CKB Builder Handbook

Welcome to the CKB Builder Handbook! This is a resource for all cohort members participating on the **Community Keeps Building** Builders' track.

Here, you'll find quick links for important resources that will help structure your study, starting from the basics, towards more advanced concepts.

Please note: some documents may refer to Lumos or Capsule for performing tasks. These tools are now deprecated and the best tools are mentioned below.

Introduction

Introduction to Nervos CKB - basic technical concepts and terminology explained

Getting started on CKB -

      Quick start - set up your dev environment and project quickly using OffCKB. Further information on networks and RPCs are available from this page.

CKB Academy lessons 1 and 2 - further learning to build your theoretical knowledge

Introduction to Script (smart contracts) - a introduction to smart contracts on CKB

As well as working through the above, **it is essential to start your own dev log on GitHub to record your own learning and progress.** Example: GitHub - Methemeticz/ckb-builder-track: I log what I learn as part of the Nervos Builder track.

Beginner

Basic exercises - once you have set up your environment and understood the technical concepts, get started on these tutorials:

Please note: for all tutorials and practical exercises, please remember to retain screenshots as proof of participation and completion and record on your dev log

      Transfer CKB
      Store Data on Cell

Create Fungible Token
Create DOB (digital object)
Build a Simple Lock

The **L1 developer course** is another resource to understand the theoretical aspects of CKB. Many Lab exercises here use Lumos or Capsule which you can ignore.

Learn building on CKB (Javascript, Typescript), using CCC (Common Chain Connector), a beginner-friendly all-in-one tool to build applications with integrated support for major wallets:

Explore **CCC App**
Test your code in the **CCC Playground**
Code examples
API

For other programming languages:

Rust SDK
        CKB-CLI - a CKB command-line tool developed in Rust
Go
Java

Other devtools, including:

Testnet faucet
CKB Debugger
CKB Tools website

Task: create your own basic application

Now you can put what you've learned above to the test!

Use your previous study and the tools above to create a basic application. It could, for example, allow users to generate their own tokens or mint spores (dobs). Discuss your ideas with the programme director (Neon) and with members of CKB DevRel, they will be able to help guide you. If you have some development background already, you may wish to be more adventurous.

There is always the chance a good idea opens the door to funding from the Spark programme or the CKB Community Fund DAO!

Further reading

Deepen your understanding of CKB using this collection of core documents:
GitHub - nervosnetwork/rfcs: This repository contains proposals, standards and documentations related to Nervos Network.

Intermediate

Want to further your understanding of CKB Development? Here are some modules to get you going:

Script development course

    Class 1: Validation Model
    Class 2: Script Basics
    Class 3: UDT
    Class 4: WebAssembly on CKB
    Class 5: Debugging
    Class 6: Type ID
    Class 7: Advanced Duktape Examples
    Class 8: Performant WASM
    Class 9: Cycle Reductions in Duktape Script
    Class 10: Language Choices

Detailed Rust scripting

Detailed JS scripting

Ecosystem scripts and libraries

Understanding Molecule and serialization

Simple UDTs (sUDT)
A standard that defines the most basic implementation of a UDT (user defined token) fungible token on Nervos. An sUDT on Nervos is the equivalent of Ethereum tokens standards ERC20 or ERC777.

    Standards document
    Tutorial

Nervos DAO
Nervos DAO enable users to lock CKBytes to get compensation from Nervos CKB secondary issuance. This process is similar to staking on other platforms. Nervos DAO provides a "virtual hardcap" for CKByte holders to insulate them from inflation.

[Introduction](#)
[Deposit and Withdraw in Nervos DAO](#)
[Source code](#)
[NervDAO - a browser implementation](#)

Spore (DOBs)
Spore Protocol infuses digital objects (DOBs) with enduring value backed by tokenomics, redeemable at any time. Ensures true on-chain ownership, privacy, creative freedom and frictionless interaction.

[Spore protocol introduction](#)
[Comprehensive Spore documentation and tutorials](#)
[DOB Cookbook](#)
[How-to Recipes](#)
[SDK](#), **[Examples](#)**, **[Demos](#)**, **[Contracts](#)**
[DOB Decoder Standalone Server](#)

Advanced

If you want to challenge yourself further and really grow your skills, study the following resources.

Script-Sourced Rich Information (SSRI)
SSRI is is a protocol for strong binding of relevant information and conventions to the Script itself on CKB. It allows developers to embed arbitrary information/logic into the Script, describing its behavior and managing data to off-chain applications.

[Introduction](#)
[SDK for implementing SSRI-compliant smart contracts on CKB](#)
    [First prototype of SSRI-Compliant contract](#)
    [Server for calling SSRI methods](#)
[Further Documentation](#)
[Pausable UDT: The first fully SSRI compliant and production ready contract that exemplifies all use cases that SSRI protocool covers.](#)
    [Audit report](#)

RGB++ - Bitcoin's Turing Catalyst

RGB++ is an **asset issuance protocol for Bitcoin that leverages the flexibility and programmability of CKB** to enable dynamic use cases for Bitcoin assets. It unlocks the full potential of Bitcoin for a rich application ecosystem without altering its foundational security. This production-ready protocol offers a trustless method for dApps to interact directly with native Bitcoin transactions, facilitated by a lightweight, decentralized verifier. Website: www.rgbpp.com

> **[Introduction](#)**
> **[RGB++ Protocol Light Paper](#)**
> **[RGB++ Resource Hub](#)**, including Script Standard, CKB Bitcoin SPV, API service for asset management, API endpoints and development tools
> **[RGB++ SDK](#)**
> **[RGB++ Explorer](#)**

xUDT - Extensible User Defined Token standard

xUDT (Extensible User-Defined Token) is a token standard for creating and managing fungible tokens on Nervos CKB. xUDT builds upon the foundational Simple User-Defined Token (sUDT), which uses a minimal, predefined validation model. In contrast, xUDT introduces extensibility by allowing developers to attach custom validation logic via external Scripts — enabling more advanced governance, minting rules, and token behaviors.

> **[Introduction](#)**
> [How xUDT works](#)
> [Enhance sUDT's Programmability with xUDT](#)
> [xUDT RFC](#)

iCKB

iCKB is a protocol that aims to solve the illiquidity problem of the NervosDAO by tokenizing NervosDAO deposits into a liquid iCKB token. Website: www.ickb.org

> **[Whitepaper](#)**
> **[Github repository](#)**
> **[Audit report](#)**
> **[iCKB SDK](#)**