="4">C#</option>
            <option value="5">C++</option>
            <option value="6">C</option>
        </select>
        <span class="label label-info">Selected service</span>
        <input type="text" onchange="updateBASEPATHS()" id="neoscan_path" />
    </div>
    <!--
    <div class="col-md-2">
        <span class="label label-success">REST Notifications</span>
        <input type="text" onchange="updateBASEPATHS()" id="rest_notifications_path" />
        <span class="label label-info">Selected service</span>
        <select id="selectionBoxRestNotifications" class="custom-select form-control">
            <option value="0">None</option>
            <option value="1">C#</option>
            <option value="2">C++</option>
            <option value="3">C</option>
            <option value="4">C#</option>
            <option value="5">C++</option>
            <option value="6">C</option>
        </select>
    </div>
-->
</div>
</div>
<div class="panel panel-primary">
    <div class="panel-heading">
        Automatic selection
    </div>
    <div class="panel-body">
        <div class="col-md-2">
            <span class="label label-success">Compiler server</span>
            <br/><input type="checkbox" id="cbx_automatic_compiler" checked="" />
        </div>
        <div class="col-md-2">
            <span class="label label-success">RPC c# neo-cli</span>
            <br/><input type="checkbox" id="cbx_automatic_node_best_height" />
        </div>
    </div>
</div>
<div class="panel panel-primary">
    <div class="panel-heading">
        Consensus Node
    </div>
    <div class="panel-body">
        <div class="form-group">
            <div class="row">
                <div class="col-md-2">
                    <span class="label label-success">Seconds per block</span>
                    <br/><input type="text" id="cbx_spb_cn" placeholder="15" size="15" />
                </div>
            </div>
        </div>
    </div>
</div>

```

```

        </div>
        <div class="col-md-2">
            <span class="label label-success">Password</span>
            <br/><input type="text" id="cbx_pwd_cn" placeholder="neorese
        </div>
    </div>
    <div class="row">
        <div class="col-md-2">
            <button type="button" onClick="onClickBtnSetCNBlockTime()" ic
        </div>
    </div>
</div>
<div class="panel panel-primary">
    <div class="panel-heading">
        Explorers
    </div>
    <div class="panel-body">
        <div class="form-group">
            <div class="row">
                <div class="col-md-2">
                    <span class="label label-success">View NEOSCAN</span>
                    <br/><input type="checkbox" id="cbx_enable_neoscan">View lir
                </div>
                <div class="col-md-2">
                    <span class="label label-success">Perform Queries using NEOS
                    <br/><input type="checkbox" id="cbx_query_neoscan">Queries (
                </div>
            </div>
        </div>
    </div>
</div>
<div class="panel panel-primary">
    <div class="panel-heading">
        Configs
    </div>
    <div class="panel-body">
        <div class="form-group">
            <div class="row">
                <div class="col-md-2">
                    <span type="checkbox" class="label label-success"><i class="
                    <br/><input type="checkbox" id="cbx_ecoNotifications"></inpu
                </div>
                <div class="col-md-3">
                    <span class="label label-success">Password</span>
                    <br/><input type="text" id="cbx_pwd_reset_service" placehol
                    <br/><button type="button" onClick="onClickPostServerToResetI
                </div>
            </div>
        </div>
    </div>
</div>
<div class="panel panel-primary">
    <div class="panel-heading">
        Automatic refreshing (ms)

```

```

    </div>
    <div class="panel-body">
        <div class="form-group">
            <div class="row">
                <div class="col-md-2">
                    <span class="label label-success">Wallet balances</span>
                    <br/><input type="text" id="cbx_refresh_wallet" placeholder="1000000" />
                </div>
                <div class="col-md-2">
                    <span class="label label-success">Headers info</span>
                    <br/><input type="text" id="cbx_refresh_headers" placeholder="1300000" />
                </div>
                <div class="col-md-2">
                    <span class="label label-success">Ping compilers</span>
                    <br/><input type="text" id="cbx_refresh_compilers" placeholder="1000000" />
                </div>
                <div class="col-md-2">
                    <span class="label label-success">Ping neo-clients</span>
                    <br/><input type="text" id="cbx_refresh_neocli" placeholder="1000000" />
                </div>
                <div class="col-md-2">
                    <span class="label label-success">GenesisBlock</span>
                    <br/><input type="text" id="cbx_refresh_gen" placeholder="1000000" />
                </div>
                <div class="col-md-2">
                    <span class="label label-success">Eco metadata</span>
                    <br/><input type="text" id="cbx_refresh_ecometadata" placeholder="1000000" />
                </div>
            </div>
        </div>
    </div>
</div>
<!-- END CONFIGURATIONS INFO -->
<!-- ===== -->

<!-- ===== -->
<!-- BEGIN INCREMENTAL STORAGE -->
    <div class="tab-pane fade" id="incrementalstorage">
        <div class="panel panel-primary">
            <div class="panel-heading">
                Incremental storage
                <button type="submit" onClick="getIncrementalStorageInfo()" class="btn btn-primary">Get Info</button>
                <input id="heightForIncStorage" class="form-control" value="0" placeholder="Height for incremental storage" />
            </div>
            <div class="panel-body">
                <div class="row">
                    <div class="col-md-12">
                        <textarea id="textAreaOfIncrementalStorage" readonly="readonly" />
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
<!-- END DIV INCREMENTAL STORAGE -->
<!-- ===== -->

```



```

<!-- ===== -->
<!-- BEGIN DIV COMPILERS -->
<div class="tab-pane fade in active" id="compilers">
    <div class="panel panel-success">
        <div class="panel-heading">Compile</div>
        <div class="panel-body">
            <form id="formCompile" action="/compilex" style="margin-top:
            <div class="tab-container">
                <ul class="nav nav-tabs" id="compilersnavtabs">
                    <li class="active"><a data-target="#csharpcode" c
                    <li><a data-target="#pythoncode" data-toggle="tak
                    <li><a data-target="#golangcode" data-toggle="tak
                    <li><a data-target="#javacode" data-toggle="tab">
                </ul>

                <!-- //////////////////////////////////////
                <!-- csharpcode -->
                <div class="tab-content">
                    <div class="tab-pane active cont" id="csharpcode"
                        <div class="dropdown btn-group" style="ma
                            <a class="btn btn-sucess dropdown-to
                                Select Example
                                <span class="caret"></span>
                            </a>
                            <ul class="dropdown-menu" id="Example
                                <li><a onclick="setExample('cshar
                                <li><a onclick="setExample('cshar
                                <li><a onclick="setExample('cshar
                                <li><a onclick="setExample('cshar
                                <li><a onclick="setExample('cshar
                                <li><a onclick="setExample('cshar
                                <li><a onclick="setExample('cshar
                            </ul>
                        </div>
                        <input type="hidden" id="code_cs" name="c
                    </div>
                <!-- //////////////////////////////////////
                <!-- end csharpcode -->

                <!-- //////////////////////////////////////
                <!-- java div -->
                <div class="tab-pane cont" id="javacode">
                    <input type="hidden" id="code_java" name="c
                </div>
                <!-- //////////////////////////////////////
                <!-- end java -->

                <!-- //////////////////////////////////////
                <!-- golang div -->
                <div class="tab-pane cont" id="golangcode">
                    <input type="hidden" id="code_golang" name="
                </div>
                <!-- //////////////////////////////////////
                <!-- golang java -->

```

```
<!-- ////////////////////////////////////////
<!-- python div -->
<div class="tab-pane cont" id="pythoncode">
  <input type="hidden" id="code_python" name=
</div>
```

```
<pre id="aceEditor"></pre>
</div>
</div>
<!-- end python div -->
<!-- ////////////////////////////////////////

<!-- ////////////////////////////////////////
<!-- common submit button -->
<p><font size="1">Use CTRL+ALT+H to access the hotkeys in
<br><span id="span_numberCompilations" class="label label
</p>
<div class="dropdown btn-group" style="margin-bottom: 10px">
  <label for="compilers_versions-selection-box">Compiler
  <select name="compilers_versions" id="compilers_versions"
</div>
<br><input type="checkbox" id="cbx_optimize">Optimize output
<br><input type="checkbox" id="cbx_compatible" name="cbx_compatible">Compatible
<button type="submit" id="compilebtn" style="margin-top: 5px;">Compile
<h4>Compile Warning/Errors:</h4>
<!-- end common submit button -->
<!-- ////////////////////////////////////////

<textarea id="codewarnerr" readonly="readonly" rows="7" class="form-control">
<h4>AVM (in hex):</h4>
<textarea id="codeavm" name="codeavm" readonly="readonly" rows="10" class="form-control">
<div class="dropdown btn-group">
  <a class="btn btn-warning dropdown-toggle" data-toggle="dropdown" href="#">
    <i class="fas fa-download"></i> AVM
    <span class="caret"></span>
  </a>
  <ul class="dropdown-menu" id="downloadMenu">
    <li><a onclick="downloadAVM('0')">binary file</a>
    <li><a onclick="downloadAVM('1')">hex code</a></li>
  </ul>
</div>
<button data-toggle="collapse" data-target="#opcodesDiv" class="btn btn-warning">
<i class="fas fa-upload"></i> AVM: <input type="file" id="codeavm">
<div id="opcodesDiv" class="collapse">
  <textarea id="txt_opcodes" readonly="readonly" rows="10" class="form-control">
  <button id="btn_downloadOpCode" style="margin-bottom: 5px;">Download
</div>
<h4>Compiled contract info:</h4>
<button type="button" onClick="addContractToWallet($('#contractInfo_ScriptHash').val())">Add
<label for="contractInfo_ScriptHash">ScriptHash</label>
<input id="contractInfo_ScriptHash" readonly="readonly" type="text">
<label for="contractInfo_Address">Address</label>
<input id="contractInfo_Address" readonly="readonly" type="text">
```

```

        <label for="contractInfo_AVMSize">AVM size (in bytes)</label>
        <input id="contractInfo_AVMSize" readonly="readonly" name="contractInfo_AVMSize" type="text" value="" />

        <h4>ABI:</h4>
        <textarea id="codeabi" readonly="readonly" rows="5" cols="50" />
    </form> <!-- id="formCompile" -->
    <!-- end of complex form -->
    <!-- ////////////////////////////////////////
</div>
</div>
</div>
<!-- end compilers -->
<!-- begin activity -->
<div class="tab-pane fade" id="activity">
    <div class="panel panel-info" id="divNetworkRelayed">
        <div class="panel-heading">Historical Relayed TX's
            <button type="submit" onClick="searchForTXs()" id="btndrawRelayedTXs">Draw Relayed TX's
        </div>
        <div class="panel-body">
            <div class="row">
                <div class="col-md-2">
                    <span for="cbx_fullHistory" type="checkbox" class="label label-success">Full History
                    <input type="checkbox" id="cbx_fullHistory"></input>
                </div>
            </div>
            <div class="row">
                <div class="col-md-12">
                    <div id="divRelayedTXs" class="table-responsive"></div>
                </div>
            </div>
            <div class="row">
                <div class="col-md-2">
                    <button onClick="moveLastTab()" id="btnActivityMoveBackLastTab">Move Back Last Tab
                </div>
                <div class="col-md-2">
                    <button type="submit" onClick="exportHistory()" id="btnExportHistory">Export History
                </div>
                <div class="col-md-2">
                    <span class="label label-info"><i class="fas fa-file-upload"></i>
                    <input type="file" id="importActivityFile" accept=".json,.JSON" />
                </div>
                <div class="col-md-2">
                    <button type="submit" onClick="searchForTXs()" id="btnGenerateTXs">Generate TXs
                </div>
            </div>
            <span type="checkbox" class="label label-success">Open Replay TX's Tool
            <input type="checkbox" data-toggle="collapse" data-target="#replayTXsDiv" />
            <div id="replayTXsDiv" class="collapse">
                <textarea id="txt_replayOrder" readonly="readonly" rows="5" cols="50" />
                <button id="btnReplayTXs" onClick="replayTXs()" style="margin-bottom: 10px;">Replay TXs
            </div>
        </div>
    </div>
</div>
</div>
<!-- end activity -->

```

```

<!-- begin network -->
<div class="tab-pane fade" id="network">

  <div class="panel panel-primary" id="divNetworkDeploy">
    <div class="panel-heading">Deploy contract into the Blockchain</div>
    <div class="panel-body">

      <div class="tab-container">
        <ul class="nav nav-tabs">
          <li class="active"><a data-target="#tabdeployjs" data-toggle="tab" href="#tabdeployjs">Deploy Contract</a></li>
        </ul>

        <div class="tab-content">

          <!-- begin deploy js -->
          <div class="tab-pane active cont" id="tabdeployjs">
            <form id="formdeployjs" method="post">
              <div class="row">
                <div class="col-md-6">
                  <div class="form-group">
                    <label for="contracthash">Contract ScriptHash</label>
                    <input id="contracthashjs" name="contracthash" class="form-control" type="text">
                  </div>
                </div>

                <div class="col-md-3">
                  <div class="form-group">
                    <label for="contractparams">Deploy Params</label>
                    <input id="contractparamsjs" name="contractparams" value="" type="text">
                  </div>
                </div>

                <div class="col-md-3">
                  <label for="contractparamnames">Param Type Names</label>
                  <input id="contractparamnamesjs" name="contractparamnames" value="" type="text">
                </div>

                <div class="col-md-6 form-inline" style="margin-bottom: 10px;">
                  <div class="form-group">
                    <select name="contractreturn" id="contractreturnjs" class="form-control">
                      <option title="00 - Signature" value="00">00 - Signature</option>
                      <option title="01 - Boolean" value="01">01 - Boolean</option>
                      <option title="02 - Integer" value="02">02 - Integer</option>
                      <option title="03 - Hash160" value="03">03 - Hash160</option>
                      <option title="04 - Hash256" value="04">04 - Hash256</option>
                      <option title="05 - ByteArray" value="05">05 - ByteArray</option>
                      <option title="06 - PublicKey" value="06">06 - PublicKey</option>
                      <option title="07 - String" value="07">07 - String</option>
                      <option title="10 - Array" value="10">10 - Array</option>
                      <option title="f0 - InteropInterface" value="f0">f0 - InteropInterface</option>
                      <option title="ff - Void" value="ff">ff - Void</option>
                    </select>
                  </div>

                  <div class="form-group" style="margin-right: 8px;">
                    <div class="checkbox">
                      <input type="checkbox"/>
                      <label>

```

```

        <input type="checkbox" name="cbx_storage" id="cbx_s
    </label>
</div>
</div>
<div class="form-group" style="margin-right: 8px">
    <div class="checkbox">
        <label>
            <input type="checkbox" name="cbx_dynamicinvoke" id=
        </label>
    </div>
</div>
<div class="form-group" style="margin-right: 8px">
    <div class="checkbox">
        <label>
            <input type="checkbox" name="cbx_ispayable" id="cbx
        </label>
    </div>
</div>
</div>
</div>

<div class="col-md-3">
    <label for="contractparamnames">Contract Name</label>
    <input id="jsdeploy_name" name="jsdeploy_name" value='a
</div>
<div class="col-md-3">
    <label for="contractparamnames">Contract Author</label>
    <input id="jsdeploy_author" name="jsdeploy_author" value
</div>
<div class="col-md-6">
    <label for="contractparamnames">Contract Description</la
    <input id="jsdeploy_desc" name="jsdeploy_desc" value='a
</div>
<div class="col-md-3">
    <label for="contractparamnames">Contract e-mail</label>
    <input id="jsdeploy_email" name="jsdeploy_email" value='
</div>
<div class="col-md-3">
    <label for="contractparamnames">Contract version</label>
    <input id="jsdeploy_version" name="jsdeploy_version" val
</div>

<div class="col-md-3">
    <div class="form-group">
        <label>Caller (AddressBase58)</label>
        <select name="wallet_deployjs" id="wallet_deployjs" c
    </div>
</div>

<div class="col-md-3">
    <div class="form-group">
        <label for="attachDeployGasFeeJS">Network fee (in GAS)<
        <input type="number" min="0" step="0.00000001" id="atte
    </div>
</div>

<div class="col-md-12">

```

```

        <button type="submit" id="deploybtnjs" class="btn btn-suc
    </div>
    <div class="col-md-12">
        <span id="span_numberDeploys" class="label label-info"></
    </div>

    </div>
</form>
</div> <!-- end panel deploy js -->

</div> <!-- end content -->
</div> <!-- end containers -->
</div> <!-- end body -->
</div> <!-- warning -->

<div class="panel panel-primary" id="divNetworkInvoke">
    <div class="panel-heading">Interacting with a Smart Contract - Invoke</div>
    <div class="panel-body">

        <div class="tab-container">
            <ul class="nav nav-tabs">
                <li class="active"><a data-target="#tabinvokejs" data-toggle="tab
            </ul>
            <div class="tab-content">
                <!-- begin invokejs -->
                <div class="tab-pane active cont" id="tabinvokejs">
                    <form id="forminvokejs" method="post">

                        <div class="row">
                            <div class="col-md-6"> <!-- LEFT SIDE OF INVOKE BOX -->
                                <div class="row">
                                    <div class="col-md-6">
                                        <div class="form-group">
                                            <label for="invokehashjs">Contract ScriptHash</label>
                                            <input id="invokehashjs" name="invokehashjs" class="form-cor
                                        </div>
                                    </div>
                                </div>
                            </div>

                            <div class="row">
                                <div class="col-md-6">
                                    <div class="form-group">
                                        <label for="attachSystemgasjs">System fee (GAS) </label>
                                        <input id="attachSystemgasjs" type="number" min="0" step=
                                    </div>
                                </div>
                                <div class="col-md-6">
                                    <div class="form-group">
                                        <label for="attachgasfeejs">Network fee (GAS)</label>
                                        <input id="attachgasfeejs" type="number" min="0" step="0
                                    </div>
                                </div>
                            </div>
                        </div>
                    </form>
                </div>
            </div>
        </div>
    </div>

```

```

<div class="row">
  <div class="col-md-6">
    <div class="form-group">
      <label for="attachneojis">Attach NEO</label>
      <input id="attachneojis" type="number" min="0" step="0.0001">
    </div>
  </div>
  <div class="col-md-6">
    <div class="form-group">
      <label for="attachgasjs">Attach GAS</label>
      <input id="attachgasjs" type="number" min="0" step="0.0001">
    </div>
  </div>
</div> <!-- LEFT col6 -->
<div class="col-md-6"> <!-- RIGHT SIDE OF INVOKE BOX -->
  <div class="form-group">
    <label for="invokeparamsjs">Invoke Parameters (JSON)</label><br>
    <textarea id="invokeparamsjs" name="invokeparamsjs" rows="5" style="width: 100%; height: 100%; border: 1px solid #ccc;">
  </div>
</div> <!-- RIGHT col6 -->
</div> <!-- finish row LEFT/RIGHT -->

<div class="row">
  <div class="col-md-3">
    <div class="form-group">
      <label for="invokefunctionjs">Function to Invoke</label>
      <select id="invokefunctionjs" class="custom-select form-control">
        <option>Main</option>
      </select>
    </div>
  </div>
  <!-- Param 1 -->
  <div class="col-md-1">
    <div class="form-group">
      <label>Type</label>
      <select id="invokeparamjsbox1" class="custom-select form-control">
        <option>None</option>
        <option>String</option>
        <option>Address</option>
        <option>Hex</option>
        <option>Integer</option> <!-- big integer -->
        <option>DecFixed8</option> <!-- fixed8 in decimal -->
      </select>
    </div>
  </div>
  <div class="col-md-2">
    <div class="form-group">
      <label>Param 1: </label> <input type="checkbox" id="invokeparamjs1">
      <input id="invokeparamsjs1" class="form-control" value="">
    </div>
  </div>
  <!-- Param 2 -->
  <div class="col-md-1">
    <div class="form-group">
      <label>Type</label>

```

```

        <select id="invokeparamjsbox2" class="custom-select">
            <option>None</option>
            <option>String</option>
            <option>Address</option>
            <option>Hex</option>
            <option>Integer</option> <!-- big integer -->
            <option>DecFixed8</option> <!-- fixed8 in decimal -->
        </select>
    </div>
</div>
<div class="col-md-2">
    <div class="form-group">
        <label>Param 2</label> <input type="checkbox" id="checkbox2">
        <input id="invokeparamsjs2" class="form-control" value="" type="text"/>
    </div>
</div>
<!-- Param 3 -->
<div class="col-md-1">
    <div class="form-group">
        <label>Type</label>
        <select id="invokeparamjsbox3" class="custom-select">
            <option>None</option>
            <option>String</option>
            <option>Address</option>
            <option>Hex</option>
            <option>Integer</option> <!-- big integer -->
            <option>DecFixed8</option> <!-- fixed8 in decimal -->
        </select>
    </div>
</div>
<div class="col-md-2">
    <div class="form-group">
        <label>Param 3</label> <input type="checkbox" id="checkbox3">
        <input id="invokeparamsjs3" class="form-control" value="" type="text"/>
    </div>
</div>
</div> <!-- row -->
<div class="row">
    <div class="col-md-3">
        <div class="form-group">
            <label>Caller (AddressBase58)</label>
            <select name="wallet_invokejs" id="wallet_invokejs" class="form-control">
                <option value="">None</option>
                <option value="1">1</option>
                <option value="2">2</option>
                <option value="3">3</option>
                <option value="4">4</option>
                <option value="5">5</option>
                <option value="6">6</option>
                <option value="7">7</option>
                <option value="8">8</option>
                <option value="9">9</option>
                <option value="10">10</option>
                <option value="11">11</option>
                <option value="12">12</option>
                <option value="13">13</option>
                <option value="14">14</option>
                <option value="15">15</option>
                <option value="16">16</option>
                <option value="17">17</option>
                <option value="18">18</option>
                <option value="19">19</option>
                <option value="20">20</option>
                <option value="21">21</option>
                <option value="22">22</option>
                <option value="23">23</option>
                <option value="24">24</option>
                <option value="25">25</option>
                <option value="26">26</option>
                <option value="27">27</option>
                <option value="28">28</option>
                <option value="29">29</option>
                <option value="30">30</option>
                <option value="31">31</option>
                <option value="32">32</option>
                <option value="33">33</option>
                <option value="34">34</option>
                <option value="35">35</option>
                <option value="36">36</option>
                <option value="37">37</option>
                <option value="38">38</option>
                <option value="39">39</option>
                <option value="40">40</option>
                <option value="41">41</option>
                <option value="42">42</option>
                <option value="43">43</option>
                <option value="44">44</option>
                <option value="45">45</option>
                <option value="46">46</option>
                <option value="47">47</option>
                <option value="48">48</option>
                <option value="49">49</option>
                <option value="50">50</option>
                <option value="51">51</option>
                <option value="52">52</option>
                <option value="53">53</option>
                <option value="54">54</option>
                <option value="55">55</option>
                <option value="56">56</option>
                <option value="57">57</option>
                <option value="58">58</option>
                <option value="59">59</option>
                <option value="60">60</option>
                <option value="61">61</option>
                <option value="62">62</option>
                <option value="63">63</option>
                <option value="64">64</option>
                <option value="65">65</option>
                <option value="66">66</option>
                <option value="67">67</option>
                <option value="68">68</option>
                <option value="69">69</option>
                <option value="70">70</option>
                <option value="71">71</option>
                <option value="72">72</option>
                <option value="73">73</option>
                <option value="74">74</option>
                <option value="75">75</option>
                <option value="76">76</option>
                <option value="77">77</option>
                <option value="78">78</option>
                <option value="79">79</option>
                <option value="80">80</option>
                <option value="81">81</option>
                <option value="82">82</option>
                <option value="83">83</option>
                <option value="84">84</option>
                <option value="85">85</option>
                <option value="86">86</option>
                <option value="87">87</option>
                <option value="88">88</option>
                <option value="89">89</option>
                <option value="90">90</option>
                <option value="91">91</option>
                <option value="92">92</option>
                <option value="93">93</option>
                <option value="94">94</option>
                <option value="95">95</option>
                <option value="96">96</option>
                <option value="97">97</option>
                <option value="98">98</option>
                <option value="99">99</option>
            </select>
        </div>
</div>
<div class="col-md-3">
        <span class="label label-info"><i class="fas fa-file-upload"></i> Import ABI File</span>
        <input type="file" id="importABIFile" accept=".json,.JSON" />
    </div>
</div>
<div class="row">
    <div class="col-md-2">
        <button type="submit" id="invokebtn" class="btn btn-success">Invoke</button>
    </div>
    <div class="col-md-2">
        <div class="form-group">
            <span type="checkbox" class="label label-success">Advanced</span>
        </div>
    </div>
</div>

```



```

        <input type="checkbox" data-toggle="collapse" data-target="#divAdvancedSigning" />
    </div>
</div>
<div class="col-md-2">
    <span id="span_numberInvokes" class="label label-info">Number of invokes</span>
</div>
</div> <!-- row -->
</form>
</div> <!-- end invokejs -->
</div> <!-- tab-content -->
</div> <!-- tab-container -->
</div> <!-- body -->
</div> <!-- panel warning -->

<div class="panel panel-info collapse" id="divAdvancedSigning">
    <div class="panel-heading">Advanced signing</div>
    <div class="panel-body">
        <div class="row">
            <div class="col-md-6">
                <label for="tx_AdvancedSigning_ScriptHash">ScriptHash</label>
                <input id="tx_AdvancedSigning_ScriptHash" readonly="readonly" name="tx_AdvancedSigning_ScriptHash" />
                <label for="tx_AdvancedSigning_HeaderSize">TX Header Size</label>
                <input id="tx_AdvancedSigning_HeaderSize" readonly="readonly" name="tx_AdvancedSigning_HeaderSize" />
                <label for="tx_AdvancedSigning_Size">TX Size</label>
                <input id="tx_AdvancedSigning_Size" readonly="readonly" name="tx_AdvancedSigning_Size" />
            </div>
            <div class="col-md-6">
                <label for="txScript_advanced_signing">Transaction Header</label>
                <textarea id="txScript_advanced_signing" readonly="readonly" rows="4"></textarea>
            </div>
        </div> <!-- row -->
        <div class="row">
            <div class="col-md-6">
                <select required multiple onchange="addAttributedAndTryToSign()" name="txScript_advanced_signing_scripts">
                    <option value="1">1</option>
                    <option value="2">2</option>
                    <option value="3">3</option>
                    <option value="4">4</option>
                    <option value="5">5</option>
                    <option value="6">6</option>
                    <option value="7">7</option>
                    <option value="8">8</option>
                    <option value="9">9</option>
                    <option value="10">10</option>
                </select>
            </div>
            <div class="col-md-6">
                <label for="txtAdvancedSigning">Scripts - Witness (json)</label>
                <textarea onchange="addScriptsToPendingTx()" id="txt_AdvancedSigning"></textarea>
            </div>
        </div> <!-- row -->
        <div class="row">
            <div class="col-md-2">
                <button type="submit" id="sendRawTxBtn" class="btn btn-success btn-sm">Send Raw Transaction</button>
            </div>
        </div> <!-- row -->
    </div>
</div>
<!-- end deploy invoke -->

<!-- begin rawRPC -->
<div class="tab-pane fade" id="rawRPC">
    <div class="panel panel-primary" id="divFormJsonOut">
        <div class="panel-heading">Invoke neo nodes (JSON RPC)</div>
    </div>
</div>

```

```

<div class="panel-body">
  <form id="frmRPCJson" action="">
    <div class="row">
      <div class="col-md-12">
        <div class="form-group">
          <label for="rpcMethod">Method (view <a href="https://docs.neo.c
          <select name="rpcMethod" id="rpcMethod" class="custom-select fc
            <option title="select">Select a method</option>
            <option title=":Gets the number of blocks in the main chain">
            <option title="<address> :      Export the private key of the s
            <option title="<address> :      Check account asset informatior
            <option title="<txid>[verbose=0]: Returns the contract log ba
            <option title="<asset_id> :      Query asset information accorc
            <option title="<asset_id> : Returns the balance of the corres
            <option title=": Gets the hash of the tallest block in the ma
            <option title="<hash> [verbose=0]:      Returns the correspondi
            <option title="<index> [verbose=0]:      Returns the correspor
            <option title="<index>:      Returns the hash value of the cor
            <option title="<index>:      Returns the system fees before th
            <option title=": Gets the current number of connections for t
            <option title="<script_hash>:      Returns information about t
            <option title=": Create a new address (Need to open the walle
            <option title="<number_of_blocks> <height_to_end (optional)>
            <option title=": Get a list of unconfirmed transactions in me
            <option title="<txid> [verbose=0]:      Returns the corresponc
            <option title="<script_hash> <key>:      Returns the stored va
            <option title="<txid>:      Returns the corresponding transact
            <option title="<txid> <n>:      Returns the corresponding trar
            <option title=": Get a list of nodes that are currently conne
            <option title=": Returns the current NEO consensus nodes info
            <option title=": Get version information of this node">getver
            <option title="<script_hash> <params>:      Invokes a smart cor
            <option title="<script_hash> <operation> <params>:      Invokes
            <option title="<script>:      Runs a script through the virtua
            <option title=": Lists all the addresses in the current walle
            <option title=": Lists all the plugins loaded in the RPC node
            <option title="<hex>:      Broadcast a transaction over the ne
            <option title="<asset_id> <address> <value> [fee=0]:      Trar
            <option title="<outputs_array> [fee=0] [change_address]:
            <option title="<hex>:      Submit new blocks (Needs to be a cc
            <option title="<address>:      Verify that the address is a cor
            <option title=": create your invoke function manually!"></opt
          </select>
        </div>
      </div>
    </div>
    <div class="col-md-12">
      <div class="form-group">
        <label for="txtRPCJson">RPC Json Model</label>
        <textarea id="txtRPCJson" rows="1" cols="100" class="form-contr
      </div>
    </div>
  </form>
  <form id="frmRPCJson2" action="">
    <div class="row">
      <div class="col-md-12">

```

```

        <div class="form-group">
            <label for="txtRPCJsonOut">Output RPC</label>
            <textarea id="txtRPCJsonOut" rows="10" cols="100" class="form-c
        </div>
    </div>
    <div class="col-md-12">
        <button type="submit" id="btnCallJsonRPC" class="btn btn-success k
        <button type="button" style="margin-left: 5px; margin-bottom: 5px
    </div>
</div>
</form>
<div id="divAppLogNotifications">
    <textarea id="txt_notifications" readonly="readonly" rows="5" cols=
</div>
</div>
</div>
<!-- end rawRPC -->

</div> <!-- TAB PANNEL -->
<!-- END DIV MAIN CONTENTS - EVERYTHING WAS INSIDE HERE - II-->
<!-- ===== -->
<!-- ===== -->

```

```

<script>
drawRelayedTXs();
window.onbeforeunload = function(event) {
    storePageData();
    //socket.disconnect();
    return confirm("Confirm refresh");
}

//Check current webpage and starts with localhost or neocompiler public website
$(document).ready(function() {
    // recoveringPageData(); //deprecated
    //Function that verifies if page is localhost or neocompiler.io, changing neocli
    if (window.location.href.indexOf("localhost") != -1) {
        $("#ecolabnetworkurlselection")[0].selectedIndex = 1;
    }
    updatePATHS();
    //This should be after Paths update
    automaticNeoCliNodeSelection();

```

```

    updateAllWalletData();
    updateEcoMetadataInfo();

    //The first call is done here as default
    $("#compilebtn")[0].disabled = true;
    updateCompilersSelectionBox("docker-mono-neo-compiler");

    // Setting default code example to csharp and hello world (example 0)
    aceEditor = createEditor("aceEditor", "ace/mode/csharp");
    setExample("csharp", '0');
});

function createEditor(name, mode) {
    var editor = ace.edit(name);
    editor.setTheme("ace/theme/clouds");
    editor.session.setMode(mode);
    editor.setAutoScrollEditorIntoView(true);
    editor.setOptions({
        enableBasicAutocompletion: true,
        enableSnippets: true,
        enableLiveAutocompletion: false,
        maxLines: 30
    });
    editor.commands.addCommand({
        name: "showKeyboardShortcuts",
        bindKey: {
            win: "Ctrl-Alt-h",
            mac: "Command-Alt-h"
        },
        exec: function(editor) {
            ace.config.loadModule("ace/ext/keybinding_menu", function(module) {
                module.init(editor);
                //editor.showKeyboardShortcuts() just shows when tigger the hotkey
            })
        }
    })
    return editor
}

//=====

function setWalletRefreshInterval(intervalForRefreshing)
{
    if(refreshIntervalWalletId!=0)
        clearInterval(refreshIntervalWalletId);
    $("#cbx_refresh_wallet").val(intervalForRefreshing);

    refreshIntervalWalletId = setInterval(function() {
        populateAllWalletData();
        searchForTXs();
        stopRefreshingDueToConnectionProblems();
    }, intervalForRefreshing);
}

function setEcoInfoRefreshInterval(intervalForRefreshing)
{
    if(refreshIntervalEcoMetadataStatsId!=0)

```

```

        clearInterval(refreshIntervalEcoMetadataStatsId);
        $("#cbx_refresh_ecometadata").val(intervalForRefreshing);

        refreshIntervalEcoMetadataStatsId = setInterval(function() {
            updateEcoMetadataInfo();
        }, intervalForRefreshing);
    }

function setCompilersByPing(intervalForRefreshing)
{
    if(refreshIntervalCompilers!=0)
        clearInterval(refreshIntervalCompilers);
        $("#cbx_refresh_compilers").val(intervalForRefreshing);

        refreshIntervalCompilers = setInterval(function() {
            pingCompilers();
        }, intervalForRefreshing);
    }

function setIntervalAutomaticNeoCliNodeWithRandomSample(intervalForRefreshing)
{
    if(refreshIntervalNeoCliNodes!=0)
        clearInterval(refreshIntervalNeoCliNodes);
        $("#cbx_refresh_neocli").val(intervalForRefreshing);

        refreshIntervalNeoCliNodes = setInterval(function() {
            automaticNeoCliNodeSelection();
            // Automatic node selection may also find a good node
            stopRefreshingDueToConnectionProblems();
        }, intervalForRefreshing);
    }

function setGenesisBlock(intervalForRefreshing)
{
    if(refreshGenesisBlock!=0)
        clearInterval(refreshGenesisBlock);
        $("#cbx_refresh_gen").val(intervalForRefreshing);

        refreshGenesisBlock = setInterval(function() {
            genesisBlockTransfer("AZ81H31DMWzbSnFDLFkzh9vHwaDLayV7fU", "AK2nJJpJr6o664CWJKi
        }, intervalForRefreshing);
    }

function updateEcoMetadataInfo() {
    //$("#span_numberCompilations")[0].innerHTML = "Total compile calls: " + numberCc
    //$("#span_numberDeploys")[0].innerHTML = "Total deploy calls: " + numberDeploys;
    //$("#span_numberInvokes")[0].innerHTML = "Total invoke calls: " + numberInvokes;
    $("#span_numberCompilations")[0].innerHTML = "";
    $("#span_numberDeploys")[0].innerHTML = "";
    $("#span_numberInvokes")[0].innerHTML = "";
}

function updateAutomaticNodeSelectionFlag() {
    if ($("#cbx_automatic_node_best_height")[0].checked) {
        AUTOMATIC_PIC_CSHARP_NODE_BEST_HEGITH = true;
    } else {

```

```

    AUTOMATIC_PIC_CSHARP_NODE_BEST_HEGITH = false;
}
}

function pingCompilers() {
    if ($("#cbx_automatic_compiler")[0].checked) {
        var availableCompilers = $("#selectionBoxCompilers")[0].length;
        var bestCompiler = $("#selectionBoxCompilers")[0][0].value;
        var bestPing = 1000000000;
        for (var i = 0; i < availableCompilers; i++) {
            var compilerToPing = $("#selectionBoxCompilers")[0][i].value;

            //console.log('Going to ping ' + compilerToPing + '...');
            ping(compilerToPing).then(function(response) {
                //console.log('Ping time was ' + String(response.delta) + ' ms');
                if (response.delta < bestPing) {
                    bestPing = response.delta;
                    bestCompiler = response.url;
                    if (bestCompiler != BASE_PATH_COMPILERS) {
                        //console.log('Going to update to bestCompiler ' + bestCompiler);
                        updateAllCompilerRelatedVariables(bestCompiler);
                    }
                }
                //console.log('bestCompiler ' + bestCompiler + ' with ' + bestPing +
            }).catch(function(err) {
                console.error('Could not ping remote URL', err);
            });
        }
    } //automatic compiler checkbox
}

//=====
//===== BEGIN ON CHANGE FUNCTIONS =====
$("#cbx_fullHistory").on('change', function() {
    if ($("#cbx_fullHistory")[0].checked) {
        FULL_ACTIVITY_HISTORY = true;
    } else {
        FULL_ACTIVITY_HISTORY = false;
    }
    console.log("New FULL_ACTIVITY_HISTORY flag is " + FULL_ACTIVITY_HISTORY);
});

//=====
//Neoscan selection box on loading
if (ENABLE_NEOSCAN_TRACKING) {
    $("#cbx_enable_neoscan")[0].checked = true;
} else {
    $("#cbx_enable_neoscan")[0].checked = false;
}
//onchange
$("#cbx_enable_neoscan").on('change', function() {
    if ($("#cbx_enable_neoscan")[0].checked) {
        ENABLE_NEOSCAN_TRACKING = true;
    } else {
        ENABLE_NEOSCAN_TRACKING = false;
    }
}

```

```

        updateNeoScanLinkIfEnable();
        drawRelayedTXs();
        console.log("New ENABLE_NEOSCAN_TRACKING flag is " + ENABLE_NEOSCAN_TRACKING);
    });
    //=====

$("##ecolabnetworkurlselection").on('change', function() {
    //$("##neonodeurl")[0].value = this.value;
    updatePATHS();
    startSocketIoConnections();
});

$("##createtx_input_utxo2").on('change', function() {
    console.log("input UTXO tx ID is: " + JSON.parse($("##createtx_input_utxo2").val());
});

/*
    $("##createtx_input_assetid").on('change', function() {
        verifyRPCNetworkForWithdrawUTXO();
    });
*/
/*
//$("##createtx_selRPCNetwork").on('change', function() {
//    verifyRPCNetworkForWithdrawUTXO();
//});

/*
    $("##createtx_input_utxo").on('change', function() {
        inputbox = document.getElementById("createtx_input_utxo");
        value = inputbox.options[inputbox.selectedIndex].value;
        $("##createtx_input_utxo_value")[0].value = value;
        $("##createtx_output_values").val($("##createtx_input_utxo_value")[0].value+"\n");
        updateOutputValues();
    })
*/

//=====
//===== BEGIN CONFIGUCATION SELECTION BOX ON CHANGE =====
function updateAllCompilerRelatedVariables(url) {
    BASE_PATH_COMPILERS = url;
    $("##compilers_path")[0].value = url;

    updateCompilersSelectionBox("docker-mono-neo-compiler");
}

$("##selectionBoxCompilers").on('change', function() {
    updateAllCompilerRelatedVariables(this.value);
});

$("##selectionBoxEcoServices").on('change', function() {
    BASE_PATH_ECOSERVICES = this.value;
    $("##eco_services_path")[0].value = this.value;
});

$("##selectionBoxRPCNodes").on('change', function() {
    BASE_PATH_CLI = this.value;
    $("##rpc_nodes_path")[0].value = this.value;

```

```

});

$("#selectionBoxNeoScan").on('change', function() {
    BASE_PATH_NEOSCAN = this.value;
    $("#neoscan_path")[0].value = this.value;
});

$("#cbx_refresh_wallet").on('change', function() {
    setWalletRefreshInterval(this.value);
});
$("#cbx_refresh_headers").on('change', function() {
    setRefreshingIntervalForHeadersInfo(this.value);
});
$("#cbx_refresh_compilers").on('change', function() {
    setCompilersByPing(this.value);
});
$("#cbx_refresh_neocli").on('change', function() {
    setIntervalAutomaticNeoCliNodeWithRandomSample(this.value);
});
$("#cbx_refresh_gen").on('change', function() {
    setGenesisBlock(this.value);
});
$("#cbx_refresh_ecometadata").on('change', function() {
    setEcoInfoRefreshInterval(this.value);
});

/*
//DEPRECATED SELECTION FOX FOR PYTHON REST SELECTION
$("#selectionBoxRestNotifications").on('change', function() {
    BASE_PATH_PY_REST = this.value;
    $("#rest_notifications_path")[0].value = this.value;
});
*/
//=====

//===== END ON-CHANGE FUNCTIONS =====
//=====

function updateBASEPATHS() {
    BASE_PATH_COMPILERS = $("#compilers_path")[0].value;
    BASE_PATH_ECOSERVICES = $("#eco_services_path")[0].value;
    BASE_PATH_CLI = $("#rpc_nodes_path")[0].value;
    BASE_PATH_NEOSCAN = $("#neoscan_path")[0].value;
    //BASE_PATH_PY_REST = $("#rest_notifications_path")[0].value;

    updateNeoScanLinkIfEnable();
}

function updateNeoScanLinkIfEnable() {
    if (ENABLE_NEOSCAN_TRACKING) {
        mynetworkIndex = $("#ecolabnetworkurlselection")[0].options[($("#ecolabnetwork"
        $("#headerNeoScanLink")[0].innerHTML = '<i class="fa fa-th-large" aria-hidden="true">
        //neocompiler eco
        if (mynetworkIndex == 0)
            $("#headerNeoScanLink")[0].innerHTML += "NeoScan-SharedPrivatenet";
    }

```



```

//local
if (mynetworkIndex == 1)
    $("#headerNeoScanLink")[0].innerHTML += "NeoScan-PrivateNet";

// MainNet
if (mynetworkIndex == 2)
    $("#headerNeoScanLink")[0].innerHTML += "NeoScan-Mainnet";

// Tesnet
if (mynetworkIndex == 3)
    $("#headerNeoScanLink")[0].innerHTML += "NeoScan-Testnet";

// CozTestnet TODO
if (mynetworkIndex == 4)
    $("#headerNeoScanLink")[0].innerHTML += "NeoScan-CozNet";

document.getElementById("headerNeoScanLink").setAttribute("target", "_blank")
document.getElementById("headerNeoScanLink").setAttribute("href", BASE_PATH_N
} else {
    document.getElementById("headerNeoScanLink").setAttribute("href", null);
    $("#headerNeoScanLink")[0].innerHTML = "";
}
}

//=====
//===== ADD OPTION TO SELECTION BOX =====
function addOptionToSelectionBox(textToOption, valueToOption, walletSelectionBox) {
    var option = document.createElement("option");
    option.text = textToOption;
    option.value = valueToOption;
    var select = document.getElementById(walletSelectionBox);
    select.appendChild(option);
}
//=====

function updateCompilersSelectionBox(compilerType) {
    $.get(BASE_PATH_COMPILERS + '/getcompilers',
        function(data) {
            compilerSelectionBoxID = "compilers_versions-selection-box";
            compilerSelectionBoxObj = document.getElementById(compilerSelectionBoxID)
            compilerSelectionBoxObj.options.length = 0;

            for (c = 0; c < data.length; c++) {
                if (data[c].compiler == compilerType)
                    addOptionToSelectionBox(data[c].version, compilerType + ":" + dat
            }
            $("#compilebtn")[0].disabled = false;
            //Select the latest as default
            //compilerSelectionBoxObj.selectedIndex = (compilerSelectionBoxObj.length
        },
        "json" // The format the response should be in
    );
}
//=====
//= UPDATE ALL KNOWN SERVICE TO THEIR CONFIGURATION SELECTION =

```

```

function addAllKnownServiceToSelectionBox(serviceType, networkServicesObj, selectionE
    //Clear selection box
    document.getElementById(selectionBoxToFill).options.length = 0;
    for (var kn = 0; kn < networkServicesObj.length; kn++) {
        var serviceUrlToAdd = getServiceURLByTypeAndNetwork(serviceType, networkServi

        if (serviceUrlToAdd !== '')
            addOptionToSelectionBox(serviceUrlToAdd, serviceUrlToAdd, selectionBoxToF
    }
}
//=====
function stopRefreshingDueToConnectionProblems(){
    if(NUMBER_FAILS_REQUESTS > MAX_N_FAILED_REQUEST_UNTIL_STOP)
    {
        //Check if it was already disabled
        if(!NEO_CLI_REFRESHING_STOPED)
        {
            console.log(NUMBER_FAILS_REQUESTS + "/" + MAX_N_FAILED_REQUEST_UNTIL_STOP);
            console.log("Time to disable refreshing...");
            // store the last refreshing rate for wallet
            NEO_CLI_REFRESHING_STOPED = true;
            $("#cbx_refresh_wallet").val(0);
            if(refreshIntervalWalletId!=0)
                clearInterval(refreshIntervalWalletId);

            if(refreshHeadersNeoCliNeoScan!=0)
                clearInterval(refreshHeadersNeoCliNeoScan);
        }
    }else {
        if(NEO_CLI_REFRESHING_STOPED)
        {
            console.log(NUMBER_FAILS_REQUESTS + "/" + MAX_N_FAILED_REQUEST_UNTIL_STOP);
            console.log("Recovering automatic refreshing for balances and headers");
            setWalletRefreshInterval(10000);
            setRefreshingIntervalForHeadersInfo(800); //0.8s
            NEO_CLI_REFRESHING_STOPED = false;
        }
    }
}

function updatePATHS() {
    mynetworkIndex = $("#ecolabnetworkurlselection")[0].options[$("#ecolabnetworkurls
    stopAllEcoLabFrontEndTimeoutIntervals();
    NEO_CLI_REFRESHING_STOPED = false;

    // Eco shared privatenet
    if (mynetworkIndex == 0) {
        $("#cbx_enable_neoscan")[0].checked = true;
        ENABLE_NEOSCAN_TRACKING = true;
        $("#cbx_query_neoscan")[0].checked = false;
        setWalletRefreshInterval(10000); //10000 millisec
        setEcoInfoRefreshInterval(10000);
        setRefreshingIntervalForHeadersInfo(1300); //1.3s
        $("#cbx_refresh_headers").val(1300); // Headers refreshing is declared at hea
        setCompilersByPing(20000);
        setIntervalAutomaticNeoCliNodeWithRandomSample(10000);
    }
}

```

```

        setGenesisBlock(10000); //set genesis block for both sharedprivatenet (neoc

        addAllKnownServiceToSelectionBox("ecocompilers", ecoNodes, "selectionBoxCompi
        addAllKnownServiceToSelectionBox("ecoservices", ecoNodes, "selectionBoxEcoSer
        addAllKnownServiceToSelectionBox("RPC", ecoNodes, "selectionBoxRPCNodes");
        addAllKnownServiceToSelectionBox("neoscan", ecoNodes, "selectionBoxNeoScan");
    }

    // Localhost services
    if (mynetworkIndex == 1) {
        $("#cbx_enable_neoscan")[0].checked = false;
        ENABLE_NEOSCAN_TRACKING = false;
        $("#cbx_query_neoscan")[0].checked = false;
        setWalletRefreshInterval(10000); //10000 millisec
        setEcoInfoRefreshInterval(10000);
        setRefreshingIntervalForHeadersInfo(800); //0.8s
        $("#cbx_refresh_headers").val(800);
        setCompilersByPing(25000);
        setIntervalAutomaticNeoCliNodeWithRandomSample(10000);
        setGenesisBlock(10000); //set genesis block for both sharedprivatenet (neoc

        addAllKnownServiceToSelectionBox("ecocompilers", localHostNodes, "selectionBc
        addAllKnownServiceToSelectionBox("ecoservices", localHostNodes, "selectionBox
        addAllKnownServiceToSelectionBox("RPC", localHostNodes, "selectionBoxRPCNodes
        addAllKnownServiceToSelectionBox("neoscan", localHostNodes, "selectionBoxNeoS

    }

    // MainNet
    if (mynetworkIndex == 2) {
        // TODO - Automatic enable neoscan until NeoCli clients are running plugin UT
        $("#cbx_enable_neoscan")[0].checked = true;
        ENABLE_NEOSCAN_TRACKING = true;
        $("#cbx_query_neoscan")[0].checked = true;
        setWalletRefreshInterval(15000); //10000 millisec
        setEcoInfoRefreshInterval(15000);
        setRefreshingIntervalForHeadersInfo(10000); //10s
        $("#cbx_refresh_headers").val(10000);
        setCompilersByPing(30000);
        setIntervalAutomaticNeoCliNodeWithRandomSample(15000);

        addAllKnownServiceToSelectionBox("ecocompilers", ecoNodes, "selectionBoxCompi
        addAllKnownServiceToSelectionBox("ecoservices", ecoNodes, "selectionBoxEcoSer
        addAllKnownServiceToSelectionBox("RPC", mainNetNodes, "selectionBoxRPCNodes")
        addAllKnownServiceToSelectionBox("neoscan", mainNetNodes, "selectionBoxNeoSc
        //addAllKnownServiceToSelectionBox("RESTNotifications",mainNetNodes,"selectic

    }

    // Tesnet
    if (mynetworkIndex == 3) {
        // TODO - Automatic enable neoscan until NeoCli clients are running plugin UT
        $("#cbx_enable_neoscan")[0].checked = true;
        ENABLE_NEOSCAN_TRACKING = true;
        $("#cbx_query_neoscan")[0].checked = true;
        setWalletRefreshInterval(15000); //10000 millisec
        setEcoInfoRefreshInterval(15000);
        setRefreshingIntervalForHeadersInfo(10000); //10s

```

```

        $("#cbx_refresh_headers").val(10000);
        setCompilersByPing(30000);
        setIntervalAutomaticNeoCliNodeWithRandomSample(15000);

        addAllKnownServiceToSelectionBox("ecocompilers", ecoNodes, "selectionBoxCompilers");
        addAllKnownServiceToSelectionBox("ecoservices", ecoNodes, "selectionBoxEcoServices");
        addAllKnownServiceToSelectionBox("RPC", testNetNodes, "selectionBoxRPCNodes");
        addAllKnownServiceToSelectionBox("neoscan", testNetNodes, "selectionBoxNeoScan");
        //addAllKnownServiceToSelectionBox("RESTNotifications",mainNetNodes,"selectionBoxRestNotifications");
    }

    // CozTestnet TODO
    if (mynetworkIndex == 4) {
        // TODO - Automatic enable neoscan until NeoCli clients are running plugin UT
        $("#cbx_enable_neoscan")[0].checked = true;
        ENABLE_NEOSCAN_TRACKING = true;
        $("#cbx_query_neoscan")[0].checked = true;
        setWalletRefreshInterval(15000); //10000 millisec
        setEcoInfoRefreshInterval(15000);
        setRefreshingIntervalForHeadersInfo(10000); //10s
        $("#cbx_refresh_headers").val(10000);
        setCompilersByPing(30000);
        setIntervalAutomaticNeoCliNodeWithRandomSample(15000);

        addAllKnownServiceToSelectionBox("ecocompilers", ecoNodes, "selectionBoxCompilers");
        addAllKnownServiceToSelectionBox("ecoservices", ecoNodes, "selectionBoxEcoServices");
        addAllKnownServiceToSelectionBox("RPC", cozNetNodes, "selectionBoxRPCNodes");
        addAllKnownServiceToSelectionBox("neoscan", cozNetNodes, "selectionBoxNeoScan");
        //addAllKnownServiceToSelectionBox("RESTNotifications",mainNetNodes,"selectionBoxRestNotifications");
    }

    $("#compilers_path")[0].value = $("#selectionBoxCompilers")[0].value;
    $("#eco_services_path")[0].value = $("#selectionBoxEcoServices")[0].value;
    $("#rpc_nodes_path")[0].value = $("#selectionBoxRPCNodes")[0].value;
    $("#neoscan_path")[0].value = $("#selectionBoxNeoScan")[0].value;
    //$("#rest_notifications_path")[0].value = $("#selectionBoxRestNotifications")[0].value;

    updateBASEPATHS();
    populateAllWalletData();
}

function getCurrentNetworkNickname() {
    mynetworkIndex = $("#ecolabnetworkurlselection")[0].options[$("#ecolabnetworkurlselection").selectedIndex].value;
    if (mynetworkIndex == 0)
        return "SharedPrivateNet";
    if (mynetworkIndex == 1)
        return "LocalPrivateNet";
    if (mynetworkIndex == 2)
        return "MainNet";
    if (mynetworkIndex == 3)
        return "TestNet";
}

function goToWalletCentral() {
    $('a.nav-pills a[data-target="#wallet"]').tab('show');
}

```

```

// =====
// ===== COOKIES TEST =====
function recoveringPageData() {
    /*
    // DEPRECATED AFTER ANGULAR ROUTES TO TABS
    if (checkCookie("lastactivetab")) {
        index = getCookie("lastactivetab");
        //console.log("Cookie tests. Recovering active tab...");
        if(window.location.hash.slice(10).length == 0)
            $("#pillstab").children().eq(index).find('a').tab('show');
    }
    */
}

function storePageData() {
    /*
    // DEPRECATED AFTER ANGULAR ROUTES TO TABS
    setCookie("lastactivetab", $("#ul#pillstab li.active").index(), 1);
    */
}
// =====

function str2hex() {
    var i = 0;
    $("#strhex_hex").val(Neon.u.str2hexstring($("#strhex_str").val()));
    $("#hex_length").val(($("#strhex_hex").val().length / 2).toString());
};

function hex2str() {
    $("#strhex_str").val(Neon.u.hexstring2str($("#strhex_hex").val()));
    $("#hex_length").val(($("#strhex_hex").val().length / 2).toString());
};

function hex2xeh() {
    $("#hexxeh_xeh").val(revertHexString($("#hexxeh_hex").val()));
    $("#hex_length").val($("#strhex_hex").val().length.toString());
};

function xeh2hex() {
    $("#hexxeh_hex").val(revertHexString($("#hexxeh_xeh").val()));
};

function num2fixed8() {
    const a = new Neon.u.Fixed8(parseInt($("#fx8_number").val()));
    $("#fx8_hex").val(a.toHex());
};

function fixed82num() {
    const a = new Neon.u.Fixed8.fromHex($("#fx8_hex").val());
    $("#fx8_number").val(a);
};

function address2scripthash() {
    if ($("#address").val() == "") return;
    $("#scripthash").val(fromBase58($("#address").val()));
};

```

```

}

function conv_string_hash() {
    $("#convert_string_hash256")[0].value = Neon.u.hash256($("#convert_string_hash")[
    $("#convert_string_hash160")[0].value = Neon.u.hash160($("#convert_string_hash")[
    $("#convert_string_sha256")[0].value = Neon.u.sha256($("#convert_string_hash")[0]
}

function conv_bigint2bebytearray() {
    valbigint = Number($("#convert_bigint")[0].value);
    //console.log("converting int:"+valbigint);
    $("#convert_bytearray")[0].value = bigint2bebytearray(valbigint);
}

function conv_bebytearray2bigint() {
    vbytearray = $("#convert_bytearray")[0].value;
    //console.log("converting int:"+valbigint);
    $("#convert_bigint")[0].value = "" + behex2bigint(vbytearray);
}

function scripthash2address() {
    if ($("#scripthash").val() == "") return;
    var b58 = toBase58($("#scripthash").val());
    if (b58 == "")
        b58 = "BAD FORMAT! Expects 20 bytes!";
    $("#address").val(b58);
    //CreateRawTx();
}

function frmRPCJson() {
    if ($("#txtRPCJson").val() == "Select a method")
        $("#txtRPCJson").val("");
    else
        $("#txtRPCJson").val("{ \"jsonrpc\": \"2.0\", \"id\": 5, \"method\": \"\" + $(
};

var createtx_balance = null;

function verifyRPCNetworkForWithdrawUTXO() {
    myaddress = ($("#createtx_from_address")[0].value);
    console.log('getBalance: for: ' + myaddress);
    if (myaddress != "") {
        setNeonApiProvider(getCurrentNetworkNickname());
        NEON_API_PROVIDER.getBalance(myaddress).then(res => updateUTXOCapture(res));
    }
}

function updateUTXOCapture(res) {
    if (res.address == "not found") {
        alert("address not found!");
        console.log(JSON.stringify(res));
        return;
    }
    createtx_balance = res;
    assetBox = ($("#createtx_input_assetid")[0];
    asset = assetBox.options[assetBox.selectedIndex].text;

```

```

//alert("asset:"+this.value+asset);

//inputbox = document.getElementById("createtx_input_utxo");
//for(i = inputbox.options.length - 1 ; i >= 0 ; i--)
//  inputbox.remove(i);

inputbox2 = document.getElementById("createtx_input_utxo2");
while (inputbox2.options.length > 0)
  inputbox2.remove(0);

//console.log(createtx_balance);

if (asset == "GAS") {
  for (i = 0; i < createtx_balance.assets.GAS.unspent.length; ++i) {
    option = document.createElement("option");
    option.text = "utxo: " + elipsisMiddle(createtx_balance.assets.GAS.unspent[i]);
    option.value = JSON.stringify(createtx_balance.assets.GAS.unspent[i]);
    //console.log(JSON.parse(option.value).txid);
    //console.log(JSON.parse(option.value).index);
    //console.log(JSON.parse(option.value).value);
    inputbox2.add(option);
  }
}
if (asset == "NEO") {
  for (i = 0; i < createtx_balance.assets.NEO.unspent.length; ++i) {
    option = document.createElement("option");
    option.text = "utxo: " + elipsisMiddle(createtx_balance.assets.NEO.unspent[i]);
    option.value = JSON.stringify(createtx_balance.assets.NEO.unspent[i]);
    //inputbox.add(option);
    inputbox2.add(option);
  }
}

$("#createtx_output_addresses").val("");
$("#createtx_output_values").val("");
updateOutputValues();
}

function ConvertJsonNotifications() {
  //console.log(jsonObj);
  if ($("#txtRPCJsonOut").val() != "") {
    var jsonObj = JSON.parse($("#txtRPCJsonOut").val());
    console.log(jsonObj);

    if (jsonObj.length >= 1 && jsonObj[0].result && jsonObj[0].result.executions)
      var myNotify = "";
      var notifArray = [];

      notifArray = jsonObj[0].result.executions[0].notifications;

      console.log("notifications:" + JSON.stringify(notifArray));
      for (var nSize = 0; nSize < notifArray.length; nSize++) {
        var jsonNotificationsObj = notifArray[nSize];
        myNotify += "Notification " + nSize + " typeI: " + jsonNotificationsObj.type +
        myNotify += " typeII: " + jsonNotificationsObj.state.value[0].type +
        myNotify += " value(hex): " + jsonNotificationsObj.state.value[0].value
      }
    }
  }
}

```

```

        myNotify += " value(string): " + hex2bin(jsonNotificationsObj.state.v
    }
    $("#txt_notifications").val(myNotify);
} else {
    $("#txt_notifications").val("Notifications were not found!");
}
} else {
    $("#txt_notifications").val("empty RPC call output to check notification");
}
}

$("#frmRPCJson2").submit(function(e) {
    e.preventDefault(); // Prevents the page from refreshing
    $("#txtRPCJsonOut").val("");
    //curl -s -X POST http://localhost:30335 -H 'Content-Type: application/json' -d '
    $.post(
        BASE_PATH_CLI, // Gets the URL to sent the post to
        $("#txtRPCJson").val(), // Serializes form data in standard format
        function(data) {
            $("#txtRPCJsonOut").val(JSON.stringify(data, null, ' '));
            ConvertJsonNotifications();
        },
        "json" // The format the response should be in
    ).fail(function() {
        $("#txtRPCJsonOut").val("failed to invoke network!");
    }); //End of POST for search
});

// =====
// ===== CREATE RAW UTXO WITHDRAW TRANSACTION BEGINS =====
// 14 blocks

function create_withdrawtx() {
    /*
        TODO: all the blocks that doesn't match with the expected size must do what is c
        if (value > 0xffffffff) write 8 bytes hex
        else if (value > 0xffff) write 4 bytes hex
        else if (value > 0xfc) write 2 bytes hex
        else write 1 byte hex
    */

    wtx = createtx_getWithoutWitness();

    if (wtx.length % 2 == 1) {
        console.log("Problems were detected with WTX lenght, pehaps, in terms of an c
        console.log("length: " + wtx.length);
        return;
    }

    var lscripthash = [];
    lscripthash.push($("#createtx_from_scripthash")[0].value); // IGOR

    var script_other = "";

```



```

if ($("#createtx_script_attributes")[0].value.trim().length > 0) { // if hash Scri
    var otherscript = ($("#createtx_script_attributes")[0].value);
    var shash = getScriptHashFromAVM(otherscript);
    lscripthash.push(shash);
    script_other += int2hex(0); // no invocation (2 bytes)
    // script_other += int2hex(1);
    //script_other += "51";
    script_other += uint2bytes(otherscript.length / 2);
    script_other += otherscript;

    if (script_other.length % 2 == 1) {
        console.log("Problems were detected with WTX OTHER SCRIPT");
        console.log("length: " + script_other.length);
        return;
    }
}

var wtxverif = "";

var hexavmverif = ($("#createtx_from_verification").val()); // invocation
hexavmverif = hexavmverif.replace(/(\r\n|\n|\r)/gm, "");
/*block 11*/
wtxverif += int2hex(hexavmverif.length / 2); // size of invocation script (8 byte
/*block 12*/
wtxverif += hexavmverif; // invocation script
console.log("will finish wallet script");
var hexavm = ($("#createtx_from_contract").val());
hexavm = hexavm.replace(/(\r\n|\n|\r)/gm, "");
/*block 13*/
wtxverif += uint2bytes(hexavm.length / 2); // size of verification script (8 byte
console.log("hexavm is:" + hexavm.length / 2 + " with last length:" + uint2bytes(
/*block 14*/
wtxverif += hexavm; // verification script

if (wtxverif.length % 2 == 1) {
    console.log("Problems were detected with WTX verification!");
    console.log("length: " + wtxverif.length);
    return;
}

/*block 10*/
//if($("#createtx_script_attributes")[0].value.trim().length > 0) // if hash Scri
// wtx += "02"
//else
// wtx += "01"; //witness count ( number of scripts) (8 bytes) can be reduced TC
wtx += int2hex(lscripthash.length); // 01 or 02

if (wtx.length % 2 == 1) {
    console.log("Problems were detected with WTX lenght, ERROR in WITNESS COUNT");
    console.log("length: " + wtx.length);
    return;
}

```

```

    if (lscipthash.length == 1) { // only single witness
        wtx += wtxverif;
        console.log("adding verification script: " + wtxverif);
    } else if (lscipthash.length == 2) {
        // check order
        if (lscipthash[0] < lscipthash[1]) {
            wtx += wtxverif;
            wtx += script_other;
            console.log("adding verification script: " + wtxverif);
            console.log("adding other script: " + script_other);
            console.log(lscipthash);
        } else {
            wtx += script_other;
            wtx += wtxverif;
            console.log("adding other script: " + script_other);
            console.log("adding verification script: " + wtxverif);
            console.log(lscipthash);
        }
    }
}

/*block end*/
wtx = wtx.toLowerCase();
console.log(wtx);
if (wtx.length % 2 == 1) {
    console.log("Problems were detected with WTX lenght, pehaps, in terms of an c
    console.log("length: " + wtx.length);
    return;
}
console.log(wtx.length / 2 + " bytes");

$("#createtx_txhex").val(wtx);
//$("#createtx_txhash").val(Neon.u.hash256($("#createtx_txhex").val()));
}

// =====
// =====

function createtx_sign() {
    $('#createtx_from_verification').val("00");
    //if(!document.getElementById("createtx_signature").checkValidity()) { // allow z
    if (document.getElementById("createtx_signature").value.length != 64) {
        alert("signature must be hex string of 32 bytes");
        return;
    }
    if (!$("#frm_createtx_withdraw")[0].checkValidity()) {
        $("#frm_createtx_withdraw").find(':submit').click(); // invoke form validatic
        return;
    }
}

wtx = createtx_getWithoutWitness();
var wtxHash = "0x" + revertHexString(Neon.u.hash256(wtx));
$("#createtx_txhash").val(wtxHash);
console.log("transaction (" + wtxHash + ") " + wtx);
//alert('oi'+wtx);
$('#createtx_from_verification').val('40' + Neon.wallet.generateSignature(wtx, $(
}

```

```

// =====
// =====

function createtx_getWithoutWitness() {
  /*block 0*/
  wtx = "80"; // tx type (1 byte)
  /*block 1*/
  wtx += "00"; //tx version (1 byte)

  var lattribs = $("#createtx_attributes")[0].value.trim().split("\n");
  if (lattribs[0] == "")
    lattribs.pop();
  var numattribs = lattribs.length;
  if (numattribs > 255) {
    console.log("WARNING: too many attributes! Limit is 255 (1 byte)")
    numattribs = 0;
  }

  console.log("Transaction with " + numattribs + " attributes");

  /*block 2*/
  wtx += int2hex(numattribs); //"00"; // number of attributes (1 byte)

  if (wtx.length % 2 == 1) {
    console.log("Problems were detected with WTX lenght, probably ATTRIBUTE COUNT")
    return;
  }

  // add attributes
  for (var i = 0; i < numattribs; i++) {
    console.log("attribute " + i + ":" + lattribs[i].substr(0, 2));
    wtx += lattribs[i]; // attribute code (1byte) + data
  }

  if (wtx.length % 2 == 1) {
    console.log("Problems were detected with WTX lenght, probably SOME ATTRIBUTE!")
    return;
  }

  outexpected = 0;
  intxlist = $("#createtx_input_utxo2").val();
  for (var i = 0, size = intxlist.length; i < size; i++) {
    var item = JSON.parse(intxlist[i]).value;
    outexpected += Number(item);
  }

  console.log(intxlist.length + " inputs");
  /*block 3*/
  wtx += int2hex(intxlist.length); // number of inputs (1 byte)
  for (i = 0; i < intxlist.length; i++) {
    /*block 4*/
    wtx += revertHexString(JSON.parse(intxlist[i]).txid); //input hash (32 bytes)
    /*block 5*/
    wtx += revertHexString(int2hex(JSON.parse(intxlist[i]).index, 4)); //input index (4 bytes)
  }
}

```

```

outtxvalues = $("#createtx_output_values").val();
outtxlist = outtxvalues.split("\n");
outtxlist = outtxlist.filter(function(n) {
    return n != ""
});
outtotal = 0;
for (var i = 0, size = outtxlist.length; i < size; i++) {
    var item = outtxlist[i];
    num = Number(item);
    if (num < 0) {
        console.log("warning; negative number not allowed as output! " + num + "
        num = 0;
    }
    outtotal += num;
}
//console.log("outtotal: "+outtotal);

outtxvalues = $("#createtx_output_values").val();
outtxlist = outtxvalues.split("\n");
outtxlist = outtxlist.filter(function(n) {
    return n != ""
});
outtxaddr = $("#createtx_output_addresses").val();
outtxlist2 = outtxaddr.split("\n");
outtxlist2 = outtxlist2.filter(function(n) {
    return n != ""
});
outtotal = 0;
for (var i = 0, size = outtxlist.length; i < size; i++) {
    var item = outtxlist[i];
    num = Number(item);
    if (num <= 0) {
        console.log("output value must be positive! got " + num + "... aborting!"
        return;
    }
    outtotal += num;
}

change = 0;
if ($("#createtx_cbx_change")[0].checked)
    change = outexpected - outtotal;

if ((outtxlist.length != outtxlist2.length) || (outtotal + change != outexpected)
    //alert("error building withdraw transaction. verify inputs and outputs.");
    //return;
    if (!window.confirm("inputs and outputs dont match! may become network fee"))
        return;
}

console.log(outtxlist.length + (change > 0) + " outputs");
/*block 6*/
wtx += int2hex(outtxlist.length + (change > 0)); // number of outputs (including
if (wtx.length % 2 == 1) {
    console.log("Problems were detected with WTX lenght, pehaps, in terms of an c
    return;

```

```

    }

    for (var i = 0, size = outtxlist.length; i < size; i++) {
        /*block 7*/
        wtx += revertHexString($("#createtx_input_assetid")[0].value); //asset type (
        const txvalue = new Neon.u.Fixed8(Number(outtxlist[i]));
        /*block 8*/
        wtx += revertHexString(txvalue.toHex()); //value of the asset (8 bytes)
        /*block 9*/wtx += revertHexString(outtxlist2[i]); // address to send the ass
        if (Neon.wallet.isAddress(outtxlist2[i]))
            wtx += revertHexString(Neon.wallet.getScriptHashFromAddress(outtxlist2[i]
        else {
            wtx += revertHexString(outtxlist2[i]); // writes scripthash in little-enc
        }
        if (wtx.length % 2 == 1) {
            console.log("Problems were detected with WTX lenght, pehaps, in terms of
            return;
        }
    }
}
if (change > 0) {
    wtx += revertHexString($("#createtx_input_assetid")[0].value); //asset type (
    const txvalue = new Neon.u.Fixed8(Number(change));
    wtx += revertHexString(txvalue.toHex());
    wtx += revertHexString($("#createtx_from_scripthash")[0].value);
}
console.log(wtx);
if (wtx.length % 2 == 1) {
    console.log("Problems were detected with WTX lenght, pehaps, in terms of an c
    return;
}
return wtx;
}
// =====
// =====

$("#frm_createtx_withdraw").submit(function(e) {
    e.preventDefault(); // Prevents the page from refreshing
    //create_withdrawtx();
    wtx = $("#createtx_txhex").val();
    sendRawTXToTheRPCNetwork(wtx);
});
// ===== CREATE RAW UTXO WITHDRAW TRANSACTION FINISHES=====
// =====

$("#createtx_from_contract").on('keyup', function() {
    var hexavm = $("#createtx_from_contract").val();
    hexavm = hexavm.replace(/(\r\n|\n|\r)/gm, "");
    var shash = getScriptHashFromAVM(hexavm);
    $("#createtx_from_scripthash")[0].value = shash;
    $("#createtx_from_address")[0].value = toBase58(shash);
});

function updateOutputValues() {
    outtxvalues = $("#createtx_output_values").val();
    outtxlist = outtxvalues.split("\n");
    outtxlist = outtxlist.filter(function(n) {
        return n != ""
    });
}

```

```

});
outtotal = 0;

for (var i = 0, size = outtxlist.length; i < size; i++) {
    var item = outtxlist[i];
    num = Number(item);
    if (num < 0) {
        console.log("warning; negative number not allowed as output! " + num + "
        num = 0;
    }
    outtotal += num;
}
//console.log("outtotal: "+outtotal);

outexpected = 0;
intxlist = $("#createtx_input_utxo2").val();
for (var i = 0, size = intxlist.length; i < size; i++) {
    var item = JSON.parse(intxlist[i]).value;
    outexpected += Number(item);
}

if (outtotal > outexpected)
    $("#createtx_output_total").text("Total: " + outtotal + " (OUTPUT EXCEEDED: "
else {
    if ($("#createtx_cbx_change")[0].checked) {
        $("#createtx_output_total").text("Total OK: " + outtotal + " (CHANGE: " +
        var wtx = createtx_getWithoutWitness();
        var wtxHash = "0x" + revertHexString(Neon.u.hash256(wtx));
        $("#createtx_txhash").val(wtxHash);
    } else
        $("#createtx_output_total").text("Total: " + outtotal + (outtotal != oute
    }
}

$("#createtx_output_values").on('keyup', function() {
    updateOutputValues();
});

$("#createtx_from_verification").on('keyup', function() {
    var hexavm = $("#createtx_from_verification").val();
    hexavm = hexavm.replace(/(\r\n|\n|\r)/gm, "");
    $("#createtx_from_verification_opcodes").val("");
    $("#createtx_from_verification_opcodes").text("" + (hexavm.length / 2) + " bytes\
    printOpcode(hexavm, $("#createtx_from_verification_opcodes"));
});

//This uses local routes
function getNodeInfo() {
    $.get(BASE_PATH_ECOSERVICES + '/statusnode/1', function(data) {
        $("#node1data").val(data.replace(/^[^x00-\x7F]/g, ""));
        $("#node1data").scrollTop($("#node1data")[0].scrollHeight);
    });
    $.get(BASE_PATH_ECOSERVICES + '/statusnode/2', function(data) {
        $("#node2data").val(data.replace(/^[^x00-\x7F]/g, ""));
        $("#node2data").scrollTop($("#node2data")[0].scrollHeight);
    });
}

```

```

$.get(BASE_PATH_ECOSERVICES + '/statusnode/3', function(data) {
    $("#node3data").val(data.replace(/[\x00-\x7F]/g, ""));
    $("#node3data").scrollTop($("#node3data")[0].scrollHeight);
});
$.get(BASE_PATH_ECOSERVICES + '/statusnode/4', function(data) {
    $("#node4data").val(data.replace(/[\x00-\x7F]/g, ""));
    $("#node4data").scrollTop($("#node4data")[0].scrollHeight);
});
$.get(BASE_PATH_ECOSERVICES + '/statusnode/0', function(data) {
    $("#noderpcdata").val(data.replace(/[\x00-\x7F]/g, ""));
    $("#noderpcdata").scrollTop($("#noderpcdata")[0].scrollHeight);
});
}

// btnfunction for calling CN set time from form
function onClickBtnSetCNBlockTime() {
    setConsensusNodesBlockTime($("#cbx_spb_cn").val(), ($("#cbx_pwd_cn").val()));
}

// Set CN blockTime
function setConsensusNodesBlockTime(blockTime, pwd) {
    $.get(BASE_PATH_ECOSERVICES + '/setconsensusnodesblocktime/1/' + blockTime + '/' +
        console.log(data)
    });
    $.get(BASE_PATH_ECOSERVICES + '/setconsensusnodesblocktime/2/' + blockTime + '/' +
        console.log(data)
    });
    $.get(BASE_PATH_ECOSERVICES + '/setconsensusnodesblocktime/3/' + blockTime + '/' +
        console.log(data)
    });
    $.get(BASE_PATH_ECOSERVICES + '/setconsensusnodesblocktime/4/' + blockTime + '/' +
        console.log(data)
    });
}

function onClickPostServerToResetDockerService() {
    postServerToResetDockerService($("#cbx_pwd_reset_service").val());
}

function postServerToResetDockerService(pwd) {
    if(getCurrentNetworkNickname() == "SharedPrivateNet")
    {
        $.get(BASE_PATH_ECOSERVICES + '/resetdockerservice/' + pwd, function(data) {
            console.log(data);
        });
    }else{
        console.error("Reset only works for SharedPrivateNet EcoService.")
    }
}

//Route for getting incremental storage from node with statedumper
function getIncrementalStorageInfo() {
    var heightToGet = ($("#heightForIncStorage").val());
    $.get(BASE_PATH_ECOSERVICES + '/incstorage/' + heightToGet, function(data) {
        $("#textAreaOfIncrementalStorage").val(data);
        $("#textAreaOfIncrementalStorage").scrollTop($("#textAreaOfIncrementalStorage

```

```

    });
}

function setExample(language, selected_index) {
    //console.log("selecting example:" + selected_index);
    aceEditor.getSession().setValue("");
    getfile(language, selected_index, 0);
}

function getfile(language, selected_index, index = 0) {
    var vExamples = cSharpFiles;
    if(language === "python")
        vExamples = cPythonFiles;
    if(language === "java")
        vExamples = cJavaFiles;
    if(language === "golang")
        vExamples = cGolangFiles;

    var numfiles = vExamples[selected_index].length;
    if (index < numfiles) {
        var file = vExamples[selected_index][index];
        //console.log("getting example file: " + file);
        $.get(file, function(data) {
            aceEditor.getSession().setValue(aceEditor.getSession().getValue() + data);
            getfile(language, selected_index, index + 1);
        });
    }
}

//=====
//===== SEND TX FUNCTIONS FROM FORM =====
//=====
function createSendTxForm() {
    var addrFromIndex = $("#createtx_from")[0].selectedOptions[0].index;
    var addrToIndex = $("#createtx_to")[0].selectedOptions[0].index;
    var addrToValue = ECO_WALLET[addrToIndex].account.address;
    if (ECO_WALLET[addrFromIndex].account.isMultiSig) {
        var jsonArrayWithPrivKeys = getMultiSigPrivateKeys(addrFromIndex);
        //Multi-sig address
        createTxFromMSAccount(addrFromIndex, addrToValue, $("#createtx_NEO").val(), $
    } else {
        //TODO add verification for watch only
        createTxFromAccount(addrFromIndex, addrToValue, $("#createtx_NEO").val(), $
    }
};

function createGasTxForm() {
    var addrFromIndex = $("#createtx_from")[0].selectedOptions[0].index;
    if (ECO_WALLET[addrFromIndex].account.isMultiSig) {
        var jsonArrayWithPrivKeys = getMultiSigPrivateKeys(addrFromIndex);
        //Multi-sig address
        createClaimMSGasTX(addrFromIndex, jsonArrayWithPrivKeys, getCurrentNetworkNic
    } else {
        //TODO add verification for watch only
        createClaimGasTX(addrFromIndex, BASE_PATH_CLI, getCurrentNetworkNickname());
    }
};

```



```

    }
};
//=====
//===== FORMS =====
//=====
$("#formCompile").submit(function(e) {
    $("#compilebtn")[0].disabled = true;
    $("#code_cs").val("");
    $("#code_python").val("");
    $("#code_golang").val("");
    $("#code_java").val("");
    if ($("#csharpcode.tab-pane.active.cont").length != 0) {
        $("#code_cs").val(ace.edit("aceEditor").getSession().getValue());
        console.log("Compiling C# code...");
    }
    if ($("#pythoncode.tab-pane.active.cont").length != 0) {
        $("#code_python").val(ace.edit("aceEditor").getSession().getValue());
        console.log("Compiling Python code...");
    }
    if ($("#golangcode.tab-pane.active.cont").length != 0) {
        $("#code_golang").val(ace.edit("aceEditor").getSession().getValue());
        console.log("Compiling Golang code...");
    }
    if ($("#javacode.tab-pane.active.cont").length != 0) {
        $("#code_java").val(ace.edit("aceEditor").getSession().getValue());
        console.log("Compiling Java code...");
    }
}

$("#codewarnerr").val("Remote compiling at "+BASE_PATH_COMPILERS+"...");
$("#codeavm").val("");
$("#opcodes").val("");
$("#codeabi").val("");
$("#invokehashjs").val("");
$("#contracthashjs").val("");
$("#contractparams").val("\n\n");
e.preventDefault(); // Prevents the page from refreshing
var $this = $(this); // `this` refers to the current form element
var indata = $(this).serialize();
$.post(
    BASE_PATH_COMPILERS + $this.attr("action"), // Gets the URL to sent the post
    indata, // Serializes form data in standard format
    function(data) {
        console.log("finished compiling");
        $("#compilebtn")[0].disabled = false;
        var coderr = atob(data.output);
        $("#codewarnerr").val(coderr);
        var hexcodeavm = atob(data.avm);
        $("#codeavm").val(hexcodeavm);
        hexcodeavm = hexcodeavm.replace(/(\r\n|\n|\r)/gm, "");
        $("#opcodes").val("");
        //printOpcode(hexcodeavm, $("#opcodes"));
        //console.log("GRAVANDO BINARIO: "+typeof(datacontent)+"-"+datacontent);
        localStorage.setItem('avmFile', hexcodeavm); //, {type: 'application/octet
        //datacontent = localStorage.getItem('avmFile', {type: 'application/octet
        //console.log("LENDO BINARIO: "+typeof(datacontent)+"-"+datacontent);
        //console.log(localStorage.getItem('avmFile').charCodeAt(0));
    }
);

```

```

        $("#btn_download")[0].style = "";

        //filling hashes
        var contractScriptHash = getScriptHashFromAVM(hexcodeavm);
        $("#contractInfo_ScriptHash")[0].value = contractScriptHash;
        $("#contractInfo_Address")[0].value = toBase58(contractScriptHash);
        updateScriptHashesBoxes(contractScriptHash);

        // -----
        // calculating extra fee according to AVM size
        var fixedAVGTxSize = 500; //TODO
        var avmSize = Math.ceil(hexcodeavm.length/2) + fixedAVGTxSize;
        $("#contractInfo_AVMSize")[0].value = avmSize;
        // TODO - Create a call on neo-plugins SimplePolicy
        var freeSize = 1024;
        var feePerExtraByte = 0.00001;
        var extraGasNetFeeForBigTx = 0;
        if (avmSize > freeSize)
            extraGasNetFeeForBigTx = (avmSize-freeSize)*feePerExtraByte;
        $("#attachDeployGasFeeJS")[0].value = extraGasNetFeeForBigTx;
        // -----

        // Loading all ABI related boxes
        if(data.abi != "")
        {
            var codeabi = atob(data.abi);
            console.log(codeabi);
            updateAllABIDependencies(JSON.parse(codeabi));
        }
    },
    "json" // The format the response should be in
); //End of POST for Compile

//Send info to EcoServices for counting number of compilations
$.post(
    BASE_PATH_ECOSERVICES + "/compileCounter", // Gets the URL to sent the post t
    null, // Serializes form data in standard format
    function(data) {},
    "json" // The format the response should be in
); //End of POST for Compile counter

}); //End of form Compile function
//=====

//=====
//Invoke JS
$("#forminvokejs").submit(function(e) {
    $("#contractmessages").text("");
    $("#contractmessagesnotify").text("");
    e.preventDefault(); // Prevents the page from refreshing
    var $this = $(this); // `this` refers to the current form element
    var indata = $(this).serialize();

    var wI = $("#wallet_invokejs")[0].selectedOptions[0].index;
    var attachgasfeejs = Number($("#attachgasfeejs").val());
    var attachsystemgasjs = Number($("#attachSystemgasjs").val());

```

```

var attachneojis = Number($("#attachneojis").val());
var attachgasjs = Number($("#attachgasjs").val());
var invokeScripthash = $("#invokehashjs").val();

var invokefunc = "";
// INVOKE function is already passed as a String parameter!!

//if($("#invokefunctionjs")[0].value != "Main")
//    invokefunc = $("#invokefunctionjs")[0].value;

var neonJSParams = [];
if($("#invokeparamsjs")[0].value != "")
    neonJSParams = JSON.parse($("#invokeparamsjs")[0].value);

InvokeFromAccount(wI, attachgasfeejs, attachsystemgasjs, attachneojis, attachgasjs

//Send info to EcoServices for counting number of invokes
$.post(
    BASE_PATH_ECOSERVICES + "/invokecounter", // Gets the URL to sent the post to
    null, // Serializes form data in standard format
    function(data) {},
    "json" // The format the response should be in
); //End of POST for Compile counter
}); //End of invoke function
//=====

//=====
//Deploy JS
$("#formdeployjs").submit(function(e) {
    //$("#contractmessages").text("");
    //$("#contractmessagesnotify").text("");
    e.preventDefault(); // Prevents the page from refreshing
    var $this = $(this); // `this` refers to the current form element
    var indata = $(this).serialize();
    var contractGasCost = 90;
    var wI = $("#wallet_deployjs")[0].selectedOptions[0].index;
    var storage = 0;

    var rT = $("#contractreturnjs").val();
    if ($("#contractreturnjs").val() == "f0" || $("#contractreturnjs").val() == "ff")
        rT += "00"; // ff becomes 'ff00' (biginteger representation)

    var params = $("#contractparamsjs").val();

    //console.log("Parameter list: "+params);
    //console.log("Return type: "+rT);

    var script = $("#codeavm").val().replace(/(\r\n|\n|\r)/gm, "");

    console.log("Deploying contract: '" + script + "' scripthash: '" + $("#contractha

    if ($("#cbx_storagejs")[0].checked) {
        contractGasCost += 400;
        storage += 1;
    }
}

```

```

    if ($("#cbx_dynamicinvokejs")[0].checked) {
        contractGasCost += 500;
        storage += 2;
    }

    if ($("#cbx_ispayablejs")[0].checked) {
        storage += 4;
    }

    console.log("Final attached gas should be:" + contractGasCost)

    var contract_name = $("#jsdeploy_name").val();
    var contract_desc = $("#jsdeploy_desc").val();
    var contract_email = $("#jsdeploy_email").val();
    var contract_author = $("#jsdeploy_author").val();
    var contract_version = $("#jsdeploy_version").val();

    //console.log(contract_name+contract_desc+contract_email+contract_author+contract

    var attachDeploygasfeejs = Number($("#attachDeployGasFeeJS").val());

    DeployFromAccount(wI, attachDeploygasfeejs, contractGasCost, BASE_PATH_CLI, getCu

    //Send info to EcoServices for counting number of deploys
    $.post(
        BASE_PATH_ECOSERVICES + "/deploycounter", // Gets the URL to sent the post to
        null, // Serializes form data in standard format
        function(data) {},
        "json" // The format the response should be in
    ); //End of POST for Compile counter

}); //End of deploy function
//=====

$("#formgetstorage").submit(function(e) {
    e.preventDefault(); // Prevents the page from refreshing
    //Clean previous storage values
    $("#gsf_contractvaluehex")[0].value = "";
    $("#gsf_contractvaluestr")[0].value = "";

    //Get the hash and key
    sthash = $("#gsf_contracthash")[0].value;
    stkey = $("#gsf_contractkey")[0].value;
    if ($("#getStorageFormat")[0].value == "String (no quotes)") {
        stkey = str2hexstring(stkey);
    }

    console.log("looking for storage key '" + stkey + "' at contract '" + sthash + "'");
    getStorage(sthash, stkey, $("#rpc_nodes_path")[0].value).then(function(data) {
        $("#gsf_contractvaluehex")[0].value = data.result;
        if (data.result != null)
            $("#gsf_contractvaluestr")[0].value = hexstring2str(data.result);
        else {
            $.post(
                $("#rpc_nodes_path")[0].value, // Gets the URL to sent the post to
                '{ "jsonrpc": "2.0", "id": 5, "method": "getcontractstate", "params":

```

```

        function(data2) {
            if (data2.result) { // contract exists
                if (data2.result.properties.storage)
                    $("#gsf_contractvaluehex")[0].value = "contract exists (k
                else
                    $("#gsf_contractvaluehex")[0].value = "contract exists (k
            } else if (data2.error.code == -100)
                $("#gsf_contractvaluehex")[0].value = "contract does not exist
            else {
                $("#gsf_contractvaluehex")[0].value = "some strange error happened
                console.log(data2);
            }
        },
        "json" // The format the response should be in
    ).fail(function() {
        $("#gsf_contractvaluehex")[0].value = "failed to invoke network!";
    }); //End of POST for search
    }
    });
}); //End of storage read function
//=====

$('#a[data-toggle="tab"]').on('shown.bs.tab', function(e) {
    //console.log(e.target.dataset.target)
    if ($("#csharpcode.tab-pane.active.cont").length != 0) {
        aceEditor.getSession().setMode("ace/mode/csharp");
        setExample("csharp", '0');
        updateCompilersSelectionBox("docker-mono-neo-compiler");
    }
    if ($("#pythoncode.tab-pane.active.cont").length != 0) {
        aceEditor.getSession().setMode("ace/mode/python");
        setExample("python", '0');
        updateCompilersSelectionBox("docker-neo-boa");
    }
    if ($("#golangcode.tab-pane.active.cont").length != 0) {
        aceEditor.getSession().setMode("ace/mode/golang");
        setExample("golang", '0');
        updateCompilersSelectionBox("docker-neo-go");
    }
    if ($("#javacode.tab-pane.active.cont").length != 0) {
        aceEditor.getSession().setMode("ace/mode/java");
        setExample("java", '0');
        updateCompilersSelectionBox("docker-java-neo-compiler");
    }
});

//=====
$("#btn_downloadOpCode").click(function(e) {
    e.preventDefault();
    mydata = $('#frm_disassembly').contents().find('#opcodes').val();
    console.log(mydata);
    var file = new Blob([mydata], {
        type: 'text/plain'
    });
    download(file, ".opcodes");
});

```

```
//=====

function moveLastTab(){
    $('.nav-pills a[data-target="#" + LAST_ACTIVE_TAB_BEFORE_ACTIVITY + "']").tab('show')
}
//=====
//===== END FORMS =====
//=====
</script>
```

