

Fixed Subscription License Documentation

Contract Overview:

The FixedSubscriptionLicense contract is an ERC-721 based contract designed for managing fixed-price software subscription licenses. It handles the process of purchasing, minting, transferring, updating, and canceling licenses, as well as providing necessary information and functionality for managing subscription licenses.

Inherited Functions:

This contract inherits the following contracts and their respective functions:

ERC721 - Non-Fungible Token Standard functions, such as:

- **balanceOf(address owner) public view** - Get the balance (number of tokens) of a specified owner.
- **ownerOf(uint256 tokenId) public view** - Get the owner of a specified token ID.
- **approve(address to, uint256 tokenId) public** - Approve a specified address to manage a specified token ID.
- **getApproved(uint256 tokenId) public view** - Get the approved address for a specified token ID.
- **setApprovalForAll(address operator, bool approved) public** - Set or unset approval for an address to manage all tokens of the caller.
- **isApprovedForAll(address owner, address operator) public view** - Check if an address is approved to manage all tokens of another address.

For additional functionalities, please refer to the official documentation of the ERC721 token standard available [here](#).

Ownable - Functions for restricting access to contract owner-only

SafeMath - Functions for safe mathematical operations, preventing integer overflows and underflows.

Contract Functions:

- **constructor(string memory companyName, string memory licenseName, uint256 licensePrice, uint256 subscriptionPeriodSecond) public** - Initializes the contract with company name, license name, license price, and subscription period (in seconds).
- **buyToken() public payable** - Allows a user to purchase a subscription license by sending the required amount of Ether. A new token will be minted, and its start and expiration timestamps will be set.
- **mintToken(address customer) public onlyOwner** - Allows the contract owner to mint a new token for a specified customer address.
- **updateSubscription(uint256 tokenId) public payable** - Allows a user to update their subscription by sending the required amount of Ether. The expiration timestamp of the license will be updated accordingly.
- **_baseURI() internal pure override returns (string memory)** - Returns the base URI used for constructing token URIs.
- **tokenURI(uint256 tokenId) public view virtual override returns (string memory)** - Returns a JSON-formatted token URI containing the license data, such as company name, license name, license type, price, start and expiration timestamps, and token ID.
- **getTokenCounter() public view returns (uint256)** - Returns the current value of the token counter.
- **getLicensePrice() public view returns (uint256)** - Returns the current price of a license.
- **getSubscriptionTimePeriod() public view returns (uint256)** - Returns the subscription period in seconds.
- **getExpirationTime(uint256 tokenId) public view returns (uint256)** - Returns the expiration timestamp of a specified token ID.
- **isSubscriptionActive(uint256 tokenId) public view returns (bool)** - Checks if the subscription for a specified token ID is still active.
- **isTransferAllowed(uint256 tokenId) public view returns (bool)** - Checks if the transfer is allowed for a specified token ID.
- **withdraw() public onlyOwner** - Allows the contract owner to withdraw the contract's balance.

- **_beforeTokenTransfer(address from, address to, uint256 firstTokenId, uint256 batchSize) internal virtual override** - Checks if the transfer is allowed for a specified token ID and calls the super implementation of the function.
- **transferFrom(address from, address to, uint256 tokenId) public virtual override** - Allows an approved address to transfer a token on behalf of its owner or by owner.
- **safeTransferFrom(address from, address to, uint256 tokenId) public virtual override** - Allows an approved address to securely transfer a token on behalf of its owner or by owner.
- **safeTransferFrom(address from, address to, uint256 tokenId, bytes memory _data) public virtual override** - Allows an approved address to securely transfer a token on behalf of its owner or by owner, with additional data.
- **updateLicensePrice(uint256 newPrice) public onlyOwner** - Allows the contract owner to update the license price.
- **allowTransfer(uint256 tokenId) public onlyOwner** - Allows the contract owner to enable transfers for a specified token ID.
- **restrictTransfer(uint256 tokenId) public onlyOwner** - Allows the contract owner to disable transfers for a specified token ID.
- **cancelSubscription(uint256 tokenId) public onlyOwner** - Allows the contract owner to cancel a subscription by setting its expiration timestamp to the current timestamp.
- **getRefundEligible(uint256 tokenId, uint256 timestamp) public view returns (bool)** - Returns a boolean indicating if the token is eligible for a refund, based on the given timestamp.

Events:

- **CreatedSubscriptionToken(uint256 indexed tokenId, uint256 licensePrice)** - Emitted when a new subscription token is created, contains the token ID and license price.
- **UpdatedSubscriptionToken(uint256 indexed tokenId, uint256 licensePrice)** - Emitted when a subscription token is updated, contains the token ID and license price.