Solidity API

Plant

Plant generated NFTs

_plantContractAddress

```
address _plantContractAddress
```

The address of the plantDrawer Contract

SELL_PRICE

```
uint256 SELL_PRICE
```

Price for one Plant

SEEDING_PRICE

```
uint256 SEEDING_PRICE
```

Price for seeding a Plant

WATERING_PRICE

```
uint256 WATERING_PRICE
```

Price for watering a Plant

MAX_WATERING_COUNTER

```
uint8 MAX_WATERING_COUNTER
```

Amount of how many times a plant needs watering befor harvesting or levelup is possible

MAX_HARVEST_LEVEL_COUNTER

```
uint8 MAX_HARVEST_LEVEL_COUNTER
```

Maximum actions on a plant

HARVEST_AMOUNT

```
uint8 HARVEST_AMOUNT
```

Amount that can be harvested with level 1 and healthyness 100%

_details

```
mapping(uint256 &#x3D;&gt; struct IPlant.Details) _details
```

The token ID plant details

_distributedTokens

```
uint256 _distributedTokens
```

Sum of all Tokens distrubuted

_distributedTokensPerWallet

```
mapping(address &#x3D;&gt; uint256) _distributedTokensPerWallet
```

Tokens distrubuted per wallet

_isTimerActive

```
bool _isTimerActive
```

Flag to enable/disable timer function for demonstration

_timeUntilNextAction

```
uint256 _timeUntilNextAction
```

Time in seconds when next action(water/harvest/levelUp) on the plant is possible

_timeUntilRotted

```
uint256 _timeUntilRotted
```

Time in seconds when plant dies without an action

withinTimeSpan

```
modifier withinTimeSpan(uint256 tokenId)
```

Modifier to check if the next action is within the desired timespan

LEVEL_FACTOR_PER_LEVEL

```
uint256[] LEVEL_FACTOR_PER_LEVEL
```

List of Factor per level times 1000

constructor

```
constructor(address plantContractAddress) public
```

getDetails

```
function getDetails(uint256 tokenId) public view returns (struct IPlant.Details
details)
```

Returns the details associated with a given token ID.

Throws if the token ID is not valid.

Name	Type	Description
tokenId	uint256	The ID of the token that represents the Plant

Name	Type	Description
details	struct IPlant.Details	memory

getIsTimerActive

```

```

```
function getIsTimerActive() external view returns (bool)
```

Method to get the timer for demonstration

Name	Type	Description
[0]	bool	The IsTimerActive boolean

setIsTimerActive

```
function setIsTimerActive(bool active) external
```

Method to enable/disable timer for demonstration

Name	Type	Description
active	bool	flag to enable disable timer

getTimeUntilNextAction

```
function getTimeUntilNextAction() external view returns (uint256)
```

_Method to get the *timeUntilNextAction*

Name	Type	Description
[0]	uint256	The time until next action in seconds

setTimeUntilNextAction

```
function setTimeUntilNextAction(uint256 timeUntilNextAction) external
```

_Method to set the *timeUntilNextAction for demonstration*

Name	Type	Description
timeUntilNextAction	uint256	in seconds until next action is allowed

getTimeUntilRotted

```
function getTimeUntilRotted() external view returns (uint256)
```

_Method to get the *timeUntilRotted*

Name	Type	Description
[0]	uint256	The time until the plant is rotted in seconds

setTimeUntilRotted

```
function setTimeUntilRotted(uint256 timeUntilRotted) external
```

_Method to set the *timeUntilRotted for demonstration*

Name	Type	Description
timeUntilRotted	uint256	in seconds until the plant is rotted

getIsRotted

```
function getIsRotted(uint256 tokenId) public view returns (bool)
```

Check if a plant is rotted

Name	Type	Description
tokenId	uint256	Id of the token to check

Name	Type	Description
[0]	bool	A boolean if the plan is rotted

getDistributedTokens

```
function getDistributedTokens() external view returns (uint256)
```

Return Sum of all Tokens distrubuted

Name	Type	Description
[0]	uint256	Sum of all Tokens distrubuted

getDistributedTokensOfAddress

```
function getDistributedTokensOfAddress() external view returns (uint256)
```

Return Tokens distrubuted for one wallet

Name	Type	Description
[0]	uint256	Tokens distrubuted for one wallet

mint

```
function mint() public payable
```

Mints a Plant with start details

seed

```
function seed(uint256 tokenId) external payable
```

Seeding a Plant to start playing

Name	Type	Description
tokenId	uint256	Id of the token to seed

water

```
function water(uint256 tokenId) external payable
```

Watering a Plant

Name	Type	Description
tokenId	uint256	Id of the token to water

levelUp

```
function levelUp(uint256 tokenId) external
```

LevelUp a Plant

Name	Type	Description
tokenId	uint256	Id of the token to levelUp

harvest

```
function harvest(uint256 tokenId) external
```

harvest a Plant

Name	Type	Description
tokenId	uint256	Id of the token to harvest

sellTokens

```
function sellTokens(uint256 tokenAmount) external payable
```

Sells owned tokens and transfers funds to sender address

Name	Type	Description
tokenAmount	uint256	Amount of tokens to be sold

calculateTokenPrice

```
function calculateTokenPrice() public view returns (uint256)
```

Calculates the token price based on the current and potential supply

Name	Type	Description
[0]	uint256	The price of a single token

calculateTotalTokenSupplyPotential

```
function calculateTotalTokenSupplyPotential() public view returns (uint256)
```

Calculates the potential harvest amount for all tokens

Name	Type	Description
[0]	uint256	The potential harvest amount of all plants if all future actions are harvest

calculateTokenSupplyPotential

```
function calculateTokenSupplyPotential(uint256 tokenId) public view returns
(uint256)
```

Calculates the potential harvest amount for single token

Name	Type	Description
tokenId	uint256	Id of the token to harvest

Name	Type	Description
[0]	uint256	The potential harvest amount if all future actions are harvest

tokenURI

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

See {IERC721Metadata-tokenURI}.

PlantDrawer

Produces a string containing the data URI for a JSON metadata string

tokenURI

```
function tokenURI(contract IPlant plant, uint256 tokenId) public view returns
(string)
```

Produces the URI describing a particular Plant (token id)

Note this URI may be a data: URI with the JSON contents directly inlined

Name	Type	Description
plant	contract IPlant	The Plant contract
tokenId	uint256	The ID of the token for which to produce a description

Name	Type	Description
[0]	string	The URI of the ERC721-compliant metadata

generateName

```
function generateName(uint256 tokenId) private pure returns (string)
```

generate Json Metadata name

generateDescription

```
function generateDescription(address minter, uint256 timestamp) private pure
returns (string)
```

generate Json Metadata description

generateAttributes

```
function generateAttributes(struct IPlant.Details details) private pure returns
(string)
```

generate Json Metadata attributes

getJsonAttribute

```
function getJsonAttribute(string trait, string value, bool end) private pure
returns (string json)
```

Get the json attribute as { "trait_type": "Level", "value": "22" }

generateSVGImage

```
function generateSVGImage(struct IPlant.Details details) private pure returns
(string)
```

Combine all the SVGs to generate the final image

generateSVGHead

```
function generateSVGHead() private pure returns (string)
```

generate SVG header

generatePlant

```
function generatePlant(struct IPlant.Details details) private pure returns
(string)
```

generate SVG plant

generatePot

```
function generatePot() private pure returns (string)
```

generate SVG pot

IPlant

Details

```
struct Details {
  uint8 level;
  uint8 harvestCounter;
  uint8 wateringCounter;
  uint256 lastActionTimestamp;
  uint256 timestamp;
  address minter;
}
```

getDetails

```
function getDetails(uint256 tokenId) external view returns (struct IPlant.Details
details)
```

Returns the details associated with a given token ID.

Throws if the token ID is not valid.

Name	Type	Description
tokenId	uint256	The ID of the token that represents the Plant

Name	Type	Description
details	struct IPlant.Details	memory

IPlantDrawer

tokenURI

```
function tokenURI(contract IPlant plant, uint256 tokenId) external view returns
(string)
```

Produces the URI describing a particular Plant (token id)

Note this URI may be a data: URI with the JSON contents directly inlined

Name	Type	Description
plant	contract IPlant	The Plant contract
tokenId	uint256	The ID of the token for which to produce a description

Name	Type	Description
[0]	string	The URI of the ERC721-compliant metadata