

Run the testrpc in the terminal with:

```
testrpc
```

Create a new .sol file in REMIX IDE with:

```
pragma solidity ^0.4.21;

contract Coursetro_lab4 {
    string fName;
    uint age;

    event Instructor(
        string name,
        uint age
    );

    function setInstructor(string _fName, uint _age) public {
        fName = _fName;
        age = _age;
        // To actually use the event, we need to call the event and pass
        // in the submitted name and age within the existing
        // setInstructor() method, like so:
        emit Instructor(_fName, _age);
    }

    function getInstructor() public constant returns (string, uint) {
        return (fName, age);
    }
}
```

Also create a new .html file with (change to your own address and ABI code):

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Document</title>
    <link rel="stylesheet" type="text/css" href="main_lab4.css">
    <script src="./node_modules/web3/dist/web3.min.js"></script>
</head>
<body>
    <div class="container">
        <h1>Coursetro Instructor</h1>
        <h2 id="instructor">
```

```


</h2>
<label for="name" class="col-lg-2 control-label">Instructor Name</label>
<input id="name" type="text">
<label for="age" class="col-lg-2 control-label">Instructor Age</label>
<input id="age" type="text">
<button id="button">Update Instructor</button>
</div>
<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"></script>
<script>
var web3;
if (typeof web3 !== 'undefined') {
    web3 = new Web3(web3.currentProvider);
} else {
    // set the provider you want from Web3.providers
    web3 = new Web3(new Web3.providers.HttpProvider("http://localhost:8545"));
    if(!web3.isConnected()) {
        console.log('not-connected');
    } else {
        console.log('connected');
    }
}

web3.eth.defaultAccount = web3.eth.accounts[0];
// var CoursetroContract = web3.eth.contract('PASTE ABI CODE HERE');
var CoursetroContract = web3.eth.contract([ { "constant": false, "inputs":
    [ { "name": "_fName", "type": "string" }, { "name": "_age", "type":
    "uint256" } ], "name": "setInstructor", "outputs": [], "payable":
    false, "stateMutability": "nonpayable", "type": "function" }, {
    "constant": true, "inputs": [], "name": "getInstructor", "outputs":
    [ { "name": "", "type": "string" }, { "name": "", "type": "uint256" } ],
    "payable": false, "stateMutability": "view", "type": "function" },
    { "anonymous": false, "inputs": [ { "indexed": false, "name": "name",
    "type": "string" }, { "indexed": false, "name": "age", "type":
    "uint256" } ], "name": "Instructor", "type": "event" } ]]);

console.log('created CoursetroContract');
// var Coursetro = CoursetroContract.at('PASTE CONTRACT ADDRESS HERE');
console.log('Creating Coursetro');
var Coursetro = CoursetroContract.at('0xce3a32a54dbb4ff2f26d80fc2261ed4b02292ca2');
console.log(Coursetro);

// Since we no longer need to call Coursetro.getInstructor(),
// so we're going to remove that.

```

```

////////////////////////////////////
//   Coursetro.getInstructor(function(error, result){
//       console.log('Inside Coursetro.getInstructor');
//
//       if (!error)
//       {
//           console.log('Inside !error');
//           $('#instructor').html(result[0]+' ('+result[1]+' years old)');
//           console.log(result);
//       }
//       else
//           console.log(error);
//   });

// Replace the Coursetro.getInstructor() lines by creating an
//   instructorEvent
var instructorEvent = Coursetro.Instructor();

// Use the .watch() method on instructorEvent with a callback.
instructorEvent.watch(function(error, result){
    if (!error)
    {
        $('#loader').hide();
        $('#instructor').html(result.args.name
            + ' (' + result.args.age + ' years old)');
    } else {
        $('#loader').hide();
        console.log(error);
    }
});

// var instructorEvent = Coursetro.Instructor({}, {fromBlock: 1, toBlock: 'latest'});
// instructorEvent.watch(function(error, result){
//     alert(1);
// });

$('#button').click(function() {
    console.log('Inside click');
    $('#loader').show();
    Coursetro.setInstructor($('#name').val(), $('#age').val());
});
</script>
</body>

```

</html>

Create a .css file for the layout with:

```
body {
  background-color:#F0F0F0;
  padding: 2em;
  font-family: 'Raleway','Source Sans Pro', 'Arial';
}
.container {
  width: 50%;
  margin: 0 auto;
}
label {
  display:block;
  margin-bottom:10px;
}
input {
  padding:10px;
  width: 50%;
  margin-bottom: 1em;
}
button {
  margin: 2em 0;
  padding: 1em 4em;
  display:block;
}
#instructor {
  padding:1em;
  background-color:#fff;
  margin: 1em 0;
}
#loader {
  width: 100px;
  display:none;
}
```

In SOLIDITY COMPILER page, click **Compile xxx.sol**

Then we can copy the **ABI code**

Change the **ENVIRONMENT** to **Web3 Provider** in DEPLOY & RUN TRANSACTIONS, Change the Web3 Provider Endpoint to <http://localhost:8545> , after click **Deploy** we will get a **Deployed Contracts**, we click to copy this as our address

If all stuff going right, we enter the data in our .html web page, we can click **getInstructor** to get our input from .html web page

The image displays a web application interface on the right and its development environment on the left. The web application, titled "Coursetro Instructor", features a form with two input fields: "Instructor Name" (containing "Quan") and "Instructor Age" (containing "25"). A red circle highlights these fields and the "Update Instructor" button below them. The development environment on the left shows the "Coursetro\_lab4 - lab4.sol" contract code. The code includes a `setInstructor` function that takes `_fName` and `_age` as arguments and emits an `Instructor` event. The `getInstructor` function is a public constant function that returns the stored name and age. The "setInstructor" function is also visible in the "Deployed Contracts" section, with input fields for `_fName` (Oscar) and `_age` (22). The "getInstructor" function is highlighted with a red circle in the "Deployed Contracts" section, showing its input fields for `0:` (string: Quan) and `1:` (uint256: 25). The console on the right shows a call to `Coursetro_lab4.getInstructor` with the following data: `value: 0 wei data: 0x22f...00000 logs: 1 hash: 0xe89...b4b10`. The console also shows a call to `Coursetro_lab4.setInstructor` with the following data: `from: 0x43E605fc633958236a7Ae027882E380609a2D86F to: Coursetro_lab4.getInstructor() data: 0x3c1...b81a5`. The console also shows a call to `Coursetro_lab4.getInstructor` with the following data: `from: 0x43E605fc633958236a7Ae027882E380609a2D86F to: Coursetro_lab4.getInstructor() data: 0x3c1...b81a5`. The console also shows a call to `Coursetro_lab4.setInstructor` with the following data: `from: 0x43E605fc633958236a7Ae027882E380609a2D86F to: Coursetro_lab4.getInstructor() data: 0x3c1...b81a5`.