

# Remix Storage Factory

The screenshot displays the Remix IDE interface. On the left, the 'FILE EXPLORERS' sidebar shows a workspace named 'default\_workspace' containing a folder 'contracts' and several files: 'StorageFactory.sol', 'ExtraStorage.sol' (which is selected), 'README.txt', and 'SimpleStorage.sol'. The main editor area shows the code for 'StorageFactory.sol'. The code includes a Solidity pragma statement, an import for 'SimpleStorage.sol', and a contract definition for 'StorageFactory'. The contract contains a public array of 'SimpleStorage' objects, a function to create and add new 'SimpleStorage' contracts, a function to store data in a specific contract, and a function to retrieve data from a specific contract.

```
1 // SPDX-License-Identifier: MIT
2
3 pragma solidity ^0.8.0;
4
5 import "./SimpleStorage.sol";
6
7 contract StorageFactory {
8
9     SimpleStorage[] public simpleStorageArray;
10
11     function createSimpleStorageContract() public {
12         SimpleStorage simpleStorage = new SimpleStorage();
13         simpleStorageArray.push(simpleStorage);
14     }
15
16     function sfStore(uint256 _simpleStorageIndex, uint256 _simpleStorageNumber) public {
17         // Address
18         // ABI
19         // SimpleStorage(address(simpleStorageArray[_simpleStorageIndex])).store(_simpleStorageNumber);
20         simpleStorageArray[_simpleStorageIndex].store(_simpleStorageNumber);
21     }
22
23     function sfGet(uint256 _simpleStorageIndex) public view returns (uint256) {
24         // return SimpleStorage(address(simpleStorageArray[_simpleStorageIndex])).retrieve();
25         return simpleStorageArray[_simpleStorageIndex].retrieve();
26     }
27 }
28
```

