

generate_candidate_account
- add_candidate(program_id: &Pubkey, accounts: &[AccountInfo], first_name: String, last_name: String) - initialize_candidate_account(pda_account: &AccountInfo, first_name: String, last_name: String)

generate_candidate_list_account
- generate_candidate_list_account(program_id: &Pubkey, accounts: &[AccountInfo], first_name: String, last_name: String name_vote_account: String)

generate_vote_account
- add_vote_account(program_id: &Pubkey, accounts: &[AccountInfo], name: String, start_date: NaiveDateTime, end_date: NaiveDateTime) - initialize_vote_account(pda_account: &AccountInfo, name: String, start_date: String, end_date: String) - try_update(program_id: &Pubkey, accounts: &[AccountInfo], name: String)

lib
- process_instruction(program_id: &Pubkey, accounts: &[AccountInfo], data_instructions: &[u8])

instruction
- pub enum ChainDemocracyInstruction{ AddVoteAccount, UpdateVoteAccount, AddCandidate } - impl ChainDemocracyInstruction{ pub fn unpack(input: &[u8])

vote_account_utilities
- check_dates(start_date: DateTime<Utc>, end_date: DateTime<Utc>)