======================================================================================================================

## **\*\*\*\* SAMPLE JAVASCRIPT CODE AND STRUCTURE BUILT \*\*\*\***

======================================================================================================================

await spCoinAddMethods.addAgentSponsorship(

SPONSOR\_ACCOUNT\_SIGNERS[0],

RECIPIENT\_ACCOUNT\_KEYS[1],

RECIPIENT\_RATES[5],

AGENT\_ACCOUNT\_KEYS[2],

AGENT\_RATES[9],

"1.000000000000000008"

);

await spCoinAddMethods.depositAgentStakingRewards(

RECIPIENT\_ACCOUNT\_KEYS[1],

AGENT\_ACCOUNT\_KEYS[2],

AGENT\_RATES[4],

444

);

======================================================================================================================

## **\*\*\*\* AS ASSIGNED IN SOLIDITY \*\*\*\***

======================================================================================================================

sponsorKey = 0xf39Fd6e51aad88F6F4ce6aB8827279cffFb92266

recipientKey = 0x70997970c51812dc3a010c7d01b50e0d17dc79c8

agentKey = 0x3c44cdddb6a900fa2b585dd299e03d12fa4293bc

rate = 4

amount = 444

rewardRateList.length = 1SOL=>2.6 rewardRateRecord.rate = 4

stakingRewards = 444

======================================================================================================================

## **\*\*\*\* RESULTING SOLIDITY DATA STRUCTURES \*\*\*\***

======================================================================================================================

struct AccountStruct { // **RECIPIENT ACCOUNT**

address accountKey; // **0x70997970c51812dc3a010c7d01b50e0d17dc79c8**

address[] recipientAccountList; // If Sponsor List of Recipient Account

address[] sponsorAccountList; // **[ 0xf39Fd6e51aad88F6F4ce6aB8827279cffFb92266 ]**

address[] agentAccountList; // If Recipient? List of Agent Account

address[] agentsParentRecipientAccountList; // If Agent? List of Recipient Accounts

mapping(string => RewardsStruct) rewardsMap; // **{ ALL\_REWARDS** **}**

}

struct RewardsStruct {

// contains Recipient Keys

mapping(address => RewardAccountStruct) agentRewardsMap; // { **0xf39Fd6e51aad88F6F4ce6aB8827279cffFb92266** };

}

struct RewardAccountStruct {

uint256[] rewardRateList; // [ **9** ];

mapping(uint256 => RewardRateStruct) rewardRateMap; // { **9** };

}

struct RewardRateStruct {

uint256 rate; // [4];

RewardsTransactionStruct[] rewardTransactionList; // [ **9** ]

}

struct RewardsTransactionStruct {

uint256 stakingRewards; // **999;**

}

# Add/Update Reward Transaction Algorithm

Case REWARD AGENT(\_amount, transactionDate) {

If (RewardDate > Agent transactionDate) {

Calculate Outstanding Agent Reward Rate Amount

Update Rate Record lastUpdate = transactionDate

Update (OutStandingAgentReward) with add Transaction(lastUpdate)

}

Update Amount

}

# Inheritance Order

spCoin.Sol

./spCoin.sol

./Token.sol

***./***rewardsManagement

./RewardsManager.sol

***.***/StakingManager.sol

./accounts

./UnSubscribe.sol

./Transactions.sol

./AgentRates.sol

./RecipientRates.sol

./Sponsor.sol

./Account.sol

./utils/StructSerialization.sol

./Utils.sol

./security.sol ./TimeUtils.sol

./SpCoinDataTypes.sol ./StringUtils.sol