

CE340
31 ม.ค. 66

Basic Decentralized Application
(Dapp)

Web App/Page Components

- 3 Parts
 - Content (HTML)
 - Static
 - Dynamic
 - Behavior / Operation / Processing
 - Javascript
 - Look & Feel (Theme)
 - CSS
- Model-View-Control (MVC)
 - Model => Data / Information
 - View => User Interfaction (UI)
 - Control => Program

Simple Web App

- Libraries
 - jQuery (Behavior)
 - Bootstrap (CSS)
 - Truffle / Smart Contract (Model / Data / Information)

jQuery

- Input/Output
 - Text/number
- Event handling
 - Mouse click

Ready Function

- `$(document).ready(function () {`
 - Read input (text)
 - Processing
 - Display output (text)
 - Create event callback (click)
- `});`
- Short form
- `$(() => {`
- `});`

Read Input

- Select target element to read
 - `<input>`
 - Define “id”
 - Select with “id” e.g. `$('#input1')`
 - Command =>
 - `const Input = $('#input1').val()`
- Clear input
 - `$('#input1').val('')`

Display Output

- Select target element to display
 - `<p class="output"></p>`
 - `text()` function
 - `$('.output').text('hello') => <p class="output">hello</p>`
 - `html()` function
 - `$('.output').html('Hello there!')`
 - `<p class="output">`
 - `Hello there!`
 - `</p>`

Event Handling

- Select target element to attach callback
- `<button id="buttonA" /> | <input type="submit" id="btn2" />`
- `$(<target-element>).on(<event_name>, <call_back>)`
- `$('#buttonA').on('click', e => {`
 - `alert(e);`
 - `console.log(e);`
- `});`

Truffle

- Dapp Development Framework
- Create new smart contract
 - `"truffle create contract <contract_name>"`
- Compile
 - `"truffle compile"`
- Deployment to Blockchain (Real network/simulator)
 - `"truffle migrate"`
- Start Blockchain
 - `"truffle develop"`

Development Process

- Install Truffle (Once)
 - “npm install -g truffle”
- Create new project folder
- Populates starting files
 - “truffle unbox <project_name>” project_name => “metacoin”
- Install required libraries
 - “npm install jquery bootstrap lite-server web3 @truffle/contract”
- Create front-end webpage
 - HTML (.html) + jQuery / javascript (.js) + Bootstrap (.css)

Development Process (cont'd)

- Create new smart contract
 - “truffle create contract <contract_name>”
- Compile
 - “truffle compile”
- Start Blockchain Simulator
 - “truffle develop”
- Deploy Smart contract
 - Requires migration script
 - Requires smart contract interface (web app)
 - “truffle migrate”
- Start Web server
 - “npm run dev” ➔ lite-server

Contract Migration

- Requires
 - Migration script (javascript)
 - Default resides in “migrations” folder
 - Default filename is “deploy_contracts.js”

Smart contract Development

- Create “web3” object
 - Provided by Crypto wallet e.g. “metamask”
 - Built-in from Blockchain simulator (truffle + ganache)
- Communicate with web3 provider
- Web3 Object
 - Create smart contract (proxy) object on webapp
- Interac with Smart contract through proxy functions