**BuyContract:**[**https://github.com/hemanthbhushan/SnippperBot/tree/master**](https://github.com/hemanthbhushan/SnippperBot/tree/master)

The BuyContract is a Solidity smart contract that allows users to perform token swaps on the Uniswap v2 decentralized exchange.

**State Variables**

**UNISWAP\_V2\_ROUTER**: This variable stores the address of the Uniswap v2 router contract.

**WETH**: This variable stores the address of the Wrapped Ether (WETH) token contract.

**Functions:**

**swap:**

function swap(

address \_tokenIn,

address \_tokenOut,

uint256 \_amountIn,

uint256 \_amountOutMin,

address \_to

) external

This function performs a token swap from one token to another. It accepts the following parameters:

**\_tokenIn**: The address of the token to trade out of.

**\_tokenOut**: The address of the token to receive in the trade.

**\_amountIn**: The amount of tokens to send in.

**\_amountOutMin**: The minimum amount of tokens expected to receive.

**\_to**: The address to send the output tokens to.

The function performs the following steps:

Transfers the specified amount of tokens \_amountIn from the caller to the contract using the transferFrom function of the ERC20 token contract \_tokenIn.

Approves the Uniswap router contract to spend the transferred tokens using the approve function of the ERC20 token contract \_tokenIn.

Constructs the token swap path based on the provided tokens \_tokenIn and \_tokenOut. If either token is WETH, the path is [tokenIn, tokenOut]. Otherwise, the path is [tokenIn, WETH, tokenOut].

Calls the swapExactTokensForTokens function of the Uniswap router contract to perform the token swap, passing \_amountIn, \_amountOutMin, path, \_to, and the current block timestamp as arguments.

**getAmountOutMin:**

function getAmountOutMin(

address \_tokenIn,

address \_tokenOut,

uint256 \_amountIn

) external view returns (uint256)

This function calculates the minimum amount of \_tokenOut tokens that will be received for a given input amount of \_tokenIn tokens. It accepts the following parameters:

**\_tokenIn**: The address of the token to trade out of.

**\_tokenOut**: The address of the token to receive in the trade.

**\_amountIn**: The amount of tokens to send in.

The function performs the following steps:

Constructs the token swap path based on the provided tokens \_tokenIn and \_tokenOut. If either token is WETH, the path is [tokenIn, tokenOut]. Otherwise, the path is [tokenIn, WETH, tokenOut].

Calls the getAmountsOut function of the Uniswap router contract to get an array of token amounts that will be received for the given input amount \_amountIn and the constructed path.

Returns the minimum amount of \_tokenOut tokens, which is the last element of the amountOutMins array.

Dependencies

**The BuyContract relies on the following external Solidity contracts:**

**IUniswapV2Router02**: This contract provides the interface for the Uniswap v2 router, allowing token swaps and retrieval of token amounts.

IUniswapV2Factory: This contract provides the interface for the Uniswap v2 factory, allowing retrieval of the pair address for two tokens.

IUniswapV2Pair: This contract provides the interface for Uniswap v2 pairs, allowing retrieval of the token reserves and other pair information.

**IERC20**: This contract provides the interface for ERC20 tokens, allowing token transfers and approvals.

The BuyContract contract uses the addresses of these external contracts to interact with them and perform the desired token swaps.

Please note:

(IUniswapV2Router02, IUniswapV2Factory, IUniswapV2Pair, ) are already deployed and available at the specified addresses.