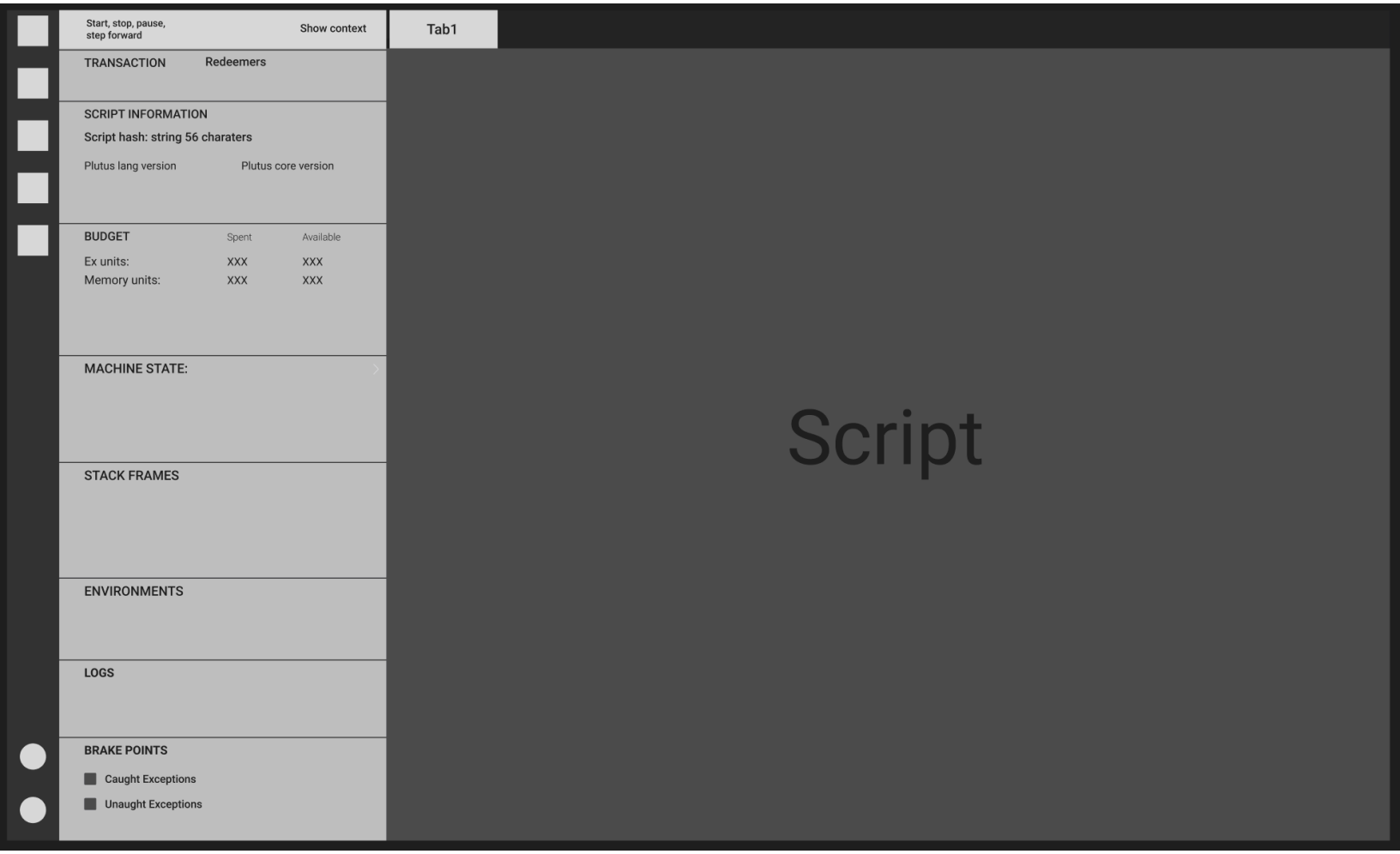
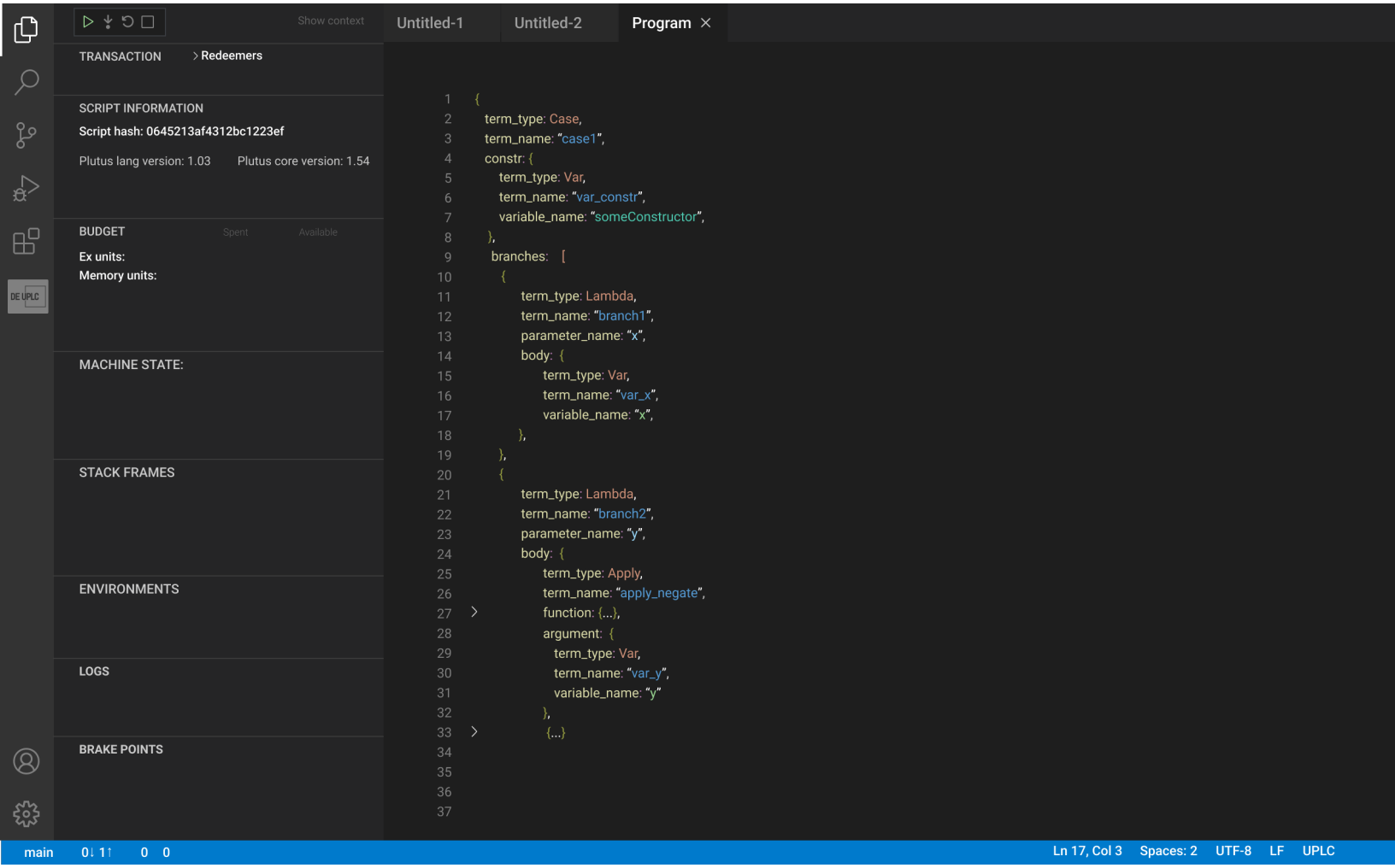
NOTE: you can always zoom it to see details.

Wireframe:
Initial idea of UI based in VS Code design



Mockups:
No active debugging example:



Left Sidebar

1. Debug Controls
   - Flow control buttons (Play, Step, Stop)
   - "Show context" toggle
2. Transaction Settings
   - Redeemer selector for script execution
3. Script Info
   - Hash
   - Plutus versions: lang and core
4. Resource Monitor
   - Ex units usage
   - Memory units usage
5. Machine State
   - Current execution state display
6. Stack Frames
   - Execution context list

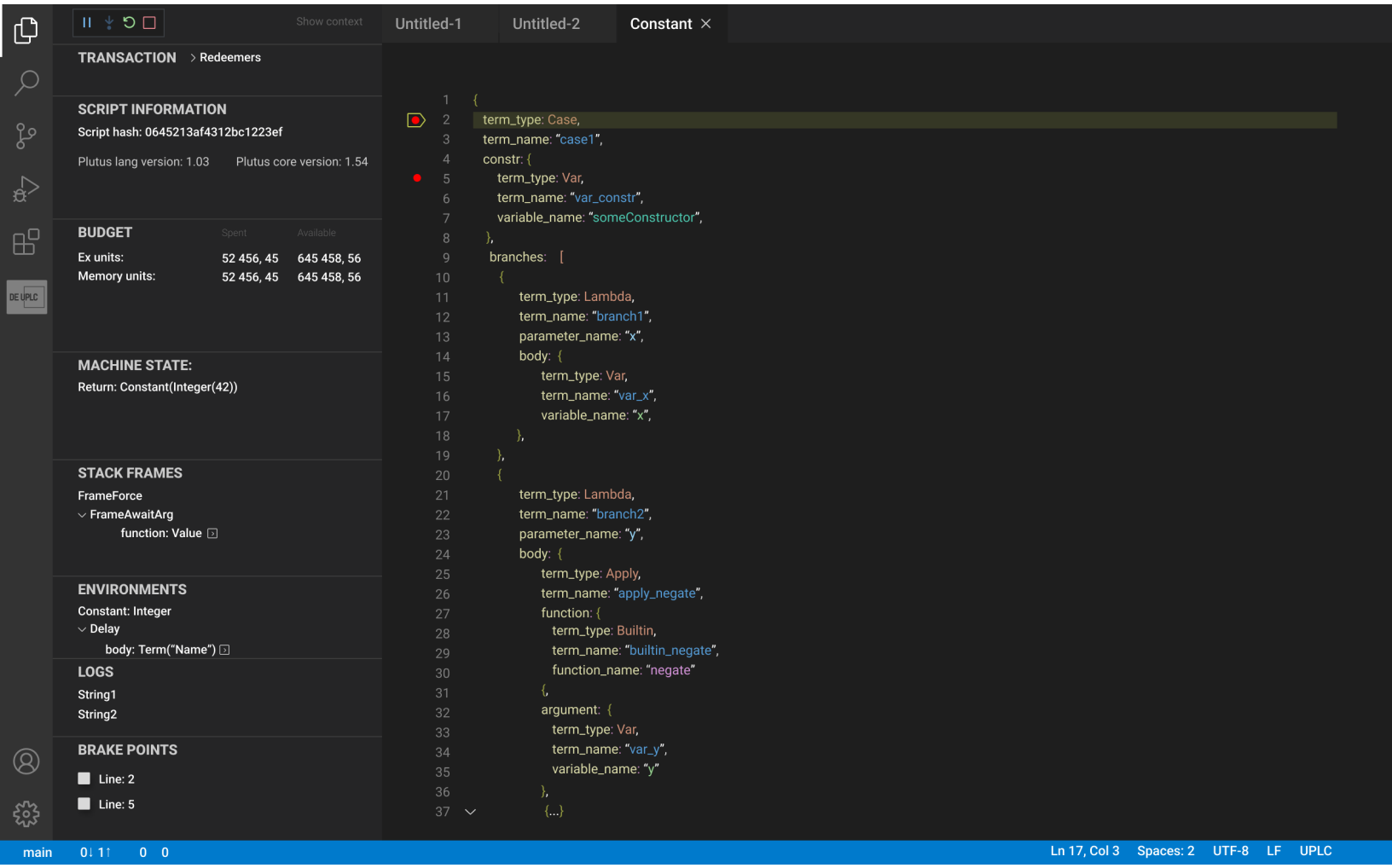
7. Environments
    - Variable values list
8. Logs
    - "trace" instruction outputs
9. Breakpoints
    - Debug stop points list

Main Area (Right)

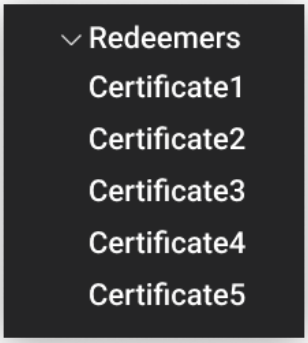
- UPLC code display with syntax highlighting
- Line numbers and current execution point
- Tab-based file navigation

DE-UPLC logo on in the left area.

Example for active debug session

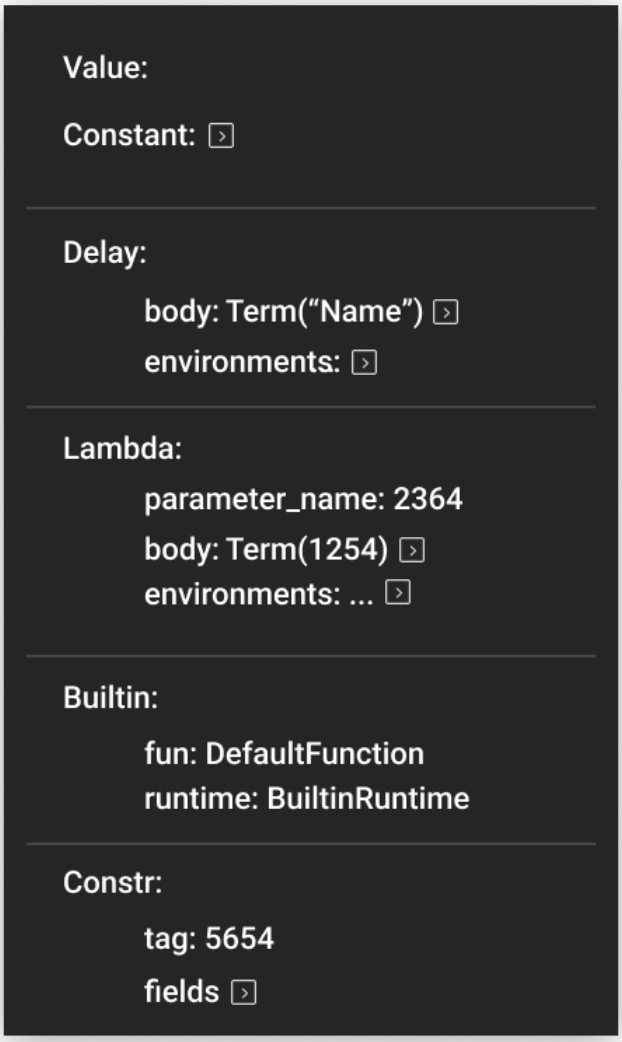


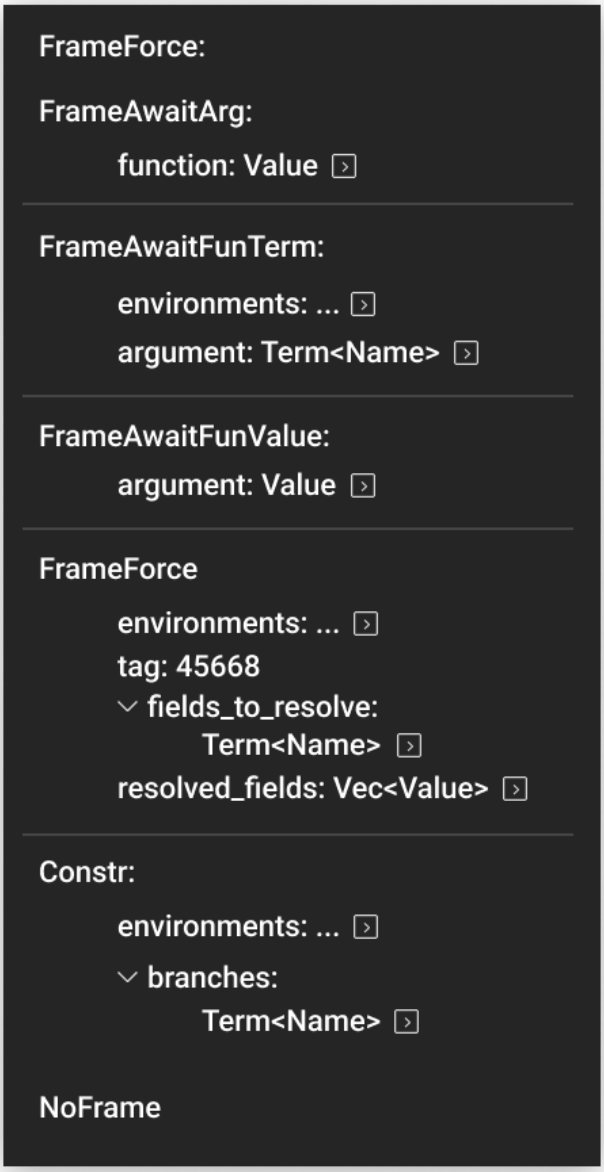
Example of Redeemers selector:



Next UI mockups are representation of most of components that we expect to see on left sidebar:

Note: this  button with arrow means that component has a data that will be opened by pressing the button. Data will be shown in separate tab in json-like representation.




FrameForce:
FrameAwaitArg:
function: Value
FrameAwaitFunTerm:
environments: ...
argument: Term<Name>
FrameAwaitFunValue:
argument: Value
FrameForce
environments: ...
tag: 45668
fields_to_resolve:
Term<Name>
resolved_fields: Vec<Value>
Constr:
environments: ...
branches:
Term<Name>
NoFrame

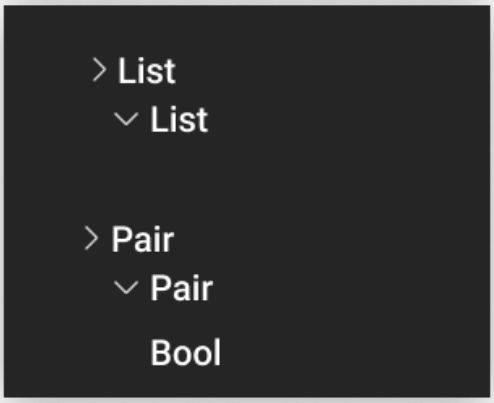
DefaultFunction:

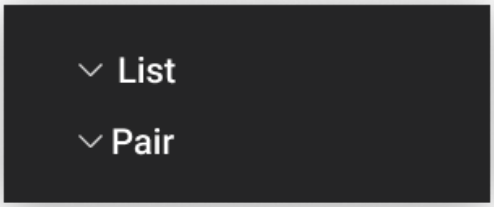
- AddInteger
- SubtractInteger
- MultiplyInteger
- DivideInteger
- QuotientInteger
- RemainderInteger
- ModInteger
- EqualsInteger
- LessThanInteger
- LessThanEqualsInteger

Type:

- Bool
- Integer
- String
- ByteString
- Unit
- List (Data)
- Pair (Bool, Intenger)
- Data
- Bls12_381G1Element
- Bls12_381G2Element
- Bls12_381MlResult

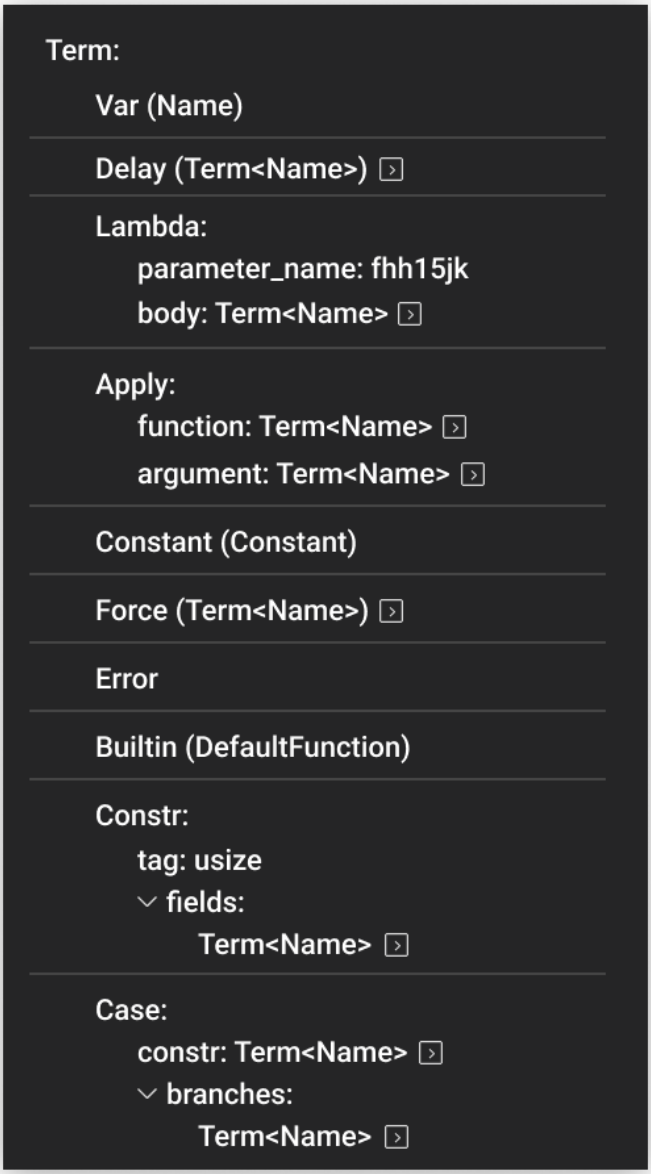

List
List
Data
Pair
Pair
Bool
Intenger
Bool


List
List
Pair
Pair
Bool


List
Pair

Constant:

- Integer (42)
- ByteString (0645213af4312bc1223ef)
- String (“kuggjkjkhkj”)
- UsignedInteger (67)
- Bool (true)
- ProtoList (Bool, Vec <Constant>)
- ProtoPair (String, Type, Constant, Constant)
- Data (PlutusData)
- Bls12_381G1Element
- Bls12_381G1Element
- Bls12_381MlResult


Term:
Var (Name)
Delay (Term<Name>)
Lambda:
parameter_name: fhh15jk
body: Term<Name>
Apply:
function: Term<Name>
argument: Term<Name>
Constant (Constant)
Force (Term<Name>)
Error
Builtin (DefaultFunction)
Constr:
tag: usize
fields:
Term<Name>
Case:
constr: Term<Name>
branches:
Term<Name>

Next UI mockups are representation of most of components that we expect to see on a tab with json-like representation:

Terms:

Var

```
{
  term_type: Var,
  term_name: “unique_term_name”,
  variable_name: “variable_name”
}
```

Force

```
{
  term_type: Force,
  term_name: “unique_term_name”,
  term: { /* Nested Term */ }
}
```

Delay

```
{
  term_type: Delay,
  term_name: “unique_term_name”,
  term: { /* Nested Term */ }
}
```

Error

```
{
  term_type: Error,
  term_name: “unique_term_name”,
}
```

Lambda

```
{
  term_type: Lambda,
  term_name: “unique_term_name”,
  parameter_name: “parameter_name“,
  body: { /* Nested Term */ }
}
```

Builtin

```
{
  term_type: Builtin,
  term_name: “unique_term_name”,
  function_name: “function_name”
}
```

Apply

```
{
  term_type: Apply,
  term_name: "unique_term_name",
  function: { /* Nested Term */ }
  argument: { /* Nested Term */ }
}
```

Constr

```
{
  term_type: Constr,
  term_name: "unique_term_name",
  tag: integer,
  fields: [ /* Array of Term */ ]
}
```

Constant

```
{
  term_type: Constant,
  term_name: "unique_term_name",
  value: /* Constant value */
}
```

Case

```
{
  term_type: Case,
  term_name: "unique_term_name",
  constr: { /* Constructor Term */ },
  branches: [ /* Array of Branch Term */ ]
}
```

Constants

Integer

```
{
  constant_type: Integer,
  value: "string_representation_of_bigint"
}
```

Bool

```
{
  constant_type: Bool,
  value: "true_or_false"
}
```

ByteString

```
{
  constant_type: ByteString,
  value: "hex_encoded_string"
}
```

ProtoList

```
{
  constant_type: ProtoList,
  element_type: { /* Type */ },
  elements: [ /* Array of Constants */ ]
}
```

String

```
{
  constant_type: String,
  value: “your_string_valuet”
}
```

Unit

```
{
  constant_type: Unit,
}
```

ProtoPair

```
{
  constant_type: ProtoPair,
  left_type: { /* Type */ },
  right_type: { /* Type */ },
  left_value: { /* Constant */ },
  right_value: { /* Constant */ }
}
```

Data

```
{
  constant_type: Data,
  value: { /* PlutusData */ }
}
```

BLS Constans:

Bls12_381G1Element

```
{
  constant_type: Bls12_381G1Element,
  value: { /* blst_p1 */ }
}
```

blst_p1

```
{
    x: { /* blst_fp */ },
    y: { /* blst_fp */ },
    z: { /* blst_fp */ }
}
```

Bls12_381G2Element

```
{
  constant_type: Bls12_381G2Element,
  value: { /* blst_p2 */ }
}
```

blst_fp

```
{
    l: [ 1, 3 ,66 , 99, 33, 123456,  ]
}
```

Bls12_381MlResult

```
{
  constant_type: Bls12_381MlResult,
  value: { /* blst_fp12 */ }
}
```

blst_fp2

```
{
    fp: [
      { /* blst_fp */ },
      { /* blst_fp */ }
    ]
}
```

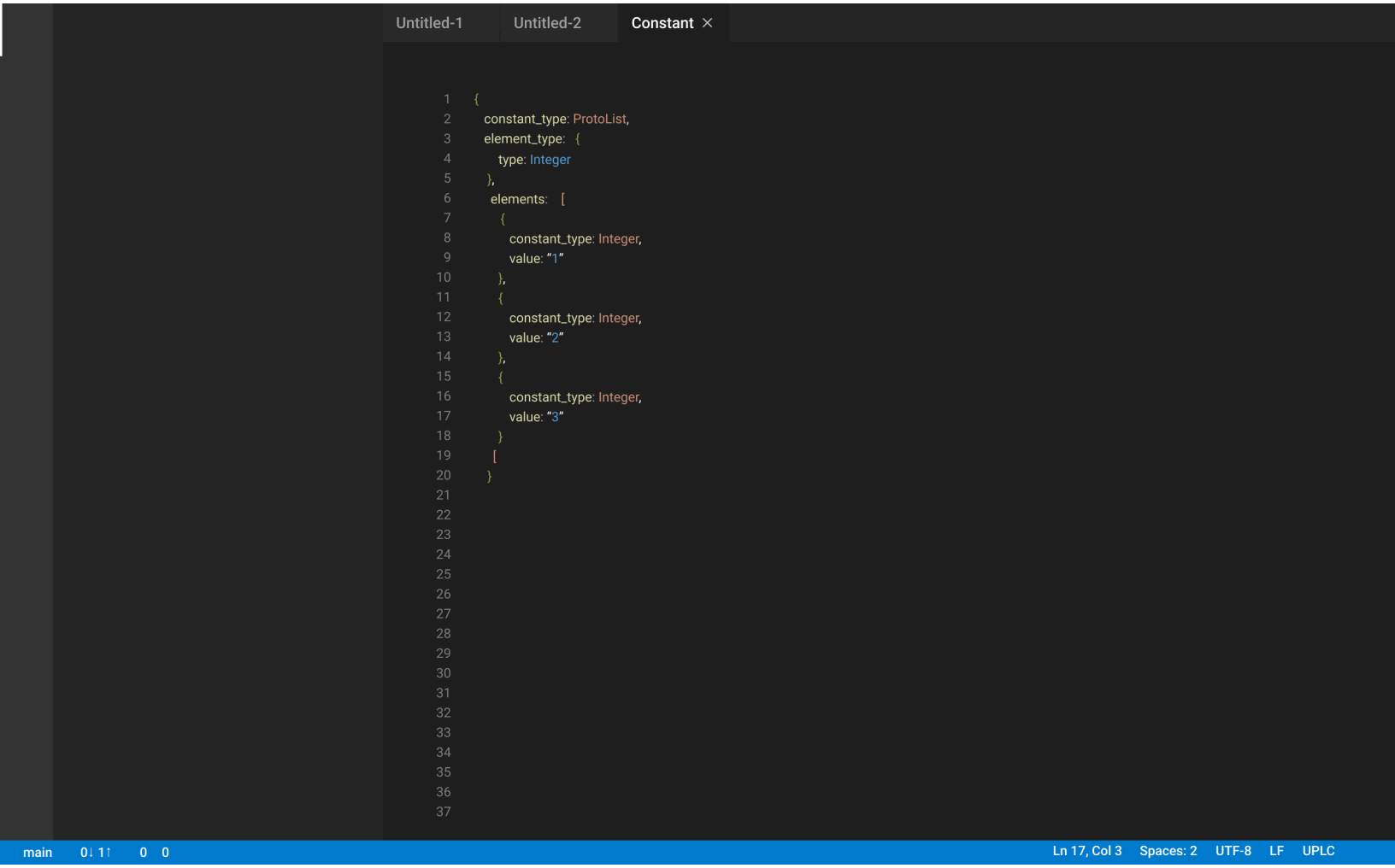
blst_p2

```
{
    x: { /* blst_fp2 */ },
    y: { /* blst_fp2 */ },
    z: { /* blst_fp2 */ }
}
```

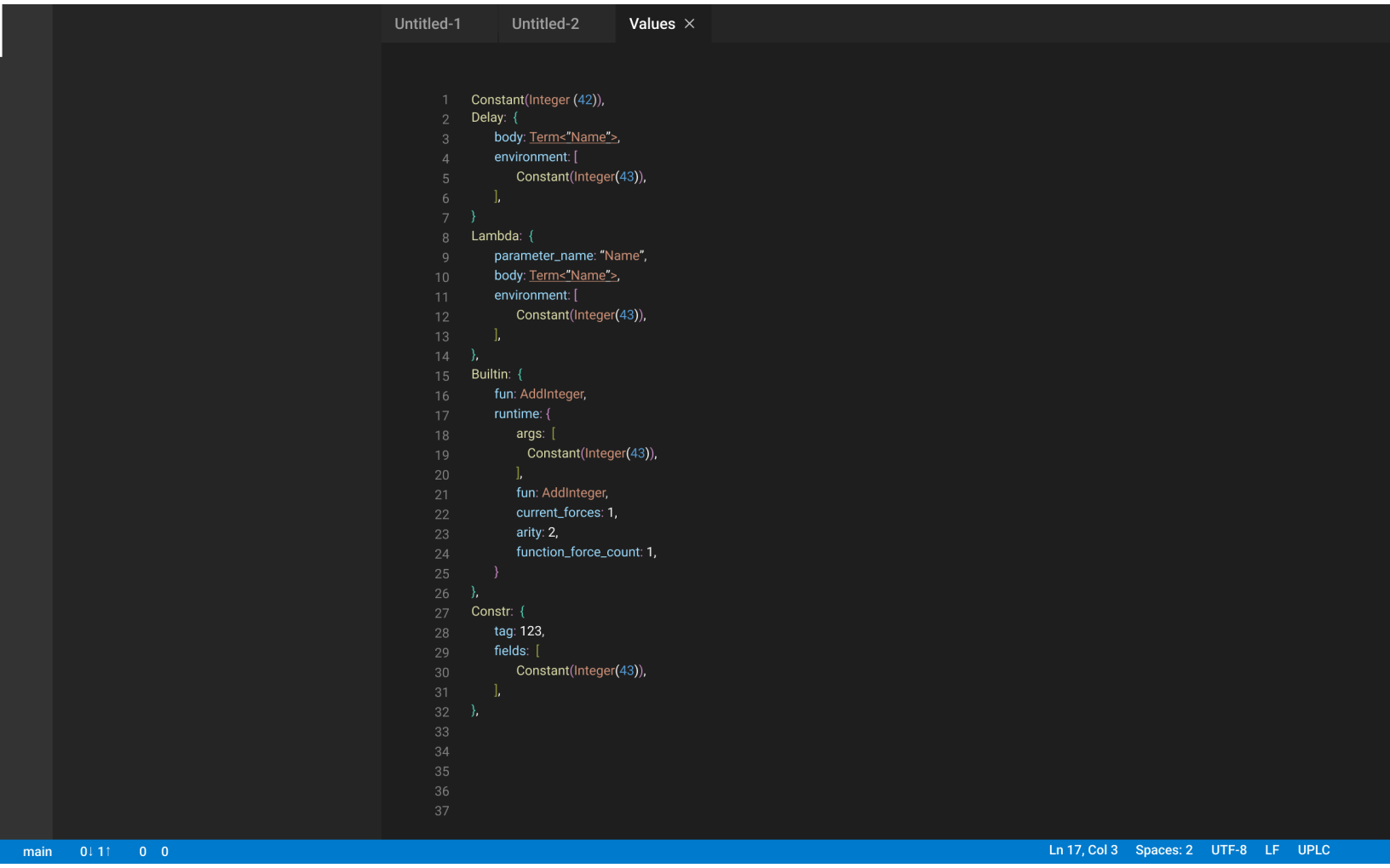
blst_fp6

```
{
    fp: [
      { /* blst_fp2 */ },
      { /* blst_fp2 */ },
      { /* blst_fp2 */ }
    ]
}
```

Example tab with constant:



Example tab with Value:



Example of script context:

Script context

```
 },
 inputs: [
  {
   out_ref: {
     tx_id: 1ab2c3d4e5f67890a1b2c3d4e5f67890a1b2c3d4e5f67890a1b2c3d4e5f6789
     output_index: 0,
   },
    resolved: {
      address: addr1qxy2k8hrkgeqmnapgx7jvg9rc0gkrmgstx4yjk9k9kxnvnq7m8a9fu9x7dzkg9amgvf2v5hz42zc7glf3kz7df4m4mass8nafx,
      value: {
        coin: 1000000
        assets: {
          b0d07d45fe9514f80213f4020e5a61241458be626841cde717cb38a76e7574686572756d: 50
          c0ffd7cb14cf0cba9c862d0aa6d0d96ceece28c9e5bf7e7f6a1d58d426974636f696e: 100
        },
      },
      datum_hash: ab12cd34ef56gh78ij90kl12mn34op56qr78st90uv12wx34yz56ab12cd34ef56
   },
  },
 ],
 outputs: [
  {
   address: addr1qx8phkx6acpnf7275k6vs3fkpzxk7zsxw2wuwqafy788txxm8a9fu9x7dzkg9amgvf2v5hz42zc7glf3kz7df4m4masfjx5p7,
   value: {
     coin: 500000
     assets: {
       b0d07d45fe9514f80213f4020e5a61241458be626841cde717cb38a76e7574686572756d: 25
     }
   },
   datum_hash: ff34de5678ab90cd12ef34gh56ij78kl90mn12op34qr56st78uv90wx12yz34ab56
  }
 ],
 fee: {
  coin: 180000
 },
 mint: {
  tokens: {
   d9312da562da182b02322fd8acb536f37eb9d29fba7c49dc12784d0f446e77546f6b656e: 1000
  }
 },
 certificates: [
  {
   type: StakeRegistration,
   stake_credential: e1f2a3b4c5d6e7f8a9b0c1d2e3f4a5b6c7d8e9f0a1b2c3d4e5f6a7b8c9d0e1f2
  }
 ],
 withdrawals: [
  [
   addr1qxy2k8hrkgeqmnapgx7jvg9rc0gkrmgstx4yjk9k9kxnvnq7m8a9fu9x7dzkg9amgvf2v5hz42zc7glf3kz7df4m4mass8nafx,
   1000000
  ]
 ],
 valid_range: {
    lower_bound: 41000000,
    upper_bound: 42000000
 },
 signatories: [
    a1b2c3d4e5f6a7b8c9d0e1f2a3b4c5d6e7f8a9b0c1d2e3f4a5b6c7d8e9f0a1b2,
    b2c3d4e5f6a7b8c9d0e1f2a3b4c5d6e7f8a9b0c1d2e3f4a5b6c7d8e9f0a1b2c3
 ],
 data: [
  [
   d1e2f3a4b5c6d7e8f9a0b1c2d3e4f5a6b7c8d9e0f1a2b3c4d5e6f7a8b9c0d1e2,
   {
    constructor: 0,
    fields: [
      {int: 42},
      {bytes: 48656c6c6f20576f726c6421}
    ]
   }
  [
 ],
 redeemers: {
  Spend: 0
    purpose: Spend,
    index: 0,
    data: {
      constructor: 0,
      fields: [,
        {int: 1}
      ]
    },
    ex_units: }
      mem: 1000000,
      steps: 700000
    }
  }
 },
 id: 3abc4def5678901234567890abcdef1234567890abcdef1234567890abcdef12
 }
```