# Substrate Kitties Overview

Learn how to make a dApp on Polkadot

Polkadot Intermediate Rust 2 hours +

Written by Kresna Sucandra

☰ Show Table of Contents

## Introduction

Welcome to the Substrate Kitties tutorial. This 4 part tutorial series will teach you everything you need to know to build a blockchain designed to handle the creation and ownership management of Substrate Kitties. Before jumping into the next section, let's have a look at what we'll be doing.

*This is a 4 part tutorial series that steps you through building a dApp for managing Substrate Kitties from scratch. Each part could take 30-60 minutes to complete depending on your level of experience. This tutorial only covers the blockchain part and does not include the front-end part.*

> ℹ You may find it useful to come back to this page as you progress through each part — just to keep track of the bigger picture.

Part I: Basic setup

Part II: Uniqueness, custom types & storage maps

Part III: Dispatchables, Events and Errors

Part IV: Interacting with your Kitties

## Learning outcomes

➡️ Write and integrate a custom FRAME pallet to your runtime.

➡️ Use structs in storage and how to create and update storage items.

➡️ Write extrinsics and helper functions.

# What we're building

In this tutorial, we'll intentionally keep things simple so that you can decide on how you'd like to improve your Substrate Kitties chain. For the purposes of what we're building, Kitties really can only do the following things:

😺 Be created either by some original source or by being bred using existing Kitties.

😼 Be sold at a price set by their owner.

😿 Be transferred from one owner to another.

# What we won't cover

The following fall outside the scope of this tutorial:

- Writing tests for our pallet.
- Declaring a configuration for the genesis of our chain.
- Customize frontend for our Substrate Kitties.

You can refer to the Substrate how-to guides on how to do this once you've completed this tutorial series.

---

Bringing things down to a more granular level, this translates to the following application design:

1. **Basic setup**. We'll need to spin up a Substrate node and create a custom pallet

2. **Runtime storage**. We'll need a total of 9 storage items in our pallet to keep track of the amount of Kitties; their index; their owners and their owner account IDs.

3. **Dispatchable functions**. We'll need a total of 5 dispatchable functions: `create` , `set_price` , `transfer` , `buy_kitty` and `breed_kitty`

4. **Private functions**. We'll write 2 helper functions to handle randomness: `increment_nonce` and `random_hash`

5. **Helper functions**. We'll write 2 helper functions for our dispatchable functions: `mint` and `transfer_from` .

Follow each step at your own pace — the goal is for you to learn and the best way to do that is to try it yourself! Before moving on from one section to the next, make sure your pallet builds without any error. You'll be writing most of the code yourself! Use the template files here to help you complete each part.

Click here to go to get started with the project setup!