

Documentation:

Program Title: Web3.js and Solidity Contract Demo

Objective:

This program demonstrates how to use Web3.js and a Solidity contract to initiate a payment and send an email. Users will input their wallet details, email subject, email body, and optionally attach a file. Upon submission, the program will interact with the smart contract to initiate the payment and then send the email using an SMTP library (Python implementation not included).

Dependencies:

- Web3.js version 1.5.3
- ethers.js version 5.2.umd.min.js
- jQuery version 3.3.1

HTML Structure:

- `<!DOCTYPE html>`: Document type declaration.
- `<html>`: HTML root element.
- `<head>`: Contains metadata and included scripts.
- `<meta charset="UTF-8">`: Specifies the character encoding.
- `<title>`: Title of the webpage.
- `<script src="https://cdn.jsdelivr.net/npm/web3@1.5.3/dist/web3.min.js">`: Includes the Web3.js library.
- `<script src="https://cdn.ethers.io/lib/ethers-5.2.umd.min.js" integrity="sha384-PcxkuaG68T6Rr+3e1GnKcTLhXdO8o3HP20UZrT3LMKi/bAylG9MT2LZ8zDGsjVUx" crossorigin="anonymous">`: Includes the ethers.js library.
- `<script src="https://cdn.jsdelivr.net/npm/jquery/3.3.1/jquery.min.js">`: Includes the jQuery library.
- `</head>`: End of the head section.
- `<body>`: The main content of the webpage.
- `<h1>`: Heading for the payment and email sending form.
- `<form id="paymentForm" enctype="multipart/form-data">`: Form element with an id and enctype attributes.
- `<label for="walletDetails">`: Label for the wallet details input.
- `<input type="text" id="walletDetails" required>`: Input field for wallet details, which is required.
- `
`: Line break.
- `<label for="emailSubject">`: Label for the email subject input.
- `<input type="text" id="emailSubject" required>`: Input field for the email subject, which is required.
- `
`: Line break.
- `<label for="emailBody">`: Label for the email body input.
- `<textarea id="emailBody" rows="4" required>`: Text area for the email body, which is required and has 4 rows.
- `</textarea>`: End of the text area.
- `
`: Line break.
- `<label for="attachment">`: Label for the attachment input.
- `<input type="file" id="attachment">`: File input for attaching files.
- `
`: Line break.
- `<input type="submit" value="Submit Payment and Send Email">`: Submit button to initiate the payment and email sending process.

- `</form>`: End of the form.
- `
`: Line break.
- `<h2>`: Heading for the transaction and email sending result.
- `<div id="result">`: Empty div element with an id to display the result of the transaction and email sending process.
- `</div>`: End of the div.
- `<script>`: JavaScript code starts here.
- `const web3 = new Web3('<your_testnet_rpc_endpoint>');`: Creates a new instance of Web3.js with the RPC endpoint of the Rootstock testnet.
- `const contractAddress = '<your_contract_address>';`: Assigns the address of the deployed Solidity contract to a variable.
- `const contractABI = <contract_abi>`: Assigns the ABI (Application Binary Interface) of the contract to a variable.
- `const contract = new web3.eth.Contract(contractABI, contractAddress);`: Creates a new instance of the contract using the ABI and contract address.
- `$('#paymentForm').submit(async function(event) {`: Listens for the form submission event and performs the following actions:
 - `event.preventDefault();`: Prevents the default form submission behavior.
 - `const walletDetails = $('#walletDetails').val();`: Retrieves the value of the wallet details input field.
 - `const emailSubject = $('#emailSubject').val();`: Retrieves the value of the email subject input field.
 - `const emailBody = $('#emailBody').val();`: Retrieves the value of the email body textarea.
 - `const file = $('#attachment')[0].files[0];`: Retrieves the attached file from the file input field.
 - `try {`: Starts a try-catch block to handle errors.
 - `const accounts = await web3.eth.getAccounts();`: Retrieves the user's accounts using Web3.js.
 - `const sender = accounts[0];`: Assigns the first account as the sender of the transaction.
 - `const transaction = await contract.methods.<your_method_name>(walletDetails, emailSubject, emailBody).send({from: sender, value: web3.utils.toWei('<your_payment_amount_ether>', 'ether')});`: Calls the specified method in the contract to initiate the payment and email sending process.
 - `$('#result').text('Payment and email sent successfully!');`: Displays a success message in the result div.
 - `} catch (error) {`: Catches and handles any errors that occur.
 - `console.error('Error:', error);`: Logs the error to the console.
 - `$('#result').text('Error: ' + error.message);`: Displays the error message in the result div.
- `</script>`: End of the JavaScript code.
- `</body>`: End of the body section.
- `</html>`: End of the HTML document.

Note:

- Placeholders within the code should be replaced with appropriate values, as mentioned in the program.
- Implement your own back-end code and external dependencies (like SMTP) to handle the email sending process.
- Serve the HTML file using a web server or a local development server like Live Server extension in Visual Studio Code.