Blockchain & Αποκεντρωμένες Εφαρμογές

3η Εργαστηριακή Δραστηριότητα

Μαυρίδης Ιωάννης, Φουληράς Παναγιώτης
Μάστορας Θεόδωρος
ΤΜΗΜΑ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ

MULTIMEDIA SECURITY NETWORKING

Α. Συμβόλαιο μεσεγγύησης Approval.sol.

1. Επιστρέψτε στον “File explorer” του Remix και δημιουργήστε ένα νέο αρχείο solidity με το όνομα “Approval.sol”.

Πληκτρολογήστε ή αντιγράψτε το παρακάτω πρόγραμμα Solidity:

link

```
// SPDX-License-Identifier: GPL-3.0
pragma solidity >=0.8.0;

contract Approval {

    address payable public sender = payable(0x0);
    address payable public receiver = payable(0x0);
    address private approver;

    event LogDeposit(
        address payable _sender,
        address payable _receiver,
        uint amount
    );

    constructor () {
        approver = msg.sender;
    }

    modifier restricted() {
        require(msg.sender == approver);
        _;
    }

    modifier credited() {
        require(address(this).balance > 0);
        _;
    }

    function deposit(address payable _receiver) public payable {
        require(msg.value > 0);
        sender = payable(msg.sender);
        receiver = _receiver;

        emit LogDeposit(sender, receiver, msg.value);
    }

    function viewApprover() public view returns(address, uint256) {
        return(approver, address(this).balance);
    }

    function approve() public restricted credited{
        require(receiver != payable(0x0));
        receiver.transfer(address(this).balance);
```

```
    }

    function reset() public restricted credited {
        sender.transfer(address(this).balance);
        sender = payable(0x0);
        receiver = payable(0x0);
    }
}
```

Το πρόγραμμα αυτό υλοποιεί ένα smart contract «μεσάζοντα» σε μια πολύ απλοϊκή μορφή. Έχει αρκετά θέματα που χρήζουν βελτίωσης, αλλά ως παράδειγμα για τις δοκιμές που ακολουθούν, είναι ικανοποιητικό.

Στη γραμμή 30 δηλώνεται η public μέθοδος deposit. Αυτός που την καλεί (ο αποστολέας), της περνά ως παράμετρο τη δημόσια διεύθυνση κάποιου τρίτου (του παραλήπτη). Επειδή είναι μια payable μέθοδος κατά την κλήση, αυτόματα περνάνε άλλες δύο τιμές, ένα ποσό σε wei ('msg.value') και η διεύθυνση αυτού που την καλεί ('msg.sender'). Αν αυτός που την καλεί δεν στείλει λεφτά, δεν εκτελείται (λόγω require). Αν εκτελεστεί, πυροδοτεί το συμβάν 'LogDeposit'.

Στις γραμμές 6-8 δηλώνονται 3 δημόσιες σφαιρικές μεταβλητές. Στην **approver** αποθηκεύεται η διεύθυνση του «εγγυητή». Στην **receiver** η διεύθυνση του «παραλήπτη». Και στη **sender** η διεύθυνση του «αποστολέα» χρημάτων.

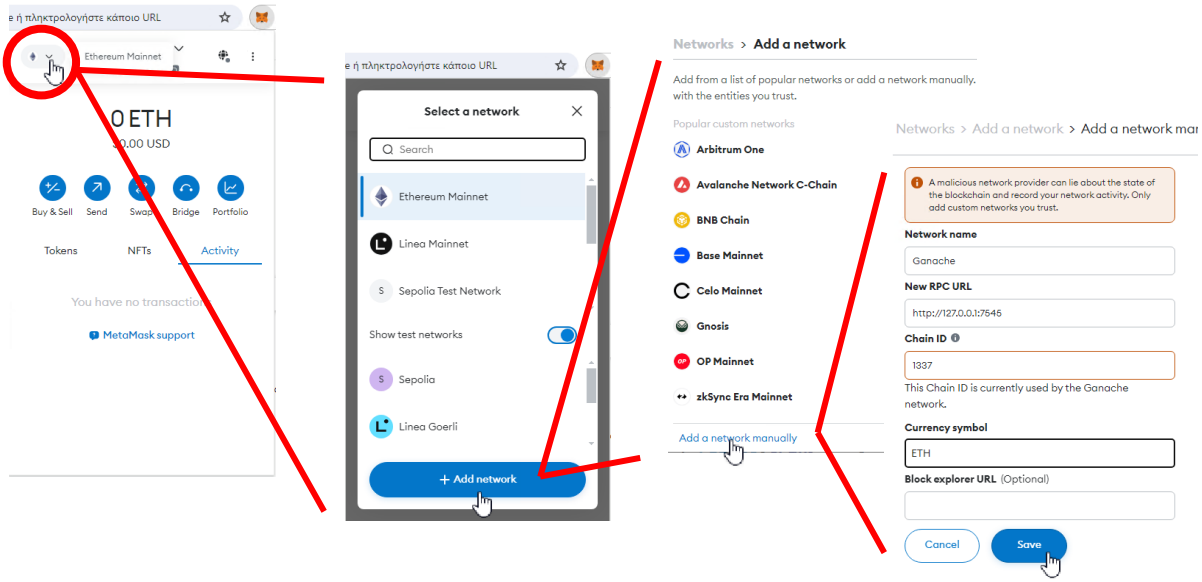
Στη γραμμή 38 δηλώνεται η μέθοδος **viewApprover** η οποία επιστρέφει τη διεύθυνση του «εγγυητή».

Στη γραμμή 42 δηλώνεται η μέθοδος **approve** η οποία, (ΜΟΝΟ) όταν καλείται από τον εγγυητή, στέλνει όλο το ποσό του contract στον παραλήπτη. Αν την καλέσει κάποιος άλλος, δεν εκτελείται (λόγω require).

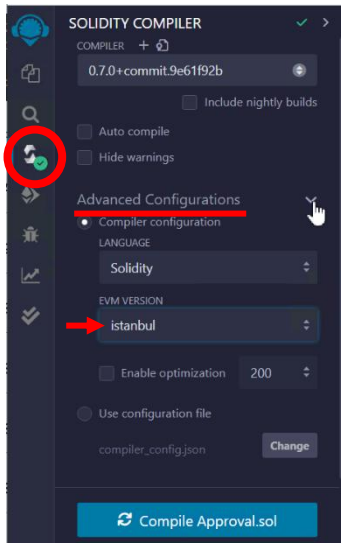
Ένα τυπικό σενάριο χρήσης έχει ως εξής: Ο λογαριασμός Α είναι ο εγγυητής. Ο λογαριασμός Β (ο αποστολέας) ζητά κάποιο προϊόν ή υπηρεσία από τον λογαριασμό C. Για να αποδείξει πως έχει το ποσό, καλεί την deposit και το στέλνει στο contract δηλώνοντας ως παραλήπτη τον C. Το contract δεσμεύει το ποσό στο λογαριασμό του. Μόλις ο C «παραδώσει», ο Α καλεί την approve και το ποσό μεταφέρεται στο λογαριασμό του (του C). Αν δεν «παραδώσει», ο Α καλεί τη μέθοδο reset που στέλνει το ποσό πίσω στον Β.

Παρατήρηση: Το contract θα οδηγηθεί σε λάθος αν η deposit κληθεί περισσότερες από μία φορές με διαφορετικούς παραλήπτες. Σε μια τέτοια περίπτωση, η approve θα έστελνε ΟΛΟ το ποσό στον πιο πρόσφατο receiver.

2. Πειραματιστείτε με το συμβόλαιο αφού το μεταγλωττίσετε και το δημοσιεύσετε στο τοπικό "Ganache". Στο REMIX επιλέξτε "Injected Web3" και συνδέστε το Metamask με το τοπικό δίκτυο Ganache.
(New RPC URL: **http://127.0.0.1:7545**, Chain ID: **1337**, Currency symbol: **ETH**).



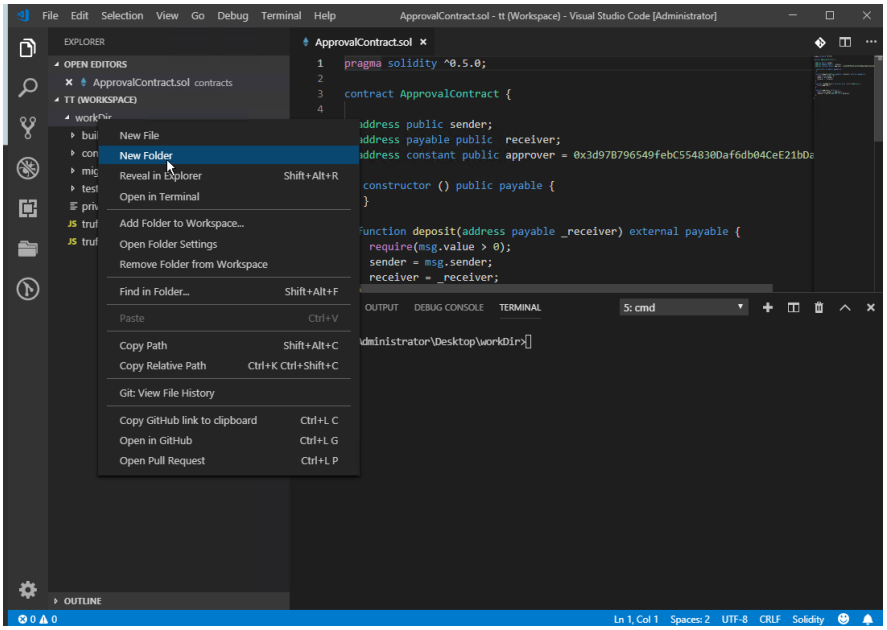
Προσοχή! Για να δημοσιευτεί το συμβόλαιο στο Ganache, ανάλογα της έκδοσης του Ganache που έχετε εγκαταστήσει, θα χρειαστεί να ρυθμίσετε το Remix, ώστε να εκτελέσει compile σε διαφορετική έκδοση EVM (π.χ. Istanbul ή Berlin – δείτε την παρακάτω εικόνα).



Με τα πορτοφόλια του metamask να έχουν δοκιμαστικά ETH του Ganache, όταν πραγματοποιηθεί η δημοσίευση του συμβολαίου χωρίς λάθη, δοκιμάστε με έναν διαφορετικό λογαριασμό από του approver να «στείλετε» χρήματα με την deposit σε κάποιον τρίτο. Πηγαίνετε μετά ως approver και εγκρίνετε τη μεταφορά. Παρακολουθήστε την πορεία των χρημάτων.

B. Υλοποίηση σε JavaScript της διασύνδεσης χρήστη (front end). Αυτή η εφαρμογή όταν δημοσιευτεί (στο swarm ή στο IPFS) θα είναι η DApp.

1. Στο panel EXPLORER του VS Code, κάντε δεξί κλικ και δημιουργήστε μέσα στα έγγραφα τον φάκελο "**web**".



2. Στο panel EXPLORER του VS Code, κάντε δεξί κλικ στο φάκελο **web** και δημιουργήστε ένα νέο αρχείο με το όνομα "**index.html**".

Πληκτρολογήστε ή αντιγράψτε από το PDF το παρακάτω πηγαίο HTML

link

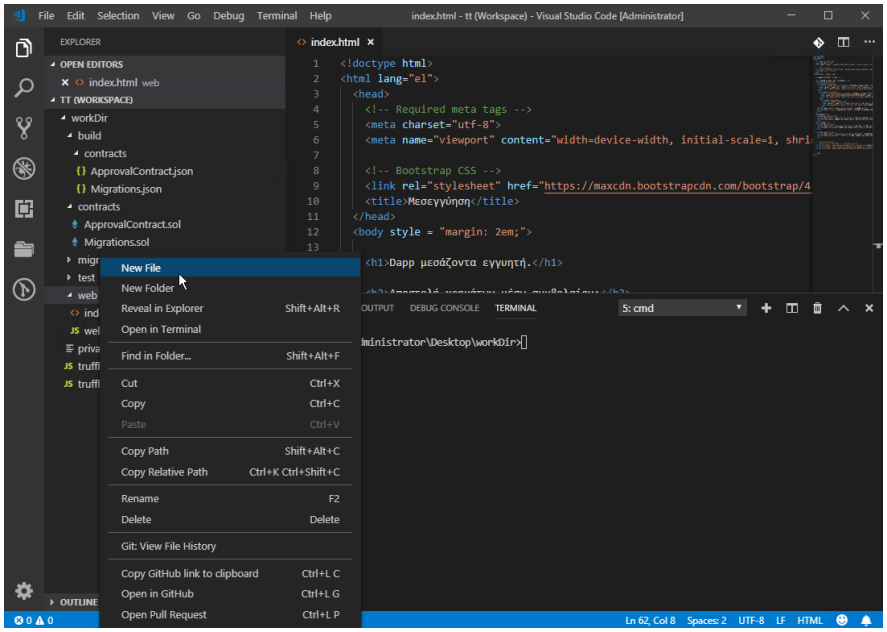
```
<!DOCTYPE html>
<html lang="el">

<head>
    <!-- Required meta tags -->
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no" />

    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
        integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous" />
    <title>Μεσεγγύηση</title>
</head>

<body style="margin: 2em">
    <h1>DApp μεσάζοντα εγγυητή.</h1>

    <h2>Αποστολή χρημάτων μέσω συμβολαίου:</h2>
    <form id="contract-form">
        <div class="form-group">
```

```
        <label for="Sender Address">Διεύθυνση αποστολέα χρημάτων (ETH)</label>
        <div id="from-Address">
            Όταν συνδεθεί το Metamask, εδώ εμφανίζεται η διεύθυνσή σας
        </div>
        <small id="fromAddressHelp" class="form-text text-muted">Θα πρέπει τελικά να δώσετε έγκριση της μεταφοράς
            από το Metamask.</small>
    </div>
    <div class="form-group">
        <label for="Receiver Address">Διεύθυνση παραλήπτη χρημάτων (ETH)</label>
        <input value="0x965B0F50762C8c88d58b6B4877c233461631b586" type="text" class="form-control" id="toAddress"
            aria-describedby="toAddressHelp" placeholder="Enter the receipient address" required="true" />
        <small id="toAddressHelp" class="form-text text-muted">Εισάγετε την διεύθυνση (δημόσιο κλειδί) του
            πορτοφολιού του παραλήπτη.</small>
    </div>
    <div class="form-group">
        <label for="Amount">Ποσό</label>
        <input value="2" type="text" class="form-control" id="amount" aria-describedby="amountHelp"
            placeholder="Amount to send in ETH" required="true" />
        <small id="amountHelp" class="form-text text-muted">Πόσα χρήματα θα στείλετε σε Ether.</small>
    </div>
    <button type="submit" class="btn btn-primary">Αποστολή</button>
    <div id="deposit-result">
        Πατώντας το κουμπί "Αποστολή" τα χρήματά σας θα πιστωθούν στο συμβόλαιο.
    </div>
</form>
<hr />
<h2>Υπόλοιπο:</h2>
<form id="get-balance-form">
    <button type="submit" class="btn btn-primary">Δείτε το υπόλοιπο</button>
    <div id="the-balance">
        Πατήστε το κουμπί για να δείτε το ποσό που είναι αποταμιευμένο στο συμβόλαιο.
    </div>
</form>
<h2>Εγγυητής:</h2>
<form id="approver-form">
    <button type="submit" class="btn btn-primary">Δείτε τον εγγυητή</button>
    <div id="approver-display">
        Πατήστε το κουμπί για να δείτε τη διεύθυνση του Εγγυητή.
    </div>
</form>

<form id="approve-form">
    <button type="submit" class="btn btn-primary">
        Επιβεβαίωση συναλλαγής
    </button>
    <div id="approval-display">
        Αν είστε ο εγγυητής, πατήστε το κουμπί για να εγκρίνετε τη συναλλαγή και
        τα χρήματα να μεταφερθούν στον παραλήπτη.
    </div>
</form>

<script src="https://cdn.jsdelivr.net/npm/web3@latest/dist/web3.min.js"></script>
```

```
<script src="Contract_abi.js"></script>
<!-- var abi = [...] -->
<!-- jQuery -->
<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
    integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
    crossorigin="anonymous"></script>

<script>
    window.addEventListener("load", async () => {
        try {
            const accounts = await window.ethereum.request({
                method: "eth_requestAccounts",
            });
            fromAddress = accounts[0];
            $("#from-Address").html("<b>" + fromAddress + "</b>");
        } catch (error) {
            console.error("MetaMask connection error:", error);
        }
    });

    if (typeof web3 != "undefined") {
        // if web3.js library is available (imported?)
        web3 = new Web3(window.ethereum);
    }

    window.ethereum.on("accountsChanged", async (new_accounts) => {
        fromAddress = new_accounts[0];
        $("#from-Address").html("<b>" + fromAddress + "</b>");
    });

    //change this to the ACTUAL contract address that you created on truffle migrate
    var contractAddress = "0x747543937777816868543ECF7Ae20b4aADE12a519";
    var ApprovalContract = new web3.eth.Contract(abi, contractAddress);
    var fromAddress;

    //make sure that addresses are legit
    $("#contract-form").submit(async () => {
        event.preventDefault();
        $("#from-Address").html("<b>" + fromAddress + "</b>");
        var toAddress = $("#toAddress").val();
        if (web3.utils.isAddress(toAddress) != true) {
            alert("Δεν δώσατε σωστή διεύθυνση παραλήπτη.");
            return;
        }
        // make sure the ETH is > 0
        var amount = parseFloat($("#amount").val());
        if (isNaN(amount) || amount <= 0) {
            alert("Πρέπει να στείλετε περισσότερα από 0 ETH");
            return;
        }
        // all is good, let's call our contract deposit
        ApprovalContract.methods
```

```
            .deposit(toAddress)
            .send(
                {
                    from: fromAddress,
                    gas: 100000, gasPrice: 1000000000,
                    value: web3.utils.toWei(amount.toString(), "ether"),
                },
                (error, result) => {
                    if (error) {
                        console.log("error: " + error);
                        $("#deposit-result").html(
                            "Error: <b>" + error.message + "</b>"
                        );
                    } else {
                        $("#deposit-result").html("Success TX: <b>" + result + "</b>");
                    }
                }
            );
    });

    $("#get-balance-form").submit(async () => {
        event.preventDefault();
        const result = await web3.eth.getBalance(contractAddress);
        console.log("balance: " + result);
        $("#the-balance").html(
            "Τρέχον υπόλοιπο: <b>" + web3.utils.fromWei(result, "ether") + "</b>"
        );
    });

    $("#approve-form").submit(async () => {
        event.preventDefault();
        const result = await ApprovalContract.methods
            .approve()
            .send({ from: fromAddress, gas: 100000, gasPrice: 1000000000 });
        console.log("transaction: " + result.transactionHash);
        $("#approval-display").html(
            "Transaction Approved. TX: <b>" + result.transactionHash + "</b>"
        );
    });

    $("#approver-form").submit(async () => {
        event.preventDefault();
        const result = await ApprovalContract.methods.viewApprover().call();
        console.log("approver: " + result[0]);
        $("#approver-display").html(
            "Approver Address: <b>" + result[0] + "</b>"
        );
    });
</script>
</body>

</html>
```

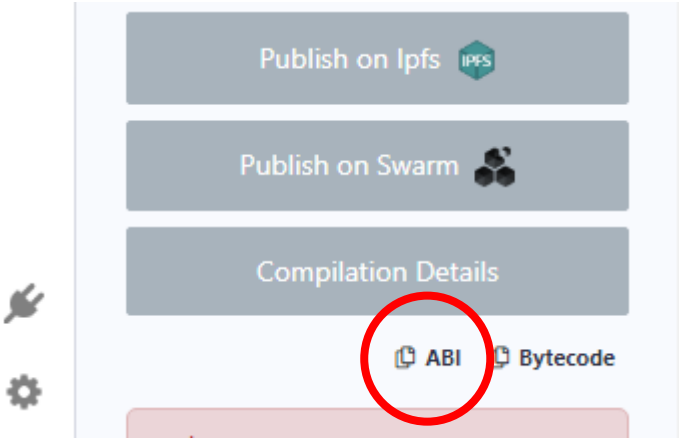
3. Στο panel EXPLORER του VS Code, κάντε δεξί κλικ στο φάκελο **web** και δημιουργήστε ένα νέο αρχείο με το όνομα "**Contract_abi.js**".



Πληκτρολογήστε (στο αρχείο **Contract_abi.js**) την εντολή:

```
var abi = [];
```

Όταν το Remix έκανε compile το `Approval.sol`, εξήγαγε το αρχείο `Approval.json` στον φάκελο «`artifacts`». Στο αρχείο αυτό υπάρχει ένα τμήμα που περιέχει το ABI. Στο αρχείο **Contract_Abi.js** αντικαταστήστε τον πίνακα **`[]`**, **κάνοντας αντιγραφή και επικόλληση** από το αρχείο `Approval.json`. Μπορείτε επίσης να το αντιγράψετε από το εικονίδιο κάτω αριστερά στο παράθυρο του Remix.



Στο VS Code επικολλήστε και σώστε το **Contract_abi.js**.

4. Στις γραμμές 133 και 134 γίνεται η δήλωση του contract ώστε να είναι δυνατή η επικοινωνία με αυτό. Πρέπει να πληκτρολογήσετε **την δική σας διεύθυνση** του contract.

```
//change this to the ACTUAL contract address that you created on Remix
var contractAddress = "0x74754393777816868543ECF7Ae20b4aADE12a519";
```

Για να είναι μικρός σε έκταση ο κώδικας, γίνεται εκτεταμένη χρήση της εντολής **`console.log`**(). Για να μπορέσετε να δοκιμάσετε τη λειτουργικότητα της σελίδας καλό είναι να την προβάλετε στον

Chrome με την επιλογή «Εργαλεία για προγραμματιστές». Η επιλογή αυτή κάνει ορατή την κονσόλα της JavaScript. Εκτός από το μενού «Περισσότερα εργαλεία», μπορείτε να ενεργοποιήσετε την κονσόλα πατώντας στο πληκτρολόγιο το συνδυασμό **Ctrl+Shift+I** ή το **F12** και διαλέγοντας την καρτέλα **“Console”**.

C. Η λειτουργία της εφαρμογής web

- Πολλοί απομακρυσμένοι χρήστες.
- Απαιτείται **συγχρονισμός** των συναλλαγών (π.χ. στοιχηματισμός σε ακολουθίες κληρώσεων λαχείου).

Διαβάστε:

1. https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Classes

1. https://www.digitalocean.com/community/tutorials/understanding-javascript-promises
2. https://eloquentjavascript.net/11_async.html
3. https://web.dev/promises/

1. https://nordicapis.com/13-node-js-frameworks-to-build-web-apis/

- Χωρίς jquery

```
betAmount = document.getElementById('betAmount').value;
document.getElementById('from-Address').innerHTML = '<b>' + …
```

- Με jquery

```
betAmount = $('#betAmount').val();
$('#from-Address').html('<b>' + fromAddress + '</b>');
```

web3.js - Επικοινωνία με metamask

1 – ορισμός του metamask ως provider

```
window.addEventListener("load", async () => {
try {
    const accounts = await window.ethereum.request({
    method: "eth_requestAccounts",
    });
    fromAddress = accounts[0];
    $("#from-Address").html("<b>" + fromAddress + "</b>");
} catch (error) {
    console.error("MetaMask connection error:", error);
}
});

if (typeof web3 != "undefined") {
// if web3.js library is available
web3 = new Web3(window.ethereum);
}
```

2 – σύνδεση με το τρέχον πορτοφόλι κάθε φορά που αλλάζει

```
var fromAddress;
window.ethereum.on("accountsChanged", async (new_accounts) => {
fromAddress = new_accounts[0];
$("#from-Address").html("<b>" + fromAddress + "</b>");
});
```

web3.js - Επικοινωνία με blockchain

1 - Έλεγχος αν μια διεύθυνση (εδώ η *toAddress*) είναι έγκυρη διεύθυνση (πορτοφολιού ή συμβολαίου) με τη χρήση της "**web3.utils.isAddress**" (εκτελείται τοπικά - δεν χρειάζεται την await):

```
// The wallet address to be checked
const toAddress = '0x1234567890123456789012345678901234567890';

// Check if the address is valid
const isValidAddress = web3.utils.isAddress(toAddress);

if (isValidAddress) {
  console.log(`${toAddress} is a valid Ethereum wallet address.`);
} else {
  console.log(`${toAddress} is NOT a valid Ethereum wallet address.`);
}

// OR
```

```
if (web3.utils.isAddress(toAddress) != true) {
  alert(`Δεν δώσατε σωστή διεύθυνση.`)
}
```

2 - Ερώτηση για το υπόλοιπο σε Ethers μιας διεύθυνσης πορτοφολιού ή συμβολαίου με χρήση της **web3.eth.getBalance**. Η μέθοδος getBalance() παίρνει ως πρώτο όρισμα μια διεύθυνση Ethereum και επιστρέφει το ποσό αυτής της διεύθυνσης σε μονάδες wei. Το ποσό που επιστρέφεται μπορεί να μετατραπεί σε Ether χρησιμοποιώντας τη μέθοδο **web3.utils.fromWei**(), η οποία παίρνει ως πρώτο όρισμα το ποσό και ως δεύτερο τη μονάδα μετατροπής (π.χ. 'ether').

```
// The contract address to check the balance of
const contractAddress = '0x1234567890123456789012345678901234567890';

// Get the balance of the contract address
web3.eth.getBalance(contractAddress, function(error, balance) {
  if (!error) {
    console.log(`The balance of ${contractAddress} is ${web3.utils.fromWei(balance, 'ether')} ETH.`);
  } else {
    console.log(`Error getting balance; ${error}`);
  }
});
// OR

result = await web3.eth.getBalance(contractAddress);
```

3 - Κλήση μιας **payable** μεθόδου. Μεταφέρονται λεφτά στο συμβόλαιο μέσω της value και πληρώνονται τέλη μέσω του gas limit.

```
ApprovalContract.methods.deposit(toAddress).send(
    { from: fromAddress, gas: 100000, gasPrice: 1000000000,
      value: web3.utils.toWei(amount.toString(), 'ether') }, (error, result) => {
        if (error) {
            console.log('error: ' + error);
            $('#deposit-result').html('Error: <b>' + error.message + '</b>');
        }
        else {
            $('#deposit-result').html('Success TX: <b>' + result + '</b>');
        }
    });
```

4 - Κλήση μιας μεθόδου που αλλάζει τιμές μεταβλητών. Πληρώνονται ΜΟΝΟ τέλη μέσω του gas limit.

```
result = await ApprovalContract.methods.approve().send({ from: fromAddress,
                                                          gas: 100000, gasPrice: 1000000000 });
```

Προσοχή! Η δήλωση τιμών για gas ΚΑΙ gasPrice είναι απαραίτητη ΜΟΝΟ σε παλιά δίκτυα (π.χ. εκδόσεις του Ganache), τα οποία δεν υποστηρίζουν τα νεότερα πρωτόκολλα καθορισμού των fees.

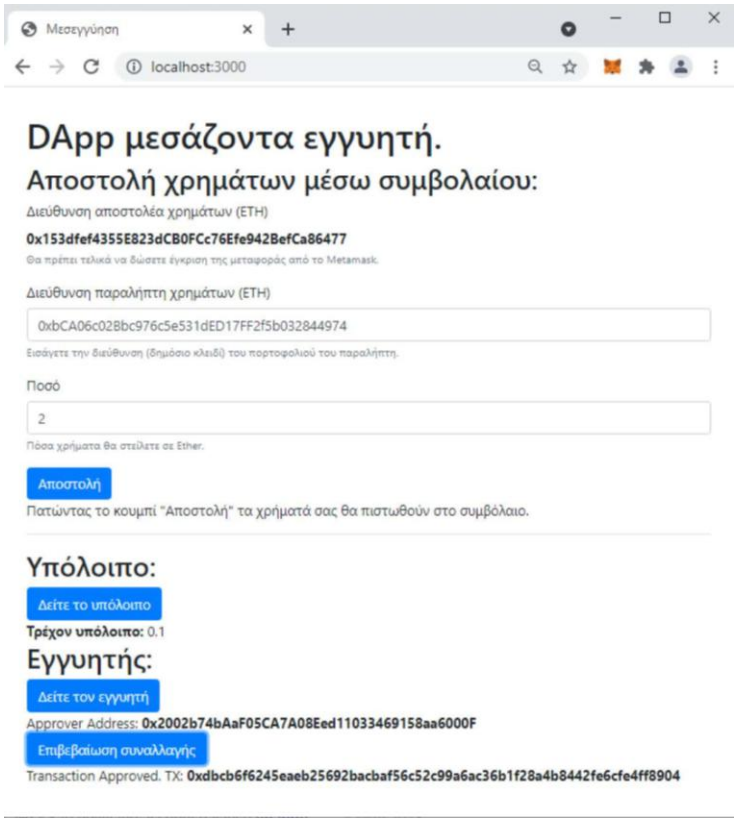
5 - Κλήση μιας δωρεάν μεθόδου.

```
result = await ApprovalContract.methods.viewApprover().call();
```

6 - Έστω στο συμβόλαιο έχουμε ένα συμβάν… `event LogDeposit(address payable _sender, address payable _receiver, uint amount);`

Για να πυροδοτείται ένα συμβάν της JavaScript, όποτε πυροδοτείται ένα συμβάν στο συμβόλαιο (δηλ. στην blockchain), χρησιμοποιούμε το παρακάτω:

```
ApprovalContract.events.LogDeposit().on("data", async (data) => {
  const curAddress = data.returnValues._receiver.toString();
  $("#events_list").append(`<li>${curAddress}</li>`);
  console.log(curAddress);
});
```

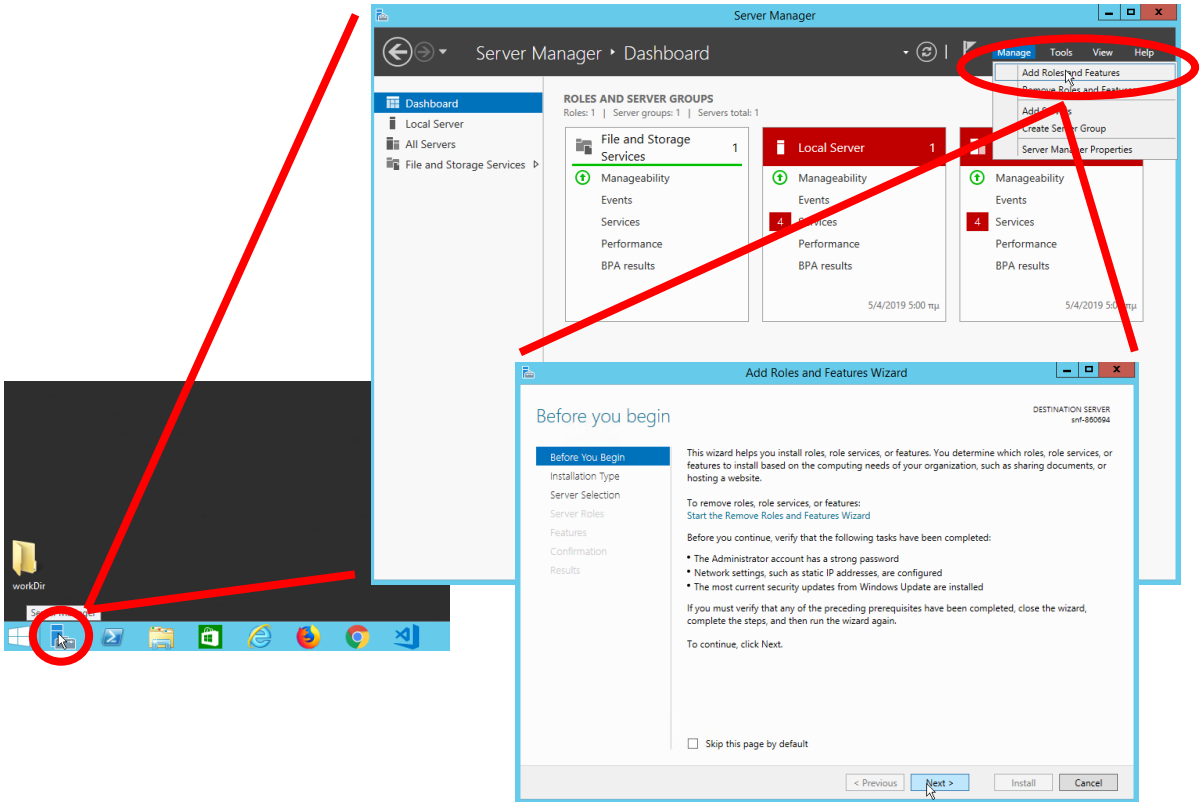


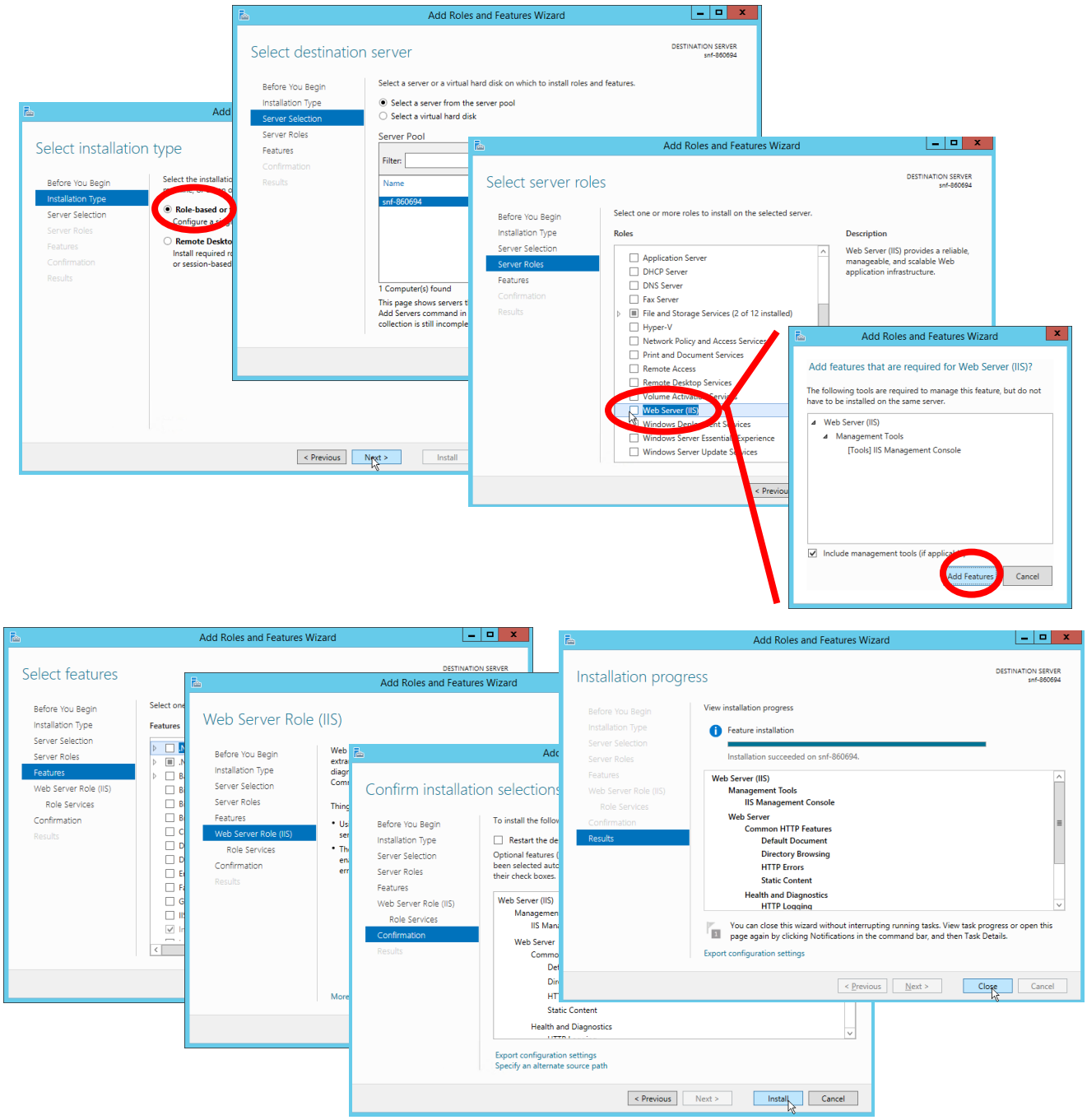
D. Δημοσίευση τοπικά στον IIS

Για να μπορεί οποιοσδήποτε να χρησιμοποιήσει το συμβόλαιο, θα πρέπει η ιστοσελίδα διασύνδεσης, να φιλοξενηθεί σε κάποιον Web Server. Έτσι ο καθένας θα μπορεί να την ανοίγει στον υπολογιστή του ή σε οποιαδήποτε άλλη συσκευή και να χρησιμοποιεί το δικό του πορτοφόλι για τις όποιες συναλλαγές.

1. Εγκαταστήστε τον Internet Information Server (IIS) στη VM. Ο IIS είναι ο Web Server που διατίθεται με τα Windows και μπορείτε να τον εγκαταστήσετε από την προσθήκη ρόλων του Server Manager.

Κάντε κλικ στη συντόμευση του Server Manager της γραμμής εργασιών (δίπλα στο κουμπί «Έναρξη»). Και στη συνέχεια επιλέξτε **Manage -> Add Roles and Features**





2. Αντιγράψτε τον φάκελο web στον φάκελο «`C:\inetpub\wwwroot`». Σύρετε τον φάκελο από παράθυρο σε παράθυρο Windows Explorer κρατώντας πατημένο το πλήκτρο Ctrl. Ο φάκελος «`wwwroot`» είναι ο προεπιλεγμένος ριζικός φάκελος του IIS.

3. Ρυθμίστε το firewall των windows ώστε να επιτρέπονται οι εισερχόμενες κλήσεις στο port 80. Μπορείτε πλέον να ανοίξτε την ιστοσελίδα http://~~83.212.102.23~~/web/. Δώστε την ip της δικής σας VM.

Ε. Δημοσίευση τοπικά στον lite-server του node.js (http://localhost:3000/)

Μέσα στον φάκελο web, σε παράθυρο γραμμής εντολών, δώστε την εντολή:

npm install lite-server --save-dev

Δημιουργείται ο φάκελος “node_modules” με τον lite-server.

Στον φάκελο web δημιουργήστε το αρχείο “package.json’ με το εξής περιεχόμενο

link

```
{
    "scripts": {
        "dev": "lite-server"
    },
    "devDependencies": {
        "lite-server": "^2.6.1"
    }
}
```

Στον φάκελο web, σε παράθυρο γραμμής εντολών, δώστε την εντολή:

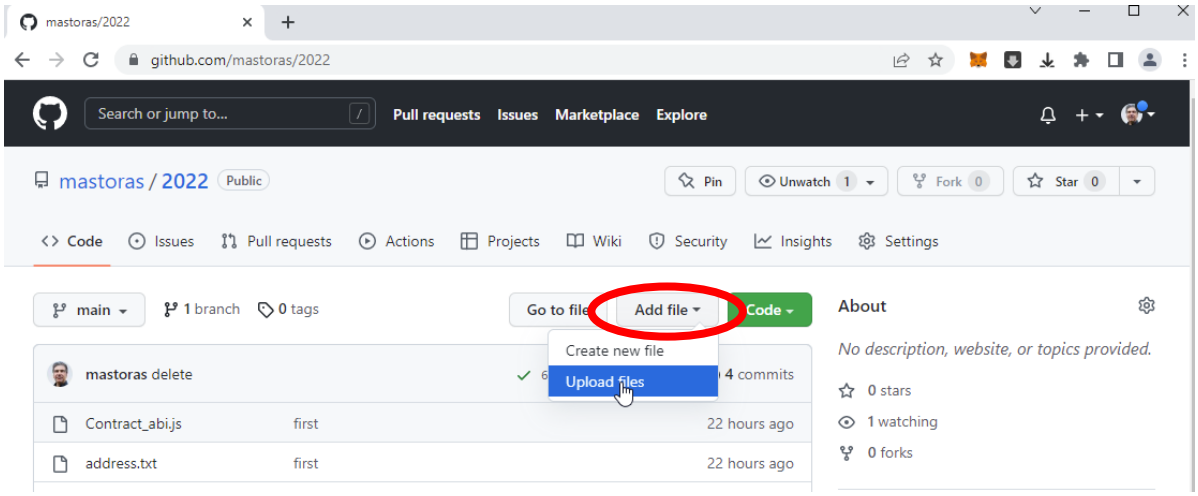
npm run dev

Αν οι οδηγίες του “package.json’ είναι σωστές, εκτελείται ο lite-server.

Αν θέλετε να ανοίγετε τη σελίδα απομακρυσμένα, ρυθμίστε το firewall των windows ώστε να επιτρέπονται οι εισερχόμενες κλήσεις στο port 3000. (Προσοχή, αν αργότερα δημοσιεύσετε αλλού τον φάκελο web δεν χρειάζεστε τον φάκελο “node_modules”)

F. Δημοσίευση στο Github – (ΜΟΝΟ Στατικές σελίδες)

1. «Ανεβάστε» τα αρχεία από τον φάκελο web στο repository σας (δεν επιτρέπει το ανέβασμα φακέλων). Πηγαίνετε στο βήμα 2.



Εναλλακτικά (προτιμότερο επειδή επιτρέπει μαζικά push – commit)

1. Κατεβάστε από https://gitforwindows.org/ το Git. Εγκαταστήστε το. Ενώ βρίσκεστε στον φάκελο έγγραφα, δώστε στο Git Bash την εντολή

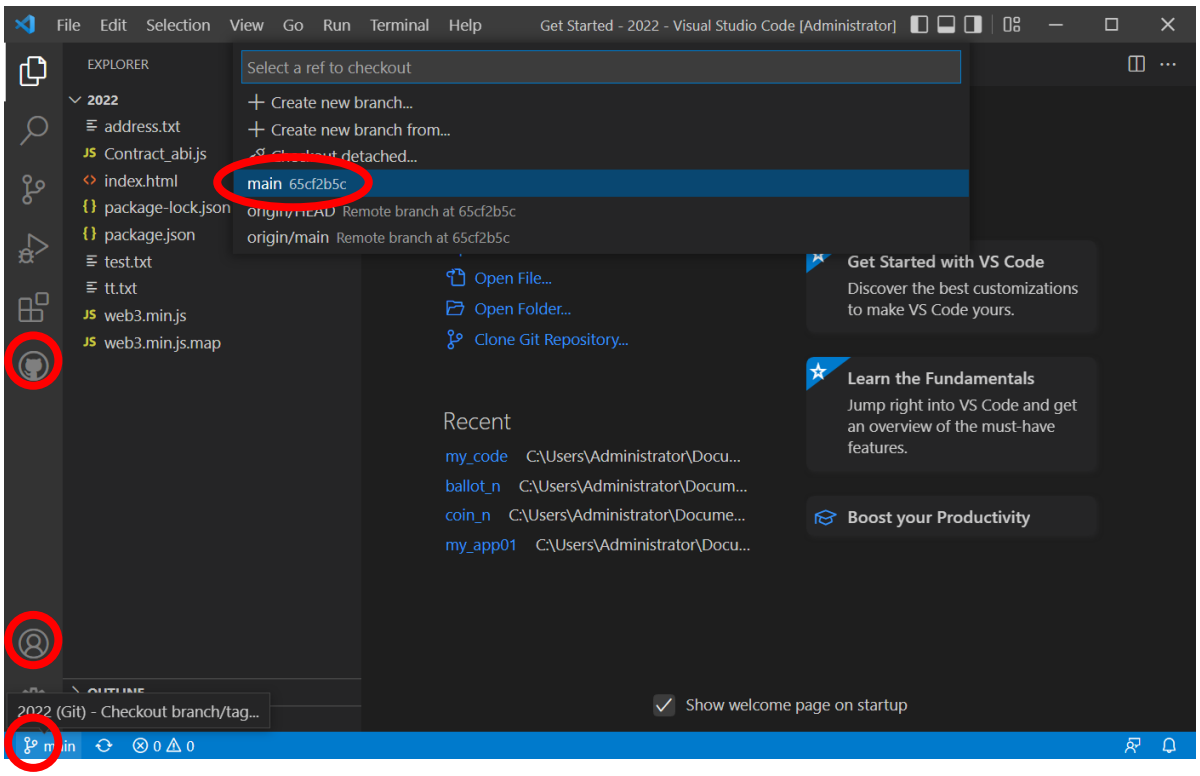
```
git clone https://github.com/<username>/<folder>
```

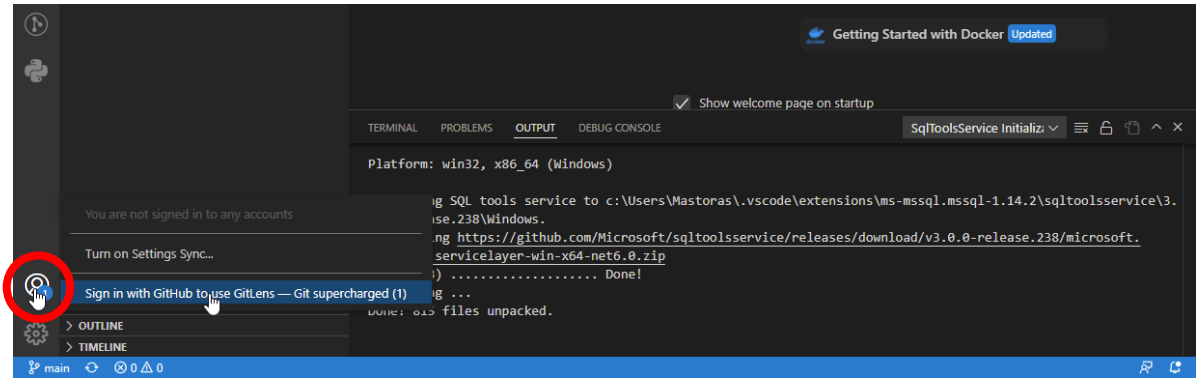
Θα δείτε να δημιουργείται ένα αντίγραφο του repository σας από το github.

Είναι χρήσιμο επίσης να δώσετε μία φορά τις εντολές που ρυθμίζουν καθολικά την git

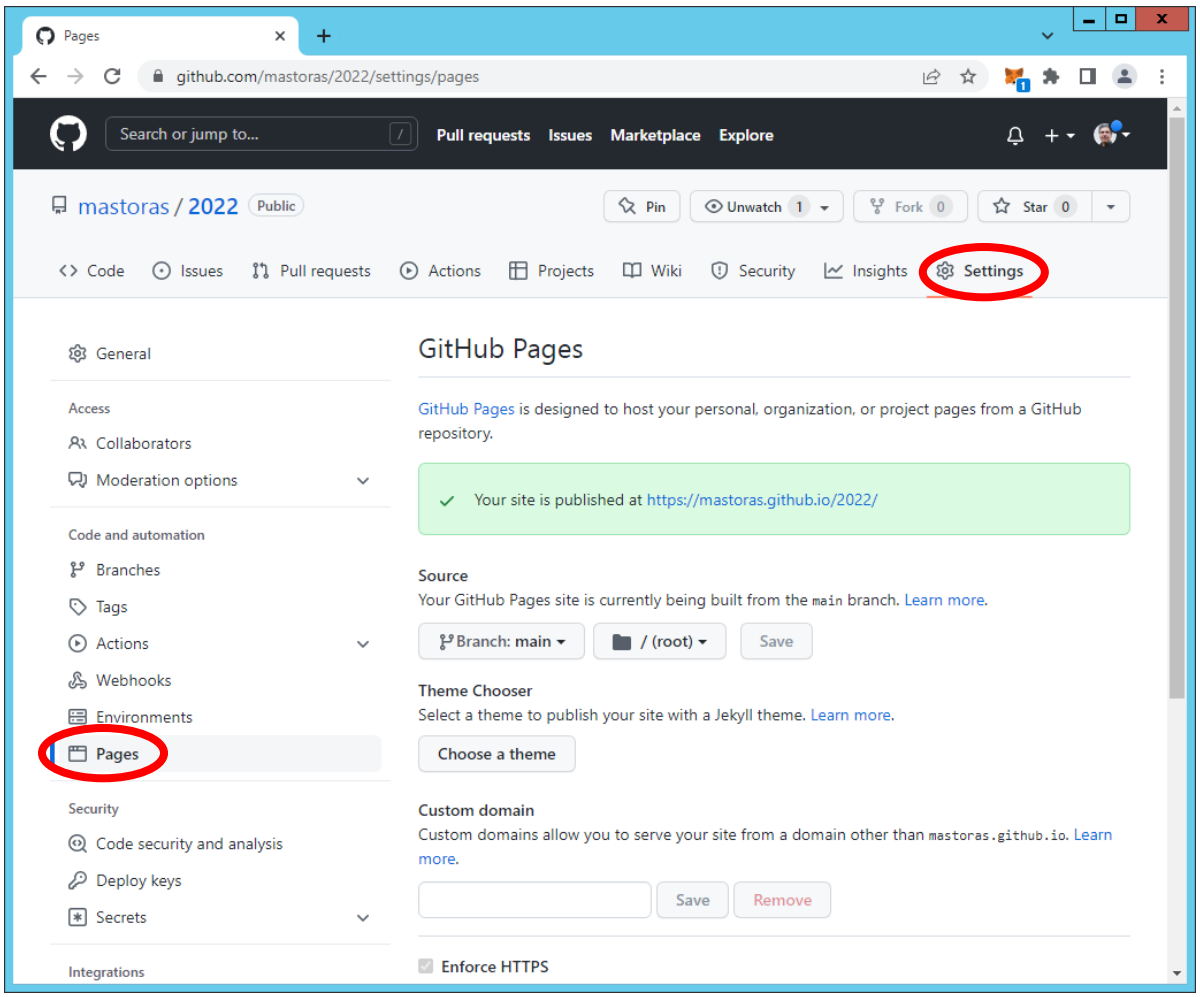
```
git config --global user.name "<username>"
git config --global user.email "<email>"
```

Εγκαταστήστε στο VSCode το πρόσθετο “GitHub Pull Requests and Issues”. Ανοίξτε στο VSCode τον repository που αντιγράψατε. Κάνετε είσοδο στο github από το VSCode πατώντας στο εικονίδιο του profile. Επιλέξτε το branch όπου θα κάνετε push πατώντας στο πρώτο εικονίδιο της γραμμής κατάστασης. Αντιγράψτε όλο το περιεχόμενο του web μέσα στον αντεγραμμένο φάκελο. Επιλέξτε το εικοονίδιο του git και κάντε stage και push.





2. Μεταβείτε στο αντίστοιχο repository στο github και από το Settings -> Pages ενεργοποιήστε τον web server για όλο τον φάκελο. (Προσοχή, χρειάζονται περίπου 5 λεπτά μέχρι να αδειοδοτηθεί προς δημοσίευση ο φάκελος).



Η σελίδα σας βρίσκεται στο link https://<user>.github.io/<repository>/

Προσοχή! Ιστότοποι του node (π.χ. σε React) που «εκτελούνται» τοπικά στον υπολογιστή σας, μπορούν να μετατραπούν σε «συμπυκνωμένη» στατική μορφή και να δημοσιευτούν στο Github. Χρειάζεται η εγκατάσταση του module **gh-pages**, με την εντολή:

npm install gh-pages --save-dev

η τροποποίηση του αρχείου package.json στον φάκελο του site,

```
{
{
  "name": "smart09",
  "version": "0.1.0",
  "homepage": "https://<user>.github.io/smart09",
  "private": true,
  "dependencies": {
    "@testing-library/jest-dom": "^5.16.5",
    "@testing-library/react": "^13.4.0",
    "@testing-library/user-event": "^13.5.0",
    "bootstrap": "^5.2.3",
    "react": "^18.2.0",
    "react-dom": "^18.2.0",
```

```
    "react-scripts": "5.0.1",
    "web-vitals": "^2.1.4",
    "web3": "^1.8.2"
  },
  "scripts": {
    "predeploy": "npm run build",
    "deploy": "gh-pages -b master -d build",
    "start": "react-app-rewired start",
    "build": "react-app-rewired build",
    "test": "react-app-rewired test",
    "eject": "react-scripts eject"
  },
```

Να ζητήσετε ένα νέο token από το Github στη διεύθυνση https://github.com/settings/tokens.

Να δώσετε στο φάκελο του site την εντολή:

git remote set-url origin https://<user>:<token>@github.com/<user>/<repo>

και τέλος την εντολή:

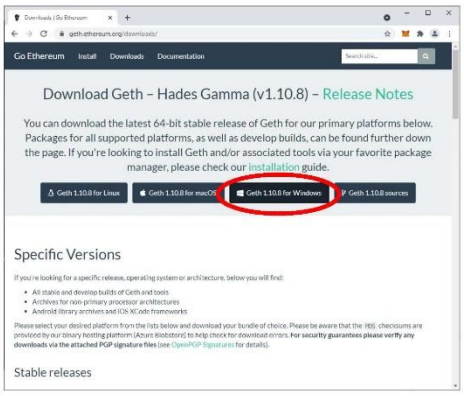
npm run deploy

Στην περίπτωση αυτή αντί του branch “main”, στο Settings -> Pages πρέπει να ενεργοποιήστε τον web server για το branch “master”. Περισσότερες λεπτομέρειες θα δοθούν στο εργαστήριο της React.

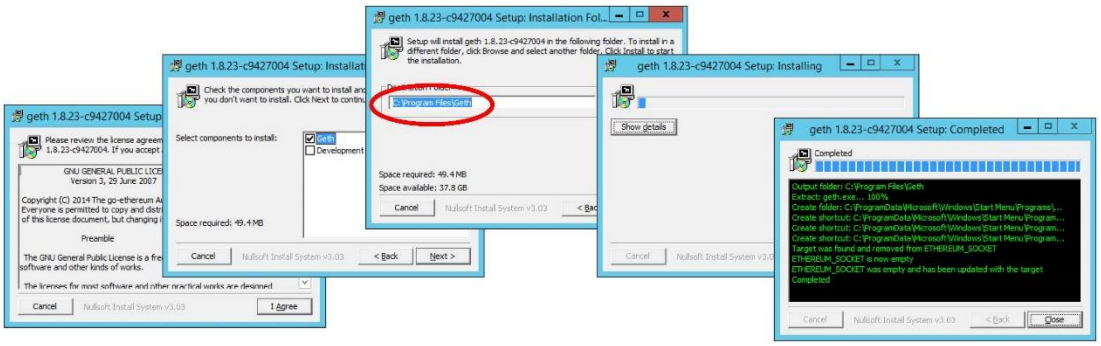
G. Δημοσίευση στο swarm με τη βοήθεια του Geth (Πραγματικό DApp)

Geth download

- ιστοσελίδα https://geth.ethereum.org/downloads/



Geth εγκατάσταση (στο c:\geth)



Geth εκτέλεση (σε μία γραμμή – χωρίς κενά)

```
c:\geth\geth --ropsten
    --syncmode "light"
    --rpc
    --rpcapi  db,eth,net,web3,personal,admin
    --cache=1024
    --rpcport 8545
```

Σε άλλο terminal

```
    c:\geth\geth attach http://127.0.0.1:8545
        web3.eth.syncing
           (false)
```

Geth (δημιουργία νέου πορτοφολιού)

```
c:\geth\geth account new
```

```
INFO [09-08|13:47:55.647] Maximum peer count                       ETH=50 LES=0 total=50
Your new account is locked with a password. Please give a password. Do not forget this password.
Password:
Repeat password:

Your new key was generated

Public address of the key:   0xD3B3742aE6AE5D6D455421a66e5345098c64De7e
Path of the secret key file: C:\Users\user\AppData\Local\Ethereum\keystore\UTC--2021-09-08T10-48-08.650232200Z--
d3b3742ae6ae5d6d455e421a66e5345098c64de7e

- You can share your public address with anyone. Others need it to interact with you.
- You must NEVER share the secret key with anyone! The key controls access to your funds!
- You must BACKUP your key file! Without the key, it's impossible to access account funds!
- You must REMEMBER your password! Without the password, it's impossible to decrypt the key!
```

Η διεύθυνση του νέου πορτοφολιού του Geth. Χωρίς καθόλου Ether. Δεν θα μας χρειαστούν.

Swarm εκτέλεση

- swarm download από

https://ethersphere.github.io/swarm-home/downloads/

- Μετακίνηση του swarm.exe στον φάκελο geth (προαιρετικά)
- Εκτέλεση…

Το πορτοφόλι του Geth που φτιάξαμε στην προηγούμενη διαφάνεια.

c:\geth\swarm --ens-api "" --bzzaccount 0xD3B3742aE0AE9D6D45E421a06eE345098c64De7e

- Δοκιμή του τοπικού Gateway του swarm δίνοντας στον browser http://localhost:8500/

Swarm δημοσίευση φακέλου

```
cd <φάκελος με όλο το προς δημοσίευση υλικό>
c:\geth\swarm --recursive up ./
```

Όταν η μεταφορά ολοκληρωθεί, εμφανίζεται ένας 256bit αριθμός. Αυτός αποτελεί το «όνομα» του site μας αν συνδυαστεί με πύλη Swarm

```
b68d103738e5181422a161e663d52dff94eeefe862c3174e6f0f45fff5c8fe15
```

- Δοκιμή στον browser τοπικά (όσο λειτουργεί ο τοπικός κόμβος Geth)

http://localhost:8500/bzz:/b68d103738e5181422a161e663d52dff94eeefe862c3174e6f0f45fff5c8fe15/index.html

- Δοκιμή παγκόσμια

https://swarm-gateways.net/bzz:/b68d103738e5181422a161e663d52dff94eeefe862c3174e6f0f45fff5c8fe15/index.html

Αν ο υπολογιστής σας (ο κόμβος Geth) μείνει εκτός Swarm για μεγάλο διάστημα, οι σελίδες σας θα διαγραφούν.

Swarm & IPFS

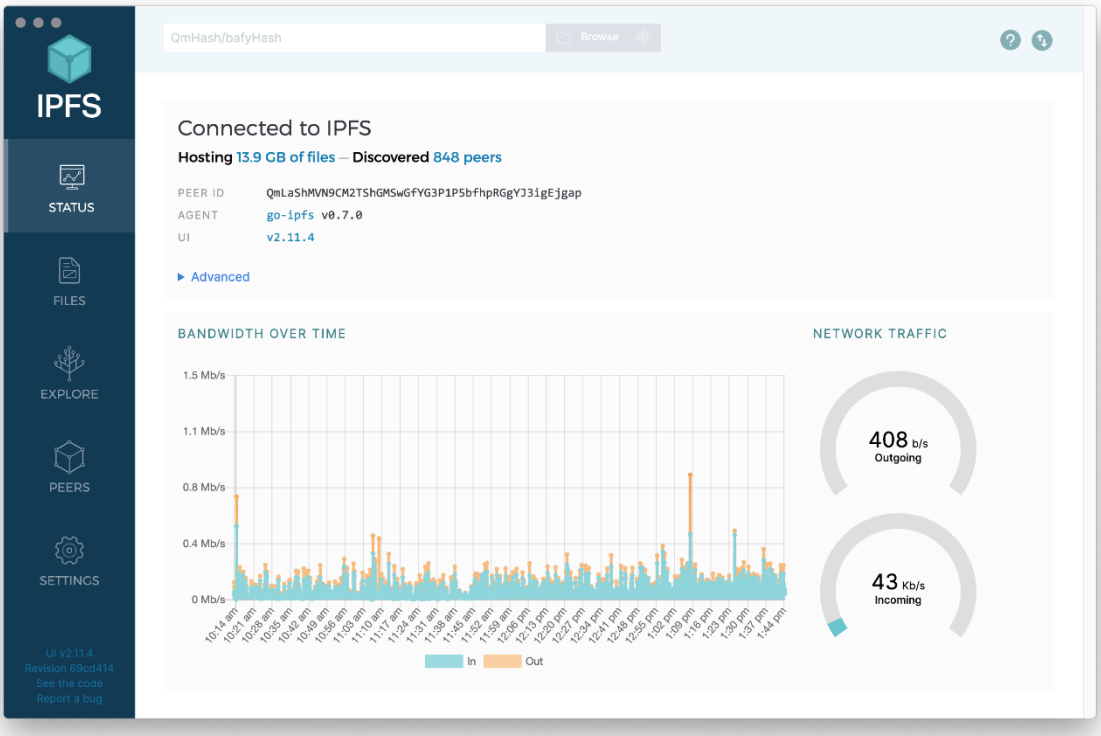
- Παλιό (προς κατάργηση;)… https://swarm-gateways.net/
- Νέο αλλά με πληρωμή (bee)… https://gateway.ethswarm.org/share
  - Μορφή:
    bzz-raw://77ef9a8e2f1b328164084b64229f94d40602ab58fddc80c15ae60122dabe06bf ή
    bzz:/a5d36b1726674f84f662a37cde13662b0ab17bd4f338c5fb1f6c8f2e7389c38a

    Πριν το bzz:/ ή το bzz-raw:// (ens vs dns)
    https://swarm-gateways.net/
- IPFS: https://anarkrypto.github.io/upload-files-to-ipfs-from-browser-panel/public/

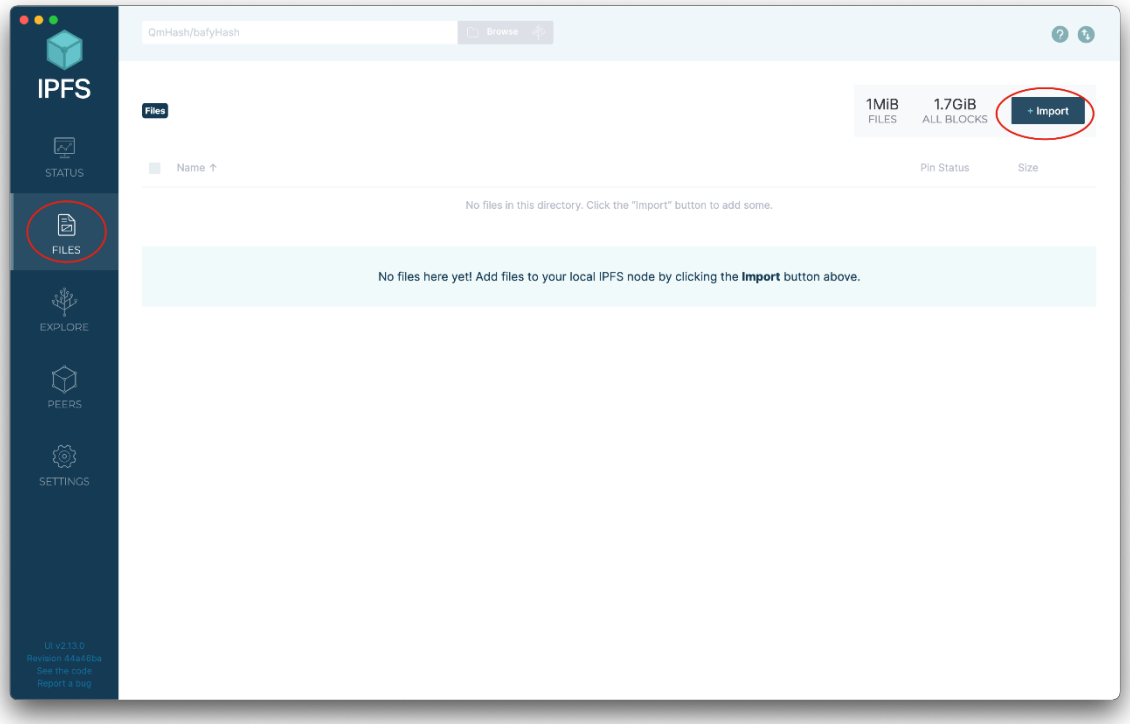
- https://app.ens.domains/, https://enslisting.com/
- https://www.toptal.com/dapp/ethereum-name-service-dapp-tutorial

Η. Δημοσίευση στο IPFS (Πραγματικό DApp)

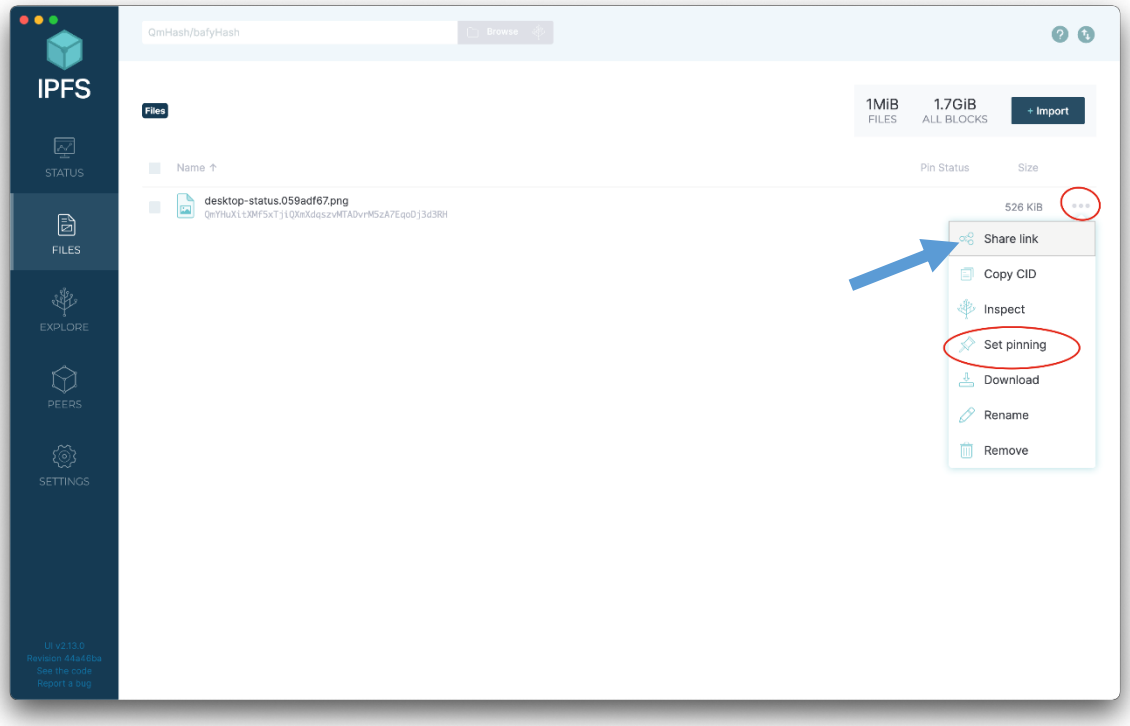
Κατεβάστε από https://github.com/ipfs/ipfs-desktop/releases και εγκαταστήστε το IPFS Desktop.

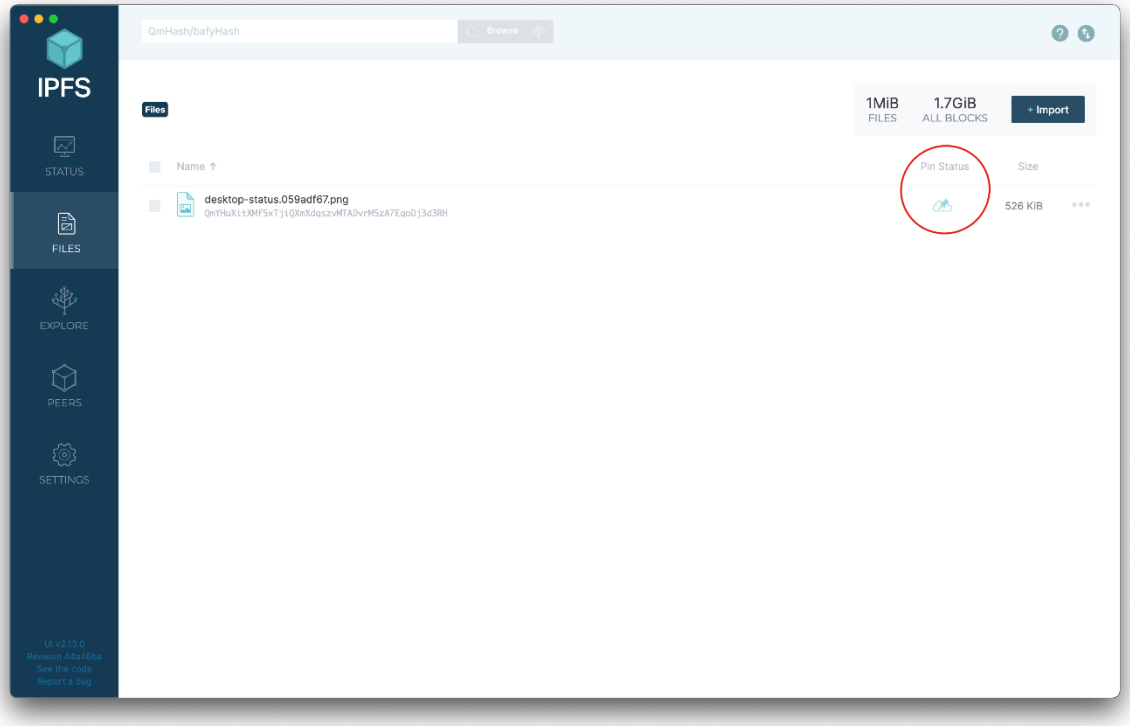


Εισάγετε τον φάκελο web.



Καρφιτσώστε τον.





Στον firewall δώστε δικαιώματα εισερχομένων κλήσεων στην εφαρμογή IPFS desktop. Η σελίδα σας θα βρίσκεται στη διεύθυνση:

https://ipfs.io/ipfs/QmaANroi53RkARqXP8dPJJfbHzfNw9xfqnQuNwJBPd1Gc4 όπου το τελευταίο τμήμα του URL είναι το CID. Όλο το link δίνεται αν κάνετε share link. Αν ο υπολογιστής σας μείνει εκτός IPFS για μεγάλο διάστημα, οι σελίδες σας θα διαγραφούν.