

Developer's Playbook Outline

ID	MAR-2766
Issue Type	Sub-task
Add to Content Calendar	<input type="checkbox"/>
Design needed?	<input type="checkbox"/>
Legal required?	<input type="checkbox"/>
Status	Approved
Due Date	@August 13, 2025
Assignee	(Y) Yulia Petukhova
Watchers	 Ginevra Maggi  Vanina I
Created by	(Y) Yulia Petukhova
Platform/Category	
Parent task	<u>Neon Dev Bootcamp - Retention Phase</u>
Level	

▼ About the Project

What is Developer's Playbook

Developer's Playbook is an open-source, community-driven knowledge hub tailored specifically for blockchain developers.

The purpose of this project

- Boost retention and maintain engagement by giving existing developers a platform to publish their ideas and showcase their

How it works

Contributor flow:

1. Developer discovers this on X or Linkedin as an opportunity to build portfolio by contributing to an open-source project.

It provides articles and curated resources, structured and organized by learning journeys for mastering blockchain topics, with a strong focus on Neon EVM, Ethereum, and Solana.

The idea is that it's materials that developers find helpful in their dev journeys and want to share with other devs: "from devs - by devs - to devs".

The content is collaboratively created by the Neon team and the developer community, proofread and curated by Neon team.

At the first stages, the content will be mainly provided by Neon team and bootcampers, then ideally - by new developers.

skills through meaningful contributions to an open-source project, which is visible as GitHub commits and public proof of work that they can leverage in their resumes & portfolio.

- Attract new developers by offering them opportunities to build their reputation as subject matter experts through publishing materials in the Playbook.

2. Developer goes to the Playbook and sees options to "Contribute to Playbook" or "Edit this page" and clicks on either of them.

3. Dev taken to GitHub - either to Playbook's LP with Readme where he/she reads rules of "How to contribute", or to a particular page that he/she chose to edit.

4. Dev makes edits and submits a pull request, providing his/her Twitter (to mention on the page he contributed) and Discord handles (to add to the closed VIP channel).

5. Neon's devrel reviews the pull request and accepts (and edits if needed) or declines.

6. Successful contributor is added to the VIP Discord channel.
7. Neon shares a new contribution on socials, tagging the dev.

Website Structure

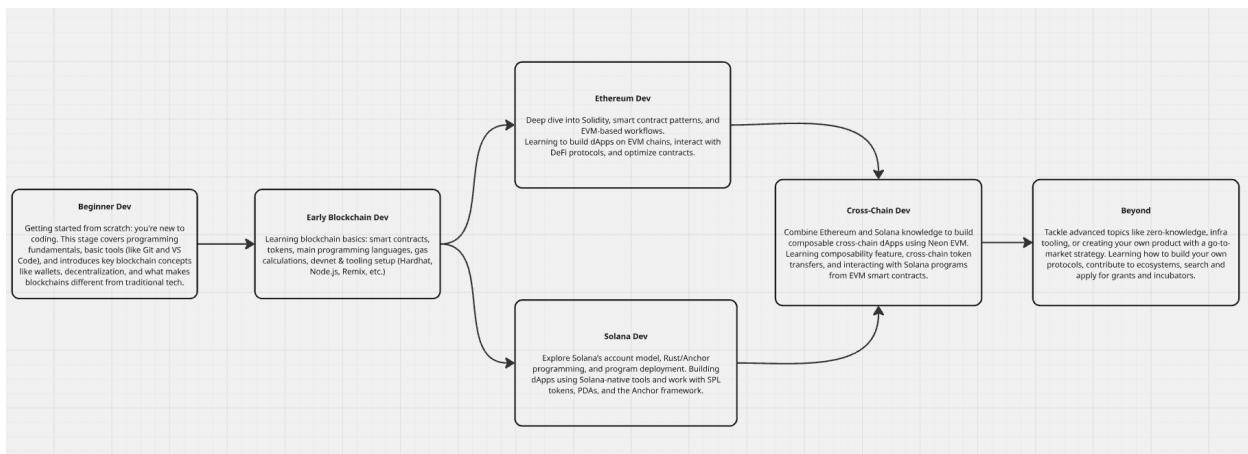
Approximate reference: <https://groundzero.superteam.fun/>

Header:

- Left: Neon Logo - leads to Neonevm.org
- Right: Contribute to Playbook button - leads to project on GitHub

Landing Page:

- What is Developer's Playbook
- Diagram of dev journey (most of the content will be structured according to these sections)



Side Menu (sections reflect dev journey; sub-sections are organized by topics):

- Beginner dev
 - Basic Programming Concepts
 - Programming Languages
 - Tools & Setup
 - Blockchain Foundations
 - Wallets and Security
 - etc.
- Early stage blockchain dev
 - Smart Contracts & Languages
 - Tooling & Architecture
 - Tokens & Transactions
 - Connecting to Frontend
- Ethereum dev
 - Advanced Solidity
 - Token Standards
 - Tooling
 - DeFi
 - Oracles
 - Data Indexing
- Solana dev
 - Program Logic
 - Account Management
 - Token Program
 - etc.
- Cross-chain dev (here we'll post about Neon)
 - Neon Composability
 - SDKs & Interfaces
 - Cross-Chain DEX Integration
 - etc.
- Beyond: advanced topics
 - Advanced Protocol Design & Cryptography

- Infrastructure & DevOps
 - Product Development & GTM
 - Grants, Funding & Growth
 - Developer Relations
 - Ecosystem Contributions & Careers
-
- Resources

Content Outline

▼ Beginner dev

Basic Programming Concepts

Programming Languages

- Intro to JavaScript
- Intro to Python
- ...etc

Tools & Setup

- Folder structure and version control
- Using a terminal & GitHub basics
- Installing Git and VS Code
- ...etc

Blockchain Foundations

- What is a blockchain?
- Decentralization & consensus
- ...etc

Wallets & Security

- What is a wallet? (Metamask, Phantom)
- Public/private keys
- Creating and managing wallets
- ...etc

▼ Early stage blockchain dev

Smart Contracts & Languages

- Intro to smart contracts
- Solidity + Hardhat basics
- Rust + Anchor basics
- ...etc

Tooling & Architecture

- Devnet setup and deployment
- EVM vs Solana: how they differ
- Remix
- Anchor CLI
- ...etc

Tokens & Transactions

- Token standards
- Sending/receiving tokens
- Understanding gas, fees, and limits
- ...etc

Connecting to Frontend

- Using Viem or Ethers.js

- Connecting smart contracts to frontend apps
- Scaffold
- ...etc

▼ Ethereum dev

Advanced Solidity

- Solidity cheat sheet
- Gas optimization techniques
- Proxy contracts
- ...etc

Token Standards

- ERC-721, ERC-1155, ERC-4626
- Custom token logic
- ...etc

Tooling

- Hardhat workflows
- Foundry
- ...etc

DeFi

- Liquidity pools and LP tokens
- How liquidity pools and automated market makers (AMMs) function under the hood
- Token swaps and pricing mechanics
- How to interact with smart contracts for lending, borrowing, swapping, and staking

Oracles

- Chainlink basics
- Price feeds and automation

Data Indexing

- Writing subgraphs using The Graph
- Querying contract data with graphQL
- Envio
- Subsquid

▼ Solana dev

Program Logic

- Lifecycle of Solana programs
- Rust programming essentials
- Anchor framework
- ...etc

Account Management

- Associated Token Accounts (ATAs)
- PDAs
- ...etc

Token Program

- Minting, transferring, and burning SPL tokens
- Creating token metadata
- ...etc

▼ Cross-chain dev

Neon Composability

- Using Neon EVM composability libraries

- Accessing Solana programs from Solidity
- ERC20ForSPL token
- ...etc

SDKs & Interfaces

- Neon Solana Native SDK
- Connecting wallets for cross-chain UX
- Scheduling and signing Solana transactions from EVM

Cross-Chain DEX Integration

- Building swap flows using Raydium program library
- Accessing Solana liquidity pools with Neon EVM

▼ Beyond: advanced topics

Advanced Protocol Design & Cryptography

- Zero-knowledge proofs
- Account abstraction & smart wallets
- Multi-party computation (MPC)
- MEV

Infrastructure & DevOps

- Running validator / RPC nodes
- Indexing with The Graph, Substreams, or custom solutions
- Monitoring tools (e.g. Tenderly, Blockscout, Prometheus + Grafana)
- Gas optimization & contract size limits
- Versioning, migrations, and upgradeability patterns

Product Development & GTM

- Designing and launching your own protocol or dApp

- Go-to-market (GTM) strategies for blockchain products
- Finding product-market fit in Web3

Grants, Funding & Growth

- How to apply for ecosystem grants (Ethereum, Solana, etc.)
- Writing winning grant proposals
- Participating in accelerators and incubators
- Managing budgets and milestones in grant-based development

Developer Relations

- Becoming a DevRel: writing, speaking, educating
- Running community calls, hackathons, and technical workshops

Ecosystem Contributions & Careers

- How to contribute to L1/L2 core development
- Web3 career paths: independent builder, protocol engineer, ecosystem lead
- Building a Web3 portfolio and proving "proof of work"

Content Types

Content in the Playbook can take different forms, including:

- Curated links with descriptions: external resources like blog posts, videos, documentation, or repos
- Original articles: tutorials, deep dives, maybe even personal dev journeys
- Mini-guides: one-topic focused pieces that explain a concept clearly

- Code examples: snippets or GitHub links showing how something works in practice
- Visual explainers with descriptions: e.g. diagrams & illustrations accompanied by short descriptions

All contributions should map to a topic and learning track. So it's a collection of resources linked to a stage of dev's journey.