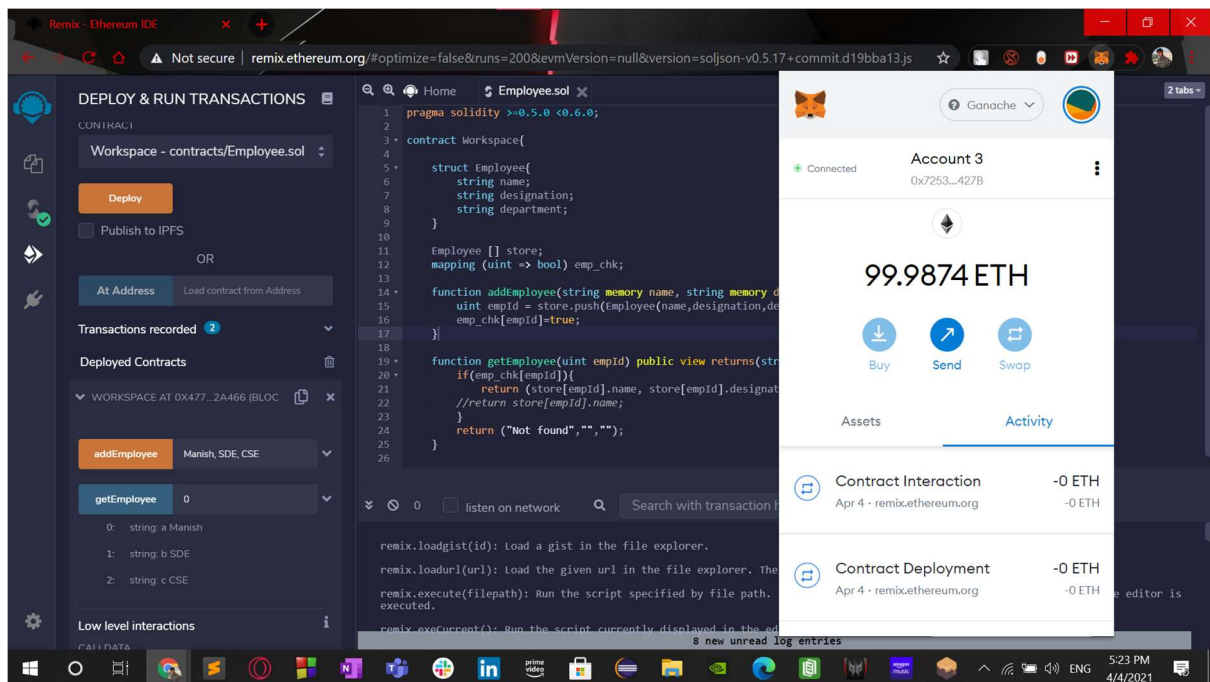


Vibhav Sharma

E18CSE206

BLOCKCHAIN LAB_9

Eg.1 When Emp_id is itself and index of the struct array (Basic Approach)



```
1 pragma solidity >=0.5.0 <0.6.0;
2
3 contract Workspace{
4
5     struct Employee{
6         string name;
7         string designation;
8         string department;
9     }
10
11     Employee [] store;
12     mapping (uint => bool) emp_chk;
13
14     function addEmployee(string memory name, string memory designation, string memory department) public{
15         uint empId = store.push(Employee(name,designation,department))-1;
16         emp_chk[empId]=true;
17     }
18
19     function getEmployee(uint empId) public view returns(string memory a, string memory b, string memory c){
20         if(emp_chk[empId]){
21             return (store[empId].name, store[empId].designation, store[empId].department);
22         }
23         //return store[empId].name;
24         return ("Not found","", "");
25     }
26 }
```

Approach 2 (EMP_ID is Unique and part of the struct employee)

The screenshot shows the Remix IDE interface with the 'Employee.sol' contract loaded. The contract code is as follows:

```
7 string name;
8 string designation;
9 string department;
10
11
12 Employee [] store;
13 mapping (uint => uint) emp_store;
14 mapping (uint => bool) emp_chk;
15
16
17 function addEmployee(uint empId, string memory name, string memory designation, string memory department) public {
18     uint index = store.push(Employee(empId,name,designation,department));
19     emp_store[empId]=index;
20     emp_chk[empId]=true;
21 }
22
23
24 function getEmployee(uint empId) public view returns(uint ID, string memory NAME, string memory DESIGNATION, string memory DEPARTMENT) {
25     if(emp_chk[empId]){
26         return (store[emp_store[empId]].emp_Id,store[emp_store[empId]].name,store[emp_store[empId]].designation,store[emp_store[empId]].department);
27     }
28     return (empId," Not found","", "");
29 }
30
```

The left sidebar shows the 'DEPLOY & RUN TRANSACTIONS' panel with the 'addEmployee' function selected. The 'getEmployee' function is also visible. The 'Low level interactions' panel shows the 'CALLDATA' field with the value 'uint256: ID 7856'. The 'MetaMask Notification' window is open, showing the transaction details for the 'http://remix.ethereum.org' contract. The transaction is confirmed with a gas fee of 0.003005 ETH.

EMP ID : 7856

The screenshot shows the Remix IDE interface with the 'Employee.sol' contract loaded. The contract code is the same as in the previous screenshot. The left sidebar shows the 'DEPLOY & RUN TRANSACTIONS' panel with the 'getEmployee' function selected. The 'Low level interactions' panel shows the results of the transaction:

```
0: uint256: ID 7856
1: string: NAME VIBHAV
2: string: DESIGNATION SDE
3: string: DEPARTMENT IT
```

The 'MetaMask Notification' window is open, showing the transaction details for the 'http://remix.ethereum.org' contract. The transaction is confirmed with a gas fee of 0.003005 ETH.

EMP ID : 8954

The screenshot shows the Remix Ethereum IDE interface. On the left, the 'DEPLOY & RUN TRANSACTIONS' panel is active, displaying a list of transactions. The 'addEmployee' transaction is highlighted, showing the input parameters: 8954, MANISH, SDE2, SE. Below it, the 'getEmployee' transaction is also highlighted, showing the input parameter: 8954. The main editor displays the 'Employee.sol' contract code, which includes functions for adding and retrieving employee data. The bottom status bar indicates '28 new unread log entries'.

```
10 }
11
12 Employee [] store;
13 mapping (uint => uint) emp_store;
14 mapping (uint => bool) emp_chk;
15
16 function addEmployee(uint empId, string memory name, string memory designation, string memory department)
17     uint index = store.push(Employee(empId,name,designation,department))-1;
18
19     emp_store[empId]=index;
20     emp_chk[empId]=true;
21 }
22
23 function getEmployee(uint empId) public view returns(uint ID,string memory NAME, string memory DESIGNATION)
24 {
25     if(emp_chk[empId]){
26         return (store[emp_store[empId]].emp_Id,store[emp_store[empId]].name, store[emp_store[empId]].desig
27     }
28     return (empId," Not found","", "");
29 }
30
31 }
```

SOME RANDOM EMP ID: NOT FOUND

The screenshot shows the Remix Ethereum IDE interface. On the left, the 'DEPLOY & RUN TRANSACTIONS' panel is active, displaying a list of transactions. The 'addEmployee' transaction is highlighted, showing the input parameters: 8954, MANISH, SDE2, SE. Below it, the 'getEmployee' transaction is also highlighted, showing the input parameter: 3956. The main editor displays the 'Employee.sol' contract code, which includes functions for adding and retrieving employee data. The bottom status bar indicates '32 new unread log entries'.

```
10 }
11
12 Employee [] store;
13 mapping (uint => uint) emp_store;
14 mapping (uint => bool) emp_chk;
15
16 function addEmployee(uint empId, string memory name, string memory designation, string memory department)
17     uint index = store.push(Employee(empId,name,designation,department))-1;
18
19     emp_store[empId]=index;
20     emp_chk[empId]=true;
21 }
22
23 function getEmployee(uint empId) public view returns(uint ID,string memory NAME, string memory DESIGNATION)
24 {
25     if(emp_chk[empId]){
26         return (store[emp_store[empId]].emp_Id,store[emp_store[empId]].name, store[emp_store[empId]].desig
27     }
28     return (empId," Not found","", "");
29 }
30
31 }
```