Solidity API

Base64

[MIT License] Provides a function for encoding some bytes in base64

TABLE

```
bytes TABLE
```

encode

```
function encode(bytes data) public pure returns (string)
```

Encodes some bytes to the base64 representation

Parameters

Name	Type	Description
data	bytes	- Data to encode

Return Values

Name	Type	Description
[0]	string	encoded string

Cast

Provides a function for casting between uint and string

uint2str

```
function uint2str(uint256 _i) public pure returns (string _uintAsString)
```

Converts uint to string

Parameters

Name	Type	Description
_i	uint256	- Number to convert

str2uint

```
function str2uint(string numString) public pure returns (uint256)
```

Converts string to uint

Parameters

Name	Type	Description
numString	string	- String to convert

nftMarketplace

balance

```
mapping(address => uint256) balance
```

listing

```
mapping(address => mapping(string => struct Listing)) listing
```

sellerIds

```
mapping(address => string[]) sellerIds
```

sellerOffers

```
mapping(address => struct SellerOffer[]) sellerOffers
```

isOwner

```
modifier isOwner(address concertAddr, uint256 tokenId, uint256 amount)
```

Modifier checks if sender is owner of proper amount tokens

Parameters

Name	Type	Description
concertAddr	address	- Address of concert
tokenId	uint256	- Token ID
amount	uint256	- Amount of tokens

isNotListed

```
modifier isNotListed(address concertAddr, uint256 tokenId)
```

Modifier checks if token is not yet listed

Parameters

Name	Type	Description
concertAddr	address	- Address of concert
tokenId	uint256	- Token ID

isListed

```
modifier isListed(address concertAddr, address owner, uint256 tokenId)
```

Modifier checks if token is listed

Parameters

Name	Type	Description
concertAddr	address	- Address of concert
owner	address	- Owner of token
tokenId	uint256	- Token ID

insertOffer

```
function insertOffer(address concertAddr, struct Listing params) external
```

Function adds offer on marketplace

Parameters

Name	Type	Description
concertAddr	address	- Address of concert
params	struct Listing	- Offer details

updateOffer

```
function updateOffer(address concertAddr, struct Listing params) external
```

Function updates offer on marketplace

Parameters

Name	Type	Description
concertAddr	address	- Address of concert
params	struct Listing	- Offer details

deleteOffer

```
function deleteOffer(address concertAddr, uint256 tokenId) external payable
```

Function deletes offer on marketplace

Parameters

Name	Type	Description
concertAddr	address	- Address of concert
tokenId	uint256	- Token ID

buyTicket

```
function buyTicket(address concertAddr, address owner, uint256 tokenId, uint256 amount) exte
```

Function for buying ticket (token) from marketplace

Parameters

Name	Type	Description
concertAddr	address	- Address of concert
owner	address	- Current owner of token
tokenId	uint256	- Token ID
amount	uint256	- Amount of tokens to buy

withdraw

```
function withdraw(address payable destAddr) public
```

Function withdraws organizator credits

Parameters

Name	Type	Description
destAddr	address payable	- Addres to send money

getSellerIds

```
function getSellerIds(address concert) public view returns (string[])
```

getOffersBySeller

```
function getOffersBySeller(address seller) public view returns (struct SellerOffer[])
```

getListedTicket

```
function getListedTicket(address concertAddr, string sellerId) external view returns (struct
```

tickeD1155

orgAddress

```
address orgAddress
```

name

```
string name
```

description

```
string description
```

date

```
uint256 date
```

image

```
string image
```

sectors

```
struct Sector[] sectors
```

soldTokenIds

```
mapping(string => uint256[]) soldTokenIds
```

ticketAttr

```
mapping(uint256 => struct Ticket) ticketAttr
```

supportsInterface

```
function supportsInterface(bytes4 interfaceId) public view virtual returns (bool)
```

constructor

```
constructor(address _owner, string _name, string _desc, uint256 _date, string _image, string
```

- Constructor initialize basic properties and creates structure of sectors

createAndMintTickets

```
function createAndMintTickets() public
```

Function for creating tickets attributes and minting if specified. This function publish tickets

buyTicket

```
function buyTicket(uint256 tokenId, uint256 amount) external payable
```

Function for buying ticket (token) for concert

Parameters

Name	Type	Description
tokenId	uint256	- ID of token
amount	uint256	- Amount of tokens to buy

uri

```
function uri(uint256 tokenId) public view returns (string)
```

Function for compatibility with openSea way of displaying SFT properties

Parameters

Name	Type	Description
tokenId	uint256	- ID of token

tokenURI

```
function tokenURI(uint256 tokenId) public view returns (string)
```

Function for compatibility with openSea way of displaying NFT properties

Parameters

Name	Type	Description
tokenId	uint256	- ID of token

addSectors

```
function addSectors(string[] _sectors) external
```

Function adds new sectors to concert

Parameters

Name	Type	Description
_sectors	string[]	- Structure of sectors at audience

withdrawOrgCredits

```
function withdrawOrgCredits(address payable destAddr) public
```

Function withdraws organizator credits

Parameters

Name	Type	Description
destAddr	address payable	- Addres to send money

addResellFee

```
function addResellFee(uint256 fee) external
```

Function adds fee for each sale on marketplace for organizer

Parameters

Name	Type	Description
fee	uint256	- Credits

setDate

```
function setDate(uint256 newDate) external
```

getSectors

```
function getSectors() public view returns (struct Sector[])
```

getSoldTokenIds

```
function getSoldTokenIds(string sectorName) public view returns (uint256[])
```

tickeDFactory

whitelist

```
mapping(address => bool) whitelist
```

deployedContracts

```
mapping(address => struct DepConcert[]) deployedContracts
```

organizers

```
address[] organizers
```

tickeD1155Created

```
event tickeD1155Created(address owner, address tokenContract)
```

setOrganizatorPermission

```
function setOrganizatorPermission(address addr, bool toggle) public
```

Function that toggles organizator permission of creating concerts

Parameters

Name	Type	Description
addr	address	- Address of organizator
toggle	bool	- Granted permissions true/false

createEvent

```
function createEvent(string _name, string _desc, uint256 _date, string _image, string[] _sec
```

Function creates new smarcontract concert

Parameters

Name	Type	Description
_name	string	- Name of concert
_desc	string	- Description of concert
_date	uint256	- Date of concert
_image	string	- Image of audience layout
_sectors	string[]	- Structure of sectors at audience

getDepContracts

```
function getDepContracts(address org) public view returns (struct DepConcert[])
```

Function that returns all concerts for given organizer

Parameters

Name	Type	Description
org	address	- Address of organizator

Return Values

Name	Type	Description
[0]	struct DepConcert[]	List of deployed concerts

getOrganizers

```
function getOrganizers() public view returns (address[])
```

Function that returns all organizers for displaying concerts

Return Values

Name	Type	Description
[0]	address[]	List of organizers

Sector

```
struct Sector {
  string name;
  bool isNumerable;
  uint256 seatStart;
  uint256 seatStop;
  bool mintedByOrg;
  uint256 price;
  uint256[] availableTokenIds;
}
```

Ticket

```
struct Ticket {
  string sectorName;
  uint256 seatNumber;
  bool minted;
  uint256 price;
  bool sold;
}
```

DepConcert

```
struct DepConcert {
  address contractAddress;
  string name;
}
```

Listing

```
struct Listing {
  uint256 tokenId;
  uint256 amount;
  uint256 price;
  address seller;
}
```

ListedTicket

```
struct ListedTicket {
  struct Listing listing;
  struct Ticket ticket;
}
```

SellerOffer

```
struct SellerOffer {
  address concertAddr;
  string sellerId;
}
```