BLOCKHAT
SECURITY

MADNFT

Smart Contract Security Audit

Prepared by BlockHat

October 15th, 2022 - October 22nd, 2022

BlockHat.io

contact@blockhat.io

Document Properties

Client	Jacob Clay
Version	1.0
Classification	Public

Scope

The MADNFT Contract in the MADNFT Repository

Repo	Owner
`https://github.com/madnfts/mad-contracts-v0.9`	MADNFTs

Files	MD5 Hash
contracts/EventsAndErrors.sol	a9dc757edc941c5f27f1472cfcc0f65d
contracts/MAD.sol	4ae2ef99d66949dc5ad8c0c34864e59b
contracts/MADFactory1155.sol	41bf8c8b3b30d13ed5a674b2f74f2f66
contracts/MADFactory721.sol	6035b3b2921cc8d7c179c1c6b3d64bc0
contracts/MADMarketplace1155.sol	145c70b2a11c21caf3177f6cb39be4a0
contracts/MADMarketplace721.sol	cc6618e0abae22734cd1d9ceea066dee
contracts/MADRouter1155.sol	a3ef028110bd4d19e0ecfbbaa0211dde
contracts/MADRouter721.sol	ccd125b57e403898929a332eb1415622

contracts/Types.sol	211938fc82b786b14a391ee5f2a8ee5f

Contacts

COMPANY	CONTACT
BlockHat	contact@blockhat.io

Contents

1 Introduction

MADNFT engaged BlockHat to conduct a security assessment on the MADNFT beginning on October 15th, 2022 and ending October 22nd, 2022. In this report, we detail our methodical approach to evaluate potential security issues associated with the implementation of smart contracts, by exposing possible semantic discrepancies between the smart contract code and design document, and by recommending additional ideas to optimize the existing code. Our findings indicate that the current version of smart contracts can still be enhanced further due to the presence of many security and performance concerns.

This document summarizes the findings of our audit.

1.1 About MADNFT

MADNFT is a nft marketplace that allows the minting of 721 and 1155 NFTs on the harmony blockchain. There is a configurable mint fee of 0.25ONE and configurable platform fee set at 10 % . User can trade other external harmony NFTs on the marketplace too.

Issuer	Jacob Clay
Website	`https://mainnet.madnfts.io`
Type	Solidity Smart Contract
Audit Method	Whitebox

1.2 Approach & Methodology

BlockHat used a combination of manual and automated security testing to achieve a balance between efficiency, timeliness, practicability, and correctness within the audit's scope. While manual testing is advised for identifying problems in logic, procedure, and implementation, automated testing techniques help to expand the coverage of smart contracts and can quickly detect code that does not comply with security best practices.

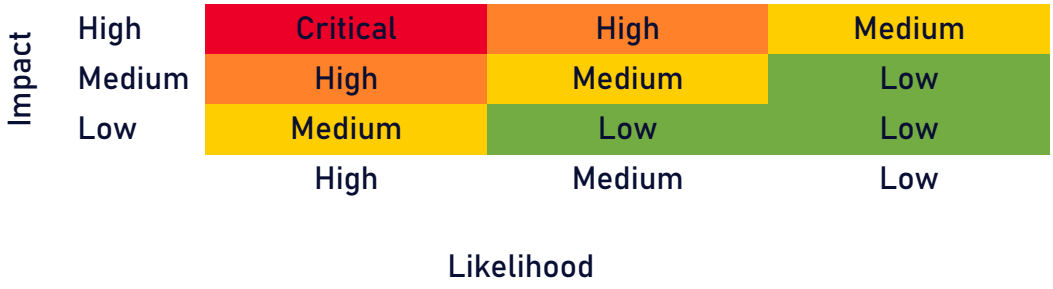
1.2.1 Risk Methodology

Vulnerabilities or bugs identified by BlockHat are ranked using a risk assessment technique that considers both the LIKELIHOOD and IMPACT of a security incident. This framework is effective at conveying the features and consequences of technological vulnerabilities.

Its quantitative paradigm enables repeatable and precise measurement, while also revealing the underlying susceptibility characteristics that were used to calculate the Risk scores. A risk level will be assigned to each vulnerability on a scale of 5 to 1, with 5 indicating the greatest possibility or impact.

- Likelihood quantifies the probability of a certain vulnerability being discovered and exploited in the untamed.
- Impact quantifies the technical and economic costs of a successful attack.
- Severity indicates the risk's overall criticality.

Probability and impact are classified into three categories: H, M, and L, which correspond to high, medium, and low, respectively. Severity is determined by probability and impact and is categorized into four levels, namely Critical, High, Medium, and Low.



2 Findings Overview

2.1 Summary

The following is a synopsis of our conclusions from our analysis of the MADNFT implementation. During the first part of our audit, we examine the smart contract source code and run the codebase via a static code analyzer. The objective here is to find known coding problems statically and then manually check (reject or confirm) issues highlighted by the tool. Additionally, we check business logics, system processes, and DeFi-related components manually to identify potential hazards and/or defects.

2.2 Key Findings

In general, these smart contracts are well-designed and constructed, but their implementation might be improved by addressing the discovered flaws, which include , 2 high-severity, 12 medium-severity vulnerabilities.

Vulnerabilities	Severity	Status
Centralization risk	HIGH	Fixed
Centralization risk	HIGH	Fixed
Owner has full control over the fees	MEDIUM	Fixed
Missing address verification	MEDIUM	Fixed
Owner has full control over the fees	MEDIUM	Fixed
Missing address verification	MEDIUM	Fixed
Missing Value Verification	MEDIUM	Fixed
Race Condition	MEDIUM	Acknowledged
Missing address verification	MEDIUM	Fixed
Missing Value Verification	MEDIUM	Fixed
Race Condition	MEDIUM	Acknowledged
Missing address verification	MEDIUM	Fixed
Missing address verification	MEDIUM	Fixed
Missing address verification	MEDIUM	Fixed

3 Finding Details

A MADRouter721.sol

A.1 Owner has full control over the fees [MEDIUM]

Description:

The owner feeMint and feeBurn can set any value in fee variable. This represent a risk on the user side.

Code:

Listing 1: MADRouter721

```
352     function setFees(
353         uint256 _feeMint,
354         uint256 _feeBurn
355     ) external onlyOwner {
356         assembly {
357             sstore(feeBurn.slot, _feeBurn)
358             sstore(feeMint.slot, _feeMint)
359         }

361         emit FeesUpdated(_feeMint, _feeBurn);
362     }
```

Risk Level:

Likelihood – 3
Impact - 3

Recommendation:

We recommend to limit the fee values by adding a require statement.

Status - Fixed

The MAD team has fixed the issue by adding require statements to make sure that the feeBurn and feeMint are less than 50.

A.2 Missing address verification [MEDIUM]

Description:

The address-type arguments newOwner and _signer should include a zero-address test, otherwise, the contract's functionality may become inaccessible. If the contract ownership is lost. You need to re-deploy the same contract again.

Code:

Listing 2: MADRouter721

```
426     function setOwner(address newOwner)
427         public
428         override
429         onlyOwner
430     {
431         // owner = newOwner;
432         assembly {
433             sstore(owner.slot, newOwner)
434         }
435
436         emit OwnerUpdated(msg.sender, newOwner);
437     }
```

Listing 3: MADRouter721

```
442     function setSigner(address _token, address _signer)
443         external
444         onlyOwner
445     {
446         ERC721Lazy(_token).setSigner(_signer);
```

```
447     }
```

Risk Level:

Likelihood – 1
Impact - 4

Recommendation:

We recommend that you make sure the addresses provided in the arguments are different from the address(0).

Status - Fixed

The MAD team has fixed the issue by adding require statements to make sure that the addresses provided in the arguments are different from the address(0).

B MADRouter1155.sol

B.1 Owner has full control over the fees [MEDIUM]

Description:

The owner can set any value in feeMint and feeBurn variables. This represent a risk on the user side.

Code:

Listing 4: MADRouter1155

```
403     function setFees(
404         uint256 _feeMint,
405         uint256 _feeBurn
406     ) external onlyOwner {
407         assembly {
```

```
408             sstore(feeBurn.slot, _feeBurn)
409             sstore(feeMint.slot, _feeMint)
410         }
411
412         emit FeesUpdated(_feeMint, _feeBurn);
413     }
```

Risk Level:

Likelihood – 3
Impact - 3

Recommendation:

We recommend to limit the fee values by adding a require statement.

Status - Fixed

The MAD team has fixed the issue by adding require statements to make sure that the feeBurn and feeMint are less than 50.

B.2 Missing address verification [MEDIUM]

Description:

The address-type arguments newOwner and _signer should include a zero-address test, otherwise, the contract's functionality may become inaccessible.

Code:

Listing 5: MADRouter1155

```
480     function setOwner(address newOwner)
481         public
482         override
483         onlyOwner
```

```
484     {
485         // owner = newOwner;
486         assembly {
487             sstore(owner.slot, newOwner)
488         }
490         emit OwnerUpdated(msg.sender, newOwner);
491     }
```

Listing 6: MADRouter721

```
496     function setSigner(address _token, address _signer)
497         external
498         onlyOwner
499     {
500         ERC721Lazy(_token).setSigner(_signer);
501     }
```

Risk Level:

Likelihood – 1
Impact - 4

Recommendation:

We recommend that you make sure the addresses provided in the arguments are different from the address(0).

Status - Fixed

The MAD team has fixed the issue by adding require statements to make sure that the addresses provided in the arguments are different from the address(0).

C MADMarketplace721.sol

C.1 Centralization risk [HIGH]

Description:

The owner can set any value in _feeVal2 and _feeVal2 variable. This represents a significant centralization risk on the user side.

Code:

Listing 7: MADMarketplace721

```
381     function setFees(
382         uint256 _feeVal2,
383         uint256 _feeVal3)
384         external
385         onlyOwner
386     {
387         assembly {
388             sstore(feeVal2.slot, _feeVal2)
389             sstore(feeVal3.slot, _feeVal3)
390         }

392         emit FeesUpdated(
393             _feeVal2,
394             _feeVal3
395         );
396     }
```

Risk Level:

Likelihood – 3
Impact - 3

Recommendation:

We recommend to limit feeVal2 and feeVal3 values by adding a require statement.

Status - Fixed

The MAD team has fixed the issue by adding require statements to make sure that the feeVal2 and feeVal3 are limited.

C.2 Missing Value Verification [MEDIUM]

Description:

Certain functions lack a safety check in the values, the values of the arguments should be verified to allow only the ones that go with the contract's logic.The variables in UpdateSettings() function should be verified otherwise, the user can't create an order , bid, or the auction won't increment.

Code:

Listing 8: MADMarketplace721

```
403     function updateSettings(
404         uint256 _minAuctionIncrement,
405         uint256 _minOrderDuration,
406         uint256 _minBidValue
407     ) public onlyOwner {
408         // minOrderDuration = _minOrderDuration;
409         // minAuctionIncrement = _minAuctionIncrement;
410         // minBidValue = _minBidValue;
411         assembly {
412             sstore(minOrderDuration.slot, _minOrderDuration)
413             sstore(
414                 minAuctionIncrement.slot,
415                 _minAuctionIncrement
416             )
```

```
417             sstore(minBidValue.slot, _minBidValue)
418         }

420         emit AuctionSettingsUpdated(
421             _minOrderDuration,
422             _minAuctionIncrement,
423             _minBidValue
424         );
425     }
```

Risk Level:

Likelihood – 2
Impact - 3

Recommendation:

Consider verifying _minAuctionIncrement _minOrderDuration and _minBidValue by adding a require statement, to allow only the ones that go with the contracts's logic.

Status - Fixed

The MAD team has fixed the issue by adding require statements to make sure that the values are limited.

C.3 Race Condition [MEDIUM]

Description:

The feeVal2 , feeVal3 variables have a setter. If the user checks the value of this variable, then calls the buy or call function, and the owner updates the Fees Value , the order of the transaction might overturn and the user's transaction in this case will be executed with the new fees without him knowing about it.

Code:

Listing 9: MADMarketplace721

```
381     function setFees(
382         uint256 _feeVal2,
383         uint256 _feeVal3)
384         external
385         onlyOwner
386     {
387         assembly {
388             sstore(feeVal2.slot, _feeVal2)
389             sstore(feeVal3.slot, _feeVal3)
390         }
391
392         emit FeesUpdated(
393             _feeVal2,
394             _feeVal3
395         );
396     }
```

Risk Level:

Likelihood – 2
Impact - 3

Recommendation:

Consider adding the feeVal2 , feeVal3 in in the arguments of the _feeResolver function then add require statements that verifies that the values provided in the arguments are the same as the one that is stored in the smart contract .In the other hand, add FeeVal3 in the arguments of the _extPath0 , _extPath1 functions then a require statement that verifies that feeVal3 is the same as the one that is stored in the contract

Status - Acknowledged

The MAD team has acknowledged the risk.

C.4 Missing address verification [MEDIUM]

Description:

The address-type arguments newOwner and _recipient should include a zero-address test, otherwise, the contract's functionality may become inaccessible. If the contract ownership is lost. You need to re-deploy the same contract again.

Code:

Listing 10: MADMarketplace721

```
441     function setRecipient(address _recipient)
442         public
443         onlyOwner
444     {
445         // recipient = _recipient;
446         assembly {
447             sstore(recipient.slot, _recipient)
448         }
449
450         emit RecipientUpdated(_recipient);
451     }
```

Listing 11: MADMarketplace721

```
454     function setOwner(address newOwner)
455         public
456         override
457         onlyOwner
458     {
459         // owner = newOwner;
460         assembly {
```

```
461             sstore(owner.slot, newOwner)
462         }
464         emit OwnerUpdated(msg.sender, newOwner);
465     }
```

Listing 12: MADMarketplace721

```
83      constructor(
84          address _recipient,
85          uint256 _minOrderDuration,
86          FactoryVerifier _factory
87      ) {
88          setFactory(_factory);
89          setRecipient(_recipient);
90          updateSettings(
91              300, // 5 min
92              _minOrderDuration,
93              20 // 5% (1/20th)
94          );
95      }
```

Risk Level:

Likelihood – 1
Impact - 4

Recommendation:

We recommend that you make sure the addresses provided in the arguments are different from the address(0).

Status - Fixed

The MAD team has fixed the issue by adding require statements to make sure that the addresses provided in the arguments are different from the address(0).

D MADMarketplace1155.sol

D.1 Centralization risk [HIGH]

Description:

The owner can set any value in _feeVal2 and _feeVal2 variable. This represents a significant centralization risk on the user side.

Code:

Listing 13: MADMarketplace1155

```
405     function setFees(
406         uint256 _feeVal2,
407         uint256 _feeVal3)
408         external
409         onlyOwner
410     {
411         assembly {
412             sstore(feeVal2.slot, _feeVal2)
413             sstore(feeVal3.slot, _feeVal3)
414         }

416         emit FeesUpdated(
417             _feeVal2,
418             _feeVal3
419         );
420     }
```

Risk Level:

Likelihood – 3
Impact - 4

Recommendation:

We recommend to limit feeVal2 and feeVal3 values by adding a require statement.

Status - Fixed

The MAD team has fixed the issue by adding require statements to make sure that the feeVal2 and feeVal3 are limited.

D.2 Missing Value Verification [MEDIUM]

Description:

Certain functions lack a safety check in the values, the values of the arguments should be verified to allow only the ones that go with the contract's logic.The variables in UpdateSettings() function should be verified otherwise, the user can't create an order , bid, or the auction won't increment.

Code:

Listing 14: MADMarketplace1155

```
428     function updateSettings(
429         uint256 _minAuctionIncrement,
430         uint256 _minOrderDuration,
431         uint256 _minBidValue
432     ) public onlyOwner {
433         // minOrderDuration = _minOrderDuration;
434         // minAuctionIncrement = _minAuctionIncrement;
435         // minBidValue = _minBidValue;
436         assembly {
437             sstore(minOrderDuration.slot, _minOrderDuration)
438             sstore(
439                 minAuctionIncrement.slot,
440                 _minAuctionIncrement
441             )
```

```
442             sstore(minBidValue.slot, _minBidValue)
443         }

445         emit AuctionSettingsUpdated(
446             _minOrderDuration,
447             _minAuctionIncrement,
448             _minBidValue
449         );
450     }
```

Risk Level:

Likelihood – 2
Impact - 3

Recommendation:

Consider verifying _minAuctionIncrement _minOrderDuration and _minBidValue by adding a require statement, to allow only the ones that go with the contracts's logic.

Status - Fixed

The MAD team has fixed the issue by adding require statements to make sure that the values are limited.

D.3 Race Condition [MEDIUM]

Description:

The feeVal2 , feeVal3 variables have a setter. If the user checks the value of this variable, then calls the buy or call function, and the owner updates the Fees Value , the order of the transaction might overturn and the user's transaction in this case will be executed with the new fees without him knowing about it.

Code:

Listing 15: MADMarketplace1155

```
405     function setFees(
406         uint256 _feeVal2,
407         uint256 _feeVal3)
408         external
409         onlyOwner
410     {
411         assembly {
412             sstore(feeVal2.slot, _feeVal2)
413             sstore(feeVal3.slot, _feeVal3)
414         }

416         emit FeesUpdated(
417             _feeVal2,
418             _feeVal3
419         );
420     }
```

Risk Level:

Likelihood – 2
Impact - 3

Recommendation:

Consider adding the feeVal2 , feeVal3 in in the arguments of the _feeResolver function then add require statements that verifies that the values provided in the arguments are the same as the one that is stored in the smart contract .In the other hand, add FeeVal3 in the arguments of the _extPath0 , _extPath1 functions then a require statement that verifies that feeVal3 is the same as the one that is stored in the contract

Status - Acknowledged

The MAD team has acknowledged the risk.

D.4 Missing address verification [MEDIUM]

Description:

The address-type arguments newOwner and _recipient should include a zero-address test, otherwise, the contract's functionality may become inaccessible. If the contract ownership is lost. You need to re-deploy the same contract again.

Code:

Listing 16: MADMarketplace1155

```
466     function setRecipient(address _recipient)
467         public
468         onlyOwner
469     {
470         // recipient = _recipient;
471         assembly {
472             sstore(recipient.slot, _recipient)
473         }
474
475         emit RecipientUpdated(_recipient);
476     }
```

Listing 17: MADMarketplace1155

```
479     function setOwner(address newOwner)
480         public
481         override
482         onlyOwner
483     {
484         // owner = newOwner;
485         assembly {
```

```
486             sstore(owner.slot, newOwner)
487         }

489         emit OwnerUpdated(msg.sender, newOwner);
490     }
```

Listing 18: MADMarketplace1155

```
83      constructor(
84          address _recipient,
85          uint256 _minOrderDuration,
86          FactoryVerifier _factory
87      ) {
88          setFactory(_factory);
89          setRecipient(_recipient);
90          updateSettings(
91              300, // 5 min
92              _minOrderDuration,
93              20 // 5% (1/20th)
94          );
95      }
```

Risk Level:

Likelihood – 1
Impact - 4

Recommendation:

We recommend that you make sure the addresses provided in the arguments are different from the address(0).

Status - Fixed

The MAD team has fixed the issue by adding require statements to make sure that the addresses provided in the arguments are different from the address(0).

E MADFactory721.sol

E.1 Missing address verification [MEDIUM]

Description:

The address-type arguments newOwner _market _router _signer should include a zero-address test, otherwise, the contract's functionality may become inaccessible. If the contract ownership is lost. You need to re-deploy the same contract again.

Code:

Listing 19: MADFactory721

```
487     function setOwner(address newOwner)
488         public
489         override
490         onlyOwner
491     {
492         // owner = newOwner;
493         assembly {
494             sstore(owner.slot, newOwner)
495         }
497         emit OwnerUpdated(msg.sender, newOwner);
498     }
```

Listing 20: MADFactory721

```
502     function setMarket(address _market) public onlyOwner {
503         assembly {
504             sstore(market.slot, _market)
505         }
507         emit MarketplaceUpdated(_market);
508     }
```

Listing 21: MADFactory721

```
512    function setRouter(address _router) public onlyOwner {
513        // router = _router;
514        assembly {
515            sstore(router.slot, _router)
516        }

518        emit RouterUpdated(_router);
519    }
```

Listing 22: MADFactory721

```
523    function setSigner(address _signer) public onlyOwner {
524        // signer = _signer;
525        assembly {
526            sstore(signer.slot, _signer)
527        }

529        emit SignerUpdated(_signer);
530    }
```

Listing 23: MADFactory721

```
99     constructor
100    (
101        address _marketplace,
102        address _router,
103        address _signer
104    )
105    {
106        setMarket(_marketplace);
107        setRouter(_router);
108        setSigner(_signer);
109    }
```

Risk Level:

Likelihood – 1
Impact - 4

Recommendation:

We recommend that you make sure the addresses provided in the arguments are different from the address(0).

Status - Fixed

The MAD team has fixed the issue by adding require statements to make sure that the addresses provided in the arguments are different from the address(0).

F MADFactory1155.sol

F.1 Missing address verification [MEDIUM]

Description:

The address-type arguments newOwner _market _router _signer should include a zero-address test, otherwise, the contract's functionality may become inaccessible. If the contract ownership is lost. You need to re-deploy the same contract again.

Code:

Listing 24: MADFactory1155

```
480     function setOwner(address newOwner)
481         public
482         override
483         onlyOwner
484     {
485         // owner = newOwner;
486         assembly {
```

```
            sstore(owner.slot, newOwner)
        }

        emit OwnerUpdated(msg.sender, newOwner);
    }
```

Listing 25: MADFactory1155

```
    function setMarket(address _market) public onlyOwner {
        assembly {
            sstore(market.slot, _market)
        }

        emit MarketplaceUpdated(_market);
    }
```

Listing 26: MADFactory1155

```
    function setRouter(address _router) public onlyOwner {
        // router = _router;
        assembly {
            sstore(router.slot, _router)
        }

        emit RouterUpdated(_router);
    }
```

Listing 27: MADFactory1155

```
    function setSigner(address _signer) public onlyOwner {
        // signer = _signer;
        assembly {
            sstore(signer.slot, _signer)
        }

        emit SignerUpdated(_signer);
    }
```

Listing 28: MADFactory1155

```
100     constructor
101     (
102         address _marketplace,
103         address _router,
104         address _signer
105     )
106     {
107         setMarket(_marketplace);
108         setRouter(_router);
109         setSigner(_signer);
110     }
```

Risk Level:

Likelihood – 1
Impact - 4

Recommendation:

We recommend that you make sure the addresses provided in the arguments are different from the address(0).

Status - Fixed

The MAD team has fixed the issue by adding require statements to make sure that the addresses provided in the arguments are different from the address(0).

4 Best Practices

BP.1 ERC721 ERC1155 Token withdrawal

Description:

As declared in comments, the payees should only share the royalties not the token contract balance. If it is the case the function must be modified.

Code:

Listing 29: ERC721 ERC1155 Deployed Tokens

```
159     function withdraw() external onlyOwner {
160         uint256 len = splitter.payeesLength();
161         address[] memory addrs = new address[](len);
162         uint256[] memory values = new uint256[](len);
163         uint256 _val = address(this).balance;
164         uint256 i;
165         for (i; i < len; ) {
166             address addr = splitter._payees(i);
167             uint256 share = splitter._shares(addr);
168             addrs[i] = addr;
169             values[i] = ((_val * (share * 1e2)) / 10_000);
170             unchecked {
171                 ++i;
172             }
173         }
174         uint256 j;
175         while (j < len) {
176             SafeTransferLib.safeTransferETH(
177                 addrs[j],
178                 values[j]
179             );
180             unchecked {
```

```
                ++j;
            }
        }
    }

    function withdrawERC20(ERC20 _token) external onlyOwner {
        uint256 len = splitter.payeesLength();
        address[] memory addrs = new address[](len);
        uint256[] memory values = new uint256[](len);
        uint256 i;
        uint256 _val = _token.balanceOf(address(this));
        for (i; i < len; ) {
            address addr = splitter._payees(i);
            uint256 share = splitter._shares(addr);
            addrs[i] = addr;
            values[i] = ((_val * (share * 1e2)) / 10_000);
            unchecked {
                ++i;
            }
        }
        uint256 j;
        while (j < len) {
            SafeTransferLib.safeTransfer(
                _token,
                addrs[j],
                values[j]
            );
            unchecked {
                ++j;
            }
        }
    }
```

c

Listing 30: MADMarketplace1155.sol

```
123     function dutchAuction(
124         IERC1155 _token,
125         uint256 _id,
126         uint256 _amount,
127         uint256 _startPrice,
128         uint256 _endPrice,
129         uint256 _endTime
130     ) public whenNotPaused {
131         _exceedsMaxEP(_startPrice, _endPrice);
132         _makeOrder(
133             1,
134             _token,
135             _id,
136             _amount,
137             _startPrice,
138             _endPrice,
139             _endTime
140         );
141     }
```

Listing 31: MADMarketplace1155.sol

```
151     function englishAuction(
152         IERC1155 _token,
153         uint256 _id,
154         uint256 _amount,
155         uint256 _startPrice,
156         uint256 _endTime
157     ) public whenNotPaused {
158         _makeOrder(
159             2,
160             _token,
161             _id,
162             _amount,
163             _startPrice,
```

```
            0,
            _endTime
        );
    }
```

Listing 32: MADMarketplace721.sol

```
    function fixedPrice(
        IERC1155 _token,
        uint256 _id,
        uint256 _amount,
        uint256 _price,
        uint256 _endTime
    ) public whenNotPaused {
        _makeOrder(
            0,
            _token,
            _id,
            _amount,
            _price,
            0,
            _endTime
        );
    }
```

Listing 33: MADMarketplace721.sol

```
    function dutchAuction(
        IERC1155 _token,
        uint256 _id,
        uint256 _amount,
        uint256 _startPrice,
        uint256 _endPrice,
        uint256 _endTime
    ) public whenNotPaused {
        _exceedsMaxEP(_startPrice, _endPrice);
```

```
123         _makeOrder(
124             1,
125             _token,
126             _id,
127             _amount,
128             _startPrice,
129             _endPrice,
130             _endTime
131         );
132     }
```

Listing 34: MADMarketplace721.sol

```
134     function englishAuction(
135         IERC1155 _token,
136         uint256 _id,
137         uint256 _amount,
138         uint256 _startPrice,
139         uint256 _endTime
140     ) public whenNotPaused {
141         _makeOrder(
142             2,
143             _token,
144             _id,
145             _amount,
146             _startPrice,
147             0,
148             _endTime
149         );
150     }
```

5 Tests

Results:

```
Downloading compiler 0.8.16
Compiled 48 Solidity files successfully
/// ... .. .. ..
/// x*8888x.:*8888: -"888: dF
/// X 48888X `8888H 8888 '88bu.
/// X8x. 8888X 8888X !888> u '*88888bu
/// X8888 X8888 88888 "*8%- us888u. ^"*8888N
/// '*888!X8888> X8888 xH8> .@88 "8888" beWE "888L
/// `?8 `8888 X888X X888> 9888 9888 888E 888E
/// -^ '888" X888 8888> 9888 9888 888E 888E
/// dx '88~x. !88~ 8888> 9888 9888 888E 888F
/// .8888Xf.888x:! X888X.: 9888 9888 .888N..888
/// :""888":~"888" `888*" "888*""888" `"888*""
/// "~' "~ "" ^Y" ^Y' "" MADNFTs © 2022.


  ERC1155Basic
    Init
      Splitter and ERC1155 should initialize (147ms)
      accounts have been funded
    Only owner setters
      Should set base URI, emit event and revert if not owner (118ms)
      Should set public mint state, emit event & revert if not owner (79
        ↪ ms)
    Mint
      Should revert if public mint is turned off (38ms)
      Should revert if max supply has reached max (5272ms)
      Should revert if price is wrong (44ms)
      Should mint, update storage and emit events (82ms)
      Should handle multiple mints (4769ms)
```

```
    Batch mint
      Should revert if supply has reached max (5284ms)
      Should revert if public mint is turned off
      Should revert if price is wrong (38ms)
      Should batch mint, update storage and emit events (111ms)
      Should handle multiple batch mints (210ms)
    Burn
      Should revert if not owner
      Should revert if id is already burnt/hasn't been minted (125ms)
      Should revert if ids length is less than 2 (42ms)
      Should burn tokens, update storage and emit event (189ms)
    Batch burn
      Should revert if caller is not the owner (68ms)
      Should revert if id is already burnt/hasn't been minted (101ms)
      Should batch burn tokens, update storage and emit event (190ms)
      Should handle multiple batch burns (328ms)
    Withdraw
      Should withdraw contract's funds (170ms)
      Should withdraw contract's ERC20s (213ms)
    Public getters
      Should query royalty info
      Should query token uri and revert if not yet minted (85ms)
      Should query total supply
      Should query base uri
    Interface IDs
      Should support interfaces (44ms)


  ERC1155Lazy
    Init
      Splitter and ERC1155 should initialize (64ms)
      accounts have been funded
    Lazy mint
      Should mint, update storage and emit events (378ms)
      Should revert if voucher has already been used (232ms)
```

```
      Should revert if signature is invalid (38ms)
      Should revert if price is wrong
    Lazy batch mint
      Should mint, update storage and emit events (148ms)
      Should revert if voucherId has already been used (83ms)
      Should revert if signature is invalid
      Should revert if price is wrong
    Only owner functions
      Should set URI and emit event (54ms)
      Should withdraw and update balances (523ms)
    Burn
      Should revert if not owner
      Should revert if id is already burnt/hasn't been minted (247ms)
      Should revert if ids length is less than 2 (52ms)
      Should burn update storage and emit events (284ms)
    Batch burn
      Should revert if caller is not the owner (211ms)
      Should revert if id is already burnt/hasn't been minted (212ms)
      Should batch burn tokens, update storage and emit event (268ms)
      Should handle multiple batch burns (413ms)
    Public getters
      Should query royalty info
      Should retrieve the domain separator
      Should retrive URI and total supply (293ms)
      Should retrive tokenURI and revert if not yet minted (204ms)
      Should support interfaces (40ms)

  ERC1155Minimal
    Init
      Splitter and ERC1155 should initialize (57ms)
      accounts have been funded
    Safe Minting
      Should revert if not the owner
      Should mint, update storage and emit events (46ms)
```

```
      Should revert if already minted (62ms)
    Burning
      Should revert if has not been minted
      Should revert if not the owner (50ms)
      Should burn, update storage and emit events (85ms)
      Should revert if already burned (83ms)
    Public Minting
      Should update public mint state (48ms)
      Should revert if public mint is off
      Should revert if price is wrong (50ms)
      Should revert if already minted (72ms)
      Should mint, update storage and emit events (77ms)
    Withdrawing
      Should revert if not the owner (75ms)
      Should update balances of contract and owner (132ms)
      Should withdraw contract's ERC20s (214ms)
    Royalties
      Should retrive royalty info
    Token URI
      Should revert if ID is not 1
      Should revert if token was not minted
      Should retrieve tokenURI (47ms)
    Interface IDs
      Should support interfaces (41ms)

  ERC1155Whitelist
    Init
      Splitter and ERC721 should initialize (136ms)
      accounts have been funded
    Only owner setters
      Should check for whitelist & freeclaim event emitting/error
        ↪ handling (100ms)
      Should set URI and emit event (60ms)
      Should set mint states (105ms)
```

```
Public mint
  Should revert if value under/overflows
  Should revert if public mint state is off
  Should revert if available supply has reached max (5747ms)
  Should revert if price is wrong (42ms)
  Should mint, update storage and emit events (118ms)
Batch mint
  Should revert if supply has reached max (5686ms)
  Should revert if public mint is turned off
  Should revert if price is wrong (53ms)
  Should batch mint, update storage and emit events (104ms)
  Should handle multiple batch mints (206ms)
Whitelist mint
  Should revert if value under/overflows
  Should revert if whitelist mint state is off
  Should revert if whitelist supply has reached max (6370ms)
  Should revert if price is wrong (43ms)
  Should revert if address is not whitelisted (46ms)
  Should mint, update storage and emit events (133ms)
Whitelist batch mint
  Should revert if value under/overflows
  Should revert if whitelist mint state is off
  Should revert if whitelist supply has reached max (6270ms)
  Should revert if price is wrong (51ms)
  Should revert if address is not whitelisted (42ms)
  Should mint, update storage and emit events (134ms)
Free claim
  Should revert if free claim state is off
  Should revert if available supply has reached max (6416ms)
  Should revert if address is not whitelisted (39ms)
  Should revert if user has already claimed (73ms)
  Should mint, update storage and emit events (97ms)
  Should gift tokens (219ms)
Mint and batch mint to creator
```

```
      Should mint to creator (164ms)
      Should batch mint to creator (180ms)
    Burn
      Should revert if not owner
      Should revert if id is already burnt/hasn't been minted (115ms)
      Should revert if ids length is less than 2
      Should burn tokens, update storage and emit event (187ms)
    Batch burn
      Should revert if caller is not the owner (80ms)
      Should revert if id is already burnt/hasn't been minted (96ms)
      Should batch burn tokens, update storage and emit event (200ms)
      Should handle multiple batch burns (346ms)
    Withdraw
      Should withdraw contract's funds (165ms)
      Should withdraw contract's ERC20s (204ms)
    Public getters
      Should query royalty info
      Should query token uri and revert if not yet minted (80ms)
      Should query total supply
      Should query base uri
    Interface IDs
      Should support interfaces (43ms)

  ERC721Basic
    Init
      Splitter and ERC721 should initialize (84ms)
      accounts have been funded
    Only owner setters
      Should set base URI, emit event and revert if not owner (73ms)
      Should set public mint state, emit event & revert if not owner (60
        ↪ ms)
    Mint
      Should revert if public mint is turned off
      Should revert if max supply has reached max (6721ms)
```

```
      Should revert if price is wrong (47ms)
      Should mint, update storage and emit events (84ms)
      Should handle multiple mints (6393ms)
    Burn
      Should revert if not owner
      Should revert if id is already burnt/hasn't been minted (105ms)
      Should revert if ids length is less than 2
      Should burn tokens, update storage and emit event (202ms)
    Withdraw
      Should withdraw contract's funds (146ms)
      Should withdraw contract's ERC20s (208ms)
    Public getters
      Should query royalty info
      Should query token uri and revert if not yet minted (78ms)
      Should query total supply
      Should query base uri
      Should support interfaces (45ms)

  ERC721Lazy
    Init
      Splitter and ERC721 should initialize (78ms)
      accounts have been funded
    Lazy mint
      Should mint, update storage and emit events (354ms)
      Should revert if voucher has already been used (224ms)
      Should revert if signature is invalid
      Should revert if price is wrong
    Only owner functions
      Should set baseURI and emit event (53ms)
      Should withdraw and update balances (486ms)
    Burn
      Should revert if not owner
      Should revert if id is already burnt/hasn't been minted (214ms)
      Should revert if ids length is less than 2
```

```
      Should burn update storage and emit events (297ms)
    Public getters
      Should retrieve the domain separator
      Should retrive baseURI and total supply (270ms)
      Should retrive tokenURI and revert if not yet minted (210ms)
      Should query royalty info
      Should support interfaces

  ERC721Minimal
    Init
      Splitter and ERC721 should initialize (79ms)
      accounts have been funded
    Safe Minting
      Should revert if not the owner
      Should mint, update storage and emit events (52ms)
      Should revert if already minted (48ms)
    Burning
      Should revert if has not been minted
      Should revert if not the owner (51ms)
      Should burn, update storage and emit events (78ms)
      Should revert if already burned (66ms)
    Public Minting
      Should update public mint state (53ms)
      Should revert if public mint is off
      Should revert if price is wrong (49ms)
      Should revert if already minted (92ms)
      Should mint, update storage and emit events (71ms)
    Withdrawing
      Should revert if not the owner (73ms)
      Should update balances of contract and owner (120ms)
      Should withdraw contract's ERC20s (209ms)
    Royalties
      Should retrive royalty info
    Token URI
```

```
      Should revert if ID is not 1
      Should revert if token was not minted
      Should retrieve tokenURI (51ms)
    Interface IDs
      Should support interfaces

  ERC721Whitelist
    Init
      Splitter and ERC721 should initialize (159ms)
      accounts have been funded
    Only owner setters
      Should check for whitelist & freeclaim event emitting/error
        ↪ handling (89ms)
      Should set baseURI and emit event (39ms)
      Should set mint states (103ms)
    Public mint
      Should revert if value under/overflows
      Should revert if public mint state is off
      Should revert if available supply has reached max (5434ms)
      Should revert if price is wrong
      Should mint, update storage and emit events (124ms)
    Whitelist mint
      Should revert if value under/overflows
      Should revert if whitelist mint state is off
      Should revert if whitelist supply has reached max (6799ms)
      Should revert if price is wrong
      Should revert if address is not whitelisted (41ms)
      Should mint, update storage and emit events (133ms)
    Free claim
      Should revert if free claim state is off
      Should revert if available supply has reached max (6480ms)
      Should revert if address is not whitelisted (46ms)
      Should revert if user has already claimed (58ms)
      Should mint, update storage and emit events (124ms)
```

```
      Should mint to creator (131ms)
      Should gift tokens (237ms)
    Burn
      Should revert if not owner
      Should revert if id is already burnt/hasn't been minted (108ms)
      Should revert if ids length is less than 2
      Should burn update storage and emit events (183ms)
    Public getters
      Should retrive baseURI and total supply (136ms)
      Should retrive tokenURI and revert if not yet minted (47ms)
      Should support interfaces
    Withdrawing
      Should revert if not the owner (82ms)
      Should update balances of contract and owner (124ms)
      Should withdraw contract's ERC20s (199ms)

  MADFactory1155
    Init
      Factory should initialize
    Splitter check
      Should revert if repeated salt is provided (204ms)
      Should deploy splitter without ambassador, update storage and emit
        ↪  events (172ms)
      Should deploy splitter with ambassador, update storage and emit
        ↪ events (197ms)
    Create collection
      Should deploy ERC1155Minimal, update storage and emit events (455
        ↪ ms)
      Should deploy ERC1155Basic, update storage and emit events (485ms)
        ↪
      Should deploy ERC1155Whitelist, update storage and emit events
        ↪ (947ms)
      Should deploy ERC1155Lazy, update storage and emit events (485ms)
    Only owner functions
```

```
      Should update contract's owner (61ms)
      Should set new marketplace instance (62ms)
      Should update ERC1155Lazy signer (45ms)
      Should update router's address (44ms)
      Should initialize paused and unpaused states (111ms)
    Helpers
      Should retrieve user's colID indexes (1270ms)
      Should get collection ID from address
      Should retrieve collection type (453ms)
      Should enable marketplace no-fee listing (1009ms)
      Should verify a collection's creator (395ms)

  MADFactory721
    Init
      Factory should initialize
    Splitter check
      Should revert if repeated salt is provided (183ms)
      Should deploy splitter without ambassador, update storage and emit
        ↪  events (187ms)
      Should deploy splitter with ambassador, update storage and emit
        ↪ events (195ms)
      Should deploy splitter with ambassador and project, update storage
        ↪  and emit events (215ms)
    Create collection
      Should deploy ERC721Minimal, update storage and emit events (431ms
        ↪ )
      Should deploy ERC721Basic, update storage and emit events (693ms)
      Should deploy ERC721Whitelist, update storage and emit events (480
        ↪ ms)
      Should deploy ERC721Lazy, update storage and emit events (452ms)
    Only owner functions
      Should update contract's owner (52ms)
      Should set new marketplace instance (65ms)
      Should update ERC721Lazy signer
```

```
      Should update router's address (42ms)
      Should initialize paused and unpaused states (102ms)
    Helpers
      Should retrieve user's colID indexes (1160ms)
      Should get collection ID from address
      Should retrieve collection type (445ms)
      Should enable marketplace no-fee listing (906ms)
      Should verify a collection's creator (383ms)

  MADMarketplace1155
    Init
      Marketplace should initialize (39ms)
    Owner Functions
      Should update factory address (66ms)
      Should update marketplace settings (39ms)
      Should initialize paused and unpaused states (203ms)
      Should update recipient (42ms)
      Should update contract's owner (44ms)
      Should withdraw to owner (110ms)
      Should delete order (686ms)
    Fixed Price Listing
      Should revert if transaction approval hasn't been set (902ms)
      Should revert if duration is less than min allowed (447ms)
      Should revert if price is invalid (444ms)
      Should list fixed price order, update storage and emit event (541
        ↪ ms)
      Should handle multiple fixed price orders (1462ms)
    Dutch Auction Listing
      Should revert if transaction approval hasn't been set (492ms)
      Should revert if duration is less than min allowed (437ms)
      Should revert if startPrice is invalid (794ms)
      Should list dutch auction order, update storage and emit event
        ↪ (573ms)
      Should handle multiple dutch auction orders (1452ms)
```

```
English Auction Listing
  Should revert if transaction approval hasn't been set (485ms)
  Should revert if duration is less than min allowed (435ms)
  Should revert if startPrice is invalid (451ms)
  Should list english auction order, update storage and emit event
    ↪ (891ms)
  Should handle multiple english auction orders (1162ms)
Bidding
  Should revert if price is wrong (936ms)
  Should revert if not English Auction (513ms)
  Should revert if order was canceled (564ms)
  Should revert if order has timed out (515ms)
  Should revert if bidder is the seller (810ms)
  Should bid, update storage and emit events (566ms)
Buying
  Should revert if price is wrong (495ms)
  Should revert if order is an English Auction (501ms)
  Should revert if order was canceled (948ms)
  Should revert if order has timed out (510ms)
  Should revert if token has already been sold (602ms)
  Should buy inhouse minted tokens, update storage and emit events
    ↪ (1506ms)
  Should verify inhouse minted tokens balance changes (1118ms)
  Should buy third party minted tokens with ERC2981 support (463ms)
  Should buy third party minted tokens without ERC2981 support (398
    ↪ ms)
  Should verify inhouse minted tokens balance changes - set fees
    ↪ (1404ms)
  Should buy third party minted tokens with ERC2981 support - set
    ↪ fees (482ms)
  Should buy third party minted tokens without ERC2981 support - set
    ↪  fees (431ms)
Claim
  Should revert if caller is seller or bidder (565ms)
```

```
      Should revert if token has already been claimed (630ms)
      Should revert if orderType is not an english auction (246ms)
      Should revert if auction hasn't ended (508ms)
      Should claim inhouse minted tokens, update storage and emit events
        ↪  (1036ms)
      Should verify inhouse minted tokens balance changes (636ms)
      Should claim third party minted tokens with ERC2981 support (362ms
        ↪ )
      Should claim third party minted tokens without ERC2981 support
        ↪ (288ms)
    Order Cancelling
      Should revert due to already sold fixed price order (564ms)
      Should revert due to already sold dutch auction order (597ms)
      Should revert due to already sold english auction order (992ms)
      Should cancel fixed price order (580ms)
      Should cancel dutch auction order (567ms)
      Should cancel english auction order (572ms)
    Public Helpers
      Should fetch the length of orderIds for a token (1142ms)
      Should fetch the length of orderIds for a seller (1139ms)

  MADMarketplace721
    Init
      Marketplace should initialize (47ms)
    Owner Functions
      Should update factory address (43ms)
      Should update marketplace settings (39ms)
      Should initialize paused and unpaused states (186ms)
      Should update recipient (38ms)
      Should update contract's owner (51ms)
      Should withdraw to owner (98ms)
      Should delete order (648ms)
    Fixed Price Listing
      Should revert if transaction approval hasn't been set (524ms)
```

```
      Should revert if duration is less than min allowed (587ms)
      Should revert if price is invalid (450ms)
      Should list fixed price order, update storage and emit event (540
         ↪ ms)
      Should handle multiple fixed price orders (1123ms)
   Dutch Auction Listing
      Should revert if transaction approval hasn't been set (775ms)
      Should revert if duration is less than min allowed (441ms)
      Should revert if startPrice is invalid (449ms)
      Should list dutch auction order, update storage and emit event
         ↪ (539ms)
      Should handle multiple dutch auction orders (1488ms)
   English Auction Listing
      Should revert if transaction approval hasn't been set (486ms)
      Should revert if duration is less than min allowed (432ms)
      Should revert if startPrice is invalid (426ms)
      Should list english auction order, update storage and emit event
         ↪ (525ms)
      Should handle multiple english auction orders (1432ms)
   Bidding
      Should revert if price is wrong (553ms)
      Should revert if not English Auction (485ms)
      Should revert if order was canceled (534ms)
      Should revert if order has timed out (863ms)
      Should revert if bidder is the seller (494ms)
      Should bid, update storage and emit events (577ms)
   Buying
      Should revert if price is wrong (507ms)
      Should revert if order is an English Auction (496ms)
      Should revert if order was canceled (871ms)
      Should revert if order has timed out (489ms)
      Should revert if token has already been sold (582ms)
      Should buy inhouse minted tokens, update storage and emit events
         ↪ (1476ms)
```

```
      Should verify inhouse minted tokens balance changes (1072ms)
BigNumber { value: "347222222222222264" } BigNumber { value:
   ↪ "8680555555555556" }
      Should buy third party minted tokens with ERC2981 support (520ms)
      Should buy third party minted tokens without ERC2981 support (432
         ↪ ms)
      Should verify inhouse minted tokens balance changes - fee change
         ↪ update (1091ms)
BigNumber { value: "347222222222222264" } BigNumber { value:
   ↪ "17361111111111113" }
      Should buy third party minted tokens with ERC2981 support - fee
         ↪ change update (927ms)
      Should buy third party minted tokens without ERC2981 support - fee
         ↪  change update (441ms)
   Claim
      Should revert if caller is seller or bidder (546ms)
      Should revert if token has already been claimed (624ms)
      Should revert if orderType is not an english auction (292ms)
      Should revert if auction hasn't ended (534ms)
      Should claim inhouse minted tokens, update storage and emit events
         ↪  (650ms)
      Should verify inhouse minted tokens balance changes (942ms)
      Should claim third party minted tokens with ERC2981 support (350ms
         ↪ )
      Should claim third party minted tokens without ERC2981 support
         ↪ (283ms)
   Order Cancelling
      Should revert due to already sold fixed price order (610ms)
      Should revert due to already sold dutch auction order (589ms)
      Should revert due to already sold english auction order (635ms)
      Should cancel fixed price order (593ms)
      Should cancel dutch auction order (942ms)
      Should cancel english auction order (598ms)
   Public Helpers
```

```
      Should fetch the length of orderIds for a token (769ms)
      Should fetch the length of orderIds for a seller (1164ms)

  MADRouter1155
    Init
      Router should initialize
    Set URI
      Should revert for invalid collection type (414ms)
      Should set URI for 1155Basic collection type (518ms)
      Should set URI for 1155Whitelist collection type (961ms)
      Should set URI for 1155Lazy collection type (502ms)
    Whitelist Settings
      Should revert for invalid collection type (840ms)
      Should set whitelist config for 1155Whitelist collection type (519
        ↪ ms)
    FreeClaim Settings
      Should revert for invalid collection type (775ms)
      Should set freeClaim config for 1155Whitelist collection type (502
        ↪ ms)
    Minimal SafeMint
      Should revert for invalid collection type (831ms)
(node:2115) PromiseRejectionHandledWarning: Promise rejection was
  ↪ handled asynchronously (rejection id: 14)
(Use `node --trace-warnings ...` to show where the warning was created)
      Should call safeMint for 1155Minimal collection type (440ms)
    Burn
      Should burn token for 1155Minimal collection type (467ms)
      Should burn tokens for 1155Basic collection type (556ms)
      Should burn tokens for 1155Whitelist collection type (977ms)
      Should burn tokens for 1155Lazy collection type (628ms)
    Batch Burn
      Should revert for invalid collection type (429ms)
      Should batch burn token for 1155Basic collection type (817ms)
      Should batch burn tokens for 1155Whitelist collection type (668ms)
```

```
        ↪
      Should batch burn tokens for 1155Lazy collection type (983ms)
  Set MintState
    Should revert for invalid stateType
    Should revert for invalid tokenType (357ms)
    Should set publicMintState for minimal, basic and whitelist
      ↪ colTypes (1448ms)
    Should set whitelistMintState for whitelist colType (554ms)
    Should set freeClaimState for whitelist colType (854ms)
  Whitelist Creator Mint
    Should revert for invalid coltype (425ms)
    Should mint to creator (893ms)
  Whitelist Creator Batch Mint
    Should revert for invalid coltype (426ms)
    Should batch mint to creator (411ms)
    Should mint to creator (965ms)
  Whitelist token gifting
    Should revert for invalid coltype (412ms)
    Should gift tokens (954ms)
  Creator Withdraw
    Should withdraw balance and ERC20 for all colTypes (3690ms)
  Only Owner
    Should update contract's owner (43ms)
    Should initialize paused and unpaused states (213ms)
  Minimal SafeMint
    Should call safeMint for 1155Minimal collection type (519ms)
  Burn-setfees
    Should burn token for 1155Minimal collection type (578ms)
fee is BigNumber { value: "5000000000000000000" }
    Should burn tokens for 1155Basic collection type (1149ms)
    Should burn tokens for 1155Whitelist collection type (783ms)
    Should burn tokens for 1155Lazy collection type (997ms)
  Batch Burn
    Should revert for invalid collection type (468ms)
```

```
      Should batch burn token for 1155Basic collection type (681ms)
      Should batch burn tokens for 1155Whitelist collection type (1244ms
        ↪ )
      Should batch burn tokens for 1155Lazy collection type (692ms)
    Whitelist Creator Mint
      Should revert for invalid coltype (761ms)
      Should mint to creator (644ms)
    Whitelist Creator Batch Mint
      Should mint to creator (962ms)
    Whitelist token gifting
      Should gift tokens (671ms)


  MADRouter721
    Init
      Router should initialize
    Set baseURI
      Should revert for invalid collection type (407ms)
      Should set baseURI for 721Basic collection type (491ms)
      Should set baseURI for 721Whitelist collection type (504ms)
      Should set baseURI for 721Lazy collection type (858ms)
    Whitelist Settings
      Should revert for invalid collection type (408ms)
      Should set whitelist config for 721Whitelist collection type (483
        ↪ ms)
    FreeClaim Settings
      Should revert for invalid collection type (689ms)
      Should set freeClaim config for 721Whitelist collection type (459
        ↪ ms)
    Minimal SafeMint
      Should revert for invalid collection type (400ms)
(node:2115) PromiseRejectionHandledWarning: Promise rejection was
  ↪ handled asynchronously (rejection id: 15)
BigNumber { value: "250000000000000000" }
minted successfully
```

```
    Should call safeMint for 721Minimal collection type (868ms)
  Burn
    Should burn token for 721Minimal collection type (479ms)
    Should burn tokens for 721Basic collection type (485ms)
    Should burn tokens for 721Whitelist collection type (569ms)
    Should burn tokens for 721Lazy collection type (926ms)
  Set MintState
    Should revert for invalid stateType
    Should revert for invalid tokenType (360ms)
    Should set publicMintState for minimal, basic and whitelist
      ↪ colTypes (1395ms)
    Should set whitelistMintState for whitelist colType (475ms)
    Should set freeClaimState for whitelist colType (489ms)
  Whitelist Creator Mint
    Should revert for invalid coltype (718ms)
    Should mint to creator (552ms)
  Whitelist token gifting
    Should revert for invalid coltype (381ms)
    Should gift tokens (973ms)
  Creator Withdraw
    Should withdraw balance and ERC20 for all colTypes (3234ms)
  Only Owner
    Should update contract's owner (50ms)
    Should initialize paused and unpaused states (209ms)
  Minimal SafeMint-setBaseFee
    Should call safeMint for 721Minimal collection type (508ms)
  Burn-setBaseFee
    Should burn tokens for 721Basic collection type (574ms)
    Should burn tokens for 721Whitelist collection type (966ms)
    Should burn tokens for 721Lazy collection type (628ms)
  Whitelist Creator Mint-setBaseFee
    Should mint to creator (604ms)
  Whitelist token gifting-setBaseFee
    Should gift tokens (857ms)
```

```
Royalties
   Royalties should initialize
   Should retrive royalty info
   Should accept recipient and fee change (95ms)
   Should support interfaces

Splitter
  Init
     Splitter should initialize (65ms)
     accounts have been funded
  Reverts
     should revert if no payees are provided
     should revert if more payees than shares are provided (39ms)
     should revert if more shares than payees are provided
     should revert if dead address is provided as payee
     should revert if a share is set to zero
     should revert if a provided payees are duplicated
     should revert if a provided payees are duplicated (44ms)
     should revert if account has no shares to claim
     should revert if there are no funds to claim
     should revert if account has no ERC20 shares to claim (94ms)
     should revert if there is no ERC20 to claim (99ms)
  Receive Payments
     should accept value and autodistribute to payees (165ms)
     should accept ERC20 (102ms)
  Release Payments
     should release value to payee (69ms)
     should release all pending balance to payees (71ms)
     should release ERC20 to payee (168ms)


471 passing (4m)
```

6 Static Analysis (Slither)

Description:

Block Hat expanded the coverage of the specific contract areas using automated testing methodologies. Slither, a Solidity static analysis framework, was one of the tools used. Slither was run on all-scoped contracts in both text and binary formats. This tool can be used to test mathematical relationships between Solidity instances statically and variables that allow for the detection of errors or inconsistent usage of the contracts' APIs throughout the entire codebase.

Results:

```
MADFactory1155.router (contracts/MADFactory1155.sol#85) is never
    ↪ initialized. It is used in:
        - MADFactory1155.createCollection(uint8,string,string,string,
            ↪ uint256,uint256,string,address,uint256) (contracts/
            ↪ MADFactory1155.sol#324-473)
        - MADFactory1155.setRouter(address) (contracts/MADFactory1155.sol
            ↪ #505-512)
        - MADFactory1155._isRouter() (contracts/MADFactory1155.sol
            ↪ #756-766)
MADFactory1155.market (contracts/MADFactory1155.sol#88) is never
    ↪ initialized. It is used in:
        - MADFactory1155.setMarket(address) (contracts/MADFactory1155.sol
            ↪ #495-501)
        - MADFactory1155._isMarket() (contracts/MADFactory1155.sol
            ↪ #770-777)
MADFactory1155.signer (contracts/MADFactory1155.sol#94) is never
    ↪ initialized. It is used in:
        - MADFactory1155.createCollection(uint8,string,string,string,
            ↪ uint256,uint256,string,address,uint256) (contracts/
            ↪ MADFactory1155.sol#324-473)
```

```
        - MADFactory1155.setSigner(address) (contracts/MADFactory1155.sol
            ↪ #516-523)
MADFactory721.router (contracts/MADFactory721.sol#84) is never
    ↪ initialized. It is used in:
        - MADFactory721.createCollection(uint8,string,string,string,
            ↪ uint256,uint256,string,address,uint256) (contracts/
            ↪ MADFactory721.sol#323-480)
        - MADFactory721.setRouter(address) (contracts/MADFactory721.sol
            ↪ #512-519)
        - MADFactory721._isRouter() (contracts/MADFactory721.sol#764-774)
MADFactory721.market (contracts/MADFactory721.sol#87) is never
    ↪ initialized. It is used in:
        - MADFactory721.setMarket(address) (contracts/MADFactory721.sol
            ↪ #502-508)
        - MADFactory721._isMarket() (contracts/MADFactory721.sol#778-785)
MADFactory721.signer (contracts/MADFactory721.sol#93) is never
    ↪ initialized. It is used in:
        - MADFactory721.createCollection(uint8,string,string,string,
            ↪ uint256,uint256,string,address,uint256) (contracts/
            ↪ MADFactory721.sol#323-480)
        - MADFactory721.setSigner(address) (contracts/MADFactory721.sol
            ↪ #523-530)
MADMarketplace1155.feeSelector (contracts/MADMarketplace1155.sol#69-70)
    ↪ is never initialized. It is used in:
        - MADMarketplace1155.buy(bytes32) (contracts/MADMarketplace1155.
            ↪ sol#224-284)
        - MADMarketplace1155.claim(bytes32) (contracts/MADMarketplace1155
            ↪ .sol#289-350)
        - MADMarketplace1155._feeResolver(uint256,uint256,uint256) (
            ↪ contracts/MADMarketplace1155.sol#788-812)
MADMarketplace1155.minOrderDuration (contracts/MADMarketplace1155.sol
    ↪ #72) is never initialized. It is used in:
        - MADMarketplace1155.updateSettings(uint256,uint256,uint256) (
            ↪ contracts/MADMarketplace1155.sol#428-450)
```

```
        - MADMarketplace1155._makeOrderChecks(uint256,uint256) (contracts
            ↪ /MADMarketplace1155.sol#856-885)
MADMarketplace1155.minAuctionIncrement (contracts/MADMarketplace1155.sol
    ↪ #73) is never initialized. It is used in:
        - MADMarketplace1155.bid(bytes32) (contracts/MADMarketplace1155.
            ↪ sol#168-219)
        - MADMarketplace1155.updateSettings(uint256,uint256,uint256) (
            ↪ contracts/MADMarketplace1155.sol#428-450)
MADMarketplace1155.minBidValue (contracts/MADMarketplace1155.sol#74) is
    ↪ never initialized. It is used in:
        - MADMarketplace1155.updateSettings(uint256,uint256,uint256) (
            ↪ contracts/MADMarketplace1155.sol#428-450)
        - MADMarketplace1155._bidChecks(uint8,uint256,address,uint256,
            ↪ uint256) (contracts/MADMarketplace1155.sol#911-966)
MADMarketplace1155.recipient (contracts/MADMarketplace1155.sol#76) is
    ↪ never initialized. It is used in:
        - MADMarketplace1155.setRecipient(address) (contracts/
            ↪ MADMarketplace1155.sol#466-476)
        - MADMarketplace1155._intPath(Types.Order1155,uint256,bytes32,
            ↪ address,uint256) (contracts/MADMarketplace1155.sol
            ↪ #645-695)
        - MADMarketplace1155._extPath0(Types.Order1155,uint256,bytes32,
            ↪ address) (contracts/MADMarketplace1155.sol#697-747)
        - MADMarketplace1155._extPath1(Types.Order1155,uint256,bytes32,
            ↪ address) (contracts/MADMarketplace1155.sol#749-786)
MADMarketplace1155.MADFactory1155 (contracts/MADMarketplace1155.sol#77)
    ↪ is never initialized. It is used in:
        - MADMarketplace1155.buy(bytes32) (contracts/MADMarketplace1155.
            ↪ sol#224-284)
        - MADMarketplace1155.claim(bytes32) (contracts/MADMarketplace1155
            ↪ .sol#289-350)
        - MADMarketplace1155.setFactory(FactoryVerifier) (contracts/
            ↪ MADMarketplace1155.sol#394-403)
```

```
MADMarketplace721.feeSelector (contracts/MADMarketplace721.sol#69-70) is
    ↪  never initialized. It is used in:
        - MADMarketplace721.buy(bytes32) (contracts/MADMarketplace721.sol
            ↪ #204-267)
        - MADMarketplace721.claim(bytes32) (contracts/MADMarketplace721.
            ↪ sol#272-335)
        - MADMarketplace721._feeResolver(uint256,uint256) (contracts/
            ↪ MADMarketplace721.sol#718-738)
MADMarketplace721.minOrderDuration (contracts/MADMarketplace721.sol#72)
    ↪ is never initialized. It is used in:
        - MADMarketplace721.updateSettings(uint256,uint256,uint256) (
            ↪ contracts/MADMarketplace721.sol#403-425)
        - MADMarketplace721._makeOrderChecks(uint256,uint256) (contracts/
            ↪ MADMarketplace721.sol#782-811)
MADMarketplace721.minAuctionIncrement (contracts/MADMarketplace721.sol
    ↪ #73) is never initialized. It is used in:
        - MADMarketplace721.bid(bytes32) (contracts/MADMarketplace721.sol
            ↪ #148-199)
        - MADMarketplace721.updateSettings(uint256,uint256,uint256) (
            ↪ contracts/MADMarketplace721.sol#403-425)
MADMarketplace721.minBidValue (contracts/MADMarketplace721.sol#74) is
    ↪ never initialized. It is used in:
        - MADMarketplace721.updateSettings(uint256,uint256,uint256) (
            ↪ contracts/MADMarketplace721.sol#403-425)
        - MADMarketplace721._bidChecks(uint8,uint256,address,uint256,
            ↪ uint256) (contracts/MADMarketplace721.sol#837-892)
MADMarketplace721.recipient (contracts/MADMarketplace721.sol#76) is
    ↪ never initialized. It is used in:
        - MADMarketplace721.setRecipient(address) (contracts/
            ↪ MADMarketplace721.sol#441-451)
        - MADMarketplace721._intPath(Types.Order721,uint256,bytes32,
            ↪ address,uint256) (contracts/MADMarketplace721.sol#596-639)
        - MADMarketplace721._extPath0(Types.Order721,uint256,bytes32,
            ↪ address) (contracts/MADMarketplace721.sol#641-683)
```

```
        - MADMarketplace721._extPath1(Types.Order721,uint256,bytes32,
            ↪ address) (contracts/MADMarketplace721.sol#685-716)
MADMarketplace721.MADFactory721 (contracts/MADMarketplace721.sol#77) is
    ↪ never initialized. It is used in:
        - MADMarketplace721.buy(bytes32) (contracts/MADMarketplace721.sol
            ↪ #204-267)
        - MADMarketplace721.claim(bytes32) (contracts/MADMarketplace721.
            ↪ sol#272-335)
        - MADMarketplace721.setFactory(FactoryVerifier) (contracts/
            ↪ MADMarketplace721.sol#370-379)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation
    ↪ #uninitialized-state-variables

MADMarketplace1155.getCurrentPrice(bytes32) (contracts/
    ↪ MADMarketplace1155.sol#1034-1100) performs a multiplication on
    ↪ the result of a division:
        -_tick_getCurrentPrice_asm_0 = _startPrice_getCurrentPrice_asm_0
            ↪ - _endPrice_getCurrentPrice_asm_0 /
            ↪ _endTime_getCurrentPrice_asm_0 -
            ↪ _startTime_getCurrentPrice_asm_0 (contracts/
            ↪ MADMarketplace1155.sol#1073-1076)
        -price = _startPrice_getCurrentPrice_asm_0 - timestamp()() -
            ↪ _startTime_getCurrentPrice_asm_0 *
            ↪ _tick_getCurrentPrice_asm_0 (contracts/MADMarketplace1155.
            ↪ sol#1077-1080)
MADMarketplace721.getCurrentPrice(bytes32) (contracts/MADMarketplace721.
    ↪ sol#960-1026) performs a multiplication on the result of a
    ↪ division:
        -_tick_getCurrentPrice_asm_0 = _startPrice_getCurrentPrice_asm_0
            ↪ - _endPrice_getCurrentPrice_asm_0 /
            ↪ _endTime_getCurrentPrice_asm_0 -
            ↪ _startTime_getCurrentPrice_asm_0 (contracts/
            ↪ MADMarketplace721.sol#999-1002)
```

```
        -price = _startPrice_getCurrentPrice_asm_0 - timestamp()() -
            ↪ _startTime_getCurrentPrice_asm_0 *
            ↪ _tick_getCurrentPrice_asm_0 (contracts/MADMarketplace721.
            ↪ sol#1003-1006)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation
    ↪ #divide-before-multiply

Contract locking ether found:
        Contract MADMarketplace1155 (contracts/MADMarketplace1155.sol
            ↪ #22-1127) has payable functions:
         - MADMarketplace1155.bid(bytes32) (contracts/MADMarketplace1155.
             ↪ sol#168-219)
         - MADMarketplace1155.buy(bytes32) (contracts/MADMarketplace1155.
             ↪ sol#224-284)
         - MADMarketplace1155.receive() (contracts/MADMarketplace1155.sol
             ↪ #386)
        But does not have a function to withdraw the ether
Contract locking ether found:
        Contract MADMarketplace721 (contracts/MADMarketplace721.sol
            ↪ #22-1053) has payable functions:
         - MADMarketplace721.bid(bytes32) (contracts/MADMarketplace721.
             ↪ sol#148-199)
         - MADMarketplace721.buy(bytes32) (contracts/MADMarketplace721.
             ↪ sol#204-267)
         - MADMarketplace721.receive() (contracts/MADMarketplace721.sol
             ↪ #362)
        But does not have a function to withdraw the ether
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation
    ↪ #contracts-that-lock-ether

Reentrancy in MADFactory1155.createCollection(uint8,string,string,string
    ↪ ,uint256,uint256,string,address,uint256) (contracts/
    ↪ MADFactory1155.sol#324-473):
        External calls:
```

```
        - (tokenSalt,deployed) = ERC1155MinimalDeployer.
            ↪ _1155MinimalDeploy(_tokenSalt,_uri,_price,_splitter,router
            ↪ ,_royalty) (contracts/MADFactory1155.sol#344-352)
        - (tokenSalt,deployed) = ERC1155BasicDeployer._1155BasicDeploy(
            ↪ _tokenSalt,_uri,_price,_maxSupply,_splitter,router,
            ↪ _royalty) (contracts/MADFactory1155.sol#376-385)
        State variables written after the call(s):
        - userTokens[tx.origin].push(colId_scope_2) (contracts/
            ↪ MADFactory1155.sol#388)
    Reentrancy in MADFactory1155.createCollection(uint8,string,string,string
        ↪ ,uint256,uint256,string,address,uint256) (contracts/
        ↪ MADFactory1155.sol#324-473):
        External calls:
        - (tokenSalt,deployed) = ERC1155MinimalDeployer.
            ↪ _1155MinimalDeploy(_tokenSalt,_uri,_price,_splitter,router
            ↪ ,_royalty) (contracts/MADFactory1155.sol#344-352)
        - (tokenSalt,deployed) = ERC1155BasicDeployer._1155BasicDeploy(
            ↪ _tokenSalt,_uri,_price,_maxSupply,_splitter,router,
            ↪ _royalty) (contracts/MADFactory1155.sol#376-385)
        - (tokenSalt,deployed) = ERC1155WhitelistDeployer.
            ↪ _1155WhitelistDeploy(_tokenSalt,_uri,_price,_maxSupply,
            ↪ _splitter,router,_royalty) (contracts/MADFactory1155.sol
            ↪ #409-418)
        State variables written after the call(s):
        - userTokens[tx.origin].push(colId_scope_5) (contracts/
            ↪ MADFactory1155.sol#421)
    Reentrancy in MADFactory1155.createCollection(uint8,string,string,string
        ↪ ,uint256,uint256,string,address,uint256) (contracts/
        ↪ MADFactory1155.sol#324-473):
        External calls:
        - (tokenSalt,deployed) = ERC1155MinimalDeployer.
            ↪ _1155MinimalDeploy(_tokenSalt,_uri,_price,_splitter,router
            ↪ ,_royalty) (contracts/MADFactory1155.sol#344-352)
```

```
        - (tokenSalt,deployed) = ERC1155BasicDeployer._1155BasicDeploy(
            ↪ _tokenSalt,_uri,_price,_maxSupply,_splitter,router,
            ↪ _royalty) (contracts/MADFactory1155.sol#376-385)
        - (tokenSalt,deployed) = ERC1155WhitelistDeployer.
            ↪ _1155WhitelistDeploy(_tokenSalt,_uri,_price,_maxSupply,
            ↪ _splitter,router,_royalty) (contracts/MADFactory1155.sol
            ↪ #409-418)
        - (tokenSalt,deployed) = ERC1155LazyDeployer._1155LazyDeploy(
            ↪ _tokenSalt,_uri,_splitter,router,signer,_royalty) (
            ↪ contracts/MADFactory1155.sol#442-450)
        State variables written after the call(s):
        - userTokens[tx.origin].push(colId_scope_8) (contracts/
            ↪ MADFactory1155.sol#453)
Reentrancy in MADFactory721.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256) (contracts/MADFactory721.
    ↪ sol#323-480):
        External calls:
        - (tokenSalt,deployed) = ERC721MinimalDeployer._721MinimalDeploy(
            ↪ _tokenSalt,_name,_symbol,_baseURI,_price,_splitter,router,
            ↪ _royalty) (contracts/MADFactory721.sol#343-353)
        - (tokenSalt,deployed) = ERC721BasicDeployer._721BasicDeploy(
            ↪ _tokenSalt,_name,_symbol,_baseURI,_price,_maxSupply,
            ↪ _splitter,router,_royalty) (contracts/MADFactory721.sol
            ↪ #377-388)
        State variables written after the call(s):
        - userTokens[tx.origin].push(colId_scope_2) (contracts/
            ↪ MADFactory721.sol#391)
Reentrancy in MADFactory721.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256) (contracts/MADFactory721.
    ↪ sol#323-480):
        External calls:
        - (tokenSalt,deployed) = ERC721MinimalDeployer._721MinimalDeploy(
            ↪ _tokenSalt,_name,_symbol,_baseURI,_price,_splitter,router,
            ↪ _royalty) (contracts/MADFactory721.sol#343-353)
```

```
        - (tokenSalt,deployed) = ERC721BasicDeployer._721BasicDeploy(
            ↪ _tokenSalt,_name,_symbol,_baseURI,_price,_maxSupply,
            ↪ _splitter,router,_royalty) (contracts/MADFactory721.sol
            ↪ #377-388)
        - (tokenSalt,deployed) = ERC721WhitelistDeployer.
            ↪ _721WhitelistDeploy(_tokenSalt,_name,_symbol,_baseURI,
            ↪ _price,_maxSupply,_splitter,router,_royalty) (contracts/
            ↪ MADFactory721.sol#412-423)
        State variables written after the call(s):
        - userTokens[tx.origin].push(colId_scope_5) (contracts/
            ↪ MADFactory721.sol#426)
    Reentrancy in MADFactory721.createCollection(uint8,string,string,string,
        ↪ uint256,uint256,string,address,uint256) (contracts/MADFactory721.
        ↪ sol#323-480):
        External calls:
        - (tokenSalt,deployed) = ERC721MinimalDeployer._721MinimalDeploy(
            ↪ _tokenSalt,_name,_symbol,_baseURI,_price,_splitter,router,
            ↪ _royalty) (contracts/MADFactory721.sol#343-353)
        - (tokenSalt,deployed) = ERC721BasicDeployer._721BasicDeploy(
            ↪ _tokenSalt,_name,_symbol,_baseURI,_price,_maxSupply,
            ↪ _splitter,router,_royalty) (contracts/MADFactory721.sol
            ↪ #377-388)
        - (tokenSalt,deployed) = ERC721WhitelistDeployer.
            ↪ _721WhitelistDeploy(_tokenSalt,_name,_symbol,_baseURI,
            ↪ _price,_maxSupply,_splitter,router,_royalty) (contracts/
            ↪ MADFactory721.sol#412-423)
        - (tokenSalt,deployed) = ERC721LazyDeployer._721LazyDeploy(
            ↪ _tokenSalt,_name,_symbol,_baseURI,_splitter,router,signer,
            ↪ _royalty) (contracts/MADFactory721.sol#447-457)
        State variables written after the call(s):
        - userTokens[tx.origin].push(colId_scope_8) (contracts/
            ↪ MADFactory721.sol#460)
    Reference: https://github.com/crytic/slither/wiki/Detector-Documentation
        ↪ #reentrancy-vulnerabilities-1
```

```
MADFactory1155.creatorCheck(bytes32) (contracts/MADFactory1155.sol
    ↪ #729-752) uses tx.origin for authorization: creator == origin()()
    ↪  (contracts/MADFactory1155.sol#743-745)
MADFactory721.creatorCheck(bytes32) (contracts/MADFactory721.sol
    ↪ #737-760) uses tx.origin for authorization: creator == origin()()
    ↪  (contracts/MADFactory721.sol#751-753)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation
    ↪ #dangerous-usage-of-txorigin

MADFactory721.createCollection(uint8,string,string,string,uint256,
    ↪ uint256,string,address,uint256).deployed_scope_1 (contracts/
    ↪ MADFactory721.sol#377) is a local variable never initialized
MADFactory1155.createCollection(uint8,string,string,string,uint256,
    ↪ uint256,string,address,uint256).deployed_scope_7 (contracts/
    ↪ MADFactory1155.sol#442) is a local variable never initialized
MADFactory721.createCollection(uint8,string,string,string,uint256,
    ↪ uint256,string,address,uint256).tokenSalt_scope_6 (contracts/
    ↪ MADFactory721.sol#447) is a local variable never initialized
MADFactory721.createCollection(uint8,string,string,string,uint256,
    ↪ uint256,string,address,uint256).deployed_scope_7 (contracts/
    ↪ MADFactory721.sol#447) is a local variable never initialized
MADFactory1155.createCollection(uint8,string,string,string,uint256,
    ↪ uint256,string,address,uint256).tokenSalt_scope_3 (contracts/
    ↪ MADFactory1155.sol#409) is a local variable never initialized
MADFactory1155.createCollection(uint8,string,string,string,uint256,
    ↪ uint256,string,address,uint256).deployed_scope_4 (contracts/
    ↪ MADFactory1155.sol#409) is a local variable never initialized
MADFactory721.createCollection(uint8,string,string,string,uint256,
    ↪ uint256,string,address,uint256).tokenSalt_scope_3 (contracts/
    ↪ MADFactory721.sol#412) is a local variable never initialized
MADFactory721.createCollection(uint8,string,string,string,uint256,
    ↪ uint256,string,address,uint256).deployed_scope_4 (contracts/
    ↪ MADFactory721.sol#412) is a local variable never initialized
```

```
MADFactory1155.createCollection(uint8,string,string,string,uint256,
    ↪ uint256,string,address,uint256).tokenSalt_scope_0 (contracts/
    ↪ MADFactory1155.sol#376) is a local variable never initialized
MADFactory1155.createCollection(uint8,string,string,string,uint256,
    ↪ uint256,string,address,uint256).deployed_scope_1 (contracts/
    ↪ MADFactory1155.sol#376) is a local variable never initialized
MADFactory721.createCollection(uint8,string,string,string,uint256,
    ↪ uint256,string,address,uint256).tokenSalt_scope_0 (contracts/
    ↪ MADFactory721.sol#377) is a local variable never initialized
MADFactory1155.createCollection(uint8,string,string,string,uint256,
    ↪ uint256,string,address,uint256).tokenSalt_scope_6 (contracts/
    ↪ MADFactory1155.sol#442) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation
    ↪ #uninitialized-local-variables

MADRouter1155._tokenRender(address) (contracts/MADRouter1155.sol
    ↪ #419-427) ignores return value by MADFactory1155.creatorCheck(
    ↪ colID) (contracts/MADRouter1155.sol#425)
MADRouter721._tokenRender(address) (contracts/MADRouter721.sol#367-375)
    ↪ ignores return value by MADFactory721.creatorCheck(colID) (
    ↪ contracts/MADRouter721.sol#373)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation
    ↪ #unused-return

MADRouter1155.feeLookup(bytes4).fee (contracts/MADRouter1155.sol#385) is
    ↪  written in both
        fee = sload(uint256)(feeBurn) (contracts/MADRouter1155.sol#394)
        fee = 0x00 (contracts/MADRouter1155.sol#397)
MADRouter721.feeLookup(bytes4).fee (contracts/MADRouter721.sol#334) is
    ↪ written in both
        fee = sload(uint256)(feeBurn) (contracts/MADRouter721.sol#343)
        fee = 0x00 (contracts/MADRouter721.sol#346)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation
    ↪ #write-after-write
```

```
Variable 'MADFactory1155.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).tokenSalt (contracts/
    ↪ MADFactory1155.sol#344)' in MADFactory1155.createCollection(uint8
    ↪ ,string,string,string,uint256,uint256,string,address,uint256) (
    ↪ contracts/MADFactory1155.sol#324-473) potentially used before
    ↪ declaration: (tokenSalt,deployed) = ERC1155BasicDeployer.
    ↪ _1155BasicDeploy(_tokenSalt,_uri,_price,_maxSupply,_splitter,
    ↪ router,_royalty) (contracts/MADFactory1155.sol#376-385)
Variable 'MADFactory1155.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).deployed (contracts/
    ↪ MADFactory1155.sol#344)' in MADFactory1155.createCollection(uint8
    ↪ ,string,string,string,uint256,uint256,string,address,uint256) (
    ↪ contracts/MADFactory1155.sol#324-473) potentially used before
    ↪ declaration: (tokenSalt,deployed) = ERC1155BasicDeployer.
    ↪ _1155BasicDeploy(_tokenSalt,_uri,_price,_maxSupply,_splitter,
    ↪ router,_royalty) (contracts/MADFactory1155.sol#376-385)
Variable 'MADFactory1155.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).tokenSalt (contracts/
    ↪ MADFactory1155.sol#344)' in MADFactory1155.createCollection(uint8
    ↪ ,string,string,string,uint256,uint256,string,address,uint256) (
    ↪ contracts/MADFactory1155.sol#324-473) potentially used before
    ↪ declaration: (tokenSalt,deployed) = ERC1155WhitelistDeployer.
    ↪ _1155WhitelistDeploy(_tokenSalt,_uri,_price,_maxSupply,_splitter,
    ↪ router,_royalty) (contracts/MADFactory1155.sol#409-418)
Variable 'MADFactory1155.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).deployed (contracts/
    ↪ MADFactory1155.sol#344)' in MADFactory1155.createCollection(uint8
    ↪ ,string,string,string,uint256,uint256,string,address,uint256) (
    ↪ contracts/MADFactory1155.sol#324-473) potentially used before
    ↪ declaration: (tokenSalt,deployed) = ERC1155WhitelistDeployer.
    ↪ _1155WhitelistDeploy(_tokenSalt,_uri,_price,_maxSupply,_splitter,
    ↪ router,_royalty) (contracts/MADFactory1155.sol#409-418)
```

```
Variable 'MADFactory1155.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).tokenSalt (contracts/
    ↪ MADFactory1155.sol#344)' in MADFactory1155.createCollection(uint8
    ↪ ,string,string,string,uint256,uint256,string,address,uint256) (
    ↪ contracts/MADFactory1155.sol#324-473) potentially used before
    ↪ declaration: (tokenSalt,deployed) = ERC1155LazyDeployer.
    ↪ _1155LazyDeploy(_tokenSalt,_uri,_splitter,router,signer,_royalty)
    ↪  (contracts/MADFactory1155.sol#442-450)
Variable 'MADFactory1155.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).deployed (contracts/
    ↪ MADFactory1155.sol#344)' in MADFactory1155.createCollection(uint8
    ↪ ,string,string,string,uint256,uint256,string,address,uint256) (
    ↪ contracts/MADFactory1155.sol#324-473) potentially used before
    ↪ declaration: (tokenSalt,deployed) = ERC1155LazyDeployer.
    ↪ _1155LazyDeploy(_tokenSalt,_uri,_splitter,router,signer,_royalty)
    ↪  (contracts/MADFactory1155.sol#442-450)
Variable 'MADFactory721.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).deployed (contracts/
    ↪ MADFactory721.sol#343)' in MADFactory721.createCollection(uint8,
    ↪ string,string,string,uint256,uint256,string,address,uint256) (
    ↪ contracts/MADFactory721.sol#323-480) potentially used before
    ↪ declaration: (tokenSalt,deployed) = ERC721BasicDeployer.
    ↪ _721BasicDeploy(_tokenSalt,_name,_symbol,_baseURI,_price,
    ↪ _maxSupply,_splitter,router,_royalty) (contracts/MADFactory721.
    ↪ sol#377-388)
Variable 'MADFactory721.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).tokenSalt (contracts/
    ↪ MADFactory721.sol#343)' in MADFactory721.createCollection(uint8,
    ↪ string,string,string,uint256,uint256,string,address,uint256) (
    ↪ contracts/MADFactory721.sol#323-480) potentially used before
    ↪ declaration: (tokenSalt,deployed) = ERC721BasicDeployer.
    ↪ _721BasicDeploy(_tokenSalt,_name,_symbol,_baseURI,_price,
    ↪ _maxSupply,_splitter,router,_royalty) (contracts/MADFactory721.
    ↪ sol#377-388)
```

```
Variable 'MADFactory721.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).deployed (contracts/
    ↪ MADFactory721.sol#343)' in MADFactory721.createCollection(uint8,
    ↪ string,string,string,uint256,uint256,string,address,uint256) (
    ↪ contracts/MADFactory721.sol#323-480) potentially used before
    ↪ declaration: (tokenSalt,deployed) = ERC721WhitelistDeployer.
    ↪ _721WhitelistDeploy(_tokenSalt,_name,_symbol,_baseURI,_price,
    ↪ _maxSupply,_splitter,router,_royalty) (contracts/MADFactory721.
    ↪ sol#412-423)
Variable 'MADFactory721.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).tokenSalt (contracts/
    ↪ MADFactory721.sol#343)' in MADFactory721.createCollection(uint8,
    ↪ string,string,string,uint256,uint256,string,address,uint256) (
    ↪ contracts/MADFactory721.sol#323-480) potentially used before
    ↪ declaration: (tokenSalt,deployed) = ERC721WhitelistDeployer.
    ↪ _721WhitelistDeploy(_tokenSalt,_name,_symbol,_baseURI,_price,
    ↪ _maxSupply,_splitter,router,_royalty) (contracts/MADFactory721.
    ↪ sol#412-423)
Variable 'MADFactory721.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).deployed (contracts/
    ↪ MADFactory721.sol#343)' in MADFactory721.createCollection(uint8,
    ↪ string,string,string,uint256,uint256,string,address,uint256) (
    ↪ contracts/MADFactory721.sol#323-480) potentially used before
    ↪ declaration: (tokenSalt,deployed) = ERC721LazyDeployer.
    ↪ _721LazyDeploy(_tokenSalt,_name,_symbol,_baseURI,_splitter,router
    ↪ ,signer,_royalty) (contracts/MADFactory721.sol#447-457)
Variable 'MADFactory721.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).tokenSalt (contracts/
    ↪ MADFactory721.sol#343)' in MADFactory721.createCollection(uint8,
    ↪ string,string,string,uint256,uint256,string,address,uint256) (
    ↪ contracts/MADFactory721.sol#323-480) potentially used before
    ↪ declaration: (tokenSalt,deployed) = ERC721LazyDeployer.
    ↪ _721LazyDeploy(_tokenSalt,_name,_symbol,_baseURI,_splitter,router
    ↪ ,signer,_royalty) (contracts/MADFactory721.sol#447-457)
```

```
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation
    ↪ #pre-declaration-usage-of-local-variables

Reentrancy in MADFactory1155.createCollection(uint8,string,string,string
    ↪ ,uint256,uint256,string,address,uint256) (contracts/
    ↪ MADFactory1155.sol#324-473):
        External calls:
        - (tokenSalt,deployed) = ERC1155MinimalDeployer.
            ↪ _1155MinimalDeploy(_tokenSalt,_uri,_price,_splitter,router
            ↪ ,_royalty) (contracts/MADFactory1155.sol#344-352)
        State variables written after the call(s):
        - colInfo[colId] = Types.Collection1155(tx.origin,Types.
            ↪ ERC1155Type.ERC1155Minimal,tokenSalt,block.number,
            ↪ _splitter) (contracts/MADFactory1155.sol#357-363)
        - userTokens[tx.origin].push(colId) (contracts/MADFactory1155.sol
            ↪ #355)
Reentrancy in MADFactory1155.createCollection(uint8,string,string,string
    ↪ ,uint256,uint256,string,address,uint256) (contracts/
    ↪ MADFactory1155.sol#324-473):
        External calls:
        - (tokenSalt,deployed) = ERC1155MinimalDeployer.
            ↪ _1155MinimalDeploy(_tokenSalt,_uri,_price,_splitter,router
            ↪ ,_royalty) (contracts/MADFactory1155.sol#344-352)
        - (tokenSalt,deployed) = ERC1155BasicDeployer._1155BasicDeploy(
            ↪ _tokenSalt,_uri,_price,_maxSupply,_splitter,router,
            ↪ _royalty) (contracts/MADFactory1155.sol#376-385)
        State variables written after the call(s):
        - colInfo[colId_scope_2] = Types.Collection1155(tx.origin,Types.
            ↪ ERC1155Type.ERC1155Basic,tokenSalt_scope_0,block.number,
            ↪ _splitter) (contracts/MADFactory1155.sol#390-396)
Reentrancy in MADFactory1155.createCollection(uint8,string,string,string
    ↪ ,uint256,uint256,string,address,uint256) (contracts/
    ↪ MADFactory1155.sol#324-473):
        External calls:
```

```
        - (tokenSalt,deployed) = ERC1155MinimalDeployer.
            ↪ _1155MinimalDeploy(_tokenSalt,_uri,_price,_splitter,router
            ↪ ,_royalty) (contracts/MADFactory1155.sol#344-352)
        - (tokenSalt,deployed) = ERC1155BasicDeployer._1155BasicDeploy(
            ↪ _tokenSalt,_uri,_price,_maxSupply,_splitter,router,
            ↪ _royalty) (contracts/MADFactory1155.sol#376-385)
        - (tokenSalt,deployed) = ERC1155WhitelistDeployer.
            ↪ _1155WhitelistDeploy(_tokenSalt,_uri,_price,_maxSupply,
            ↪ _splitter,router,_royalty) (contracts/MADFactory1155.sol
            ↪ #409-418)
        State variables written after the call(s):
        - colInfo[colId_scope_5] = Types.Collection1155(tx.origin,Types.
            ↪ ERC1155Type.ERC1155Whitelist,tokenSalt_scope_3,block.
            ↪ number,_splitter) (contracts/MADFactory1155.sol#423-429)
    Reentrancy in MADFactory1155.createCollection(uint8,string,string,string
        ↪ ,uint256,uint256,string,address,uint256) (contracts/
        ↪ MADFactory1155.sol#324-473):
        External calls:
        - (tokenSalt,deployed) = ERC1155MinimalDeployer.
            ↪ _1155MinimalDeploy(_tokenSalt,_uri,_price,_splitter,router
            ↪ ,_royalty) (contracts/MADFactory1155.sol#344-352)
        - (tokenSalt,deployed) = ERC1155BasicDeployer._1155BasicDeploy(
            ↪ _tokenSalt,_uri,_price,_maxSupply,_splitter,router,
            ↪ _royalty) (contracts/MADFactory1155.sol#376-385)
        - (tokenSalt,deployed) = ERC1155WhitelistDeployer.
            ↪ _1155WhitelistDeploy(_tokenSalt,_uri,_price,_maxSupply,
            ↪ _splitter,router,_royalty) (contracts/MADFactory1155.sol
            ↪ #409-418)
        - (tokenSalt,deployed) = ERC1155LazyDeployer._1155LazyDeploy(
            ↪ _tokenSalt,_uri,_splitter,router,signer,_royalty) (
            ↪ contracts/MADFactory1155.sol#442-450)
        State variables written after the call(s):
        - colInfo[colId_scope_8] = Types.Collection1155(tx.origin,Types.
            ↪ ERC1155Type.ERC1155Lazy,tokenSalt_scope_6,block.number,
```

```
            ↪ _splitter) (contracts/MADFactory1155.sol#455-461)
    Reentrancy in MADFactory721.createCollection(uint8,string,string,string,
       ↪ uint256,uint256,string,address,uint256) (contracts/MADFactory721.
       ↪ sol#323-480):
            External calls:
            - (tokenSalt,deployed) = ERC721MinimalDeployer._721MinimalDeploy(
               ↪ _tokenSalt,_name,_symbol,_baseURI,_price,_splitter,router,
               ↪ _royalty) (contracts/MADFactory721.sol#343-353)
            State variables written after the call(s):
            - colInfo[colId] = Types.Collection721(tx.origin,Types.ERC721Type
               ↪ .ERC721Minimal,tokenSalt,block.number,_splitter) (
               ↪ contracts/MADFactory721.sol#358-364)
            - userTokens[tx.origin].push(colId) (contracts/MADFactory721.sol
               ↪ #356)
    Reentrancy in MADFactory721.createCollection(uint8,string,string,string,
       ↪ uint256,uint256,string,address,uint256) (contracts/MADFactory721.
       ↪ sol#323-480):
            External calls:
            - (tokenSalt,deployed) = ERC721MinimalDeployer._721MinimalDeploy(
               ↪ _tokenSalt,_name,_symbol,_baseURI,_price,_splitter,router,
               ↪ _royalty) (contracts/MADFactory721.sol#343-353)
            - (tokenSalt,deployed) = ERC721BasicDeployer._721BasicDeploy(
               ↪ _tokenSalt,_name,_symbol,_baseURI,_price,_maxSupply,
               ↪ _splitter,router,_royalty) (contracts/MADFactory721.sol
               ↪ #377-388)
            State variables written after the call(s):
            - colInfo[colId_scope_2] = Types.Collection721(tx.origin,Types.
               ↪ ERC721Type.ERC721Basic,tokenSalt_scope_0,block.number,
               ↪ _splitter) (contracts/MADFactory721.sol#393-399)
    Reentrancy in MADFactory721.createCollection(uint8,string,string,string,
       ↪ uint256,uint256,string,address,uint256) (contracts/MADFactory721.
       ↪ sol#323-480):
            External calls:
```

```
        - (tokenSalt,deployed) = ERC721MinimalDeployer._721MinimalDeploy(
            ↪ _tokenSalt,_name,_symbol,_baseURI,_price,_splitter,router,
            ↪ _royalty) (contracts/MADFactory721.sol#343-353)
        - (tokenSalt,deployed) = ERC721BasicDeployer._721BasicDeploy(
            ↪ _tokenSalt,_name,_symbol,_baseURI,_price,_maxSupply,
            ↪ _splitter,router,_royalty) (contracts/MADFactory721.sol
            ↪ #377-388)
        - (tokenSalt,deployed) = ERC721WhitelistDeployer.
            ↪ _721WhitelistDeploy(_tokenSalt,_name,_symbol,_baseURI,
            ↪ _price,_maxSupply,_splitter,router,_royalty) (contracts/
            ↪ MADFactory721.sol#412-423)
        State variables written after the call(s):
        - colInfo[colId_scope_5] = Types.Collection721(tx.origin,Types.
            ↪ ERC721Type.ERC721Whitelist,tokenSalt_scope_3,block.number,
            ↪ _splitter) (contracts/MADFactory721.sol#428-434)
    Reentrancy in MADFactory721.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256) (contracts/MADFactory721.
    ↪ sol#323-480):
        External calls:
        - (tokenSalt,deployed) = ERC721MinimalDeployer._721MinimalDeploy(
            ↪ _tokenSalt,_name,_symbol,_baseURI,_price,_splitter,router,
            ↪ _royalty) (contracts/MADFactory721.sol#343-353)
        - (tokenSalt,deployed) = ERC721BasicDeployer._721BasicDeploy(
            ↪ _tokenSalt,_name,_symbol,_baseURI,_price,_maxSupply,
            ↪ _splitter,router,_royalty) (contracts/MADFactory721.sol
            ↪ #377-388)
        - (tokenSalt,deployed) = ERC721WhitelistDeployer.
            ↪ _721WhitelistDeploy(_tokenSalt,_name,_symbol,_baseURI,
            ↪ _price,_maxSupply,_splitter,router,_royalty) (contracts/
            ↪ MADFactory721.sol#412-423)
        - (tokenSalt,deployed) = ERC721LazyDeployer._721LazyDeploy(
            ↪ _tokenSalt,_name,_symbol,_baseURI,_splitter,router,signer,
            ↪ _royalty) (contracts/MADFactory721.sol#447-457)
        State variables written after the call(s):
```

```
        - colInfo[colId_scope_8] = Types.Collection721(tx.origin,Types.
            ↪ ERC721Type.ERC721Lazy,tokenSalt_scope_6,block.number,
            ↪ _splitter) (contracts/MADFactory721.sol#462-468)
    Reentrancy in MADFactory1155.splitterCheck(string,address,address,
       ↪ uint256,uint256) (contracts/MADFactory1155.sol#124-304):
        External calls:
        - _splitter = SplitterDeployer._SplitterDeploy(_splitterSalt,
            ↪ _payees,_shares) (contracts/MADFactory1155.sol#146-150)
        State variables written after the call(s):
        - splitterInfo[tx.origin][_splitter] = Types.SplitterConfig(
            ↪ _splitter,splitterSalt,address(0),address(0),0,0,true) (
            ↪ contracts/MADFactory1155.sol#152-161)
    Reentrancy in MADFactory1155.splitterCheck(string,address,address,
       ↪ uint256,uint256) (contracts/MADFactory1155.sol#124-304):
        External calls:
        - _splitter_scope_2 = SplitterDeployer._SplitterDeploy(
            ↪ _splitterSalt,_payees_scope_0,_shares_scope_1) (contracts/
            ↪ MADFactory1155.sol#185-189)
        State variables written after the call(s):
        - splitterInfo[tx.origin][_splitter_scope_2] = Types.
            ↪ SplitterConfig(_splitter_scope_2,splitterSalt,_ambassador,
            ↪ address(0),_ambShare,0,true) (contracts/MADFactory1155.sol
            ↪ #191-200)
    Reentrancy in MADFactory1155.splitterCheck(string,address,address,
       ↪ uint256,uint256) (contracts/MADFactory1155.sol#124-304):
        External calls:
        - _splitter_scope_5 = SplitterDeployer._SplitterDeploy(
            ↪ _splitterSalt,_payees_scope_3,_shares_scope_4) (contracts/
            ↪ MADFactory1155.sol#224-228)
        State variables written after the call(s):
        - splitterInfo[tx.origin][_splitter_scope_5] = Types.
            ↪ SplitterConfig(_splitter_scope_5,splitterSalt,address(0),
            ↪ _project,0,_projectShare,true) (contracts/MADFactory1155.
            ↪ sol#230-239)
```

```
Reentrancy in MADFactory1155.splitterCheck(string,address,address,
    ↪ uint256,uint256) (contracts/MADFactory1155.sol#124-304):
        External calls:
        - _splitter_scope_8 = SplitterDeployer._SplitterDeploy(
            ↪ _splitterSalt,_payees_scope_6,_shares_scope_7) (contracts/
            ↪ MADFactory1155.sol#271-275)
        State variables written after the call(s):
        - splitterInfo[tx.origin][_splitter_scope_8] = Types.
            ↪ SplitterConfig(_splitter_scope_8,splitterSalt,_ambassador,
            ↪ _project,_ambShare,_projectShare,true) (contracts/
            ↪ MADFactory1155.sol#277-286)
Reentrancy in MADFactory721.splitterCheck(string,address,address,uint256
    ↪ ,uint256) (contracts/MADFactory721.sol#123-303):
        External calls:
        - _splitter = SplitterDeployer._SplitterDeploy(_splitterSalt,
            ↪ _payees,_shares) (contracts/MADFactory721.sol#145-149)
        State variables written after the call(s):
        - splitterInfo[tx.origin][_splitter] = Types.SplitterConfig(
            ↪ _splitter,splitterSalt,address(0),address(0),0,0,true) (
            ↪ contracts/MADFactory721.sol#151-160)
Reentrancy in MADFactory721.splitterCheck(string,address,address,uint256
    ↪ ,uint256) (contracts/MADFactory721.sol#123-303):
        External calls:
        - _splitter_scope_2 = SplitterDeployer._SplitterDeploy(
            ↪ _splitterSalt,_payees_scope_0,_shares_scope_1) (contracts/
            ↪ MADFactory721.sol#184-188)
        State variables written after the call(s):
        - splitterInfo[tx.origin][_splitter_scope_2] = Types.
            ↪ SplitterConfig(_splitter_scope_2,splitterSalt,_ambassador,
            ↪ address(0),_ambShare,0,true) (contracts/MADFactory721.sol
            ↪ #190-199)
Reentrancy in MADFactory721.splitterCheck(string,address,address,uint256
    ↪ ,uint256) (contracts/MADFactory721.sol#123-303):
        External calls:
```

```
        - _splitter_scope_5 = SplitterDeployer._SplitterDeploy(
            ↪ _splitterSalt,_payees_scope_3,_shares_scope_4) (contracts/
            ↪ MADFactory721.sol#223-227)
        State variables written after the call(s):
        - splitterInfo[tx.origin][_splitter_scope_5] = Types.
            ↪ SplitterConfig(_splitter_scope_5,splitterSalt,address(0),
            ↪ _project,0,_projectShare,true) (contracts/MADFactory721.
            ↪ sol#229-238)
    Reentrancy in MADFactory721.splitterCheck(string,address,address,uint256
        ↪ ,uint256) (contracts/MADFactory721.sol#123-303):
        External calls:
        - _splitter_scope_8 = SplitterDeployer._SplitterDeploy(
            ↪ _splitterSalt,_payees_scope_6,_shares_scope_7) (contracts/
            ↪ MADFactory721.sol#270-274)
        State variables written after the call(s):
        - splitterInfo[tx.origin][_splitter_scope_8] = Types.
            ↪ SplitterConfig(_splitter_scope_8,splitterSalt,_ambassador,
            ↪ _project,_ambShare,_projectShare,true) (contracts/
            ↪ MADFactory721.sol#276-285)
    Reference: https://github.com/crytic/slither/wiki/Detector-Documentation
        ↪ #reentrancy-vulnerabilities-2

    Reentrancy in MADMarketplace721._extPath0(Types.Order721,uint256,bytes32
        ↪ ,address) (contracts/MADMarketplace721.sol#641-683):
        External calls:
        - _order.token.safeTransferFrom(address(this),_to,_order.tokenId)
            ↪  (contracts/MADMarketplace721.sol#670-674)
        Event emitted after the call(s):
        - Claim(_order.token,_order.tokenId,_orderId,_order.seller,_to,
            ↪ _price) (contracts/MADMarketplace721.sol#675-682)
    Reentrancy in MADMarketplace721._extPath1(Types.Order721,uint256,bytes32
        ↪ ,address) (contracts/MADMarketplace721.sol#685-716):
        External calls:
```

```
        - _order.token.safeTransferFrom(address(this),_to,_order.tokenId)
            ↪  (contracts/MADMarketplace721.sol#703-707)
        Event emitted after the call(s):
        - Claim(_order.token,_order.tokenId,_orderId,_order.seller,_to,
            ↪ _price) (contracts/MADMarketplace721.sol#708-715)
Reentrancy in MADMarketplace721._intPath(Types.Order721,uint256,bytes32,
    ↪ address,uint256) (contracts/MADMarketplace721.sol#596-639):
        External calls:
        - _order.token.safeTransferFrom(address(this),_to,_order.tokenId)
            ↪  (contracts/MADMarketplace721.sol#626-630)
        Event emitted after the call(s):
        - Claim(_order.token,_order.tokenId,_orderId,_order.seller,_to,
            ↪ _price) (contracts/MADMarketplace721.sol#631-638)
Reentrancy in MADFactory1155.createCollection(uint8,string,string,string
    ↪ ,uint256,uint256,string,address,uint256) (contracts/
    ↪ MADFactory1155.sol#324-473):
        External calls:
        - (tokenSalt,deployed) = ERC1155MinimalDeployer.
            ↪ _1155MinimalDeploy(_tokenSalt,_uri,_price,_splitter,router
            ↪ ,_royalty) (contracts/MADFactory1155.sol#344-352)
        Event emitted after the call(s):
        - ERC1155MinimalCreated(_splitter,deployed,_name,_symbol,_royalty
            ↪ ,_maxSupply,_price) (contracts/MADFactory1155.sol#365-373)
Reentrancy in MADFactory1155.createCollection(uint8,string,string,string
    ↪ ,uint256,uint256,string,address,uint256) (contracts/
    ↪ MADFactory1155.sol#324-473):
        External calls:
        - (tokenSalt,deployed) = ERC1155MinimalDeployer.
            ↪ _1155MinimalDeploy(_tokenSalt,_uri,_price,_splitter,router
            ↪ ,_royalty) (contracts/MADFactory1155.sol#344-352)
        - (tokenSalt,deployed) = ERC1155BasicDeployer._1155BasicDeploy(
            ↪ _tokenSalt,_uri,_price,_maxSupply,_splitter,router,
            ↪ _royalty) (contracts/MADFactory1155.sol#376-385)
        Event emitted after the call(s):
```

```
        - ERC1155BasicCreated(_splitter,deployed_scope_1,_name,_symbol,
            ↪ _royalty,_maxSupply,_price) (contracts/MADFactory1155.sol
            ↪ #398-406)
    Reentrancy in MADFactory1155.createCollection(uint8,string,string,string
        ↪ ,uint256,uint256,string,address,uint256) (contracts/
        ↪ MADFactory1155.sol#324-473):
        External calls:
        - (tokenSalt,deployed) = ERC1155MinimalDeployer.
            ↪ _1155MinimalDeploy(_tokenSalt,_uri,_price,_splitter,router
            ↪ ,_royalty) (contracts/MADFactory1155.sol#344-352)
        - (tokenSalt,deployed) = ERC1155BasicDeployer._1155BasicDeploy(
            ↪ _tokenSalt,_uri,_price,_maxSupply,_splitter,router,
            ↪ _royalty) (contracts/MADFactory1155.sol#376-385)
        - (tokenSalt,deployed) = ERC1155WhitelistDeployer.
            ↪ _1155WhitelistDeploy(_tokenSalt,_uri,_price,_maxSupply,
            ↪ _splitter,router,_royalty) (contracts/MADFactory1155.sol
            ↪ #409-418)
        Event emitted after the call(s):
        - ERC1155WhitelistCreated(_splitter,deployed_scope_4,_name,
            ↪ _symbol,_royalty,_maxSupply,_price) (contracts/
            ↪ MADFactory1155.sol#431-439)
    Reentrancy in MADFactory1155.createCollection(uint8,string,string,string
        ↪ ,uint256,uint256,string,address,uint256) (contracts/
        ↪ MADFactory1155.sol#324-473):
        External calls:
        - (tokenSalt,deployed) = ERC1155MinimalDeployer.
            ↪ _1155MinimalDeploy(_tokenSalt,_uri,_price,_splitter,router
            ↪ ,_royalty) (contracts/MADFactory1155.sol#344-352)
        - (tokenSalt,deployed) = ERC1155BasicDeployer._1155BasicDeploy(
            ↪ _tokenSalt,_uri,_price,_maxSupply,_splitter,router,
            ↪ _royalty) (contracts/MADFactory1155.sol#376-385)
        - (tokenSalt,deployed) = ERC1155WhitelistDeployer.
            ↪ _1155WhitelistDeploy(_tokenSalt,_uri,_price,_maxSupply,
            ↪ _splitter,router,_royalty) (contracts/MADFactory1155.sol
```

```
            ↪ #409-418)
        - (tokenSalt,deployed) = ERC1155LazyDeployer._1155LazyDeploy(
            ↪ _tokenSalt,_uri,_splitter,router,signer,_royalty) (
            ↪ contracts/MADFactory1155.sol#442-450)
        Event emitted after the call(s):
        - ERC1155LazyCreated(_splitter,deployed_scope_7,_name,_symbol,
            ↪ _royalty,_maxSupply,_price) (contracts/MADFactory1155.sol
            ↪ #463-471)
Reentrancy in MADFactory721.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256) (contracts/MADFactory721.
    ↪ sol#323-480):
        External calls:
        - (tokenSalt,deployed) = ERC721MinimalDeployer._721MinimalDeploy(
            ↪ _tokenSalt,_name,_symbol,_baseURI,_price,_splitter,router,
            ↪ _royalty) (contracts/MADFactory721.sol#343-353)
        Event emitted after the call(s):
        - ERC721MinimalCreated(_splitter,deployed,_name,_symbol,_royalty,
            ↪ _maxSupply,_price) (contracts/MADFactory721.sol#366-374)
Reentrancy in MADFactory721.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256) (contracts/MADFactory721.
    ↪ sol#323-480):
        External calls:
        - (tokenSalt,deployed) = ERC721MinimalDeployer._721MinimalDeploy(
            ↪ _tokenSalt,_name,_symbol,_baseURI,_price,_splitter,router,
            ↪ _royalty) (contracts/MADFactory721.sol#343-353)
        - (tokenSalt,deployed) = ERC721BasicDeployer._721BasicDeploy(
            ↪ _tokenSalt,_name,_symbol,_baseURI,_price,_maxSupply,
            ↪ _splitter,router,_royalty) (contracts/MADFactory721.sol
            ↪ #377-388)
        Event emitted after the call(s):
        - ERC721BasicCreated(_splitter,deployed_scope_1,_name,_symbol,
            ↪ _royalty,_maxSupply,_price) (contracts/MADFactory721.sol
            ↪ #401-409)
```

```
Reentrancy in MADFactory721.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256) (contracts/MADFactory721.
    ↪ sol#323-480):
        External calls:
        - (tokenSalt,deployed) = ERC721MinimalDeployer._721MinimalDeploy(
            ↪ _tokenSalt,_name,_symbol,_baseURI,_price,_splitter,router,
            ↪ _royalty) (contracts/MADFactory721.sol#343-353)
        - (tokenSalt,deployed) = ERC721BasicDeployer._721BasicDeploy(
            ↪ _tokenSalt,_name,_symbol,_baseURI,_price,_maxSupply,
            ↪ _splitter,router,_royalty) (contracts/MADFactory721.sol
            ↪ #377-388)
        - (tokenSalt,deployed) = ERC721WhitelistDeployer.
            ↪ _721WhitelistDeploy(_tokenSalt,_name,_symbol,_baseURI,
            ↪ _price,_maxSupply,_splitter,router,_royalty) (contracts/
            ↪ MADFactory721.sol#412-423)
        Event emitted after the call(s):
        - ERC721WhitelistCreated(_splitter,deployed_scope_4,_name,_symbol
            ↪ ,_royalty,_maxSupply,_price) (contracts/MADFactory721.sol
            ↪ #436-444)
Reentrancy in MADFactory721.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256) (contracts/MADFactory721.
    ↪ sol#323-480):
        External calls:
        - (tokenSalt,deployed) = ERC721MinimalDeployer._721MinimalDeploy(
            ↪ _tokenSalt,_name,_symbol,_baseURI,_price,_splitter,router,
            ↪ _royalty) (contracts/MADFactory721.sol#343-353)
        - (tokenSalt,deployed) = ERC721BasicDeployer._721BasicDeploy(
            ↪ _tokenSalt,_name,_symbol,_baseURI,_price,_maxSupply,
            ↪ _splitter,router,_royalty) (contracts/MADFactory721.sol
            ↪ #377-388)
        - (tokenSalt,deployed) = ERC721WhitelistDeployer.
            ↪ _721WhitelistDeploy(_tokenSalt,_name,_symbol,_baseURI,
            ↪ _price,_maxSupply,_splitter,router,_royalty) (contracts/
            ↪ MADFactory721.sol#412-423)
```

```
        - (tokenSalt,deployed) = ERC721LazyDeployer._721LazyDeploy(
            ↪ _tokenSalt,_name,_symbol,_baseURI,_splitter,router,signer,
            ↪ _royalty) (contracts/MADFactory721.sol#447-457)
        Event emitted after the call(s):
        - ERC721LazyCreated(_splitter,deployed_scope_7,_name,_symbol,
            ↪ _royalty,_maxSupply,_price) (contracts/MADFactory721.sol
            ↪ #470-478)
Reentrancy in MADRouter721.setBase(address,string) (contracts/
    ↪ MADRouter721.sol#74-95):
        External calls:
        - ERC721Basic(_token).setBaseURI(_baseURI) (contracts/
            ↪ MADRouter721.sol#84)
        Event emitted after the call(s):
        - BaseURI(_colID,_baseURI) (contracts/MADRouter721.sol#85)
Reentrancy in MADRouter721.setBase(address,string) (contracts/
    ↪ MADRouter721.sol#74-95):
        External calls:
        - ERC721Whitelist(_token).setBaseURI(_baseURI) (contracts/
            ↪ MADRouter721.sol#87)
        Event emitted after the call(s):
        - BaseURI(_colID,_baseURI) (contracts/MADRouter721.sol#88)
Reentrancy in MADRouter721.setBase(address,string) (contracts/
    ↪ MADRouter721.sol#74-95):
        External calls:
        - ERC721Lazy(_token).setBaseURI(_baseURI) (contracts/MADRouter721
            ↪ .sol#90)
        Event emitted after the call(s):
        - BaseURI(_colID,_baseURI) (contracts/MADRouter721.sol#91)
Reentrancy in MADRouter1155.setMintState(address,bool,uint8) (contracts/
    ↪ MADRouter1155.sol#228-252):
        External calls:
        - _stateType0(_tokenType,_token,_state) (contracts/MADRouter1155.
            ↪ sol#239)
```

```
                - ERC1155Minimal(_token).setPublicMintState(_state) (
                    ↪ contracts/MADRouter1155.sol#437)
                - ERC1155Basic(_token).setPublicMintState(_state) (
                    ↪ contracts/MADRouter1155.sol#439)
                - ERC1155Whitelist(_token).setPublicMintState(_state) (
                    ↪ contracts/MADRouter1155.sol#441-443)
        Event emitted after the call(s):
        - PublicMintState(_colID,_tokenType,_state) (contracts/
            ↪ MADRouter1155.sol#240)
Reentrancy in MADRouter1155.setMintState(address,bool,uint8) (contracts/
    ↪ MADRouter1155.sol#228-252):
        External calls:
        - _stateType1(_tokenType,_token,_state) (contracts/MADRouter1155.
            ↪ sol#242)
                - ERC1155Whitelist(_token).setWhitelistMintState(_state) (
                    ↪ contracts/MADRouter1155.sol#455-457)
        Event emitted after the call(s):
        - WhitelistMintState(_colID,_tokenType,_state) (contracts/
            ↪ MADRouter1155.sol#243-247)
Reentrancy in MADRouter1155.setMintState(address,bool,uint8) (contracts/
    ↪ MADRouter1155.sol#228-252):
        External calls:
        - _stateType2(_tokenType,_token,_state) (contracts/MADRouter1155.
            ↪ sol#249)
                - ERC1155Whitelist(_token).setFreeClaimState(_state) (
                    ↪ contracts/MADRouter1155.sol#469-471)
        Event emitted after the call(s):
        - FreeClaimState(_colID,_tokenType,_state) (contracts/
            ↪ MADRouter1155.sol#250)
Reentrancy in MADRouter721.setMintState(address,bool,uint8) (contracts/
    ↪ MADRouter721.sol#193-217):
        External calls:
        - _stateType0(_tokenType,_token,_state) (contracts/MADRouter721.
            ↪ sol#204)
```

```
                - ERC721Minimal(_token).setPublicMintState(_state) (
                    ↪ contracts/MADRouter721.sol#385)
                - ERC721Basic(_token).setPublicMintState(_state) (
                    ↪ contracts/MADRouter721.sol#387)
                - ERC721Whitelist(_token).setPublicMintState(_state) (
                    ↪ contracts/MADRouter721.sol#389-391)
        Event emitted after the call(s):
        - PublicMintState(_colID,_tokenType,_state) (contracts/
            ↪ MADRouter721.sol#205)
Reentrancy in MADRouter721.setMintState(address,bool,uint8) (contracts/
    ↪ MADRouter721.sol#193-217):
        External calls:
        - _stateType1(_tokenType,_token,_state) (contracts/MADRouter721.
            ↪ sol#207)
                - ERC721Whitelist(_token).setWhitelistMintState(_state) (
                    ↪ contracts/MADRouter721.sol#403-405)
        Event emitted after the call(s):
        - WhitelistMintState(_colID,_tokenType,_state) (contracts/
            ↪ MADRouter721.sol#208-212)
Reentrancy in MADRouter721.setMintState(address,bool,uint8) (contracts/
    ↪ MADRouter721.sol#193-217):
        External calls:
        - _stateType2(_tokenType,_token,_state) (contracts/MADRouter721.
            ↪ sol#214)
                - ERC721Whitelist(_token).setFreeClaimState(_state) (
                    ↪ contracts/MADRouter721.sol#417)
        Event emitted after the call(s):
        - FreeClaimState(_colID,_tokenType,_state) (contracts/
            ↪ MADRouter721.sol#215)
Reentrancy in MADRouter1155.setURI(address,string) (contracts/
    ↪ MADRouter1155.sol#75-96):
        External calls:
        - ERC1155Basic(_token).setURI(_uri) (contracts/MADRouter1155.sol
            ↪ #85)
```

```
        Event emitted after the call(s):
        - BaseURI(_colID,_uri) (contracts/MADRouter1155.sol#86)
Reentrancy in MADRouter1155.setURI(address,string) (contracts/
    ↪ MADRouter1155.sol#75-96):
        External calls:
        - ERC1155Whitelist(_token).setURI(_uri) (contracts/MADRouter1155.
            ↪ sol#88)
        Event emitted after the call(s):
        - BaseURI(_colID,_uri) (contracts/MADRouter1155.sol#89)
Reentrancy in MADRouter1155.setURI(address,string) (contracts/
    ↪ MADRouter1155.sol#75-96):
        External calls:
        - ERC1155Lazy(_token).setURI(_uri) (contracts/MADRouter1155.sol
            ↪ #91)
        Event emitted after the call(s):
        - BaseURI(_colID,_uri) (contracts/MADRouter1155.sol#92)
Reentrancy in MADFactory1155.splitterCheck(string,address,address,
    ↪ uint256,uint256) (contracts/MADFactory1155.sol#124-304):
        External calls:
        - _splitter = SplitterDeployer._SplitterDeploy(_splitterSalt,
            ↪ _payees,_shares) (contracts/MADFactory1155.sol#146-150)
        Event emitted after the call(s):
        - SplitterCreated(tx.origin,_shares,_payees,_splitter,0) (
            ↪ contracts/MADFactory1155.sol#163-169)
Reentrancy in MADFactory1155.splitterCheck(string,address,address,
    ↪ uint256,uint256) (contracts/MADFactory1155.sol#124-304):
        External calls:
        - _splitter_scope_2 = SplitterDeployer._SplitterDeploy(
            ↪ _splitterSalt,_payees_scope_0,_shares_scope_1) (contracts/
            ↪ MADFactory1155.sol#185-189)
        Event emitted after the call(s):
        - SplitterCreated(tx.origin,_shares_scope_1,_payees_scope_0,
            ↪ _splitter_scope_2,1) (contracts/MADFactory1155.sol
            ↪ #202-208)
```

```
Reentrancy in MADFactory1155.splitterCheck(string,address,address,
    ↪ uint256,uint256) (contracts/MADFactory1155.sol#124-304):
        External calls:
        - _splitter_scope_5 = SplitterDeployer._SplitterDeploy(
            ↪ _splitterSalt,_payees_scope_3,_shares_scope_4) (contracts/
            ↪ MADFactory1155.sol#224-228)
        Event emitted after the call(s):
        - SplitterCreated(tx.origin,_shares_scope_4,_payees_scope_3,
            ↪ _splitter_scope_5,2) (contracts/MADFactory1155.sol
            ↪ #241-247)
Reentrancy in MADFactory1155.splitterCheck(string,address,address,
    ↪ uint256,uint256) (contracts/MADFactory1155.sol#124-304):
        External calls:
        - _splitter_scope_8 = SplitterDeployer._SplitterDeploy(
            ↪ _splitterSalt,_payees_scope_6,_shares_scope_7) (contracts/
            ↪ MADFactory1155.sol#271-275)
        Event emitted after the call(s):
        - SplitterCreated(tx.origin,_shares_scope_7,_payees_scope_6,
            ↪ _splitter_scope_8,3) (contracts/MADFactory1155.sol
            ↪ #288-294)
Reentrancy in MADFactory721.splitterCheck(string,address,address,uint256
    ↪ ,uint256) (contracts/MADFactory721.sol#123-303):
        External calls:
        - _splitter = SplitterDeployer._SplitterDeploy(_splitterSalt,
            ↪ _payees,_shares) (contracts/MADFactory721.sol#145-149)
        Event emitted after the call(s):
        - SplitterCreated(tx.origin,_shares,_payees,_splitter,0) (
            ↪ contracts/MADFactory721.sol#162-168)
Reentrancy in MADFactory721.splitterCheck(string,address,address,uint256
    ↪ ,uint256) (contracts/MADFactory721.sol#123-303):
        External calls:
        - _splitter_scope_2 = SplitterDeployer._SplitterDeploy(
            ↪ _splitterSalt,_payees_scope_0,_shares_scope_1) (contracts/
            ↪ MADFactory721.sol#184-188)
```

```
        Event emitted after the call(s):
        - SplitterCreated(tx.origin,_shares_scope_1,_payees_scope_0,
            ↪ _splitter_scope_2,1) (contracts/MADFactory721.sol#201-207)
    Reentrancy in MADFactory721.splitterCheck(string,address,address,uint256
        ↪ ,uint256) (contracts/MADFactory721.sol#123-303):
        External calls:
        - _splitter_scope_5 = SplitterDeployer._SplitterDeploy(
            ↪ _splitterSalt,_payees_scope_3,_shares_scope_4) (contracts/
            ↪ MADFactory721.sol#223-227)
        Event emitted after the call(s):
        - SplitterCreated(tx.origin,_shares_scope_4,_payees_scope_3,
            ↪ _splitter_scope_5,2) (contracts/MADFactory721.sol#240-246)
    Reentrancy in MADFactory721.splitterCheck(string,address,address,uint256
        ↪ ,uint256) (contracts/MADFactory721.sol#123-303):
        External calls:
        - _splitter_scope_8 = SplitterDeployer._SplitterDeploy(
            ↪ _splitterSalt,_payees_scope_6,_shares_scope_7) (contracts/
            ↪ MADFactory721.sol#270-274)
        Event emitted after the call(s):
        - SplitterCreated(tx.origin,_shares_scope_7,_payees_scope_6,
            ↪ _splitter_scope_8,3) (contracts/MADFactory721.sol#287-293)
    Reentrancy in MADRouter1155.withdraw(address,ERC20) (contracts/
        ↪ MADRouter1155.sol#305-375):
        External calls:
        - ERC1155Minimal(_token).withdrawERC20(_erc20) (contracts/
            ↪ MADRouter1155.sol#315-320)
        - ERC1155Minimal(_token).withdraw() (contracts/MADRouter1155.sol
            ↪ #315-320)
        Event emitted after the call(s):
        - TokenFundsWithdrawn(_colID,_tokenType,msg.sender) (contracts/
            ↪ MADRouter1155.sol#322-326)
    Reentrancy in MADRouter1155.withdraw(address,ERC20) (contracts/
        ↪ MADRouter1155.sol#305-375):
        External calls:
```

```
        - ERC1155Minimal(_token).withdrawERC20(_erc20) (contracts/
            ↪ MADRouter1155.sol#315-320)
        - ERC1155Basic(_token).withdrawERC20(_erc20) (contracts/
            ↪ MADRouter1155.sol#330-335)
        - ERC1155Minimal(_token).withdraw() (contracts/MADRouter1155.sol
            ↪ #315-320)
        - ERC1155Basic(_token).withdraw() (contracts/MADRouter1155.sol
            ↪ #330-335)
        Event emitted after the call(s):
        - TokenFundsWithdrawn(_colID,_tokenType,msg.sender) (contracts/
            ↪ MADRouter1155.sol#337-341)
    Reentrancy in MADRouter1155.withdraw(address,ERC20) (contracts/
        ↪ MADRouter1155.sol#305-375):
        External calls:
        - ERC1155Minimal(_token).withdrawERC20(_erc20) (contracts/
            ↪ MADRouter1155.sol#315-320)
        - ERC1155Basic(_token).withdrawERC20(_erc20) (contracts/
            ↪ MADRouter1155.sol#330-335)
        - ERC1155Whitelist(_token).withdrawERC20(_erc20) (contracts/
            ↪ MADRouter1155.sol#345-352)
        - ERC1155Minimal(_token).withdraw() (contracts/MADRouter1155.sol
            ↪ #315-320)
        - ERC1155Basic(_token).withdraw() (contracts/MADRouter1155.sol
            ↪ #330-335)
        - ERC1155Whitelist(_token).withdraw() (contracts/MADRouter1155.
            ↪ sol#345-352)
        Event emitted after the call(s):
        - TokenFundsWithdrawn(_colID,_tokenType,msg.sender) (contracts/
            ↪ MADRouter1155.sol#354-358)
    Reentrancy in MADRouter1155.withdraw(address,ERC20) (contracts/
        ↪ MADRouter1155.sol#305-375):
        External calls:
        - ERC1155Minimal(_token).withdrawERC20(_erc20) (contracts/
            ↪ MADRouter1155.sol#315-320)
```

```
        - ERC1155Basic(_token).withdrawERC20(_erc20) (contracts/
            ↪ MADRouter1155.sol#330-335)
        - ERC1155Whitelist(_token).withdrawERC20(_erc20) (contracts/
            ↪ MADRouter1155.sol#345-352)
        - ERC1155Lazy(_token).withdrawERC20(_erc20) (contracts/
            ↪ MADRouter1155.sol#362-367)
        - ERC1155Minimal(_token).withdraw() (contracts/MADRouter1155.sol
            ↪ #315-320)
        - ERC1155Basic(_token).withdraw() (contracts/MADRouter1155.sol
            ↪ #330-335)
        - ERC1155Whitelist(_token).withdraw() (contracts/MADRouter1155.
            ↪ sol#345-352)
        - ERC1155Lazy(_token).withdraw() (contracts/MADRouter1155.sol
            ↪ #362-367)
        Event emitted after the call(s):
        - TokenFundsWithdrawn(_colID,_tokenType,msg.sender) (contracts/
            ↪ MADRouter1155.sol#369-373)
    Reentrancy in MADRouter721.withdraw(address,ERC20) (contracts/
        ↪ MADRouter721.sol#254-324):
        External calls:
        - ERC721Minimal(_token).withdrawERC20(_erc20) (contracts/
            ↪ MADRouter721.sol#264-269)
        - ERC721Minimal(_token).withdraw() (contracts/MADRouter721.sol
            ↪ #264-269)
        Event emitted after the call(s):
        - TokenFundsWithdrawn(_colID,_tokenType,msg.sender) (contracts/
            ↪ MADRouter721.sol#271-275)
    Reentrancy in MADRouter721.withdraw(address,ERC20) (contracts/
        ↪ MADRouter721.sol#254-324):
        External calls:
        - ERC721Minimal(_token).withdrawERC20(_erc20) (contracts/
            ↪ MADRouter721.sol#264-269)
        - ERC721Basic(_token).withdrawERC20(_erc20) (contracts/
            ↪ MADRouter721.sol#279-284)
```

```
        - ERC721Minimal(_token).withdraw() (contracts/MADRouter721.sol
            ↪ #264-269)
        - ERC721Basic(_token).withdraw() (contracts/MADRouter721.sol
            ↪ #279-284)
        Event emitted after the call(s):
        - TokenFundsWithdrawn(_colID,_tokenType,msg.sender) (contracts/
            ↪ MADRouter721.sol#286-290)
    Reentrancy in MADRouter721.withdraw(address,ERC20) (contracts/
        ↪ MADRouter721.sol#254-324):
        External calls:
        - ERC721Minimal(_token).withdrawERC20(_erc20) (contracts/
            ↪ MADRouter721.sol#264-269)
        - ERC721Basic(_token).withdrawERC20(_erc20) (contracts/
            ↪ MADRouter721.sol#279-284)
        - ERC721Whitelist(_token).withdrawERC20(_erc20) (contracts/
            ↪ MADRouter721.sol#294-301)
        - ERC721Minimal(_token).withdraw() (contracts/MADRouter721.sol
            ↪ #264-269)
        - ERC721Basic(_token).withdraw() (contracts/MADRouter721.sol
            ↪ #279-284)
        - ERC721Whitelist(_token).withdraw() (contracts/MADRouter721.sol
            ↪ #294-301)
        Event emitted after the call(s):
        - TokenFundsWithdrawn(_colID,_tokenType,msg.sender) (contracts/
            ↪ MADRouter721.sol#303-307)
    Reentrancy in MADRouter721.withdraw(address,ERC20) (contracts/
        ↪ MADRouter721.sol#254-324):
        External calls:
        - ERC721Minimal(_token).withdrawERC20(_erc20) (contracts/
            ↪ MADRouter721.sol#264-269)
        - ERC721Basic(_token).withdrawERC20(_erc20) (contracts/
            ↪ MADRouter721.sol#279-284)
        - ERC721Whitelist(_token).withdrawERC20(_erc20) (contracts/
            ↪ MADRouter721.sol#294-301)
```

```
        - ERC721Lazy(_token).withdrawERC20(_erc20) (contracts/
            ↪ MADRouter721.sol#311-316)
        - ERC721Minimal(_token).withdraw() (contracts/MADRouter721.sol
            ↪ #264-269)
        - ERC721Basic(_token).withdraw() (contracts/MADRouter721.sol
            ↪ #279-284)
        - ERC721Whitelist(_token).withdraw() (contracts/MADRouter721.sol
            ↪ #294-301)
        - ERC721Lazy(_token).withdraw() (contracts/MADRouter721.sol
            ↪ #311-316)
        Event emitted after the call(s):
        - TokenFundsWithdrawn(_colID,_tokenType,msg.sender) (contracts/
            ↪ MADRouter721.sol#318-322)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation
    ↪ #reentrancy-vulnerabilities-3

MADFactory1155.name() (contracts/MADFactory1155.sol#51-62) uses assembly
        - INLINE ASM (contracts/MADFactory1155.sol#57-61)
MADFactory1155.splitterCheck(string,address,address,uint256,uint256) (
    ↪ contracts/MADFactory1155.sol#124-304) uses assembly
        - INLINE ASM (contracts/MADFactory1155.sol#298-302)
MADFactory1155.setOwner(address) (contracts/MADFactory1155.sol#480-491)
    ↪ uses assembly
        - INLINE ASM (contracts/MADFactory1155.sol#486-488)
MADFactory1155.setMarket(address) (contracts/MADFactory1155.sol#495-501)
    ↪  uses assembly
        - INLINE ASM (contracts/MADFactory1155.sol#496-498)
MADFactory1155.setRouter(address) (contracts/MADFactory1155.sol#505-512)
    ↪  uses assembly
        - INLINE ASM (contracts/MADFactory1155.sol#507-509)
MADFactory1155.setSigner(address) (contracts/MADFactory1155.sol#516-523)
    ↪  uses assembly
        - INLINE ASM (contracts/MADFactory1155.sol#518-520)
```

```
MADFactory1155.typeChecker(bytes32) (contracts/MADFactory1155.sol
    ↪ #595-603) uses assembly
        - INLINE ASM (contracts/MADFactory1155.sol#599-602)
MADFactory1155._payeesBuffer(address,address) (contracts/MADFactory1155.
    ↪ sol#606-653) uses assembly
        - INLINE ASM (contracts/MADFactory1155.sol#613-652)
MADFactory1155._sharesBuffer(uint256,uint256) (contracts/MADFactory1155.
    ↪ sol#656-701) uses assembly
        - INLINE ASM (contracts/MADFactory1155.sol#661-700)
MADFactory1155.creatorAuth(address,address) (contracts/MADFactory1155.
    ↪ sol#704-726) uses assembly
        - INLINE ASM (contracts/MADFactory1155.sol#711)
MADFactory1155.creatorCheck(bytes32) (contracts/MADFactory1155.sol
    ↪ #729-752) uses assembly
        - INLINE ASM (contracts/MADFactory1155.sol#738-751)
MADFactory1155._isRouter() (contracts/MADFactory1155.sol#756-766) uses
    ↪ assembly
        - INLINE ASM (contracts/MADFactory1155.sol#758-765)
MADFactory1155._isMarket() (contracts/MADFactory1155.sol#770-777) uses
    ↪ assembly
        - INLINE ASM (contracts/MADFactory1155.sol#771-776)
MADFactory1155._limiter(uint8,address) (contracts/MADFactory1155.sol
    ↪ #779-793) uses assembly
        - INLINE ASM (contracts/MADFactory1155.sol#787-792)
MADFactory1155._royaltyLocker(uint256) (contracts/MADFactory1155.sol
    ↪ #795-807) uses assembly
        - INLINE ASM (contracts/MADFactory1155.sol#799-806)
MADFactory1155._userRender(address) (contracts/MADFactory1155.sol
    ↪ #813-824) uses assembly
        - INLINE ASM (contracts/MADFactory1155.sol#814-823)
MADFactory721.name() (contracts/MADFactory721.sol#50-61) uses assembly
        - INLINE ASM (contracts/MADFactory721.sol#56-60)
MADFactory721.splitterCheck(string,address,address,uint256,uint256) (
    ↪ contracts/MADFactory721.sol#123-303) uses assembly
```

```
        - INLINE ASM (contracts/MADFactory721.sol#297-301)
MADFactory721.setOwner(address) (contracts/MADFactory721.sol#487-498)
    ↪ uses assembly
        - INLINE ASM (contracts/MADFactory721.sol#493-495)
MADFactory721.setMarket(address) (contracts/MADFactory721.sol#502-508)
    ↪ uses assembly
        - INLINE ASM (contracts/MADFactory721.sol#503-505)
MADFactory721.setRouter(address) (contracts/MADFactory721.sol#512-519)
    ↪ uses assembly
        - INLINE ASM (contracts/MADFactory721.sol#514-516)
MADFactory721.setSigner(address) (contracts/MADFactory721.sol#523-530)
    ↪ uses assembly
        - INLINE ASM (contracts/MADFactory721.sol#525-527)
MADFactory721.typeChecker(bytes32) (contracts/MADFactory721.sol#602-610)
    ↪  uses assembly
        - INLINE ASM (contracts/MADFactory721.sol#606-609)
MADFactory721._payeesBuffer(address,address) (contracts/MADFactory721.
    ↪ sol#613-660) uses assembly
        - INLINE ASM (contracts/MADFactory721.sol#620-659)
MADFactory721._sharesBuffer(uint256,uint256) (contracts/MADFactory721.
    ↪ sol#663-708) uses assembly
        - INLINE ASM (contracts/MADFactory721.sol#668-707)
MADFactory721.creatorAuth(address,address) (contracts/MADFactory721.sol
    ↪ #712-734) uses assembly
        - INLINE ASM (contracts/MADFactory721.sol#719)
MADFactory721.creatorCheck(bytes32) (contracts/MADFactory721.sol
    ↪ #737-760) uses assembly
        - INLINE ASM (contracts/MADFactory721.sol#746-759)
MADFactory721._isRouter() (contracts/MADFactory721.sol#764-774) uses
    ↪ assembly
        - INLINE ASM (contracts/MADFactory721.sol#766-773)
MADFactory721._isMarket() (contracts/MADFactory721.sol#778-785) uses
    ↪ assembly
        - INLINE ASM (contracts/MADFactory721.sol#779-784)
```

```
MADFactory721._limiter(uint8,address) (contracts/MADFactory721.sol
    ↪ #787-801) uses assembly
        - INLINE ASM (contracts/MADFactory721.sol#795-800)
MADFactory721._royaltyLocker(uint256) (contracts/MADFactory721.sol
    ↪ #803-815) uses assembly
        - INLINE ASM (contracts/MADFactory721.sol#807-814)
MADFactory721._userRender(address) (contracts/MADFactory721.sol#821-832)
    ↪  uses assembly
        - INLINE ASM (contracts/MADFactory721.sol#822-831)
MADMarketplace1155.name() (contracts/MADMarketplace1155.sol#32-43) uses
    ↪ assembly
        - INLINE ASM (contracts/MADMarketplace1155.sol#38-42)
MADMarketplace1155.bid(bytes32) (contracts/MADMarketplace1155.sol
    ↪ #168-219) uses assembly
        - INLINE ASM (contracts/MADMarketplace1155.sol#186-203)
MADMarketplace1155.setFactory(FactoryVerifier) (contracts/
    ↪ MADMarketplace1155.sol#394-403) uses assembly
        - INLINE ASM (contracts/MADMarketplace1155.sol#398-401)
MADMarketplace1155.setFees(uint256,uint256) (contracts/
    ↪ MADMarketplace1155.sol#405-420) uses assembly
        - INLINE ASM (contracts/MADMarketplace1155.sol#411-414)
MADMarketplace1155.updateSettings(uint256,uint256,uint256) (contracts/
    ↪ MADMarketplace1155.sol#428-450) uses assembly
        - INLINE ASM (contracts/MADMarketplace1155.sol#436-443)
MADMarketplace1155.setRecipient(address) (contracts/MADMarketplace1155.
    ↪ sol#466-476) uses assembly
        - INLINE ASM (contracts/MADMarketplace1155.sol#471-473)
MADMarketplace1155.setOwner(address) (contracts/MADMarketplace1155.sol
    ↪ #479-490) uses assembly
        - INLINE ASM (contracts/MADMarketplace1155.sol#485-487)
MADMarketplace1155.interfaceCheck(address,bytes4) (contracts/
    ↪ MADMarketplace1155.sol#605-630) uses assembly
        - INLINE ASM (contracts/MADMarketplace1155.sol#616-627)
```

```
MADMarketplace1155._feeResolver(uint256,uint256,uint256) (contracts/
    ↪ MADMarketplace1155.sol#788-812) uses assembly
        - INLINE ASM (contracts/MADMarketplace1155.sol#793-811)
MADMarketplace1155._exceedsMaxEP(uint256,uint256) (contracts/
    ↪ MADMarketplace1155.sol#818-836) uses assembly
        - INLINE ASM (contracts/MADMarketplace1155.sol#822-835)
MADMarketplace1155._isBidderOrSeller(address,address) (contracts/
    ↪ MADMarketplace1155.sol#838-854) uses assembly
        - INLINE ASM (contracts/MADMarketplace1155.sol#842-853)
MADMarketplace1155._makeOrderChecks(uint256,uint256) (contracts/
    ↪ MADMarketplace1155.sol#856-885) uses assembly
        - INLINE ASM (contracts/MADMarketplace1155.sol#860-884)
MADMarketplace1155._cancelOrderChecks(address,bool,uint256) (contracts/
    ↪ MADMarketplace1155.sol#887-909) uses assembly
        - INLINE ASM (contracts/MADMarketplace1155.sol#892-908)
MADMarketplace1155._bidChecks(uint8,uint256,address,uint256,uint256) (
    ↪ contracts/MADMarketplace1155.sol#911-966) uses assembly
        - INLINE ASM (contracts/MADMarketplace1155.sol#918-965)
MADMarketplace1155._buyChecks(uint256,uint8,bool) (contracts/
    ↪ MADMarketplace1155.sol#968-998) uses assembly
        - INLINE ASM (contracts/MADMarketplace1155.sol#973-997)
MADMarketplace1155._claimChecks(bool,uint8,uint256) (contracts/
    ↪ MADMarketplace1155.sol#1000-1025) uses assembly
        - INLINE ASM (contracts/MADMarketplace1155.sol#1005-1024)
MADMarketplace1155.getCurrentPrice(bytes32) (contracts/
    ↪ MADMarketplace1155.sol#1034-1100) uses assembly
        - INLINE ASM (contracts/MADMarketplace1155.sol#1041-1099)
MADMarketplace721.name() (contracts/MADMarketplace721.sol#32-43) uses
    ↪ assembly
        - INLINE ASM (contracts/MADMarketplace721.sol#38-42)
MADMarketplace721.bid(bytes32) (contracts/MADMarketplace721.sol#148-199)
    ↪  uses assembly
        - INLINE ASM (contracts/MADMarketplace721.sol#166-183)
```

```
MADMarketplace721.setFactory(FactoryVerifier) (contracts/
    ↪ MADMarketplace721.sol#370-379) uses assembly
        - INLINE ASM (contracts/MADMarketplace721.sol#374-377)
MADMarketplace721.setFees(uint256,uint256) (contracts/MADMarketplace721.
    ↪ sol#381-396) uses assembly
        - INLINE ASM (contracts/MADMarketplace721.sol#387-390)
MADMarketplace721.updateSettings(uint256,uint256,uint256) (contracts/
    ↪ MADMarketplace721.sol#403-425) uses assembly
        - INLINE ASM (contracts/MADMarketplace721.sol#411-418)
MADMarketplace721.setRecipient(address) (contracts/MADMarketplace721.sol
    ↪ #441-451) uses assembly
        - INLINE ASM (contracts/MADMarketplace721.sol#446-448)
MADMarketplace721.setOwner(address) (contracts/MADMarketplace721.sol
    ↪ #454-465) uses assembly
        - INLINE ASM (contracts/MADMarketplace721.sol#460-462)
MADMarketplace721.interfaceCheck(address,bytes4) (contracts/
    ↪ MADMarketplace721.sol#556-581) uses assembly
        - INLINE ASM (contracts/MADMarketplace721.sol#567-578)
MADMarketplace721._feeResolver(uint256,uint256) (contracts/
    ↪ MADMarketplace721.sol#718-738) uses assembly
        - INLINE ASM (contracts/MADMarketplace721.sol#722-737)
MADMarketplace721._exceedsMaxEP(uint256,uint256) (contracts/
    ↪ MADMarketplace721.sol#744-762) uses assembly
        - INLINE ASM (contracts/MADMarketplace721.sol#748-761)
MADMarketplace721._isBidderOrSeller(address,address) (contracts/
    ↪ MADMarketplace721.sol#764-780) uses assembly
        - INLINE ASM (contracts/MADMarketplace721.sol#768-779)
MADMarketplace721._makeOrderChecks(uint256,uint256) (contracts/
    ↪ MADMarketplace721.sol#782-811) uses assembly
        - INLINE ASM (contracts/MADMarketplace721.sol#786-810)
MADMarketplace721._cancelOrderChecks(address,bool,uint256) (contracts/
    ↪ MADMarketplace721.sol#813-835) uses assembly
        - INLINE ASM (contracts/MADMarketplace721.sol#818-834)
```

```
MADMarketplace721._bidChecks(uint8,uint256,address,uint256,uint256) (
    ↪ contracts/MADMarketplace721.sol#837-892) uses assembly
        - INLINE ASM (contracts/MADMarketplace721.sol#844-891)
MADMarketplace721._buyChecks(uint256,uint8,bool) (contracts/
    ↪ MADMarketplace721.sol#894-924) uses assembly
        - INLINE ASM (contracts/MADMarketplace721.sol#899-923)
MADMarketplace721._claimChecks(bool,uint8,uint256) (contracts/
    ↪ MADMarketplace721.sol#926-951) uses assembly
        - INLINE ASM (contracts/MADMarketplace721.sol#931-950)
MADMarketplace721.getCurrentPrice(bytes32) (contracts/MADMarketplace721.
    ↪ sol#960-1026) uses assembly
        - INLINE ASM (contracts/MADMarketplace721.sol#967-1025)
MADRouter1155.name() (contracts/MADRouter1155.sol#34-45) uses assembly
        - INLINE ASM (contracts/MADRouter1155.sol#40-44)
MADRouter1155.feeLookup(bytes4) (contracts/MADRouter1155.sol#381-401)
    ↪ uses assembly
        - INLINE ASM (contracts/MADRouter1155.sol#387-400)
MADRouter1155.setFees(uint256,uint256) (contracts/MADRouter1155.sol
    ↪ #403-413) uses assembly
        - INLINE ASM (contracts/MADRouter1155.sol#407-410)
MADRouter1155.setOwner(address) (contracts/MADRouter1155.sol#480-491)
    ↪ uses assembly
        - INLINE ASM (contracts/MADRouter1155.sol#486-488)
MADRouter721.name() (contracts/MADRouter721.sol#34-45) uses assembly
        - INLINE ASM (contracts/MADRouter721.sol#40-44)
MADRouter721.feeLookup(bytes4) (contracts/MADRouter721.sol#330-350) uses
    ↪  assembly
        - INLINE ASM (contracts/MADRouter721.sol#336-349)
MADRouter721.setFees(uint256,uint256) (contracts/MADRouter721.sol
    ↪ #352-362) uses assembly
        - INLINE ASM (contracts/MADRouter721.sol#356-359)
MADRouter721.setOwner(address) (contracts/MADRouter721.sol#426-437) uses
    ↪  assembly
        - INLINE ASM (contracts/MADRouter721.sol#432-434)
```

```
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation
    ↪ #assembly-usage

MADMarketplace1155.buy(bytes32) (contracts/MADMarketplace1155.sol
    ↪ #224-284) compares to a boolean constant:
        -ERC165Check(address(order.token)) && interfaceCheck(address(
            ↪ order.token),0x2a55205a) == true (contracts/
            ↪ MADMarketplace1155.sol#260-265)
MADMarketplace1155.claim(bytes32) (contracts/MADMarketplace1155.sol
    ↪ #289-350) compares to a boolean constant:
        -ERC165Check(address(order.token)) && interfaceCheck(address(
            ↪ order.token),0x2a55205a) == true (contracts/
            ↪ MADMarketplace1155.sol#326-331)
MADMarketplace1155.claim(bytes32) (contracts/MADMarketplace1155.sol
    ↪ #289-350) compares to a boolean constant:
        -! feeSelector[key][order.tokenId][order.amount] &&
            ↪ MADFactory1155.creatorAuth(address(order.token),order.
            ↪ seller) == true (contracts/MADMarketplace1155.sol#307-312)
MADMarketplace721.buy(bytes32) (contracts/MADMarketplace721.sol#204-267)
    ↪  compares to a boolean constant:
        -ERC165Check(address(order.token)) && interfaceCheck(address(
            ↪ order.token),0x2a55205a) == true (contracts/
            ↪ MADMarketplace721.sol#241-246)
MADMarketplace721.buy(bytes32) (contracts/MADMarketplace721.sol#204-267)
    ↪  compares to a boolean constant:
        -! feeSelector[key][order.tokenId] && MADFactory721.creatorAuth(
            ↪ address(order.token),order.seller) == true (contracts/
            ↪ MADMarketplace721.sol#228-233)
MADMarketplace721.claim(bytes32) (contracts/MADMarketplace721.sol
    ↪ #272-335) compares to a boolean constant:
        -ERC165Check(address(order.token)) && interfaceCheck(address(
            ↪ order.token),0x2a55205a) == true (contracts/
            ↪ MADMarketplace721.sol#309-314)
```

```
MADMarketplace721.claim(bytes32) (contracts/MADMarketplace721.sol
    ↪ #272-335) compares to a boolean constant:
        -! feeSelector[key][order.tokenId] && MADFactory721.creatorAuth(
            ↪ address(order.token),order.seller) == true (contracts/
            ↪ MADMarketplace721.sol#290-295)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation
    ↪ #boolean-equality

Pragma version0.8.16 (contracts/EventsAndErrors.sol#3) necessitates a
    ↪ version too recent to be trusted. Consider deploying with
    ↪ 0.6.12/0.7.6/0.8.7
Pragma version0.8.16 (contracts/MAD.sol#9) necessitates a version too
    ↪ recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
Pragma version0.8.16 (contracts/MADFactory1155.sol#3) necessitates a
    ↪ version too recent to be trusted. Consider deploying with
    ↪ 0.6.12/0.7.6/0.8.7
Pragma version0.8.16 (contracts/MADFactory721.sol#3) necessitates a
    ↪ version too recent to be trusted. Consider deploying with
    ↪ 0.6.12/0.7.6/0.8.7
Pragma version0.8.16 (contracts/MADMarketplace1155.sol#3) necessitates a
    ↪  version too recent to be trusted. Consider deploying with
    ↪ 0.6.12/0.7.6/0.8.7
Pragma version0.8.16 (contracts/MADMarketplace721.sol#3) necessitates a
    ↪ version too recent to be trusted. Consider deploying with
    ↪ 0.6.12/0.7.6/0.8.7
Pragma version0.8.16 (contracts/MADRouter1155.sol#9) necessitates a
    ↪ version too recent to be trusted. Consider deploying with
    ↪ 0.6.12/0.7.6/0.8.7
Pragma version0.8.16 (contracts/MADRouter721.sol#9) necessitates a
    ↪ version too recent to be trusted. Consider deploying with
    ↪ 0.6.12/0.7.6/0.8.7
Pragma version0.8.16 (contracts/Types.sol#3) necessitates a version too
    ↪ recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.7
solc-0.8.16 is not recommended for deployment
```

```
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation
    ↪ #incorrect-versions-of-solidity

Parameter MADFactory1155.splitterCheck(string,address,address,uint256,
    ↪ uint256)._splitterSalt (contracts/MADFactory1155.sol#125) is not
    ↪ in mixedCase
Parameter MADFactory1155.splitterCheck(string,address,address,uint256,
    ↪ uint256)._ambassador (contracts/MADFactory1155.sol#126) is not in
    ↪  mixedCase
Parameter MADFactory1155.splitterCheck(string,address,address,uint256,
    ↪ uint256)._project (contracts/MADFactory1155.sol#127) is not in
    ↪ mixedCase
Parameter MADFactory1155.splitterCheck(string,address,address,uint256,
    ↪ uint256)._ambShare (contracts/MADFactory1155.sol#128) is not in
    ↪ mixedCase
Parameter MADFactory1155.splitterCheck(string,address,address,uint256,
    ↪ uint256)._projectShare (contracts/MADFactory1155.sol#129) is not
    ↪ in mixedCase
Parameter MADFactory1155.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256)._tokenType (contracts/
    ↪ MADFactory1155.sol#325) is not in mixedCase
Parameter MADFactory1155.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256)._tokenSalt (contracts/
    ↪ MADFactory1155.sol#326) is not in mixedCase
Parameter MADFactory1155.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256)._name (contracts/
    ↪ MADFactory1155.sol#327) is not in mixedCase
Parameter MADFactory1155.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256)._symbol (contracts/
    ↪ MADFactory1155.sol#328) is not in mixedCase
Parameter MADFactory1155.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256)._price (contracts/
    ↪ MADFactory1155.sol#329) is not in mixedCase
```

```
Parameter MADFactory1155.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256)._maxSupply (contracts/
    ↪ MADFactory1155.sol#330) is not in mixedCase
Parameter MADFactory1155.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256)._uri (contracts/
    ↪ MADFactory1155.sol#331) is not in mixedCase
Parameter MADFactory1155.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256)._splitter (contracts/
    ↪ MADFactory1155.sol#332) is not in mixedCase
Parameter MADFactory1155.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256)._royalty (contracts/
    ↪ MADFactory1155.sol#333) is not in mixedCase
Parameter MADFactory1155.setMarket(address)._market (contracts/
    ↪ MADFactory1155.sol#495) is not in mixedCase
Parameter MADFactory1155.setRouter(address)._router (contracts/
    ↪ MADFactory1155.sol#505) is not in mixedCase
Parameter MADFactory1155.setSigner(address)._signer (contracts/
    ↪ MADFactory1155.sol#516) is not in mixedCase
Parameter MADFactory1155.getIDsLength(address)._user (contracts/
    ↪ MADFactory1155.sol#580) is not in mixedCase
Parameter MADFactory1155.getColID(address)._colAddress (contracts/
    ↪ MADFactory1155.sol#589) is not in mixedCase
Parameter MADFactory1155.typeChecker(bytes32)._colID (contracts/
    ↪ MADFactory1155.sol#595) is not in mixedCase
Parameter MADFactory1155.creatorAuth(address,address)._token (contracts/
    ↪ MADFactory1155.sol#704) is not in mixedCase
Parameter MADFactory1155.creatorAuth(address,address)._user (contracts/
    ↪ MADFactory1155.sol#704) is not in mixedCase
Parameter MADFactory1155.creatorCheck(bytes32)._colID (contracts/
    ↪ MADFactory1155.sol#729) is not in mixedCase
Parameter MADFactory1155.getDeployedAddr(string)._salt (contracts/
    ↪ MADFactory1155.sol#826) is not in mixedCase
Parameter MADFactory721.splitterCheck(string,address,address,uint256,
    ↪ uint256)._splitterSalt (contracts/MADFactory721.sol#124) is not
```

```
    ↪ in mixedCase
Parameter MADFactory721.splitterCheck(string,address,address,uint256,
    ↪ uint256)._ambassador (contracts/MADFactory721.sol#125) is not in
    ↪ mixedCase
Parameter MADFactory721.splitterCheck(string,address,address,uint256,
    ↪ uint256)._project (contracts/MADFactory721.sol#126) is not in
    ↪ mixedCase
Parameter MADFactory721.splitterCheck(string,address,address,uint256,
    ↪ uint256)._ambShare (contracts/MADFactory721.sol#127) is not in
    ↪ mixedCase
Parameter MADFactory721.splitterCheck(string,address,address,uint256,
    ↪ uint256)._projectShare (contracts/MADFactory721.sol#128) is not
    ↪ in mixedCase
Parameter MADFactory721.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256)._tokenType (contracts/
    ↪ MADFactory721.sol#324) is not in mixedCase
Parameter MADFactory721.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256)._tokenSalt (contracts/
    ↪ MADFactory721.sol#325) is not in mixedCase
Parameter MADFactory721.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256)._name (contracts/
    ↪ MADFactory721.sol#326) is not in mixedCase
Parameter MADFactory721.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256)._symbol (contracts/
    ↪ MADFactory721.sol#327) is not in mixedCase
Parameter MADFactory721.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256)._price (contracts/
    ↪ MADFactory721.sol#328) is not in mixedCase
Parameter MADFactory721.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256)._maxSupply (contracts/
    ↪ MADFactory721.sol#329) is not in mixedCase
Parameter MADFactory721.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256)._baseURI (contracts/
    ↪ MADFactory721.sol#330) is not in mixedCase
```

```
Parameter MADFactory721.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256)._splitter (contracts/
    ↪ MADFactory721.sol#331) is not in mixedCase
Parameter MADFactory721.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256)._royalty (contracts/
    ↪ MADFactory721.sol#332) is not in mixedCase
Parameter MADFactory721.setMarket(address)._market (contracts/
    ↪ MADFactory721.sol#502) is not in mixedCase
Parameter MADFactory721.setRouter(address)._router (contracts/
    ↪ MADFactory721.sol#512) is not in mixedCase
Parameter MADFactory721.setSigner(address)._signer (contracts/
    ↪ MADFactory721.sol#523) is not in mixedCase
Parameter MADFactory721.getIDsLength(address)._user (contracts/
    ↪ MADFactory721.sol#587) is not in mixedCase
Parameter MADFactory721.getColID(address)._colAddress (contracts/
    ↪ MADFactory721.sol#596) is not in mixedCase
Parameter MADFactory721.typeChecker(bytes32)._colID (contracts/
    ↪ MADFactory721.sol#602) is not in mixedCase
Parameter MADFactory721.creatorAuth(address,address)._token (contracts/
    ↪ MADFactory721.sol#712) is not in mixedCase
Parameter MADFactory721.creatorAuth(address,address)._user (contracts/
    ↪ MADFactory721.sol#712) is not in mixedCase
Parameter MADFactory721.creatorCheck(bytes32)._colID (contracts/
    ↪ MADFactory721.sol#737) is not in mixedCase
Parameter MADFactory721.getDeployedAddr(string)._salt (contracts/
    ↪ MADFactory721.sol#834) is not in mixedCase
Parameter MADMarketplace1155.fixedPrice(IERC1155,uint256,uint256,uint256
    ↪ ,uint256)._token (contracts/MADMarketplace1155.sol#104) is not in
    ↪  mixedCase
Parameter MADMarketplace1155.fixedPrice(IERC1155,uint256,uint256,uint256
    ↪ ,uint256)._id (contracts/MADMarketplace1155.sol#105) is not in
    ↪ mixedCase
Parameter MADMarketplace1155.fixedPrice(IERC1155,uint256,uint256,uint256
    ↪ ,uint256)._amount (contracts/MADMarketplace1155.sol#106) is not
```

```
    ↪ in mixedCase
Parameter MADMarketplace1155.fixedPrice(IERC1155,uint256,uint256,uint256
    ↪ ,uint256)._price (contracts/MADMarketplace1155.sol#107) is not in
    ↪  mixedCase
Parameter MADMarketplace1155.fixedPrice(IERC1155,uint256,uint256,uint256
    ↪ ,uint256)._endTime (contracts/MADMarketplace1155.sol#108) is not
    ↪ in mixedCase
Parameter MADMarketplace1155.dutchAuction(IERC1155,uint256,uint256,
    ↪ uint256,uint256,uint256)._token (contracts/MADMarketplace1155.sol
    ↪ #124) is not in mixedCase
Parameter MADMarketplace1155.dutchAuction(IERC1155,uint256,uint256,
    ↪ uint256,uint256,uint256)._id (contracts/MADMarketplace1155.sol
    ↪ #125) is not in mixedCase
Parameter MADMarketplace1155.dutchAuction(IERC1155,uint256,uint256,
    ↪ uint256,uint256,uint256)._amount (contracts/MADMarketplace1155.
    ↪ sol#126) is not in mixedCase
Parameter MADMarketplace1155.dutchAuction(IERC1155,uint256,uint256,
    ↪ uint256,uint256,uint256)._startPrice (contracts/
    ↪ MADMarketplace1155.sol#127) is not in mixedCase
Parameter MADMarketplace1155.dutchAuction(IERC1155,uint256,uint256,
    ↪ uint256,uint256,uint256)._endPrice (contracts/MADMarketplace1155.
    ↪ sol#128) is not in mixedCase
Parameter MADMarketplace1155.dutchAuction(IERC1155,uint256,uint256,
    ↪ uint256,uint256,uint256)._endTime (contracts/MADMarketplace1155.
    ↪ sol#129) is not in mixedCase
Parameter MADMarketplace1155.englishAuction(IERC1155,uint256,uint256,
    ↪ uint256,uint256)._token (contracts/MADMarketplace1155.sol#146) is
    ↪  not in mixedCase
Parameter MADMarketplace1155.englishAuction(IERC1155,uint256,uint256,
    ↪ uint256,uint256)._id (contracts/MADMarketplace1155.sol#147) is
    ↪ not in mixedCase
Parameter MADMarketplace1155.englishAuction(IERC1155,uint256,uint256,
    ↪ uint256,uint256)._amount (contracts/MADMarketplace1155.sol#148)
    ↪ is not in mixedCase
```

```
Parameter MADMarketplace1155.englishAuction(IERC1155,uint256,uint256,
    ↪ uint256,uint256)._startPrice (contracts/MADMarketplace1155.sol
    ↪ #149) is not in mixedCase
Parameter MADMarketplace1155.englishAuction(IERC1155,uint256,uint256,
    ↪ uint256,uint256)._endTime (contracts/MADMarketplace1155.sol#150)
    ↪ is not in mixedCase
Parameter MADMarketplace1155.bid(bytes32)._order (contracts/
    ↪ MADMarketplace1155.sol#168) is not in mixedCase
Parameter MADMarketplace1155.buy(bytes32)._order (contracts/
    ↪ MADMarketplace1155.sol#224) is not in mixedCase
Parameter MADMarketplace1155.claim(bytes32)._order (contracts/
    ↪ MADMarketplace1155.sol#289) is not in mixedCase
Parameter MADMarketplace1155.cancelOrder(bytes32)._order (contracts/
    ↪ MADMarketplace1155.sol#355) is not in mixedCase
Parameter MADMarketplace1155.setFactory(FactoryVerifier)._factory (
    ↪ contracts/MADMarketplace1155.sol#394) is not in mixedCase
Parameter MADMarketplace1155.setFees(uint256,uint256)._feeVal2 (
    ↪ contracts/MADMarketplace1155.sol#406) is not in mixedCase
Parameter MADMarketplace1155.setFees(uint256,uint256)._feeVal3 (
    ↪ contracts/MADMarketplace1155.sol#407) is not in mixedCase
Parameter MADMarketplace1155.updateSettings(uint256,uint256,uint256).
    ↪ _minAuctionIncrement (contracts/MADMarketplace1155.sol#429) is
    ↪ not in mixedCase
Parameter MADMarketplace1155.updateSettings(uint256,uint256,uint256).
    ↪ _minOrderDuration (contracts/MADMarketplace1155.sol#430) is not
    ↪ in mixedCase
Parameter MADMarketplace1155.updateSettings(uint256,uint256,uint256).
    ↪ _minBidValue (contracts/MADMarketplace1155.sol#431) is not in
    ↪ mixedCase
Parameter MADMarketplace1155.setRecipient(address)._recipient (contracts
    ↪ /MADMarketplace1155.sol#466) is not in mixedCase
Parameter MADMarketplace1155.delOrder(bytes32,IERC1155,uint256,uint256,
    ↪ address)._token (contracts/MADMarketplace1155.sol#504) is not in
    ↪ mixedCase
```

```
Parameter MADMarketplace1155.delOrder(bytes32,IERC1155,uint256,uint256,
    ↪ address)._id (contracts/MADMarketplace1155.sol#505) is not in
    ↪ mixedCase
Parameter MADMarketplace1155.delOrder(bytes32,IERC1155,uint256,uint256,
    ↪ address)._amount (contracts/MADMarketplace1155.sol#506) is not in
    ↪  mixedCase
Parameter MADMarketplace1155.delOrder(bytes32,IERC1155,uint256,uint256,
    ↪ address)._seller (contracts/MADMarketplace1155.sol#507) is not in
    ↪  mixedCase
Function MADMarketplace1155.ERC165Check(address) (contracts/
    ↪ MADMarketplace1155.sol#635-643) is not in mixedCase
Parameter MADMarketplace1155.getCurrentPrice(bytes32)._order (contracts/
    ↪ MADMarketplace1155.sol#1034) is not in mixedCase
Parameter MADMarketplace1155.tokenOrderLength(IERC1155,uint256,uint256).
    ↪ _token (contracts/MADMarketplace1155.sol#1108) is not in
    ↪ mixedCase
Parameter MADMarketplace1155.tokenOrderLength(IERC1155,uint256,uint256).
    ↪ _id (contracts/MADMarketplace1155.sol#1109) is not in mixedCase
Parameter MADMarketplace1155.tokenOrderLength(IERC1155,uint256,uint256).
    ↪ _amount (contracts/MADMarketplace1155.sol#1110) is not in
    ↪ mixedCase
Parameter MADMarketplace1155.sellerOrderLength(address)._seller (
    ↪ contracts/MADMarketplace1155.sol#1120) is not in mixedCase
Constant MADMarketplace1155.basisPoints (contracts/MADMarketplace1155.
    ↪ sol#56) is not in UPPER_CASE_WITH_UNDERSCORES
Variable MADMarketplace1155.MADFactory1155 (contracts/MADMarketplace1155
    ↪ .sol#77) is not in mixedCase
Parameter MADMarketplace721.fixedPrice(IERC721,uint256,uint256,uint256).
    ↪ _token (contracts/MADMarketplace721.sol#104) is not in mixedCase
Parameter MADMarketplace721.fixedPrice(IERC721,uint256,uint256,uint256).
    ↪ _id (contracts/MADMarketplace721.sol#105) is not in mixedCase
Parameter MADMarketplace721.fixedPrice(IERC721,uint256,uint256,uint256).
    ↪ _price (contracts/MADMarketplace721.sol#106) is not in mixedCase
```

```
Parameter MADMarketplace721.fixedPrice(IERC721,uint256,uint256,uint256).
    ↪ _endTime (contracts/MADMarketplace721.sol#107) is not in
    ↪ mixedCase
Parameter MADMarketplace721.dutchAuction(IERC721,uint256,uint256,uint256
    ↪ ,uint256)._token (contracts/MADMarketplace721.sol#115) is not in
    ↪ mixedCase
Parameter MADMarketplace721.dutchAuction(IERC721,uint256,uint256,uint256
    ↪ ,uint256)._id (contracts/MADMarketplace721.sol#116) is not in
    ↪ mixedCase
Parameter MADMarketplace721.dutchAuction(IERC721,uint256,uint256,uint256
    ↪ ,uint256)._startPrice (contracts/MADMarketplace721.sol#117) is
    ↪ not in mixedCase
Parameter MADMarketplace721.dutchAuction(IERC721,uint256,uint256,uint256
    ↪ ,uint256)._endPrice (contracts/MADMarketplace721.sol#118) is not
    ↪ in mixedCase
Parameter MADMarketplace721.dutchAuction(IERC721,uint256,uint256,uint256
    ↪ ,uint256)._endTime (contracts/MADMarketplace721.sol#119) is not
    ↪ in mixedCase
Parameter MADMarketplace721.englishAuction(IERC721,uint256,uint256,
    ↪ uint256)._token (contracts/MADMarketplace721.sol#135) is not in
    ↪ mixedCase
Parameter MADMarketplace721.englishAuction(IERC721,uint256,uint256,
    ↪ uint256)._id (contracts/MADMarketplace721.sol#136) is not in
    ↪ mixedCase
Parameter MADMarketplace721.englishAuction(IERC721,uint256,uint256,
    ↪ uint256)._startPrice (contracts/MADMarketplace721.sol#137) is not
    ↪  in mixedCase
Parameter MADMarketplace721.englishAuction(IERC721,uint256,uint256,
    ↪ uint256)._endTime (contracts/MADMarketplace721.sol#138) is not in
    ↪  mixedCase
Parameter MADMarketplace721.bid(bytes32)._order (contracts/
    ↪ MADMarketplace721.sol#148) is not in mixedCase
Parameter MADMarketplace721.buy(bytes32)._order (contracts/
    ↪ MADMarketplace721.sol#204) is not in mixedCase
```

```
Parameter MADMarketplace721.claim(bytes32)._order (contracts/
   ↪ MADMarketplace721.sol#272) is not in mixedCase
Parameter MADMarketplace721.cancelOrder(bytes32)._order (contracts/
   ↪ MADMarketplace721.sol#340) is not in mixedCase
Parameter MADMarketplace721.setFactory(FactoryVerifier)._factory (
   ↪ contracts/MADMarketplace721.sol#370) is not in mixedCase
Parameter MADMarketplace721.setFees(uint256,uint256)._feeVal2 (contracts
   ↪ /MADMarketplace721.sol#382) is not in mixedCase
Parameter MADMarketplace721.setFees(uint256,uint256)._feeVal3 (contracts
   ↪ /MADMarketplace721.sol#383) is not in mixedCase
Parameter MADMarketplace721.updateSettings(uint256,uint256,uint256).
   ↪ _minAuctionIncrement (contracts/MADMarketplace721.sol#404) is not
   ↪  in mixedCase
Parameter MADMarketplace721.updateSettings(uint256,uint256,uint256).
   ↪ _minOrderDuration (contracts/MADMarketplace721.sol#405) is not in
   ↪  mixedCase
Parameter MADMarketplace721.updateSettings(uint256,uint256,uint256).
   ↪ _minBidValue (contracts/MADMarketplace721.sol#406) is not in
   ↪ mixedCase
Parameter MADMarketplace721.setRecipient(address)._recipient (contracts/
   ↪ MADMarketplace721.sol#441) is not in mixedCase
Parameter MADMarketplace721.delOrder(bytes32,IERC721,uint256,address).
   ↪ _token (contracts/MADMarketplace721.sol#479) is not in mixedCase
Parameter MADMarketplace721.delOrder(bytes32,IERC721,uint256,address).
   ↪ _id (contracts/MADMarketplace721.sol#480) is not in mixedCase
Parameter MADMarketplace721.delOrder(bytes32,IERC721,uint256,address).
   ↪ _seller (contracts/MADMarketplace721.sol#481) is not in mixedCase
Function MADMarketplace721.ERC165Check(address) (contracts/
   ↪ MADMarketplace721.sol#586-594) is not in mixedCase
Parameter MADMarketplace721.getCurrentPrice(bytes32)._order (contracts/
   ↪ MADMarketplace721.sol#960) is not in mixedCase
Parameter MADMarketplace721.tokenOrderLength(IERC721,uint256)._token (
   ↪ contracts/MADMarketplace721.sol#1033) is not in mixedCase
```

```
Parameter MADMarketplace721.tokenOrderLength(IERC721,uint256)._id (
    ↪ contracts/MADMarketplace721.sol#1033) is not in mixedCase
Parameter MADMarketplace721.sellerOrderLength(address)._seller (
    ↪ contracts/MADMarketplace721.sol#1046) is not in mixedCase
Constant MADMarketplace721.basisPoints (contracts/MADMarketplace721.sol
    ↪ #56) is not in UPPER_CASE_WITH_UNDERSCORES
Variable MADMarketplace721.MADFactory721 (contracts/MADMarketplace721.
    ↪ sol#77) is not in mixedCase
Parameter MADRouter1155.setURI(address,string)._token (contracts/
    ↪ MADRouter1155.sol#75) is not in mixedCase
Parameter MADRouter1155.setURI(address,string)._uri (contracts/
    ↪ MADRouter1155.sol#75) is not in mixedCase
Parameter MADRouter1155.whitelistSettings(address,uint256,uint256,
    ↪ bytes32)._token (contracts/MADRouter1155.sol#103) is not in
    ↪ mixedCase
Parameter MADRouter1155.whitelistSettings(address,uint256,uint256,
    ↪ bytes32)._price (contracts/MADRouter1155.sol#104) is not in
    ↪ mixedCase
Parameter MADRouter1155.whitelistSettings(address,uint256,uint256,
    ↪ bytes32)._supply (contracts/MADRouter1155.sol#105) is not in
    ↪ mixedCase
Parameter MADRouter1155.whitelistSettings(address,uint256,uint256,
    ↪ bytes32)._root (contracts/MADRouter1155.sol#106) is not in
    ↪ mixedCase
Parameter MADRouter1155.freeSettings(address,uint256,uint256,bytes32).
    ↪ _token (contracts/MADRouter1155.sol#123) is not in mixedCase
Parameter MADRouter1155.freeSettings(address,uint256,uint256,bytes32).
    ↪ _freeAmount (contracts/MADRouter1155.sol#124) is not in mixedCase
Parameter MADRouter1155.freeSettings(address,uint256,uint256,bytes32).
    ↪ _maxFree (contracts/MADRouter1155.sol#125) is not in mixedCase
Parameter MADRouter1155.freeSettings(address,uint256,uint256,bytes32).
    ↪ _claimRoot (contracts/MADRouter1155.sol#126) is not in mixedCase
Parameter MADRouter1155.minimalSafeMint(address,address,uint256)._token
    ↪ (contracts/MADRouter1155.sol#140) is not in mixedCase
```

```
Parameter MADRouter1155.minimalSafeMint(address,address,uint256)._to (
    ↪ contracts/MADRouter1155.sol#140) is not in mixedCase
Parameter MADRouter1155.basicMintTo(address,address,uint256,uint256[]).
    ↪ _token (contracts/MADRouter1155.sol#152) is not in mixedCase
Parameter MADRouter1155.basicMintTo(address,address,uint256,uint256[]).
    ↪ _to (contracts/MADRouter1155.sol#153) is not in mixedCase
Parameter MADRouter1155.basicMintTo(address,address,uint256,uint256[]).
    ↪ _amount (contracts/MADRouter1155.sol#154) is not in mixedCase
Parameter MADRouter1155.basicMintTo(address,address,uint256,uint256[]).
    ↪ _balances (contracts/MADRouter1155.sol#155) is not in mixedCase
Parameter MADRouter1155.basicMintBatchTo(address,address,uint256[],
    ↪ uint256[])._token (contracts/MADRouter1155.sol#163) is not in
    ↪ mixedCase
Parameter MADRouter1155.basicMintBatchTo(address,address,uint256[],
    ↪ uint256[])._to (contracts/MADRouter1155.sol#164) is not in
    ↪ mixedCase
Parameter MADRouter1155.basicMintBatchTo(address,address,uint256[],
    ↪ uint256[])._ids (contracts/MADRouter1155.sol#165) is not in
    ↪ mixedCase
Parameter MADRouter1155.basicMintBatchTo(address,address,uint256[],
    ↪ uint256[])._balances (contracts/MADRouter1155.sol#166) is not in
    ↪ mixedCase
Parameter MADRouter1155.burn(address,uint256[],address[],uint256[]).
    ↪ _token (contracts/MADRouter1155.sol#178) is not in mixedCase
Parameter MADRouter1155.burn(address,uint256[],address[],uint256[])._ids
    ↪  (contracts/MADRouter1155.sol#178) is not in mixedCase
Parameter MADRouter1155.burn(address,uint256[],address[],uint256[]).
    ↪ _amount (contracts/MADRouter1155.sol#178) is not in mixedCase
Parameter MADRouter1155.batchBurn(address,address,uint256[],uint256[]).
    ↪ _token (contracts/MADRouter1155.sol#203) is not in mixedCase
Parameter MADRouter1155.batchBurn(address,address,uint256[],uint256[]).
    ↪ _from (contracts/MADRouter1155.sol#204) is not in mixedCase
Parameter MADRouter1155.batchBurn(address,address,uint256[],uint256[]).
    ↪ _ids (contracts/MADRouter1155.sol#205) is not in mixedCase
```

```
Parameter MADRouter1155.batchBurn(address,address,uint256[],uint256[]).
    ↪ _balances (contracts/MADRouter1155.sol#206) is not in mixedCase
Parameter MADRouter1155.setMintState(address,bool,uint8)._token (
    ↪ contracts/MADRouter1155.sol#229) is not in mixedCase
Parameter MADRouter1155.setMintState(address,bool,uint8)._state (
    ↪ contracts/MADRouter1155.sol#230) is not in mixedCase
Parameter MADRouter1155.setMintState(address,bool,uint8)._stateType (
    ↪ contracts/MADRouter1155.sol#231) is not in mixedCase
Parameter MADRouter1155.creatorMint(address,uint256,uint256[],uint256).
    ↪ _token (contracts/MADRouter1155.sol#256) is not in mixedCase
Parameter MADRouter1155.creatorMint(address,uint256,uint256[],uint256).
    ↪ _amount (contracts/MADRouter1155.sol#256) is not in mixedCase
Parameter MADRouter1155.creatorMint(address,uint256,uint256[],uint256).
    ↪ _balances (contracts/MADRouter1155.sol#256) is not in mixedCase
Parameter MADRouter1155.creatorBatchMint(address,uint256[],uint256[],
    ↪ uint256)._token (contracts/MADRouter1155.sol#271) is not in
    ↪ mixedCase
Parameter MADRouter1155.creatorBatchMint(address,uint256[],uint256[],
    ↪ uint256)._ids (contracts/MADRouter1155.sol#272) is not in
    ↪ mixedCase
Parameter MADRouter1155.creatorBatchMint(address,uint256[],uint256[],
    ↪ uint256)._balances (contracts/MADRouter1155.sol#273) is not in
    ↪ mixedCase
Parameter MADRouter1155.gift(address,address[],uint256[],uint256)._token
    ↪  (contracts/MADRouter1155.sol#285) is not in mixedCase
Parameter MADRouter1155.gift(address,address[],uint256[],uint256).
    ↪ _addresses (contracts/MADRouter1155.sol#286) is not in mixedCase
Parameter MADRouter1155.gift(address,address[],uint256[],uint256).
    ↪ _balances (contracts/MADRouter1155.sol#287) is not in mixedCase
Parameter MADRouter1155.withdraw(address,ERC20)._token (contracts/
    ↪ MADRouter1155.sol#305) is not in mixedCase
Parameter MADRouter1155.withdraw(address,ERC20)._erc20 (contracts/
    ↪ MADRouter1155.sol#305) is not in mixedCase
```

```
Parameter MADRouter1155.setFees(uint256,uint256)._feeMint (contracts/
    ↪ MADRouter1155.sol#404) is not in mixedCase
Parameter MADRouter1155.setFees(uint256,uint256)._feeBurn (contracts/
    ↪ MADRouter1155.sol#405) is not in mixedCase
Parameter MADRouter1155.setSigner(address,address)._token (contracts/
    ↪ MADRouter1155.sol#496) is not in mixedCase
Parameter MADRouter1155.setSigner(address,address)._signer (contracts/
    ↪ MADRouter1155.sol#496) is not in mixedCase
Variable MADRouter1155.MADFactory1155 (contracts/MADRouter1155.sol#51)
    ↪ is not in mixedCase
Parameter MADRouter721.setBase(address,string)._token (contracts/
    ↪ MADRouter721.sol#74) is not in mixedCase
Parameter MADRouter721.setBase(address,string)._baseURI (contracts/
    ↪ MADRouter721.sol#74) is not in mixedCase
Parameter MADRouter721.whitelistSettings(address,uint256,uint256,bytes32
    ↪ )._token (contracts/MADRouter721.sol#102) is not in mixedCase
Parameter MADRouter721.whitelistSettings(address,uint256,uint256,bytes32
    ↪ )._price (contracts/MADRouter721.sol#103) is not in mixedCase
Parameter MADRouter721.whitelistSettings(address,uint256,uint256,bytes32
    ↪ )._supply (contracts/MADRouter721.sol#104) is not in mixedCase
Parameter MADRouter721.whitelistSettings(address,uint256,uint256,bytes32
    ↪ )._root (contracts/MADRouter721.sol#105) is not in mixedCase
Parameter MADRouter721.freeSettings(address,uint256,uint256,bytes32).
    ↪ _token (contracts/MADRouter721.sol#122) is not in mixedCase
Parameter MADRouter721.freeSettings(address,uint256,uint256,bytes32).
    ↪ _freeAmount (contracts/MADRouter721.sol#123) is not in mixedCase
Parameter MADRouter721.freeSettings(address,uint256,uint256,bytes32).
    ↪ _maxFree (contracts/MADRouter721.sol#124) is not in mixedCase
Parameter MADRouter721.freeSettings(address,uint256,uint256,bytes32).
    ↪ _claimRoot (contracts/MADRouter721.sol#125) is not in mixedCase
Parameter MADRouter721.minimalSafeMint(address,address)._token (
    ↪ contracts/MADRouter721.sol#139) is not in mixedCase
Parameter MADRouter721.minimalSafeMint(address,address)._to (contracts/
    ↪ MADRouter721.sol#139) is not in mixedCase
```

```
Parameter MADRouter721.basicMintTo(address,address,uint256)._token (
    ↪ contracts/MADRouter721.sol#151) is not in mixedCase
Parameter MADRouter721.basicMintTo(address,address,uint256)._to (
    ↪ contracts/MADRouter721.sol#152) is not in mixedCase
Parameter MADRouter721.basicMintTo(address,address,uint256)._amount (
    ↪ contracts/MADRouter721.sol#153) is not in mixedCase
Parameter MADRouter721.burn(address,uint256[])._token (contracts/
    ↪ MADRouter721.sol#165) is not in mixedCase
Parameter MADRouter721.burn(address,uint256[])._ids (contracts/
    ↪ MADRouter721.sol#165) is not in mixedCase
Parameter MADRouter721.setMintState(address,bool,uint8)._token (
    ↪ contracts/MADRouter721.sol#194) is not in mixedCase
Parameter MADRouter721.setMintState(address,bool,uint8)._state (
    ↪ contracts/MADRouter721.sol#195) is not in mixedCase
Parameter MADRouter721.setMintState(address,bool,uint8)._stateType (
    ↪ contracts/MADRouter721.sol#196) is not in mixedCase
Parameter MADRouter721.creatorMint(address,uint256)._token (contracts/
    ↪ MADRouter721.sol#221) is not in mixedCase
Parameter MADRouter721.creatorMint(address,uint256)._amount (contracts/
    ↪ MADRouter721.sol#221) is not in mixedCase
Parameter MADRouter721.gift(address,address[])._token (contracts/
    ↪ MADRouter721.sol#236) is not in mixedCase
Parameter MADRouter721.gift(address,address[])._addresses (contracts/
    ↪ MADRouter721.sol#237) is not in mixedCase
Parameter MADRouter721.withdraw(address,ERC20)._token (contracts/
    ↪ MADRouter721.sol#254) is not in mixedCase
Parameter MADRouter721.withdraw(address,ERC20)._erc20 (contracts/
    ↪ MADRouter721.sol#254) is not in mixedCase
Parameter MADRouter721.setFees(uint256,uint256)._feeMint (contracts/
    ↪ MADRouter721.sol#353) is not in mixedCase
Parameter MADRouter721.setFees(uint256,uint256)._feeBurn (contracts/
    ↪ MADRouter721.sol#354) is not in mixedCase
Parameter MADRouter721.setSigner(address,address)._token (contracts/
    ↪ MADRouter721.sol#442) is not in mixedCase
```

```
Parameter MADRouter721.setSigner(address,address)._signer (contracts/
    ↪ MADRouter721.sol#442) is not in mixedCase
Variable MADRouter721.MADFactory721 (contracts/MADRouter721.sol#51) is
    ↪ not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation
    ↪ #conformance-to-solidity-naming-conventions

Variable MADFactory1155.splitterCheck(string,address,address,uint256,
    ↪ uint256)._payees_scope_0 (contracts/MADFactory1155.sol#178-181)
    ↪ is too similar to MADFactory1155.splitterCheck(string,address,
    ↪ address,uint256,uint256)._payees_scope_3 (contracts/
    ↪ MADFactory1155.sol#217-220)
Variable MADFactory1155.splitterCheck(string,address,address,uint256,
    ↪ uint256)._payees_scope_0 (contracts/MADFactory1155.sol#178-181)
    ↪ is too similar to MADFactory1155.splitterCheck(string,address,
    ↪ address,uint256,uint256)._payees_scope_6 (contracts/
    ↪ MADFactory1155.sol#258-261)
Variable MADFactory1155.splitterCheck(string,address,address,uint256,
    ↪ uint256)._payees_scope_3 (contracts/MADFactory1155.sol#217-220)
    ↪ is too similar to MADFactory1155.splitterCheck(string,address,
    ↪ address,uint256,uint256)._payees_scope_6 (contracts/
    ↪ MADFactory1155.sol#258-261)
Variable MADFactory1155.splitterCheck(string,address,address,uint256,
    ↪ uint256)._shares_scope_1 (contracts/MADFactory1155.sol#183) is
    ↪ too similar to MADFactory1155.splitterCheck(string,address,
    ↪ address,uint256,uint256)._shares_scope_4 (contracts/
    ↪ MADFactory1155.sol#222)
Variable MADFactory1155.splitterCheck(string,address,address,uint256,
    ↪ uint256)._shares_scope_1 (contracts/MADFactory1155.sol#183) is
    ↪ too similar to MADFactory1155.splitterCheck(string,address,
    ↪ address,uint256,uint256)._shares_scope_7 (contracts/
    ↪ MADFactory1155.sol#269)
Variable MADFactory1155.splitterCheck(string,address,address,uint256,
    ↪ uint256)._shares_scope_4 (contracts/MADFactory1155.sol#222) is
```

```
    ↪ too similar to MADFactory1155.splitterCheck(string,address,
    ↪ address,uint256,uint256)._shares_scope_7 (contracts/
    ↪ MADFactory1155.sol#269)
Variable MADFactory1155.splitterCheck(string,address,address,uint256,
    ↪ uint256)._splitter_scope_2 (contracts/MADFactory1155.sol#185-189)
    ↪  is too similar to MADFactory1155.splitterCheck(string,address,
    ↪ address,uint256,uint256)._splitter_scope_5 (contracts/
    ↪ MADFactory1155.sol#224-228)
Variable MADFactory1155.splitterCheck(string,address,address,uint256,
    ↪ uint256)._splitter_scope_2 (contracts/MADFactory1155.sol#185-189)
    ↪  is too similar to MADFactory1155.splitterCheck(string,address,
    ↪ address,uint256,uint256)._splitter_scope_8 (contracts/
    ↪ MADFactory1155.sol#271-275)
Variable MADFactory1155.splitterCheck(string,address,address,uint256,
    ↪ uint256)._splitter_scope_5 (contracts/MADFactory1155.sol#224-228)
    ↪  is too similar to MADFactory1155.splitterCheck(string,address,
    ↪ address,uint256,uint256)._splitter_scope_8 (contracts/
    ↪ MADFactory1155.sol#271-275)
Variable MADFactory1155.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).colId_scope_2 (contracts/
    ↪ MADFactory1155.sol#387) is too similar to MADFactory1155.
    ↪ createCollection(uint8,string,string,string,uint256,uint256,
    ↪ string,address,uint256).colId_scope_5 (contracts/MADFactory1155.
    ↪ sol#420)
Variable MADFactory1155.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).colId_scope_2 (contracts/
    ↪ MADFactory1155.sol#387) is too similar to MADFactory1155.
    ↪ createCollection(uint8,string,string,string,uint256,uint256,
    ↪ string,address,uint256).colId_scope_8 (contracts/MADFactory1155.
    ↪ sol#452)
Variable MADFactory1155.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).colId_scope_5 (contracts/
    ↪ MADFactory1155.sol#420) is too similar to MADFactory1155.
    ↪ createCollection(uint8,string,string,string,uint256,uint256,
```

```
    ↪ string,address,uint256).colId_scope_8 (contracts/MADFactory1155.
    ↪ sol#452)
Variable MADFactory1155.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).deployed_scope_1 (
    ↪ contracts/MADFactory1155.sol#376) is too similar to
    ↪ MADFactory1155.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).deployed_scope_4 (
    ↪ contracts/MADFactory1155.sol#409)
Variable MADFactory1155.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).deployed_scope_1 (
    ↪ contracts/MADFactory1155.sol#376) is too similar to
    ↪ MADFactory1155.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).deployed_scope_7 (
    ↪ contracts/MADFactory1155.sol#442)
Variable MADFactory1155.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).deployed_scope_4 (
    ↪ contracts/MADFactory1155.sol#409) is too similar to
    ↪ MADFactory1155.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).deployed_scope_7 (
    ↪ contracts/MADFactory1155.sol#442)
Variable MADFactory1155.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).tokenSalt_scope_0 (
    ↪ contracts/MADFactory1155.sol#376) is too similar to
    ↪ MADFactory1155.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).tokenSalt_scope_3 (
    ↪ contracts/MADFactory1155.sol#409)
Variable MADFactory1155.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).tokenSalt_scope_0 (
    ↪ contracts/MADFactory1155.sol#376) is too similar to
    ↪ MADFactory1155.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).tokenSalt_scope_6 (
    ↪ contracts/MADFactory1155.sol#442)
Variable MADFactory1155.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).tokenSalt_scope_3 (
```

```
    ↪ contracts/MADFactory1155.sol#409) is too similar to
    ↪ MADFactory1155.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).tokenSalt_scope_6 (
    ↪ contracts/MADFactory1155.sol#442)
Variable MADFactory721.splitterCheck(string,address,address,uint256,
    ↪ uint256)._payees_scope_0 (contracts/MADFactory721.sol#177-180) is
    ↪  too similar to MADFactory721.splitterCheck(string,address,
    ↪ address,uint256,uint256)._payees_scope_3 (contracts/MADFactory721
    ↪ .sol#216-219)
Variable MADFactory721.splitterCheck(string,address,address,uint256,
    ↪ uint256)._payees_scope_0 (contracts/MADFactory721.sol#177-180) is
    ↪  too similar to MADFactory721.splitterCheck(string,address,
    ↪ address,uint256,uint256)._payees_scope_6 (contracts/MADFactory721
    ↪ .sol#257-260)
Variable MADFactory721.splitterCheck(string,address,address,uint256,
    ↪ uint256)._payees_scope_3 (contracts/MADFactory721.sol#216-219) is
    ↪  too similar to MADFactory721.splitterCheck(string,address,
    ↪ address,uint256,uint256)._payees_scope_6 (contracts/MADFactory721
    ↪ .sol#257-260)
Variable MADFactory721.splitterCheck(string,address,address,uint256,
    ↪ uint256)._shares_scope_1 (contracts/MADFactory721.sol#182) is too
    ↪  similar to MADFactory721.splitterCheck(string,address,address,
    ↪ uint256,uint256)._shares_scope_4 (contracts/MADFactory721.sol
    ↪ #221)
Variable MADFactory721.splitterCheck(string,address,address,uint256,
    ↪ uint256)._shares_scope_1 (contracts/MADFactory721.sol#182) is too
    ↪  similar to MADFactory721.splitterCheck(string,address,address,
    ↪ uint256,uint256)._shares_scope_7 (contracts/MADFactory721.sol
    ↪ #268)
Variable MADFactory721.splitterCheck(string,address,address,uint256,
    ↪ uint256)._shares_scope_4 (contracts/MADFactory721.sol#221) is too
    ↪  similar to MADFactory721.splitterCheck(string,address,address,
    ↪ uint256,uint256)._shares_scope_7 (contracts/MADFactory721.sol
    ↪ #268)
```

```
Variable MADFactory721.splitterCheck(string,address,address,uint256,
    ↪ uint256)._splitter_scope_2 (contracts/MADFactory721.sol#184-188)
    ↪ is too similar to MADFactory721.splitterCheck(string,address,
    ↪ address,uint256,uint256)._splitter_scope_5 (contracts/
    ↪ MADFactory721.sol#223-227)
Variable MADFactory721.splitterCheck(string,address,address,uint256,
    ↪ uint256)._splitter_scope_2 (contracts/MADFactory721.sol#184-188)
    ↪ is too similar to MADFactory721.splitterCheck(string,address,
    ↪ address,uint256,uint256)._splitter_scope_8 (contracts/
    ↪ MADFactory721.sol#270-274)
Variable MADFactory721.splitterCheck(string,address,address,uint256,
    ↪ uint256)._splitter_scope_5 (contracts/MADFactory721.sol#223-227)
    ↪ is too similar to MADFactory721.splitterCheck(string,address,
    ↪ address,uint256,uint256)._splitter_scope_8 (contracts/
    ↪ MADFactory721.sol#270-274)
Variable MADFactory721.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).colId_scope_2 (contracts/
    ↪ MADFactory721.sol#390) is too similar to MADFactory721.
    ↪ createCollection(uint8,string,string,string,uint256,uint256,
    ↪ string,address,uint256).colId_scope_5 (contracts/MADFactory721.
    ↪ sol#425)
Variable MADFactory721.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).colId_scope_2 (contracts/
    ↪ MADFactory721.sol#390) is too similar to MADFactory721.
    ↪ createCollection(uint8,string,string,string,uint256,uint256,
    ↪ string,address,uint256).colId_scope_8 (contracts/MADFactory721.
    ↪ sol#459)
Variable MADFactory721.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).colId_scope_5 (contracts/
    ↪ MADFactory721.sol#425) is too similar to MADFactory721.
    ↪ createCollection(uint8,string,string,string,uint256,uint256,
    ↪ string,address,uint256).colId_scope_8 (contracts/MADFactory721.
    ↪ sol#459)
```

```
Variable MADFactory721.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).deployed_scope_1 (
    ↪ contracts/MADFactory721.sol#377) is too similar to MADFactory721.
    ↪ createCollection(uint8,string,string,string,uint256,uint256,
    ↪ string,address,uint256).deployed_scope_4 (contracts/MADFactory721
    ↪ .sol#412)
Variable MADFactory721.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).deployed_scope_1 (
    ↪ contracts/MADFactory721.sol#377) is too similar to MADFactory721.
    ↪ createCollection(uint8,string,string,string,uint256,uint256,
    ↪ string,address,uint256).deployed_scope_7 (contracts/MADFactory721
    ↪ .sol#447)
Variable MADFactory721.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).deployed_scope_4 (
    ↪ contracts/MADFactory721.sol#412) is too similar to MADFactory721.
    ↪ createCollection(uint8,string,string,string,uint256,uint256,
    ↪ string,address,uint256).deployed_scope_7 (contracts/MADFactory721
    ↪ .sol#447)
Variable MADFactory721.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).tokenSalt_scope_0 (
    ↪ contracts/MADFactory721.sol#377) is too similar to MADFactory721.
    ↪ createCollection(uint8,string,string,string,uint256,uint256,
    ↪ string,address,uint256).tokenSalt_scope_3 (contracts/
    ↪ MADFactory721.sol#412)
Variable MADFactory721.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).tokenSalt_scope_0 (
    ↪ contracts/MADFactory721.sol#377) is too similar to MADFactory721.
    ↪ createCollection(uint8,string,string,string,uint256,uint256,
    ↪ string,address,uint256).tokenSalt_scope_6 (contracts/
    ↪ MADFactory721.sol#447)
Variable MADFactory721.createCollection(uint8,string,string,string,
    ↪ uint256,uint256,string,address,uint256).tokenSalt_scope_3 (
    ↪ contracts/MADFactory721.sol#412) is too similar to MADFactory721.
    ↪ createCollection(uint8,string,string,string,uint256,uint256,
```

```
    ↪ string,address,uint256).tokenSalt_scope_6 (contracts/
    ↪ MADFactory721.sol#447)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation
    ↪ #variable-names-are-too-similar

MADFactory1155._limiter(uint8,address) (contracts/MADFactory1155.sol
    ↪ #779-793) uses literals with too many digits:
        - mstore(uint256,uint256)(0x00,0
            ↪ x4ca888670000000000000000000000000000000000000000000000000000000000000000
            ↪ ) (contracts/MADFactory1155.sol#789)
MADFactory721._limiter(uint8,address) (contracts/MADFactory721.sol
    ↪ #787-801) uses literals with too many digits:
        - mstore(uint256,uint256)(0x00,0
            ↪ x4ca888670000000000000000000000000000000000000000000000000000000000000000
            ↪ ) (contracts/MADFactory721.sol#797)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation
    ↪ #too-many-digits

MADFactory1155.market (contracts/MADFactory1155.sol#88) should be
    ↪ constant
MADFactory1155.router (contracts/MADFactory1155.sol#85) should be
    ↪ constant
MADFactory1155.signer (contracts/MADFactory1155.sol#94) should be
    ↪ constant
MADFactory721.market (contracts/MADFactory721.sol#87) should be constant
MADFactory721.router (contracts/MADFactory721.sol#84) should be constant
MADFactory721.signer (contracts/MADFactory721.sol#93) should be constant
MADMarketplace1155.feeVal2 (contracts/MADMarketplace1155.sol#51) should
    ↪ be constant
MADMarketplace1155.feeVal3 (contracts/MADMarketplace1155.sol#52) should
    ↪ be constant
MADMarketplace1155.minAuctionIncrement (contracts/MADMarketplace1155.sol
    ↪ #73) should be constant
```

```
MADMarketplace1155.minBidValue (contracts/MADMarketplace1155.sol#74)
    ↪ should be constant
MADMarketplace1155.minOrderDuration (contracts/MADMarketplace1155.sol
    ↪ #72) should be constant
MADMarketplace1155.recipient (contracts/MADMarketplace1155.sol#76)
    ↪ should be constant
MADMarketplace721.feeVal2 (contracts/MADMarketplace721.sol#51) should be
    ↪  constant
MADMarketplace721.feeVal3 (contracts/MADMarketplace721.sol#52) should be
    ↪  constant
MADMarketplace721.minAuctionIncrement (contracts/MADMarketplace721.sol
    ↪ #73) should be constant
MADMarketplace721.minBidValue (contracts/MADMarketplace721.sol#74)
    ↪ should be constant
MADMarketplace721.minOrderDuration (contracts/MADMarketplace721.sol#72)
    ↪ should be constant
MADMarketplace721.recipient (contracts/MADMarketplace721.sol#76) should
    ↪ be constant
MADRouter1155.feeBurn (contracts/MADRouter1155.sol#57) should be
    ↪ constant
MADRouter1155.feeMint (contracts/MADRouter1155.sol#56) should be
    ↪ constant
MADRouter721.feeBurn (contracts/MADRouter721.sol#57) should be constant
MADRouter721.feeMint (contracts/MADRouter721.sol#56) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation
    ↪ #state-variables-that-could-be-declared-constant

name() should be declared external:
        - MAD.name() (contracts/MAD.sol#34-38)
        - MADFactory1155.name() (contracts/MADFactory1155.sol#51-62)
        - MADFactory721.name() (contracts/MADFactory721.sol#50-61)
        - MADMarketplace1155.name() (contracts/MADMarketplace1155.sol
            ↪ #32-43)
```

```
        - MADMarketplace721.name() (contracts/MADMarketplace721.sol
            ↪ #32-43)
        - MADRouter1155.name() (contracts/MADRouter1155.sol#34-45)
        - MADRouter721.name() (contracts/MADRouter721.sol#34-45)
creatorCheck(bytes32) should be declared external:
        - MADFactory1155.creatorCheck(bytes32) (contracts/MADFactory1155.
            ↪ sol#729-752)
        - MADFactory721.creatorCheck(bytes32) (contracts/MADFactory721.
            ↪ sol#737-760)
getDeployedAddr(string) should be declared external:
        - MADFactory1155.getDeployedAddr(string) (contracts/
            ↪ MADFactory1155.sol#826-835)
getDeployedAddr(string) should be declared external:
        - MADFactory721.getDeployedAddr(string) (contracts/MADFactory721.
            ↪ sol#834-843)
fixedPrice(IERC1155,uint256,uint256,uint256,uint256) should be declared
    ↪ external:
        - MADMarketplace1155.fixedPrice(IERC1155,uint256,uint256,uint256,
            ↪ uint256) (contracts/MADMarketplace1155.sol#103-119)
dutchAuction(IERC1155,uint256,uint256,uint256,uint256,uint256) should be
    ↪  declared external:
        - MADMarketplace1155.dutchAuction(IERC1155,uint256,uint256,
            ↪ uint256,uint256,uint256) (contracts/MADMarketplace1155.sol
            ↪ #123-141)
englishAuction(IERC1155,uint256,uint256,uint256,uint256) should be
    ↪ declared external:
        - MADMarketplace1155.englishAuction(IERC1155,uint256,uint256,
            ↪ uint256,uint256) (contracts/MADMarketplace1155.sol
            ↪ #145-161)
fixedPrice(IERC721,uint256,uint256,uint256) should be declared external:
        - MADMarketplace721.fixedPrice(IERC721,uint256,uint256,uint256) (
            ↪ contracts/MADMarketplace721.sol#103-110)
dutchAuction(IERC721,uint256,uint256,uint256,uint256) should be declared
    ↪  external:
```

```
        - MADMarketplace721.dutchAuction(IERC721,uint256,uint256,uint256,
            ↪ uint256) (contracts/MADMarketplace721.sol#114-130)
englishAuction(IERC721,uint256,uint256,uint256) should be declared
    ↪ external:
        - MADMarketplace721.englishAuction(IERC721,uint256,uint256,
            ↪ uint256) (contracts/MADMarketplace721.sol#134-141)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation
    ↪ #public-function-that-could-be-declared-external
. analyzed (72 contracts with 78 detectors), 472 result(s) found
```

Conclusion:

Most of the vulnerabilities found by the analysis have already been addressed by the smart contract code review.

7 Conclusion

In this audit, we examined the design and implementation of MADNFT contract and discovered several issues of varying severity. Jacob Clay team addressed all the issues raised in the initial report and implemented the necessary fixes.

The present code base is well-structured and ready for the mainnet.

BLOCKHAT
SECURITY

For a Contract Audit, contact us at contact@blockhat.io