```
//We need 2 contracts and 1 web-app.
//Bytocode data volume limit is 24KB from EIP-170
//We want to use 48KB in ETH net work.so I code 2 contracts that "OTPGEN" and "OTPAUTH"


!  OTPGEN
//based https://github.com/0xcert/ethereum-erc721
//
//Author NZRI_Kaz-Naz_https://github.com/Kaz-Naz,Toshiya Y_https://github.com/Toshiya-Y-K-N
//by;0x86904339D23BF346C1FFF31Cc3Bb7262fa59d837

//ropsten ;
//rinkeby ;

//nftName = "OneTimePassward-Generator";
//nftSymbol = "OTPGEN";

//We need 2 contracts and 1 web-app.
//Bytocode data volume limit is 24KB from EIP-170
//We want to use 48KB in ETH net work.so I code 2 contracts that "OTPGEN" and "OTPAUTH"


pragma solidity 0.6.2;

import "./nf-token.sol";
import "./nf-token-metadata.sol";
import "./nf-token-enumerable.sol";
import "../ownership/ownable.sol";

/**
 *
 * @dev This is an example contract implementation of NFToken with metadata extension.
 */
contract MyERC721LTOGEN is
  NFTokenMetadata,
  Ownable
{

  /**
   * @dev Contract constructor. Sets metadata extension `name` and `symbol`.
   */
  constructor()
    public
  {
    nftName = "OneTimePassward-Generator";
    nftSymbol = "OTPGEN";
  }


    /**
    * Creator Data Section==================
    */
```

```
    //Creator data    作者の名前とウェブサイトやメールアドレス、ＳＮＳアカウント、電話番号などを記載する。
    //(有事の際はこのサイトでコントラクトの運用に関してアナウンスする)
    string private _creatorData = "Toshiya_github.com/Toshiya-Y-K-N";
    function getCreatorData() public view onlyOwner returns (string memory) {
        return _creatorData;
    }
    //Site    招待するウェブサイトアドレス
    string public _creatorSiteAddress1 = "@";//example
    //address setter getter
    function setSite(string memory _newAddress) public onlyOwner {
        _creatorSiteAddress1 = _newAddress ;
    }



//=====================================================================
    /**
    * oneTimePassCode Section==================
    */

    //secret
    string private _secret = "ty2ty";//examle. gene と auth で統一すること

    //CreatorNum blocknum
    uint256 private _creatorOneTimeNum = 20201717;//examle. gene と auth で統一すること

    //setCreatorOneTimeNum  public onlyOwner
    //手動でワンタイムパスワード（ＯＴＰ）を変える数値を入力できる関数。運営者のオーナーのみが関数を実行出来る。
    /**
    * @dev setCreatorOneTimeNum(uint256 _newNum) public onlyOwner
    */
    function setCreatorOtpNum(uint256 _newNum) public onlyOwner {
        _creatorOneTimeNum  = _newNum ;
    }

    //90 日、30 秒ブロックで 518400 後に無効になる場合 518400, 15sec で 259200 この関数は顧客の NFT が時間的に見て有効か示す。
    //web サイト側でアクセスしてきたユーザーのトークンが無効かどうか、web サイト側で決めるための関数。
    uint256 _blockTimeLimitNum = 259200;// 518400
    function setBlockTimeLimitNum(uint256 _newNum)public onlyOwner {
        _blockTimeLimitNum  = _newNum ;
    }

//////////////////////////////////////////////////////////////////

    //トークンが有効期限内であるかあらわす。 全ての人が判別できる   web アプリで有効判別するのに使う。
    function isThisNftValidDate(uint256 _tokenId) public view returns(bool){
        if (  (idToMintTimeBlockNum[_tokenId] + _blockTimeLimitNum ) > block.number  ) {
            return true ;
```

```
        } else {
            return false;
        }
    }


//7 桁のワンタイムパスワードに関してはデプロイ制限により追加困難。
//web アプリ上で搭載必要。



//オーナー変更型 OTP================================================

    //getYourOTP() それぞれの人が持つパスワードを表示させる。
    function getYourOtp(uint256 _tokenId) public view returns (bytes32) {
        require(msg.sender != address(0), ZERO_ADDRESS);//セロアドレスでないか確認
        require(idToOwner[_tokenId] == msg.sender  );//tokenId のオーナーが関数実行者か確認
        require(isThisNftValidDate(_tokenId) == true);//有効期限内のトークンか確認
        require( _getOwnerNFTCount(msg.sender) > 0 );//オーナーのトークンが一つ以上か確認

        //条件通りならばクリエイターのセットしたパスワードをリターンする
        string memory str =  _secret ;
        uint256 inte = _creatorOneTimeNum ;
        address adr = msg.sender;

        return sha256(abi.encodePacked(inte, str,adr,_tokenId));
    }
//==========================================
//ワンタイムパスワード取得時にイーサを消費させることで台帳上に履歴を残させる。
//ワンタイムパスワードの呼び出し履歴があるかどうか調べられるようにする。ハッキング対策である。
    //getYourOtpUseGas() それぞれの人が持つパスワードを表示させる。UseGas
    function getYourOtpUseGas(uint256 _tokenId) public returns (bytes32) {
        require(msg.sender != address(0), ZERO_ADDRESS);//セロアドレスでないか確認
        require(idToOwner[_tokenId] == msg.sender  );//tokenId のオーナーが関数実行者か確認
        require(isThisNftValidDate(_tokenId) == true);//有効期限内のトークンか確認
        require( _getOwnerNFTCount(msg.sender) > 0 );//オーナーのトークンが一つ以上か確認

        idToOtpCallNum[_tokenId] = idToOtpCallNum[_tokenId] + 1 ;//インクリメント　GAS 消費

        //条件通りならばクリエイターのセットしたパスワードをリターンする
        string memory str =  _secret ;
        uint256 inte = _creatorOneTimeNum ;
        address adr = msg.sender;

        return sha256(abi.encodePacked(inte, str,adr,_tokenId));
    }

//==========================================
        //時刻同期型 OTP TOTP
        function getYourTOTP(uint256 _tokenId) public view returns (bytes32) {

            require(msg.sender != address(0), ZERO_ADDRESS);//セロアドレスでないか確認
```

```
        require(idToOwner[_tokenId] == msg.sender  );//tokenId のオーナーが関数実行者か確認
        require(isThisNftValidDate(_tokenId) == true);//有効期限内のトークンか確認
        require( _getOwnerNFTCount(msg.sender) > 0 );//オーナーのトークンが一つ以上か確認

        string memory str =  _secret ;
        uint256 inte = block.number ;
        address adr = msg.sender;

        return sha256(abi.encodePacked(inte, str,adr,_tokenId));
      }
//==============================

  /**
   * @dev Removes a NFT from owner.
   * @param _tokenId Which NFT we want to remove.(burn)
   */
  function burn(
    uint256 _tokenId,
    uint256 _ownerTokenId,
    bytes32 _ownerOTP
  )
    external
    onlyOwner
  {
    require (  getYourOtp(_ownerTokenId)  ==  _ownerOTP  ) ;//念のためトークン持つ人用 OTP を要求。
    super._burn(_tokenId);
  }
  /**
   * @dev Mints a new NFT.
   * @param _to The address that will own the minted NFT.
   * @param _tokenId of the NFT to be minted by the msg.sender.
   * @param _uri String representing RFC 3986 URI.
   */
  function mint(
    address _to,
    uint256 _tokenId,
    string calldata _uri
  )
    external
    onlyOwner
  {
    super._mint(_to, _tokenId);
    super._setTokenUri(_tokenId, _uri);
    idToMintTimeBlockNum[_tokenId] = block.number;//mint 時の時刻、blocktime 追加
  }


  function writeUri2(
    uint256 _tokenId,
    string calldata _uri2
  )
    external
```

```
  onlyOwner
  validNFToken(_tokenId)
 {
  idToUri2[_tokenId] = _uri2;
 }

   /**
   * 転送関数有効化フラグ。一時的にトランスファー可能にする。
   * This section enables transfer and approve function
   * In japan , this section disabled. (if we want to use erc721LT as ticket.)
   */
  function setTransferPermission(
  bool _newState
  )
  external
  onlyOwner
  {
     _setTransferPermission(_newState);
  }

}
//////////////////

ABI (OTP Generator)
[
        {
                "inputs": [],
                "stateMutability": "nonpayable",
                "type": "constructor"
        },
        {
                "anonymous": false,
                "inputs": [
                        {
                                "indexed": true,
                                "internalType": "address",
                                "name": "_owner",
                                "type": "address"
                        },
                        {
                                "indexed": true,
                                "internalType": "address",
                                "name": "_approved",
                                "type": "address"
                        },
                        {
                                "indexed": true,
                                "internalType": "uint256",
                                "name": "_tokenId",
                                "type": "uint256"
                        }
                ],
```

```
        "name": "Approval",
        "type": "event"
    },
    {
        "anonymous": false,
        "inputs": [
            {
                "indexed": true,
                "internalType": "address",
                "name": "_owner",
                "type": "address"
            },
            {
                "indexed": true,
                "internalType": "address",
                "name": "_operator",
                "type": "address"
            },
            {
                "indexed": false,
                "internalType": "bool",
                "name": "_approved",
                "type": "bool"
            }
        ],
        "name": "ApprovalForAll",
        "type": "event"
    },
    {
        "anonymous": false,
        "inputs": [
            {
                "indexed": true,
                "internalType": "address",
                "name": "_from",
                "type": "address"
            },
            {
                "indexed": true,
                "internalType": "address",
                "name": "_to",
                "type": "address"
            },
            {
                "indexed": true,
                "internalType": "uint256",
                "name": "_tokenId",
                "type": "uint256"
            }
        ],
        "name": "LimitedTransfer",
        "type": "event"
```

```
},
{
	"anonymous": false,
	"inputs": [
		{
			"indexed": true,
			"internalType": "address",
			"name": "previousOwner",
			"type": "address"
		},
		{
			"indexed": true,
			"internalType": "address",
			"name": "newOwner",
			"type": "address"
		}
	],
	"name": "OwnershipTransferred",
	"type": "event"
},
{
	"anonymous": false,
	"inputs": [
		{
			"indexed": true,
			"internalType": "address",
			"name": "_from",
			"type": "address"
		},
		{
			"indexed": true,
			"internalType": "address",
			"name": "_to",
			"type": "address"
		},
		{
			"indexed": true,
			"internalType": "uint256",
			"name": "_tokenId",
			"type": "uint256"
		}
	],
	"name": "Transfer",
	"type": "event"
},
{
	"inputs": [],
	"name": "CANNOT_TRANSFER_TO_ZERO_ADDRESS",
	"outputs": [
		{
			"internalType": "string",
			"name": "",
```

```
                        "type": "string"
                }
        ],
        "stateMutability": "view",
        "type": "function"
},
{
        "inputs": [
                {
                        "internalType": "address",
                        "name": "_approved",
                        "type": "address"
                },
                {
                        "internalType": "uint256",
                        "name": "_tokenId",
                        "type": "uint256"
                }
        ],
        "name": "LimitedApprove",
        "outputs": [],
        "stateMutability": "nonpayable",
        "type": "function"
},
{
        "inputs": [
                {
                        "internalType": "address",
                        "name": "_from",
                        "type": "address"
                },
                {
                        "internalType": "address",
                        "name": "_to",
                        "type": "address"
                },
                {
                        "internalType": "uint256",
                        "name": "_tokenId",
                        "type": "uint256"
                }
        ],
        "name": "LimitedSafeTransferFrom",
        "outputs": [],
        "stateMutability": "nonpayable",
        "type": "function"
},
{
        "inputs": [
                {
                        "internalType": "address",
                        "name": "_from",
```

```
                        "type": "address"
                },
                {
                        "internalType": "address",
                        "name": "_to",
                        "type": "address"
                },
                {
                        "internalType": "uint256",
                        "name": "_tokenId",
                        "type": "uint256"
                },
                {
                        "internalType": "bytes",
                        "name": "_data",
                        "type": "bytes"
                }
        ],
        "name": "LimitedSafeTransferFrom",
        "outputs": [],
        "stateMutability": "nonpayable",
        "type": "function"
},
{
        "inputs": [
                {
                        "internalType": "address",
                        "name": "_operator",
                        "type": "address"
                },
                {
                        "internalType": "bool",
                        "name": "_approved",
                        "type": "bool"
                }
        ],
        "name": "LimitedSetApprovalForAll",
        "outputs": [],
        "stateMutability": "nonpayable",
        "type": "function"
},
{
        "inputs": [
                {
                        "internalType": "address",
                        "name": "_from",
                        "type": "address"
                },
                {
                        "internalType": "address",
                        "name": "_to",
                        "type": "address"
```

```
                },
                {
                        "internalType": "uint256",
                        "name": "_tokenId",
                        "type": "uint256"
                }
        ],
        "name": "LimitedTransferFrom",
        "outputs": [],
        "stateMutability": "nonpayable",
        "type": "function"
},
{
        "inputs": [],
        "name": "NOT_CURRENT_OWNER",
        "outputs": [
                {
                        "internalType": "string",
                        "name": "",
                        "type": "string"
                }
        ],
        "stateMutability": "view",
        "type": "function"
},
{
        "inputs": [],
        "name": "_creatorSiteAddress1",
        "outputs": [
                {
                        "internalType": "string",
                        "name": "",
                        "type": "string"
                }
        ],
        "stateMutability": "view",
        "type": "function"
},
{
        "inputs": [
                {
                        "internalType": "address",
                        "name": "_owner",
                        "type": "address"
                }
        ],
        "name": "balanceOf",
        "outputs": [
                {
                        "internalType": "uint256",
                        "name": "",
                        "type": "uint256"
```

```
                }
            ],
            "stateMutability": "view",
            "type": "function"
        },
        {
            "inputs": [
                {
                    "internalType": "uint256",
                    "name": "_tokenId",
                    "type": "uint256"
                },
                {
                    "internalType": "uint256",
                    "name": "_ownerTokenId",
                    "type": "uint256"
                },
                {
                    "internalType": "bytes32",
                    "name": "_ownerOTP",
                    "type": "bytes32"
                }
            ],
            "name": "burn",
            "outputs": [],
            "stateMutability": "nonpayable",
            "type": "function"
        },
        {
            "inputs": [
                {
                    "internalType": "uint256",
                    "name": "_tokenId",
                    "type": "uint256"
                }
            ],
            "name": "getApproved",
            "outputs": [
                {
                    "internalType": "address",
                    "name": "",
                    "type": "address"
                }
            ],
            "stateMutability": "view",
            "type": "function"
        },
        {
            "inputs": [],
            "name": "getCreatorData",
            "outputs": [
                {
```

```
                        "internalType": "string",
                        "name": "",
                        "type": "string"
                }
        ],
        "stateMutability": "view",
        "type": "function"
},
{
        "inputs": [],
        "name": "getTransferPermission",
        "outputs": [
                {
                        "internalType": "bool",
                        "name": "",
                        "type": "bool"
                }
        ],
        "stateMutability": "view",
        "type": "function"
},
{
        "inputs": [
                {
                        "internalType": "uint256",
                        "name": "_tokenId",
                        "type": "uint256"
                }
        ],
        "name": "getYourOtp",
        "outputs": [
                {
                        "internalType": "bytes32",
                        "name": "",
                        "type": "bytes32"
                }
        ],
        "stateMutability": "view",
        "type": "function"
},
{
        "inputs": [
                {
                        "internalType": "uint256",
                        "name": "_tokenId",
                        "type": "uint256"
                }
        ],
        "name": "getYourOtpUseGas",
        "outputs": [
                {
                        "internalType": "bytes32",
```

```
                "name": "",
                "type": "bytes32"
            }
        ],
        "stateMutability": "nonpayable",
        "type": "function"
    },
    {
        "inputs": [
            {
                "internalType": "uint256",
                "name": "_tokenId",
                "type": "uint256"
            }
        ],
        "name": "getYourTOTP",
        "outputs": [
            {
                "internalType": "bytes32",
                "name": "",
                "type": "bytes32"
            }
        ],
        "stateMutability": "view",
        "type": "function"
    },
    {
        "inputs": [
            {
                "internalType": "uint256",
                "name": "",
                "type": "uint256"
            }
        ],
        "name": "idToMintTimeBlockNum",
        "outputs": [
            {
                "internalType": "uint256",
                "name": "",
                "type": "uint256"
            }
        ],
        "stateMutability": "view",
        "type": "function"
    },
    {
        "inputs": [
            {
                "internalType": "uint256",
                "name": "",
                "type": "uint256"
            }
        ],
```

```
        ],
        "name": "idToOtpCallNum",
        "outputs": [
                {
                        "internalType": "uint256",
                        "name": "",
                        "type": "uint256"
                }
        ],
        "stateMutability": "view",
        "type": "function"
},
{
        "inputs": [
                {
                        "internalType": "address",
                        "name": "_owner",
                        "type": "address"
                },
                {
                        "internalType": "address",
                        "name": "_operator",
                        "type": "address"
                }
        ],
        "name": "isApprovedForAll",
        "outputs": [
                {
                        "internalType": "bool",
                        "name": "",
                        "type": "bool"
                }
        ],
        "stateMutability": "view",
        "type": "function"
},
{
        "inputs": [
                {
                        "internalType": "uint256",
                        "name": "_tokenId",
                        "type": "uint256"
                }
        ],
        "name": "isThisNftValidDate",
        "outputs": [
                {
                        "internalType": "bool",
                        "name": "",
                        "type": "bool"
                }
        ],
```

```
        "stateMutability": "view",
        "type": "function"
    },
    {
        "inputs": [
            {
                "internalType": "address",
                "name": "_to",
                "type": "address"
            },
            {
                "internalType": "uint256",
                "name": "_tokenId",
                "type": "uint256"
            },
            {
                "internalType": "string",
                "name": "_uri",
                "type": "string"
            }
        ],
        "name": "mint",
        "outputs": [],
        "stateMutability": "nonpayable",
        "type": "function"
    },
    {
        "inputs": [],
        "name": "name",
        "outputs": [
            {
                "internalType": "string",
                "name": "_name",
                "type": "string"
            }
        ],
        "stateMutability": "view",
        "type": "function"
    },
    {
        "inputs": [],
        "name": "owner",
        "outputs": [
            {
                "internalType": "address",
                "name": "",
                "type": "address"
            }
        ],
        "stateMutability": "view",
        "type": "function"
    },
```

```
{
	"inputs": [
		{
			"internalType": "uint256",
			"name": "_tokenId",
			"type": "uint256"
		}
	],
	"name": "ownerOf",
	"outputs": [
		{
			"internalType": "address",
			"name": "_owner",
			"type": "address"
		}
	],
	"stateMutability": "view",
	"type": "function"
},
{
	"inputs": [
		{
			"internalType": "uint256",
			"name": "_newNum",
			"type": "uint256"
		}
	],
	"name": "setBlockTimeLimitNum",
	"outputs": [],
	"stateMutability": "nonpayable",
	"type": "function"
},
{
	"inputs": [
		{
			"internalType": "uint256",
			"name": "_newNum",
			"type": "uint256"
		}
	],
	"name": "setCreatorOtpNum",
	"outputs": [],
	"stateMutability": "nonpayable",
	"type": "function"
},
{
	"inputs": [
		{
			"internalType": "string",
			"name": "_newAddress",
			"type": "string"
		}
```

```
        ],
        "name": "setSite",
        "outputs": [],
        "stateMutability": "nonpayable",
        "type": "function"
    },
    {
        "inputs": [
            {
                "internalType": "bool",
                "name": "_newState",
                "type": "bool"
            }
        ],
        "name": "setTransferPermission",
        "outputs": [],
        "stateMutability": "nonpayable",
        "type": "function"
    },
    {
        "inputs": [
            {
                "internalType": "bytes4",
                "name": "_interfaceID",
                "type": "bytes4"
            }
        ],
        "name": "supportsInterface",
        "outputs": [
            {
                "internalType": "bool",
                "name": "",
                "type": "bool"
            }
        ],
        "stateMutability": "view",
        "type": "function"
    },
    {
        "inputs": [],
        "name": "symbol",
        "outputs": [
            {
                "internalType": "string",
                "name": "_symbol",
                "type": "string"
            }
        ],
        "stateMutability": "view",
        "type": "function"
    },
    {
```

```
		"inputs": [
			{
				"internalType": "uint256",
				"name": "_tokenId",
				"type": "uint256"
			}
		],
		"name": "tokenURI",
		"outputs": [
			{
				"internalType": "string",
				"name": "",
				"type": "string"
			}
		],
		"stateMutability": "view",
		"type": "function"
	},
	{
		"inputs": [
			{
				"internalType": "address",
				"name": "_newOwner",
				"type": "address"
			}
		],
		"name": "transferOwnership",
		"outputs": [],
		"stateMutability": "nonpayable",
		"type": "function"
	},
	{
		"inputs": [
			{
				"internalType": "uint256",
				"name": "_tokenId",
				"type": "uint256"
			},
			{
				"internalType": "string",
				"name": "_uri2",
				"type": "string"
			}
		],
		"name": "writeUri2",
		"outputs": [],
		"stateMutability": "nonpayable",
		"type": "function"
	}
]
```

!!OTPAUTH

//based https://github.com/0xcert/ethereum-erc721
//
//Author NZRI_Kaz-Naz_https://github.com/Kaz-Naz,Toshiya Y_https://github.com/Toshiya-Y-K-N
//by;0x86904339D23BF346C1FFF31Cc3Bb7262fa59d837

//ropsten;
//rinkeby；
//ERC721LTO

```
pragma solidity 0.6.2;

import "./nf-token.sol";
import "./nf-token-metadata.sol";
import "./nf-token-enumerable.sol";
import "../ownership/ownable.sol";

/**
 *
 * @dev This is an example contract implementation of NFToken with metadata extension.
 */
contract MyERC721LTOAUTH is
  NFTokenMetadata,
  Ownable
{

  /**
   * @dev Contract constructor. Sets metadata extension `name` and `symbol`.
   */
  constructor()
    public
  {
    nftName = "OneTimePassward-Authenticator";
    nftSymbol = "OTPAUTH";
  }


    /**
    * Creator Data Section===================
    */
    //Creator data　作者の名前とウェブサイトやメールアドレス、ＳＮＳアカウント、電話番号などを記載する。
    //(有事の際はこのサイトでコントラクトの運用に関してアナウンスする)
    string private _creatorData = "Toshiya_github.com/Toshiya-Y-K-N";
    function getCreatorData() public view onlyOwner returns (string memory) {
        return _creatorData;
    }
    //Site　招待するウェブサイトアドレス
    string public _creatorSiteAddress1 = "@";//example
```

```
    //address setter getter
    function setSite(string memory _newAddress) public onlyOwner {
        _creatorSiteAddress1 = _newAddress ;
    }



//=====================================================================
    /**
    * oneTimePassCode Section==================
    */

    //secret
    string private _secret = "ty2ty";//examle. gene と auth で統一すること

    //CreatorNum blocknum
    uint256 private _creatorOneTimeNum = 20201717;//examle. gene と auth で統一すること

    //setCreatorOneTimeNum  public onlyOwner
    //手動でワンタイムパスワード（ＯＴＰ）を変える数値を入力できる関数。運営者のオーナーのみが関数を実行出来る。
    /**
    * @dev setCreatorOneTimeNum(uint256 _newNum) public onlyOwner
    */
    function setCreatorOtpNum(uint256 _newNum) public onlyOwner {
        _creatorOneTimeNum  = _newNum ;
    }

    //90 日、30 秒ブロックで 518400 後に無効になる場合 518400, 15sec で 259200 この関数は顧客の NFT が時間的に見て有効か示す。
    //web サイト側でアクセスしてきたユーザーのトークンが無効かどうか、web サイト側で決めるための関数。
    uint256 _blockTimeLimitNum = 259200;// 518400
    function setBlockTimeLimitNum(uint256 _newNum)public onlyOwner {
        _blockTimeLimitNum  = _newNum ;
    }


//7 桁のワンタイムパスワードに関してはデプロイ制限により追加困難。
//web アプリ上で搭載必要。



//オーナー変更型 OTP================================================

//!!web アプリの認証時にユーザーのトークンが有効かどうかを生成器トークンの関数で確認すること。
//!! use---->>>> "  function isThisNftValidDate(uint256 _tokenId)  "

    //web3 を顧客が使えなくても認証できるようにする場合の方法
    function authYourOtpByOwner(address _adr , uint256 _tokenId , bytes32 _OTP) external view returns (bool) {
        require(msg.sender != address(0), ZERO_ADDRESS);//セロアドレスでないか確認
        require( _getOwnerNFTCount(msg.sender) > 0 );//オーナーの Auth トークンが一つ以上か確認
```

```
        string memory str =  _secret ;
        uint256 inte = _creatorOneTimeNum ;
        address adr = _adr;

        if ( _OTP == sha256(abi.encodePacked(inte, str,adr,_tokenId)) ) {
            return true ;
        } else {
            return false;
        }
    }

//TOTP-時刻同期型ワンタイムパスワード=========================================
==

    //web3 を顧客が使えなくても認証できるようにする場合の方法
    function authYourTotpByOwner(address _adr , uint256 _tokenId , bytes32 _OTP) external
view returns (bool) {
        require(msg.sender != address(0), ZERO_ADDRESS);//セロアドレスでないか確認
        require( _getOwnerNFTCount(msg.sender) > 0 );//オーナーの Auth トークンが一つ以上か
確認

        string memory str =  _secret ;
        uint256 inte = block.number ;
        address adr = _adr;

        if ( _OTP == sha256(abi.encodePacked(inte, str,adr,_tokenId)) ) {
            return true ;
        } else {
            return false;
        }
    }

//==============================

    /**
     * @dev Removes a NFT from owner.
     * @param _tokenId Which NFT we want to remove.(burn)
     */
  function burn(
    uint256 _tokenId
  )
    external
    onlyOwner
  {
    super._burn(_tokenId);
  }
  /**
   * @dev Mints a new NFT.
   * @param _to The address that will own the minted NFT.
   * @param _tokenId of the NFT to be minted by the msg.sender.
   * @param _uri String representing RFC 3986 URI.
   */
```

```
    function mint(
        address _to,
        uint256 _tokenId,
        string calldata _uri
    )
        external
        onlyOwner
    {
        super._mint(_to, _tokenId);
        super._setTokenUri(_tokenId, _uri);
        idToMintTimeBlockNum[_tokenId] = block.number;//mint 時の時刻、blocktime 追加
    }



}
//////////////////

////以下実装していない関数////
/*
//Metamask など web3 経由での認証用
function authYourOtpByWeb3(uint256 _tokenId , bytes32 _yourOTP) external view returns (bool)
{
    require(msg.sender != address(0), ZERO_ADDRESS);//セロアドレスでないか確認
    require( _getOwnerNFTCount(msg.sender) > 0 );//オーナーの Auth トークンが一つ以上か確認

    //条件通りならばクリエイターのセットしたパスワードをリターンする
    string memory str =  _secret ;
    uint256 inte = _creatorOneTimeNum ;
    address adr = msg.sender;

    if ( _yourOTP == sha256(abi.encodePacked(inte, str,adr,_tokenId)) ) {
        return true ;
    } else {
        return false ;
    }
}
*/

/*
//Metamask など web3 経由での認証用　;　正規使用法
function authYourTotpByWeb3(uint256 _tokenId , bytes32 _yourOTP) external view returns (bool)
{
    require(msg.sender != address(0), ZERO_ADDRESS);//セロアドレスでないか確認
    require( _getOwnerNFTCount(msg.sender) > 0 );//オーナーの Auth トークンが一つ以上か確認

    //条件通りならばクリエイターのセットしたパスワードをリターンする
    string memory str =  _secret ;
    uint256 inte = block.number ;
    address adr = msg.sender;

    if ( _yourOTP == sha256(abi.encodePacked(inte, str,adr,_tokenId)) ) {
        return true ;
```

```
    } else {
        return false;
    }
}
*/

/*
  function writeUri2(
    uint256 _tokenId,
    string calldata _uri2
  )
    external
    onlyOwner
    validNFToken(_tokenId)
  {
    idToUri2[_tokenId] = _uri2;
  }
*/

ABI (Auth)
[
    {
        "inputs": [],
        "stateMutability": "nonpayable",
        "type": "constructor"
    },
    {
        "anonymous": false,
        "inputs": [
            {
                "indexed": true,
                "internalType": "address",
                "name": "_owner",
                "type": "address"
            },
            {
                "indexed": true,
                "internalType": "address",
                "name": "_approved",
                "type": "address"
            },
            {
                "indexed": true,
                "internalType": "uint256",
                "name": "_tokenId",
                "type": "uint256"
            }
        ],
        "name": "Approval",
        "type": "event"
    },
    {
```

```
        "anonymous": false,
        "inputs": [
                {
                        "indexed": true,
                        "internalType": "address",
                        "name": "_owner",
                        "type": "address"
                },
                {
                        "indexed": true,
                        "internalType": "address",
                        "name": "_operator",
                        "type": "address"
                },
                {
                        "indexed": false,
                        "internalType": "bool",
                        "name": "_approved",
                        "type": "bool"
                }
        ],
        "name": "ApprovalForAll",
        "type": "event"
},
{
        "anonymous": false,
        "inputs": [
                {
                        "indexed": true,
                        "internalType": "address",
                        "name": "_from",
                        "type": "address"
                },
                {
                        "indexed": true,
                        "internalType": "address",
                        "name": "_to",
                        "type": "address"
                },
                {
                        "indexed": true,
                        "internalType": "uint256",
                        "name": "_tokenId",
                        "type": "uint256"
                }
        ],
        "name": "LimitedTransfer",
        "type": "event"
},
{
        "anonymous": false,
        "inputs": [
```

```
                {
                        "indexed": true,
                        "internalType": "address",
                        "name": "previousOwner",
                        "type": "address"
                },
                {
                        "indexed": true,
                        "internalType": "address",
                        "name": "newOwner",
                        "type": "address"
                }
        ],
        "name": "OwnershipTransferred",
        "type": "event"
},
{
        "anonymous": false,
        "inputs": [
                {
                        "indexed": true,
                        "internalType": "address",
                        "name": "_from",
                        "type": "address"
                },
                {
                        "indexed": true,
                        "internalType": "address",
                        "name": "_to",
                        "type": "address"
                },
                {
                        "indexed": true,
                        "internalType": "uint256",
                        "name": "_tokenId",
                        "type": "uint256"
                }
        ],
        "name": "Transfer",
        "type": "event"
},
{
        "inputs": [],
        "name": "CANNOT_TRANSFER_TO_ZERO_ADDRESS",
        "outputs": [
                {
                        "internalType": "string",
                        "name": "",
                        "type": "string"
                }
        ],
        "stateMutability": "view",
```

```
        "type": "function"
    },
    {
        "inputs": [
            {
                "internalType": "address",
                "name": "_approved",
                "type": "address"
            },
            {
                "internalType": "uint256",
                "name": "_tokenId",
                "type": "uint256"
            }
        ],
        "name": "LimitedApprove",
        "outputs": [],
        "stateMutability": "nonpayable",
        "type": "function"
    },
    {
        "inputs": [
            {
                "internalType": "address",
                "name": "_from",
                "type": "address"
            },
            {
                "internalType": "address",
                "name": "_to",
                "type": "address"
            },
            {
                "internalType": "uint256",
                "name": "_tokenId",
                "type": "uint256"
            }
        ],
        "name": "LimitedSafeTransferFrom",
        "outputs": [],
        "stateMutability": "nonpayable",
        "type": "function"
    },
    {
        "inputs": [
            {
                "internalType": "address",
                "name": "_from",
                "type": "address"
            },
            {
                "internalType": "address",
```

```
                        "name": "_to",
                        "type": "address"
                },
                {
                        "internalType": "uint256",
                        "name": "_tokenId",
                        "type": "uint256"
                },
                {
                        "internalType": "bytes",
                        "name": "_data",
                        "type": "bytes"
                }
        ],
        "name": "LimitedSafeTransferFrom",
        "outputs": [],
        "stateMutability": "nonpayable",
        "type": "function"
},
{
        "inputs": [
                {
                        "internalType": "address",
                        "name": "_operator",
                        "type": "address"
                },
                {
                        "internalType": "bool",
                        "name": "_approved",
                        "type": "bool"
                }
        ],
        "name": "LimitedSetApprovalForAll",
        "outputs": [],
        "stateMutability": "nonpayable",
        "type": "function"
},
{
        "inputs": [
                {
                        "internalType": "address",
                        "name": "_from",
                        "type": "address"
                },
                {
                        "internalType": "address",
                        "name": "_to",
                        "type": "address"
                },
                {
                        "internalType": "uint256",
                        "name": "_tokenId",
```

```
                        "type": "uint256"
                }
        ],
        "name": "LimitedTransferFrom",
        "outputs": [],
        "stateMutability": "nonpayable",
        "type": "function"
},
{
        "inputs": [],
        "name": "NOT_CURRENT_OWNER",
        "outputs": [
                {
                        "internalType": "string",
                        "name": "",
                        "type": "string"
                }
        ],
        "stateMutability": "view",
        "type": "function"
},
{
        "inputs": [],
        "name": "_creatorSiteAddress1",
        "outputs": [
                {
                        "internalType": "string",
                        "name": "",
                        "type": "string"
                }
        ],
        "stateMutability": "view",
        "type": "function"
},
{
        "inputs": [
                {
                        "internalType": "address",
                        "name": "_adr",
                        "type": "address"
                },
                {
                        "internalType": "uint256",
                        "name": "_tokenId",
                        "type": "uint256"
                },
                {
                        "internalType": "bytes32",
                        "name": "_OTP",
                        "type": "bytes32"
                }
        ],
```

```
        "name": "authYourOtpByOwner",
        "outputs": [
                {
                        "internalType": "bool",
                        "name": "",
                        "type": "bool"
                }
        ],
        "stateMutability": "view",
        "type": "function"
},
{
        "inputs": [
                {
                        "internalType": "address",
                        "name": "_adr",
                        "type": "address"
                },
                {
                        "internalType": "uint256",
                        "name": "_tokenId",
                        "type": "uint256"
                },
                {
                        "internalType": "bytes32",
                        "name": "_OTP",
                        "type": "bytes32"
                }
        ],
        "name": "authYourTotpByOwner",
        "outputs": [
                {
                        "internalType": "bool",
                        "name": "",
                        "type": "bool"
                }
        ],
        "stateMutability": "view",
        "type": "function"
},
{
        "inputs": [
                {
                        "internalType": "address",
                        "name": "_owner",
                        "type": "address"
                }
        ],
        "name": "balanceOf",
        "outputs": [
                {
                        "internalType": "uint256",
```

```
                        "name": "",
                        "type": "uint256"
                }
        ],
        "stateMutability": "view",
        "type": "function"
},
{
        "inputs": [
                {
                        "internalType": "uint256",
                        "name": "_tokenId",
                        "type": "uint256"
                }
        ],
        "name": "burn",
        "outputs": [],
        "stateMutability": "nonpayable",
        "type": "function"
},
{
        "inputs": [
                {
                        "internalType": "uint256",
                        "name": "_tokenId",
                        "type": "uint256"
                }
        ],
        "name": "getApproved",
        "outputs": [
                {
                        "internalType": "address",
                        "name": "",
                        "type": "address"
                }
        ],
        "stateMutability": "view",
        "type": "function"
},
{
        "inputs": [],
        "name": "getCreatorData",
        "outputs": [
                {
                        "internalType": "string",
                        "name": "",
                        "type": "string"
                }
        ],
        "stateMutability": "view",
        "type": "function"
},
```

```
{
	"inputs": [],
	"name": "getTransferPermission",
	"outputs": [
		{
			"internalType": "bool",
			"name": "",
			"type": "bool"
		}
	],
	"stateMutability": "view",
	"type": "function"
},
{
	"inputs": [
		{
			"internalType": "uint256",
			"name": "",
			"type": "uint256"
		}
	],
	"name": "idToMintTimeBlockNum",
	"outputs": [
		{
			"internalType": "uint256",
			"name": "",
			"type": "uint256"
		}
	],
	"stateMutability": "view",
	"type": "function"
},
{
	"inputs": [
		{
			"internalType": "uint256",
			"name": "",
			"type": "uint256"
		}
	],
	"name": "idToOtpCallNum",
	"outputs": [
		{
			"internalType": "uint256",
			"name": "",
			"type": "uint256"
		}
	],
	"stateMutability": "view",
	"type": "function"
},
{
```

```
        "inputs": [
                {
                        "internalType": "address",
                        "name": "_owner",
                        "type": "address"
                },
                {
                        "internalType": "address",
                        "name": "_operator",
                        "type": "address"
                }
        ],
        "name": "isApprovedForAll",
        "outputs": [
                {
                        "internalType": "bool",
                        "name": "",
                        "type": "bool"
                }
        ],
        "stateMutability": "view",
        "type": "function"
},
{
        "inputs": [
                {
                        "internalType": "address",
                        "name": "_to",
                        "type": "address"
                },
                {
                        "internalType": "uint256",
                        "name": "_tokenId",
                        "type": "uint256"
                },
                {
                        "internalType": "string",
                        "name": "_uri",
                        "type": "string"
                }
        ],
        "name": "mint",
        "outputs": [],
        "stateMutability": "nonpayable",
        "type": "function"
},
{
        "inputs": [],
        "name": "name",
        "outputs": [
                {
                        "internalType": "string",
```

```
                        "name": "_name",
                        "type": "string"
                }
        ],
        "stateMutability": "view",
        "type": "function"
},
{
        "inputs": [],
        "name": "owner",
        "outputs": [
                {
                        "internalType": "address",
                        "name": "",
                        "type": "address"
                }
        ],
        "stateMutability": "view",
        "type": "function"
},
{
        "inputs": [
                {
                        "internalType": "uint256",
                        "name": "_tokenId",
                        "type": "uint256"
                }
        ],
        "name": "ownerOf",
        "outputs": [
                {
                        "internalType": "address",
                        "name": "_owner",
                        "type": "address"
                }
        ],
        "stateMutability": "view",
        "type": "function"
},
{
        "inputs": [
                {
                        "internalType": "uint256",
                        "name": "_newNum",
                        "type": "uint256"
                }
        ],
        "name": "setBlockTimeLimitNum",
        "outputs": [],
        "stateMutability": "nonpayable",
        "type": "function"
},
```

```
{
	"inputs": [
		{
			"internalType": "uint256",
			"name": "_newNum",
			"type": "uint256"
		}
	],
	"name": "setCreatorOtpNum",
	"outputs": [],
	"stateMutability": "nonpayable",
	"type": "function"
},
{
	"inputs": [
		{
			"internalType": "string",
			"name": "_newAddress",
			"type": "string"
		}
	],
	"name": "setSite",
	"outputs": [],
	"stateMutability": "nonpayable",
	"type": "function"
},
{
	"inputs": [
		{
			"internalType": "bytes4",
			"name": "_interfaceID",
			"type": "bytes4"
		}
	],
	"name": "supportsInterface",
	"outputs": [
		{
			"internalType": "bool",
			"name": "",
			"type": "bool"
		}
	],
	"stateMutability": "view",
	"type": "function"
},
{
	"inputs": [],
	"name": "symbol",
	"outputs": [
		{
			"internalType": "string",
			"name": "_symbol",
```

```
                        "type": "string"
                    }
                ],
                "stateMutability": "view",
                "type": "function"
            },
            {
                "inputs": [
                    {
                        "internalType": "uint256",
                        "name": "_tokenId",
                        "type": "uint256"
                    }
                ],
                "name": "tokenURI",
                "outputs": [
                    {
                        "internalType": "string",
                        "name": "",
                        "type": "string"
                    }
                ],
                "stateMutability": "view",
                "type": "function"
            },
            {
                "inputs": [
                    {
                        "internalType": "address",
                        "name": "_newOwner",
                        "type": "address"
                    }
                ],
                "name": "transferOwnership",
                "outputs": [],
                "stateMutability": "nonpayable",
                "type": "function"
            }
]
```