



## **RugFreeCoins Audit**



## **ADA Boy Token Audit**

## **Smart Contract Security Audit**

**August 04, 2021**



# Contents

Audit details	1
Disclaimer	2
Background	3
About the project	4
Target market and the concept	6
Potential to grow with score points	7
Total Points	7
Contract details	8
Top token holders	9
Token distribution	10
Contract interaction details	11
Contract code function details	12
Contract description table	13
Security issue checking status	23
Owner privileges	24
Audit conclusion	27

# Audit details



## **Audited project**

ADA Boy Token



## **Contract Address**

0x1E653794A6849bC8A78be50C4D48981AfAD6359d



## **Client contact**

ADA Boy Team



## **Blockchain**

Binance smart chain



## **Project website**

<https://adaboytoken.com/>

# Disclaimer

This is a limited report on our findings based on our analysis, in accordance with good industry practice as at the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, the details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report. While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn't say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the disclaimer below – please make sure to read it in full.

**DISCLAIMER:** By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice. No one shall have any right to rely on the report or its contents, and Rugfreecoins and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (Rugfreecoins) owe no duty of care towards you or any other person, nor does Rugfreecoins make any warranty or representation to any person on the accuracy or completeness of the report. The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and Rugfreecoins hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, Rugfreecoins hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against Rugfreecoins, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report. The analysis of the security is purely based on the smart contracts alone. No applications or operations were reviewed for security. No product code has been reviewed.

# Background

Rugfreecoins was commissioned by ADA Boy to perform an audit of the smart contract.

**<https://bscscan.com/token/0x1E653794A6849bC8A78be50C4D48981AfAD6359d>**

The focus of this audit is to verify that the smart contract is secure, resilient and working according to the specifications.

The information in this report should be used to understand the risk exposure of the smart contract, project feasibility, long term sustainability and as a guide to improve the security posture of the smart contract by remediating the issues that were identified.

# About the project

ADA Boy is a token built on the Binance Smart Chain. Each transaction, purchase incur a 12% fee, and sales incur an 13% fee.

- ❖ The **automatic ADA reward of 7%** is what ADA Boy's entire marketing strategy is based around: that tokens will be distributed among every holder proportional to how many tokens each individual holds.
- ❖ The liquidity fee of 3%, which is a redistribution mechanism that ensures the trading pool always has sufficient liquidity. This is a key element for decentralized exchanges like Pancakeswap.
- ❖ The **sustainability fee of 1% marketing** is what allows ADA Boy to hold the aforementioned promise. Tokens will be swapped into BNBs and will be sent to a marketing wallet per transaction. This way, ADA Boy will have enough funds to promote the coin and spend for future development without selling tokens as the traditional way.

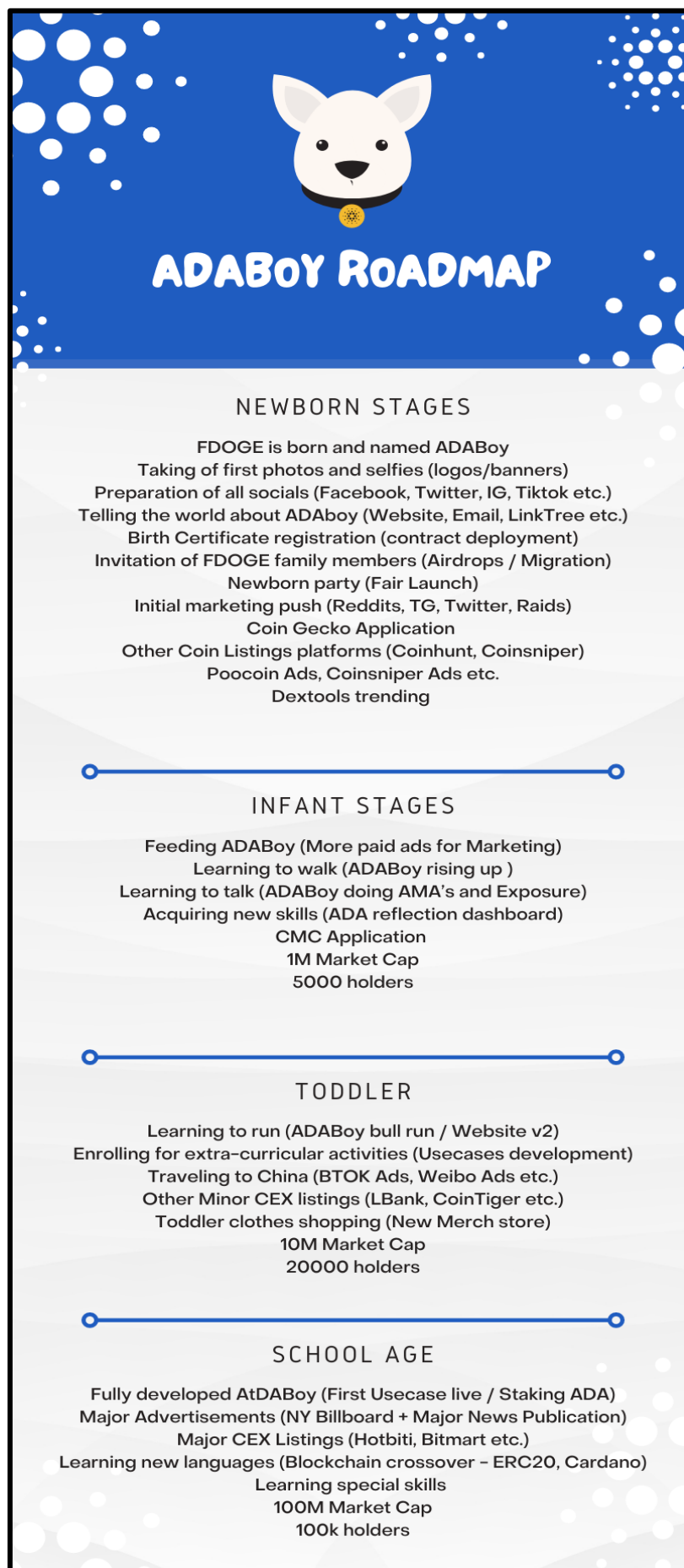
## Tokenomics

### 12% fee when buying

- 8% of trade goes to holders pockets in ADA.
- 3% of trade goes to the liquidity pool.
- 1% of trade goes to marketing.

### 13% fee when selling

- 8% of trade goes to holders pockets in ADA.
- 4% of trade goes to the liquidity pool.
- 1% of trade goes to marketing.



# Target market and the concept

## Target market

- ❖ Anyone who's interested in Crypto space with long term investment plans.
- ❖ Anyone who's ready to earn a passive income in ADA by holding tokens.
- ❖ Anyone who's interested in trading tokens.
- ❖ All Cardano inventors and fans out there.
- ❖ Anyone who's interested in stake ADA or ADA Boy and earn rewards.
- ❖ Anyone who's interested in taking part with the future lottery and games that's going to be built by the ADA Boy team.
- ❖ Anyone who's interested in making financial transactions with any other party using ADA Boy or ADA as the currency.

## Core concept

### The ADA reward system

8% of each transaction gets converted to ADA, and is split amongst all holders. The rewards are sent to holders that have at least 200,000 ADA Boy tokens, holders will be eligible to receive tokens every one hour and rewards are proportional to how many tokens each individual holds.

### Sustainable mechanism

**The liquidity fee of 3%**, which is a redistribution mechanism that ensures the trading pool always has sufficient liquidity.

The **fee of 1% marketing** is what allows ADA Boy to promote the token and use funds to further development of the platform. Tokens will be swapped into BNBs and will be sent to a marketing wallet per transaction. This way, SugarDaddyDoge will have access to the funds without selling tokens as the traditional way, which will enable them to consume funds without hurting the project.

### The future plan

#### Dapp site to stake ADA or ADA Boy

The ADA Boy team is working on building a DApp for the holders to stake their ADA Boy tokens or ADA for a longer period and earn rewards.



## Gaming and Lottery platform

The platform will allow users to use ADABoy tokens to play planned lottery-like games and earn pools of prizes.

# Potential to grow with score points

1.	Project efficiency	8/10
2.	Project uniqueness	8/10
3	Information quality	8/10
4	Service quality	8/10
5	System quality	7/10
6	Impact on the community	8/10
7	Impact on the business	8/10
8	Preparing for the future	8/10
Total Points		<b>7.88/10</b>

# Contract details

## Token contract details for 04<sup>th</sup> August 2021

<b>Contract name</b>	ADA Boy
<b>Contract address</b>	0x1E653794A6849bC8A78be50C4D48981AfAD6359d
<b>Token supply</b>	100,000,000,000
<b>Token ticker</b>	ADABoy
<b>Decimals</b>	18
<b>Token holders</b>	1,344
<b>Transaction count</b>	26,803
<b>Top 100% holders dominance</b>	86.00%
<b>Dividend Tracker</b>	0x82ba7c718bcd70338e33295e302aacb3598c29d0
<b>Marketing wallet address</b>	0xf2bbc71d8c7209d4fac76e72abdc216c0d28c7b7
<b>Contract deployer address</b>	0x252b4Ed3516F8921b36A71363b7047eb9cF069eF
<b>Contract's current owner address</b>	0x656b9875624e5bcece96910e5fb840a90386840b

# Top token holders

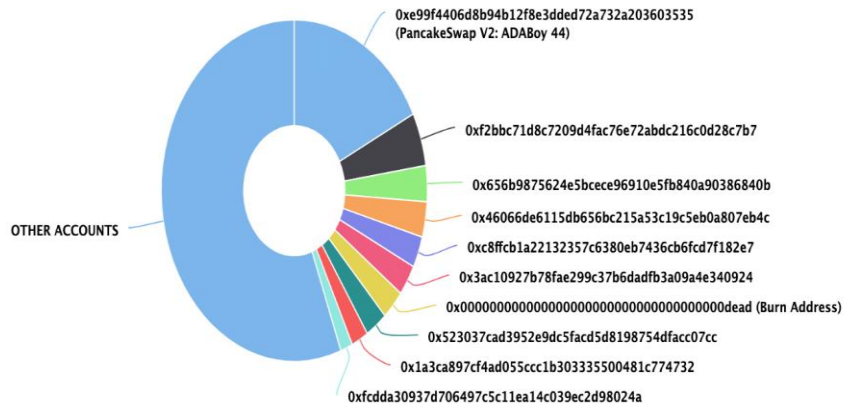
## Top 10 Token Holders

The top 10 holders collectively own 44.32% (44,317,877,590.24 Tokens) of ADA BOY

Token Total Supply: 100,000,000,000.00 Token | Total Token Holders: 1,345

ADA BOY Top 10 Token Holders

Source: BscScan.com

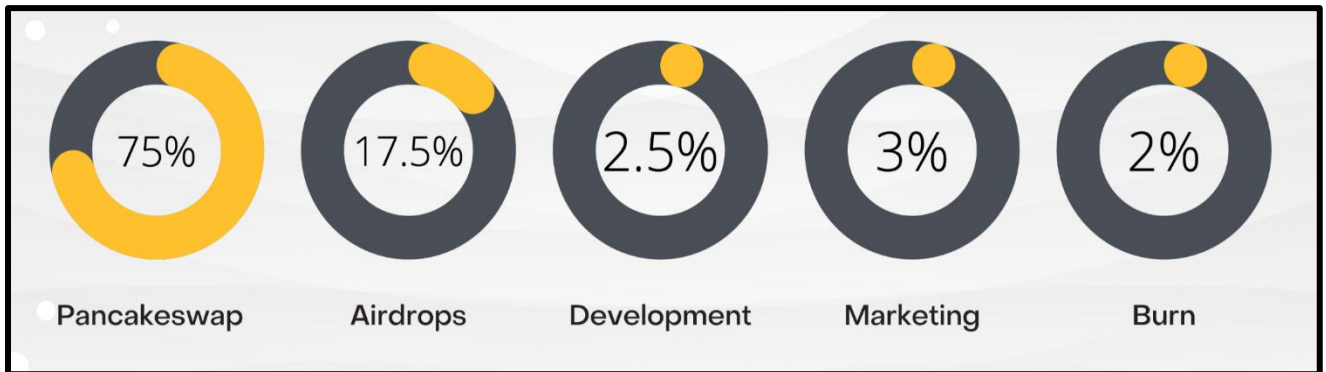


(A total of 44,317,877,590.24 tokens held by the top 10 accounts from the total supply of 100,000,000,000.00 token)

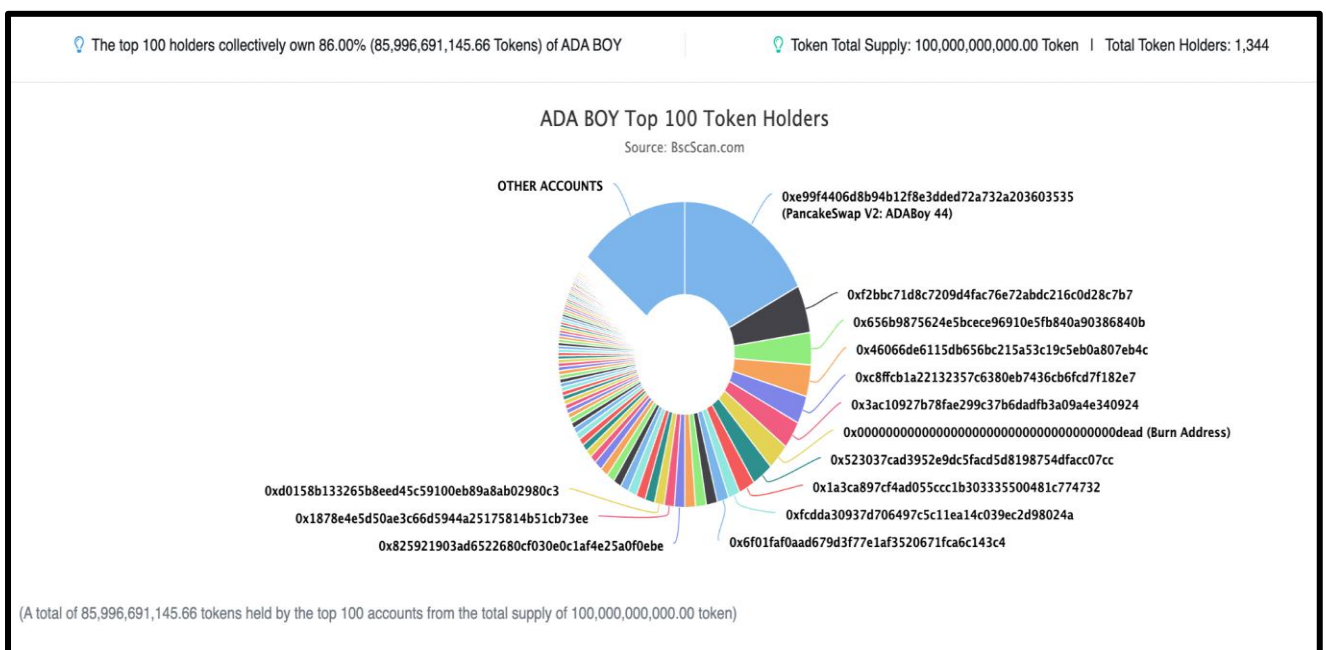
Rank	Address	Quantity (Token)	Percentage
1	<a href="#">PancakeSwap V2: ADABoy 44</a>	17,591,765,408.922960656707579969	17.5918%
2	<a href="#">0xf2bbc71d8c7209d4fac76e72abdc216c0d28c7b7</a>	5,048,748,698.204324386241564139	5.0487%
3	<a href="#">0x656b9875624e5bcece96910e5fb840a90386840b</a>	3,408,519,454.566849817546685562	3.4085%
4	<a href="#">0x46066de6115db656bc215a53c19c5eb0a807eb4c</a>	3,350,150,028.181101183375396682	3.3502%
5	<a href="#">0xc8ffcb1a22132357c6380eb7436cb6cd7f182e7</a>	2,948,000,000	2.9480%
6	<a href="#">0x3ac10927b78fae299c37b6dadfb3a09a4e340924</a>	2,922,335,928.198470844085341955	2.9223%
7	<a href="#">Burn Address</a>	2,816,000,000	2.8160%
8	<a href="#">0x523037cad3952e9dc5facd5d8198754dfacc07cc</a>	2,689,964,699.418913309245252763	2.6900%
9	<a href="#">0x1a3ca897cf4ad055ccc1b303335500481c774732</a>	2,042,072,416.657829250332636638	2.0421%
10	<a href="#">0xfcdda30937d706497c5c11ea14c039ec2d98024a</a>	1,500,320,956.087683013058714048	1.5003%

# Token distribution

Tokens are distributed as follows:

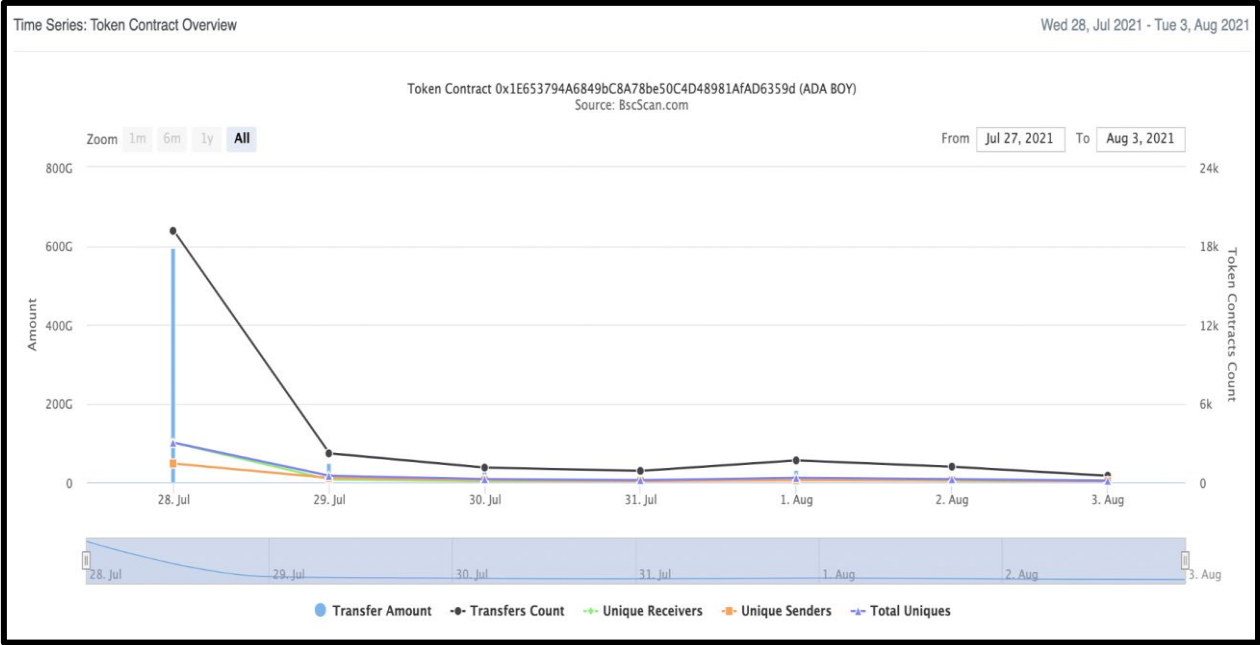


## Top 100 Token Holders





# Contract interaction details


































# Contract code function details

No	Category	Item	Result
1	Coding conventions	BRC20 Token standards	pass
		compile errors	pass
		Compiler version security	pass
		visibility specifiers	pass
		Gas consumption	pass
		SafeMath features	pass
		Fallback usage	pass
		tx.origin usage	pass
		deprecated items	pass
		Redundant code	pass
		Overriding variables	pass
2	Function call audit	Authorization of function call	pass
		Low level function (call/delegate call) security	pass
		Returned value security	pass
		Selfdestruct function security	pass
3	Business security	Access control of owners	pass
		Business logics	pass
		Business implementations	pass
4	Integer overflow/underflow		pass
5	Reentrancy		pass
6	Exceptional reachable state		pass
7	Transaction ordering dependence		pass
8	Block properties dependence		pass
9	Pseudo random number generator (PRNG)		pass
10	DoS (Denial of Service)		pass
11	Token vesting implementation		pass
12	Fake deposit		pass
13	Event security		pass



































# Contract description table

Below table represents the summary of the contracts and methods in the token contract. We scanned the whole contract and listed down all the Interfaces, functions and implementations with its visibility and mutability.





































Contract	Type	Bases		
L	Function Name	Visibility	Mutability	Modifiers
<b>ADABOYTOKEN</b>	<b>Implementation</b>	<b>ERC20, Ownable</b>		
L		Public 		ERC20
L		External 		NO 
L	updateDividendTracker	Public 		onlyOwner
L	updateUniswapV2Router	Public 		onlyOwner
L	excludeFromFees	Public 		onlyOwner
L	excludeMultipleAccountsFromFees	Public 		onlyOwner
L	setMarketingWallet	External 		onlyOwner
L	setADARewardsFee	External 		onlyOwner
L	setLiquiditFee	External 		onlyOwner
L	setMarketingFee	External 		onlyOwner
L	startTrading	External 		onlyOwner
L	setAutomatedMarketMakerPair	Public 		onlyOwner
L	blacklistAddress	External 		onlyOwner
L	_setAutomatedMarketMakerPair	Private 		
L	updateGasForProcessing	Public 		onlyOwner
























L	updateClaimWait	External ⚡	⚙️	onlyOwner
L	getClaimWait	External ⚡		NO⚡
L	getTotalDividendsDistributed	External ⚡		NO⚡
L	isExcludedFromFees	Public ⚡		NO⚡
L	withdrawableDividendOf	Public ⚡		NO⚡
L	dividendTokenBalanceOf	Public ⚡		NO⚡
L	excludeFromDividends	External ⚡	⚙️	onlyOwner
L	getAccountDividendInfo	External ⚡		NO⚡
L	getAccountDividendInfoAtIndex	External ⚡		NO⚡
L	processDividendTracker	External ⚡	⚙️	NO⚡
L	claim	External ⚡	⚙️	NO⚡
L	getLastProcessedIndex	External ⚡		NO⚡
L	getNumberOfDividendTokenHolders	External ⚡		NO⚡
L	_transfer	Internal 🔒	⚙️	
L	swapAndSendToFee	Private 🔒	⚙️	
L	swapAndLiquify	Private 🔒	⚙️	
L	swapTokensForEth	Private 🔒	⚙️	
L	swapTokensForADA	Private 🔒	⚙️	
L	addLiquidity	Private 🔒	⚙️	
L	swapAndSendDividends	Private 🔒	⚙️	



<b>ADABoyDividendTracker</b>	<b>Implementation</b>	<b>Ownable, DividendPaying Token</b>		
L		Public 		DividendPayingToken
L	_transfer	Internal 		
L	withdrawDividend	Public 		NO 
L	excludeFromDividends	External 		onlyOwner
L	updateClaimWait	External 		onlyOwner
L	getLastProcessedIndex	External 		NO 
L	getNumberOfTokenHolders	External 		NO 
L	getAccount	Public 		NO 
L	getAccountAtIndex	Public 		NO 
L	canAutoClaim	Private 		
L	setBalance	External 		onlyOwner
L	process	Public 		NO 
L	processAccount	Public 		onlyOwner
<b>DividendPayingToken</b>	<b>Implementation</b>	<b>ERC20, Ownable, DividendPaying TokenInterface, DividendPaying TokenOptionalInterface</b>		
L		Public 		ERC20
L	distributeADADividends	Public 		onlyOwner
L	withdrawDividend	Public 		NO 

L	_withdrawDividendOfUser	Internal 🔒	⬛	
L	dividendOf	Public 🔓		NO⚠
L	withdrawableDividendOf	Public 🔓		NO⚠
L	withdrawnDividendOf	Public 🔓		NO⚠
L	accumulativeDividendOf	Public 🔓		NO⚠
L	_transfer	Internal 🔒	⬛	
L	_mint	Internal 🔒	⬛	
L	_burn	Internal 🔒	⬛	
L	_setBalance	Internal 🔒	⬛	
<b>ERC20</b>	<b>Implementation</b>	<b>Context, IERC20, IERC20Metadata</b>		
L		Public 🔓	⬛	NO⚠
L	name	Public 🔓		NO⚠
L	symbol	Public 🔓		NO⚠
L	decimals	Public 🔓		NO⚠
L	totalSupply	Public 🔓		NO⚠
L	balanceOf	Public 🔓		NO⚠
L	transfer	Public 🔓	⬛	NO⚠
L	allowance	Public 🔓		NO⚠
L	approve	Public 🔓	⬛	NO⚠
L	transferFrom	Public 🔓	⬛	NO⚠
L	increaseAllowance	Public 🔓	⬛	NO⚠
L	decreaseAllowance	Public 🔓	⬛	NO⚠















L	_transfer	Internal 		
L	_mint	Internal 		
L	_burn	Internal 		
L	_approve	Internal 		
L	_beforeTokenTransfer	Internal 		
<b>IERC20</b>	<b>Interface</b>			
L	totalSupply	External 		NO 
L	balanceOf	External 		NO 
L	transfer	External 		NO 
L	allowance	External 		NO 
L	approve	External 		NO 
L	transferFrom	External 		NO 
<b>IERC20Metadata</b>	<b>Interface</b>	<b>IERC20</b>		
L	name	External 		NO 
L	symbol	External 		NO 
L	decimals	External 		NO 
<b>Context</b>	<b>Implementation</b>			
L	_msgSender	Internal 		
L	_msgData	Internal 		
<b>SafeMath</b>	<b>Library</b>			
L	add	Internal 		
L	sub	Internal 		
L	sub	Internal 		






L	mul	Internal 		
L	div	Internal 		
L	div	Internal 		
L	mod	Internal 		
L	mod	Internal 		
<b>SafeMathUint</b>	<b>Library</b>			
L	toInt256Safe	Internal 		
<b>SafeMathInt</b>	<b>Library</b>			
L	mul	Internal 		
L	div	Internal 		
L	sub	Internal 		
L	add	Internal 		
L	abs	Internal 		
L	toUint256Safe	Internal 		
<b>DividendPayingToken Interface</b>	<b>Interface</b>			
L	dividendOf	External 		NO 
L	withdrawDividend	External 		NO 
<b>DividendPayingToken OptionalInterface</b>	<b>Interface</b>			
L	withdrawableDividen dOf	External 		NO 
L	withdrawnDividendO f	External 		NO 
L	accumulativeDividen dOf	External 		NO 





Ownable	Implementation	Context		
L		Public ¶	⊗	NO¶
L	owner	Public ¶		NO¶
L	renounceOwnership	Public ¶	⊗	onlyOwner
L	transferOwnership	Public ¶	⊗	onlyOwner
IterableMapping	Library			
L	get	Public ¶		NO¶
L	getIndexOfKey	Public ¶		NO¶
L	getKeyAtIndex	Public ¶		NO¶
L	size	Public ¶		NO¶
L	set	Public ¶	⊗	NO¶
L	remove	Public ¶	⊗	NO¶
IUniswapV2Pair	Interface			
L	name	External ¶		NO¶
L	symbol	External ¶		NO¶
L	decimals	External ¶		NO¶
L	totalSupply	External ¶		NO¶
L	balanceOf	External ¶		NO¶
L	allowance	External ¶		NO¶
L	approve	External ¶	⊗	NO¶
L	transfer	External ¶	⊗	NO¶
L	transferFrom	External ¶	⊗	NO¶
L	DOMAIN_SEPARATOR	External ¶		NO¶

L	PERMIT_TYPEHAS H	External ¶		NO¶
L	nonces	External ¶		NO¶
L	permit	External ¶	⦿	NO¶
L	MINIMUM_LIQUIDIT Y	External ¶		NO¶
L	factory	External ¶		NO¶
L	token0	External ¶		NO¶
L	token1	External ¶		NO¶
L	getReserves	External ¶		NO¶
L	price0CumulativeLa st	External ¶		NO¶
L	price1CumulativeLa st	External ¶		NO¶
L	kLast	External ¶		NO¶
L	mint	External ¶	⦿	NO¶
L	burn	External ¶	⦿	NO¶
L	swap	External ¶	⦿	NO¶
L	skim	External ¶	⦿	NO¶
L	sync	External ¶	⦿	NO¶
L	initialize	External ¶	⦿	NO¶
<b>IUniswapV2Factory</b>	<b>Interface</b>			
L	feeTo	External ¶		NO¶
L	feeToSetter	External ¶		NO¶
L	getPair	External ¶		NO¶
L	allPairs	External ¶		NO¶
L	allPairsLength	External ¶		NO¶
L	createPair	External ¶	⦿	NO¶

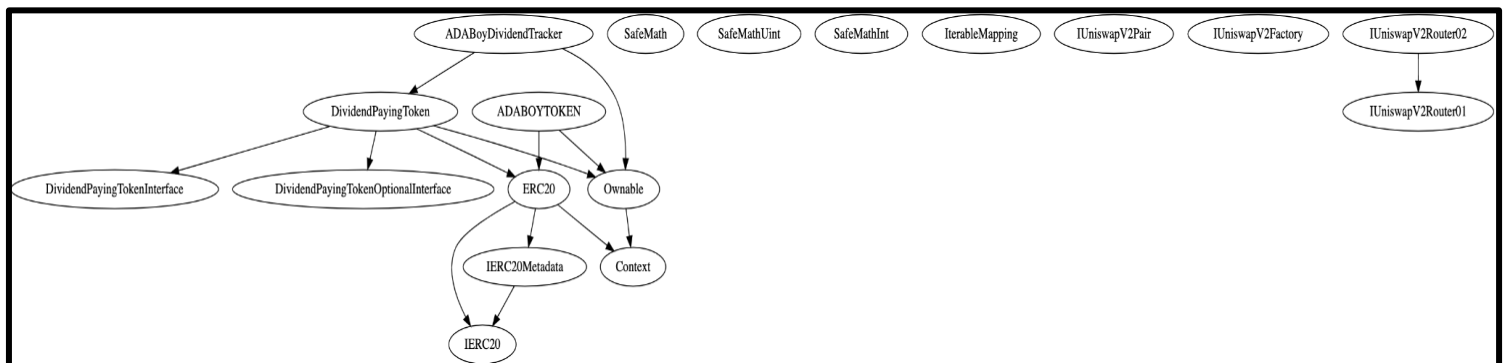
L	setFeeTo	External ¶		NO¶
L	setFeeToSetter	External ¶		NO¶
<b>IUniswapV2Router01</b>	<b>Interface</b>			
L	factory	External ¶		NO¶
L	WETH	External ¶		NO¶
L	addLiquidity	External ¶		NO¶
L	addLiquidityETH	External ¶		NO¶
L	removeLiquidity	External ¶		NO¶
L	removeLiquidityETH	External ¶		NO¶
L	removeLiquidityWithPermit	External ¶		NO¶
L	removeLiquidityETHWithPermit	External ¶		NO¶
L	swapExactTokensForTokens	External ¶		NO¶
L	swapTokensForExactTokens	External ¶		NO¶
L	swapExactETHForTokens	External ¶		NO¶
L	swapTokensForExactETH	External ¶		NO¶
L	swapExactTokensForETH	External ¶		NO¶
L	swapETHForExactTokens	External ¶		NO¶
L	quote	External ¶		NO¶
L	getAmountOut	External ¶		NO¶
L	getAmountIn	External ¶		NO¶
L	getAmountsOut	External ¶		NO¶
L	getAmountsIn	External ¶		NO¶

IUniswapV2Router02	Interface	IUniswapV2Router01		
L	removeLiquidityETH SupportingFeeOnTransferTokens	External !		NO!
L	removeLiquidityETH WithPermitSupportingFeeOnTransferTokens	External !		NO!
L	swapExactTokensForTokensSupportingFeeOnTransferTokens	External !		NO!
L	swapExactETHForTokensSupportingFeeOnTransferTokens	External !		NO!
L	swapExactTokensForETHSupportingFeeOnTransferTokens	External !		NO!

### Legend

Symbol	Meaning
	Function can modify state
	Function is payable

## Inheritance Hierarchy





# Security issue checking status

## ❖ High severity issues

No high severity issues found.

## ❖ Medium severity issues

No medium severity issues found.

## ❖ Low severity issues

No low severity issues found.

# Owner privileges

- ❖ The owner can change the router address.

```
ftrace | funcSig
function updateUniswapV2Router(address newAddress↑) public onlyOwner {
    require(
        newAddress↑ != address(uniswapV2Router),
        "ADABoy: The router already has that address"
    );
    emit UpdateUniswapV2Router(newAddress↑, address(uniswapV2Router));
    uniswapV2Router = IUniswapV2Router02(newAddress↑);
    address _uniswapV2Pair = IUniswapV2Factory(uniswapV2Router, factory())
        .createPair(address(this), uniswapV2Router.WETH());
    uniswapV2Pair = _uniswapV2Pair;
}
```

- ❖ The owner can include/exclude accounts from fees.

```
ftrace | funcSig
function excludeFromFees(address account↑, bool excluded↑) public onlyOwner {
    require(
        _isExcludedFromFees[account↑] != excluded↑,
        "ADABoy: Account is already the value of 'excluded'"
    );
    _isExcludedFromFees[account↑] = excluded↑;

    emit ExcludeFromFees(account↑, excluded↑);
}

ftrace | funcSig
function excludeMultipleAccountsFromFees(
    address[] calldata accounts↑,
    bool excluded↑
) public onlyOwner {
    for (uint256 i = 0; i < accounts↑.length; i++) {
        _isExcludedFromFees[accounts↑[i]] = excluded↑;
    }

    emit ExcludeMultipleAccountsFromFees(accounts↑, excluded↑);
}
```

- ❖ The owner can change the gas fee up to minimum 200,000 and max 500,000.

```
ftrace | funcSig
function updateGasForProcessing(uint256 newValue↑) public onlyOwner {
    require(
        newValue↑ >= 200000 && newValue↑ <= 500000,
        "ADABoy: gasForProcessing must be between 200,000 and 500,000"
    );
    require(
        newValue↑ != gasForProcessing,
        "ADABoy: Cannot update gasForProcessing to same value"
    );
    emit GasForProcessingUpdated(newValue↑, gasForProcessing);
    gasForProcessing = newValue↑;
}
```

- ❖ The owner can change the claim wait time.

```
ftrace | funcSig
function updateClaimWait(uint256 claimWait↑) external onlyOwner {
    dividendTracker.updateClaimWait(claimWait↑);
}
```

- ❖ The owner can change the marketing wallet address.

```
ftrace | funcSig
function setMarketingWallet(address payable wallet↑) external onlyOwner {
    marketingWalletAddress = wallet↑;
}
```

- ❖ The owner can change the ADA reward fee.

```
ftrace | funcSig
function setADARewardsFee(uint256 value↑) external onlyOwner {
    ADARewardsFee = value↑;
    totalFees = ADARewardsFee.add(liquidityFee).add(marketingFee);
}
```

- ❖ The owner can change the liquidity fee.

```
ftrace | funcSig
function setLiquidityFee(uint256 value↑) external onlyOwner {
    liquidityFee = value↑;
    totalFees = ADARewardsFee.add(liquidityFee).add(marketingFee);
}
```

- ❖ The owner can change the marketing fee.

```
ftrace | funcSig
function setMarketingFee(uint256 value↑) external onlyOwner {
    marketingFee = value↑;
    totalFees = ADARewardsFee.add(liquidityFee).add(marketingFee);
}
```

- ❖ The owner can enable trading.

```
ftrace | funcSig
function startTrading() external onlyOwner {
    tradingEnabled = true;
}
```

- ❖ The owner can blacklist and whitelist addresses.

```
ftrace | funcSig
function blacklistAddress(address account↑, bool value↑) external onlyOwner {
    isBlacklisted[account↑] = value↑;
}
```

- ❖ The owner can exclude accounts from dividends.

```
ftrace | funcSig
function excludeFromDividends(address account↑) external onlyOwner {
    dividendTracker.excludeFromDividends(account↑);
}
```

# Audit conclusion

While conducting the audit of the ADA Boy smart contract, it was observed that there is nothing alarming with the code.