

Custom Session Storage Tutorial

Use Custom storage for session keys module with Node JS

[## Introduction](#) The session storage module provides a way to create sessions for an App, where users can make use of the signing purpose. These session keys are stored on chain, and information for each leaf is stored in offchain data storage. By default it gets stored in the browser local storage. This tutorial covers how one can create a custom Session storage for their Dapps. It can be a database or file. In this we will go over the implementation of the File storage for their session leaf. [## Environment Set up](#) If you already through the quick start guide this section follows the same initial steps. You can skip this and go straight to the next section if you want to implement into your existing quick start codebase. [## Initialize Smart Account](#) In this step we'll set up our node js script to create a smart account in our command prompt. [## Create Custom session](#) Now users can make their own implementation of Session implementing ISessionStorage interface and pass it to the SessionKeyManager module instance. Let's create a new file called customSession and start writing our own session storage. [## Session Module](#) As described in the last section, an off-chain session is essential for the session leafs. In this context, we will employ Files for storing this data. Any customized storage layer must adhere to a specific interface, ensuring the correct implementation of methods for saving or retrieving information. Alternatively, one can develop their own implementation using a Database server layer. This guide will lead you through the process of creating a fundamental Node.js script using TypeScript that enables users to initiate a session. [## ERC20 transfer](#) We will need the bundler however the paymaster in this case. If you want to sponsor transactions later or allow users to pay for gas in erc20 tokens make sure to set this up but for the purposes of this tutorial we will not be sponsoring any transactions. The paymaster URL here will only work for tutorial contracts. [Previous Batch ERC20 Transfer with Session Router](#) [Next Introduction](#)