



PoSt-Satoshi era
dApp. Useful begin

PS: 3
Created: 2019

```
static bool init(CURL *&conn, char *url)
{
    CURLcode code;
    conn = curl_easy_init();
    if (conn == NULL)
        fprintf(stderr, "Failed to create CURL connection\n");
    exit(EXIT_FAILURE);

    code = curl_easy_setopt(conn, CURLOPT_ERRORBUFFER,
                           errorBuffer);
    if (code != CURLE_OK)
        fprintf(stderr, "Failed to set error buffer [%d]\n",
                code);
    return false;

    code = curl_easy_setopt(conn, CURLOPT_URL, url);
    if (code != CURLE_OK)
        fprintf(stderr, "Failed to set URL [%s]\n", errorBuffer);
    return false;

    code = curl_easy_setopt(conn, CURLOPT_FOLLOWLOCATION,
                           1);
    if (code != CURLE_OK)
        fprintf(stderr, "Failed to set redirect option [%s]\n",
                errorBuffer);
    return false;

    code = curl_easy_setopt(conn, CURLOPT_WRITEFUNCTION,
                           write),
    if (code != CURLE_OK)
```



```
static void StartElement(
    Context *context,
    const char *name,
    const char **attributes,
    void *userData)
{
    if (COMPARE(name, "title"))
        context->title = context->addTitle();
    else if (COMPARE(name, "description"))
        context->addDescription();
}

// libxml end element
static voidEndElement(
    Context *context,
    const char *name,
    void *userData)
{
    if (COMPARE(name, "title"))
        context->addTitle();
}

// Text handling
static void handleCharacterData(
    Context *context,
    const char *data,
    int length,
    void *userData)
{
    if (context->addTitle())
        context->title.append(data, length);
}

// libxml PCDATA callback
static void CharacterDataHandler(
    void *userData,
    const XML_Char *data,
    int length)
```

CONTENTS

I	General Instructions	4
I	PART 00: Introduction	5
II	PART 01: Wallet	6

GENERAL INSTRUCTIONS

Technical organization:

Submit your finished tasks to the GIT repository.
<https://xteams-gitlab.unit.ua/module-N-login>

Only work submitted to the repository will be considered for mentor-evaluation. Any extraneous files will be considered against you if this is not justified by any serious cause.

Deadline:

Access to the repository for making records closes after 7 days since tasks were presented at 09:01:00 AM GMT+2

XBIP: 7 - 11:03:2018 at 09:01:00 AM GMT+2

INTRODUCTION

This week you will be implement Decentralized Application on the top of Ethereum blockchain. It will be a wallet with useful functional. Further, you can push your project to GitHub with GNU license and provide the elegant solution for the community, which lack of the robust helper tools for the asset management (hello to Parity Wallet). Considering past mistakes, keep you application logic simple, especially on the smart contract side. Also, please provide ability to run the project on the localhost for the more safety usage.

PART 01: Wallet

Wallet will consist of two parts - front-end application (dApp) and smart contracts. In details below.

Dapp

Create Wallet dApp for smart contracts interaction. There is UI application from the front-end side of the development. You allow to use any SPA frameworks (or even purely javascript) and libs and helpers like web3.js for communication with Ethereum blockchain.

Wallet need to has features:

Create/Import private key and get public key and address from it
Set imported/created wallet as current wallet
Send ether to another address

Asterisk task: Wallet has to interact with ERC20 tokens. User can add the token contract address and get token information and access to send tokens to another address

Interface of this project has to be minimalistic and not disgusting. Feel free to use any of interface libs and kits.

Smart contracts

There is a part of Wallet dApp from the blockchain side of the development. Extend the wallet capabilities with a multisig functional where multiple users can:

- Initialize the new contract with signers (factory approach).
- Create proposals to send an amount ether to an address
- Sign proposals
- Reject proposals

There will be not a copy-paste gnosis wallet or something else, but you can be inspire by open-source projects.

Please add access to this functional through the Wallet dApp.