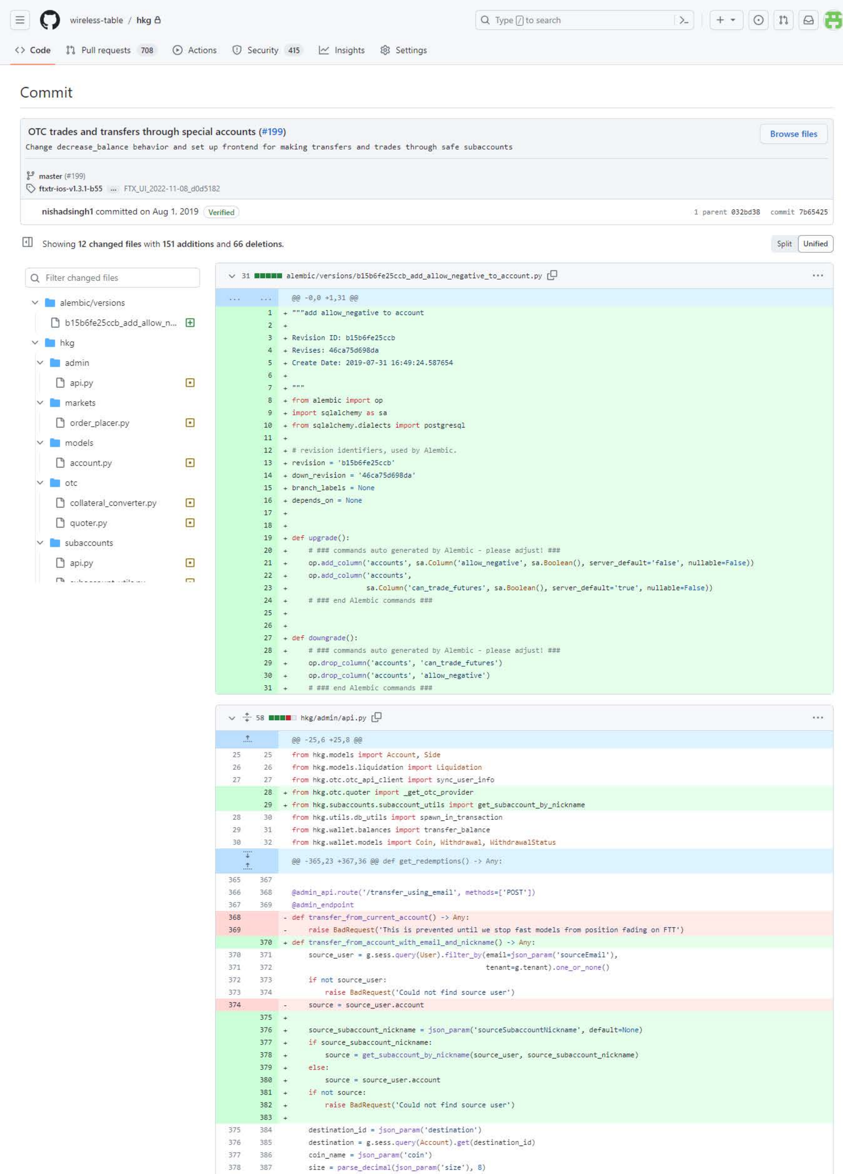


Commit

OTC trades and transfers through special accounts (#199)

`Change decrease_balance behavior and set up frontend for making transfers and trades through safe subaccounts`

master (#199)
ftxtr-ios-v1.3.1-b55 … FTX_UI_2022-11-08_d0d5182

nishadsingh1 committed on Aug 1, 2019 Verified

1 parent 032bd38 commit 7b65425

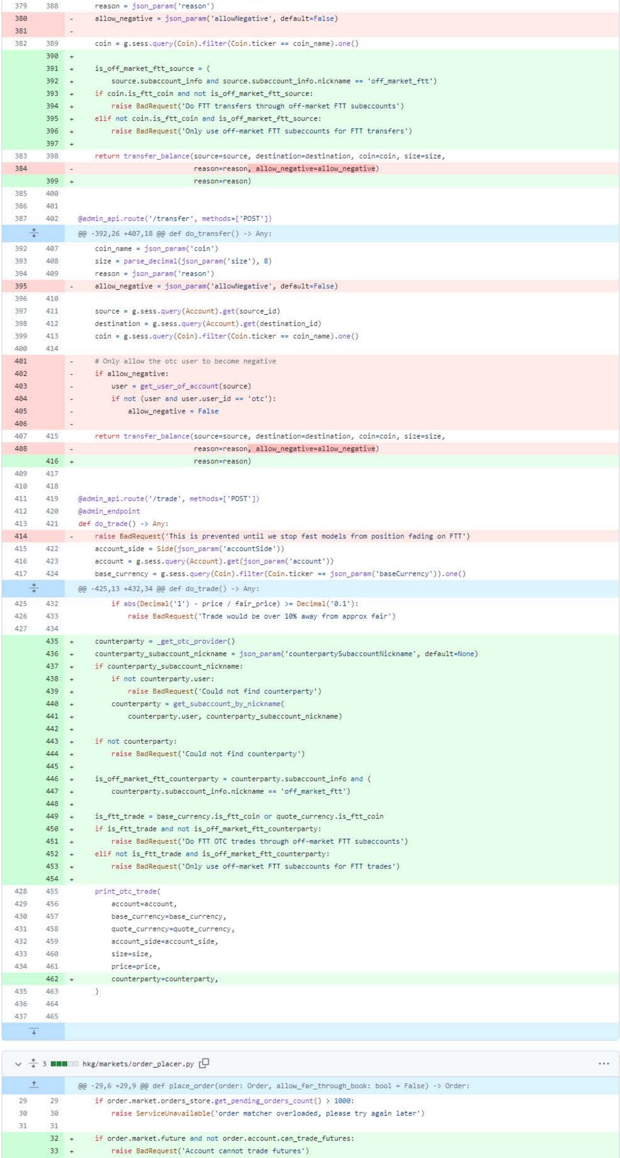
Showing **12 changed files** with **151 additions** and **66 deletions.**

31 alembic/versions/b15b6fe25ccb_add_allow_negative_to_account.py

```
@@ -0,0 +1,31 @@
+"""add allow_negative to account
+
+Revision ID: b15b6fe25ccb
+Revises: 46ca75d698da
+Create Date: 2019-07-31 16:49:24.587654
+
+"""
+from alembic import op
+import sqlalchemy as sa
+from sqlalchemy.dialects import postgresql
+
+# revision identifiers, used by Alembic.
+revision = 'b15b6fe25ccb'
+down_revision = '46ca75d698da'
+branch_labels = None
+depends_on = None
+
+
+def upgrade():
+    # ### commands auto generated by Alembic - please adjust! ###
+    op.add_column('accounts', sa.Column('allow_negative', sa.Boolean(), server_default='false', nullable=False))
+    op.add_column('accounts',
+                  sa.Column('can_trade_futures', sa.Boolean(), server_default='true', nullable=False))
+    # ### end Alembic commands ###
+
+
+def downgrade():
+    # ### commands auto generated by Alembic - please adjust! ###
+    op.drop_column('accounts', 'can_trade_futures')
+    op.drop_column('accounts', 'allow_negative')
+    # ### end Alembic commands ###
```

58 hkg/admin/api.py

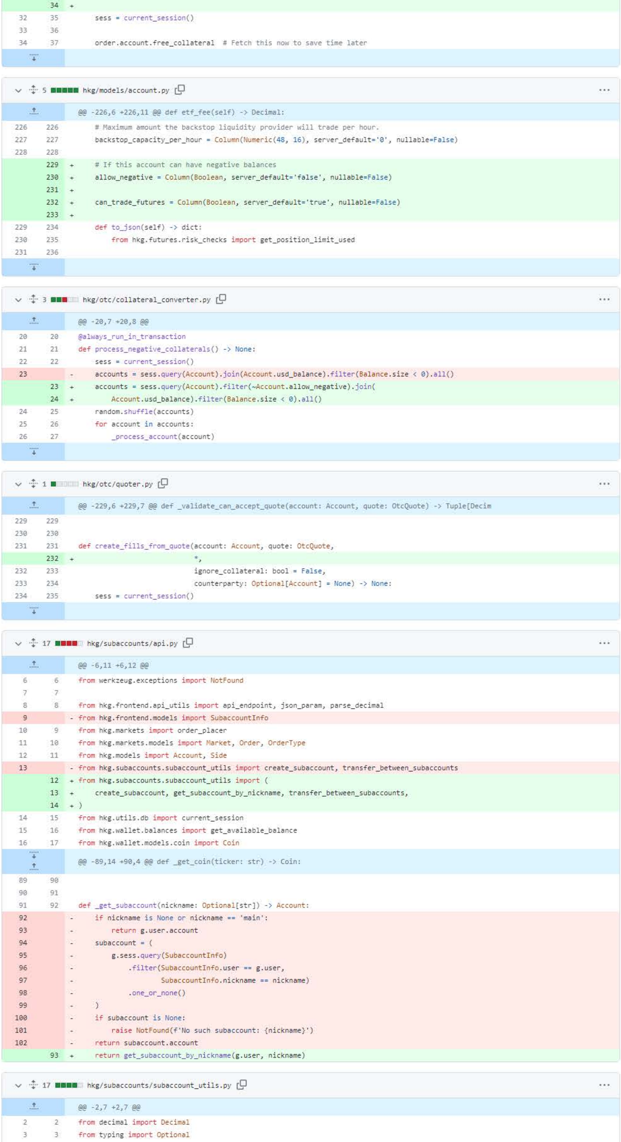
```
@@ -25,6 +25,8 @@
 from hkg.models import Account, Side
 from hkg.models.liquidation import Liquidation
 from hkg.otc.otc_api_client import sync_user_info
+from hkg.otc.quoter import _get_otc_provider
+from hkg.subaccounts.subaccount_utils import get_subaccount_by_nickname
 from hkg.utils.db_utils import spawn_in_transaction
 from hkg.wallet.balances import transfer_balance
 from hkg.wallet.models import Coin, Withdrawal, WithdrawalStatus
@@ -365,23 +367,36 @@ def get_redemptions() -> Any:
 
 @admin_api.route('/transfer_using_email', methods=['POST'])
 @admin_endpoint
-def transfer_from_current_account() -> Any:
-    raise BadRequest('This is prevented until we stop fast models from position fading on FTT')
+def transfer_from_account_with_email_and_nickname() -> Any:
     source_user = g.sess.query(User).filter_by(email=json_param('sourceEmail'),
                                                tenant=g.tenant).one_or_none()
     if not source_user:
         raise BadRequest('Could not find source user')
-    source = source_user.account
+
+    source_subaccount_nickname = json_param('sourceSubaccountNickname', default=None)
+    if source_subaccount_nickname:
+        source = get_subaccount_by_nickname(source_user, source_subaccount_nickname)
+    else:
+        source = source_user.account
+    if not source:
+        raise BadRequest('Could not find source user')
+
     destination_id = json_param('destination')
     destination = g.sess.query(Account).get(destination_id)
     coin_name = json_param('coin')
     size = parse_decimal(json_param('size'), 8)
```



```diff
     reason = json_param('reason')
-    allow_negative = json_param('allowNegative', default=False)
-
     coin = g.sess.query(Coin).filter(Coin.ticker == coin_name).one()
+
+    is_off_market_ftt_source = (
+        source.subaccount_info and source.subaccount_info.nickname == 'off_market_ftt')
+    if coin.is_ftt_coin and not is_off_market_ftt_source:
+        raise BadRequest('Do FTT transfers through off-market FTT subaccounts')
+    elif not coin.is_ftt_coin and is_off_market_ftt_source:
+        raise BadRequest('Only use off-market FTT subaccounts for FTT transfers')
+
     return transfer_balance(source=source, destination=destination, coin=coin, size=size,
-                            reason=reason, allow_negative=allow_negative)
+                            reason=reason)


 @admin_api.route('/transfer', methods=['POST'])
@@ -392,26 +407,18 @@ def do_transfer() -> Any:
     coin_name = json_param('coin')
     size = parse_decimal(json_param('size'), 8)
     reason = json_param('reason')
-    allow_negative = json_param('allowNegative', default=False)

     source = g.sess.query(Account).get(source_id)
     destination = g.sess.query(Account).get(destination_id)
     coin = g.sess.query(Coin).filter(Coin.ticker == coin_name).one()

-    # Only allow the otc user to become negative
-    if allow_negative:
-        user = get_user_of_account(source)
-        if not (user and user.user_id == 'otc'):
-            allow_negative = False
-
     return transfer_balance(source=source, destination=destination, coin=coin, size=size,
-                            reason=reason, allow_negative=allow_negative)
+                            reason=reason)


 @admin_api.route('/trade', methods=['POST'])
 @admin_endpoint
 def do_trade() -> Any:
-    raise BadRequest('This is prevented until we stop fast models from position fading on FTT')
     account_side = Side(json_param('accountSide'))
     account = g.sess.query(Account).get(json_param('account'))
     base_currency = g.sess.query(Coin).filter(Coin.ticker == json_param('baseCurrency')).one()
@@ -425,13 +432,34 @@ def do_trade() -> Any:
         if abs(Decimal('1') - price / fair_price) >= Decimal('0.1'):
             raise BadRequest('Trade would be over 10% away from approx fair')

+    counterparty = _get_otc_provider()
+    counterparty_subaccount_nickname = json_param('counterpartySubaccountNickname', default=None)
+    if counterparty_subaccount_nickname:
+        if not counterparty.user:
+            raise BadRequest('Could not find counterparty')
+        counterparty = get_subaccount_by_nickname(
+            counterparty.user, counterparty_subaccount_nickname)
+
+    if not counterparty:
+        raise BadRequest('Could not find counterparty')
+
+    is_off_market_ftt_counterparty = counterparty.subaccount_info and (
+        counterparty.subaccount_info.nickname == 'off_market_ftt')
+
+    is_ftt_trade = base_currency.is_ftt_coin or quote_currency.is_ftt_coin
+    if is_ftt_trade and not is_off_market_ftt_counterparty:
+        raise BadRequest('Do FTT OTC trades through off-market FTT subaccounts')
+    elif not is_ftt_trade and is_off_market_ftt_counterparty:
+        raise BadRequest('Only use off-market FTT subaccounts for FTT trades')
+
     print_otc_trade(
         account=account,
         base_currency=base_currency,
         quote_currency=quote_currency,
         account_side=account_side,
         size=size,
         price=price,
+        counterparty=counterparty,
     )


```

3 hkg/markets/order_placer.py

```diff
@@ -29,6 +29,9 @@ def place_order(order: Order, allow_far_through_book: bool = False) -> Order:
     if order.market.orders_store.get_pending_orders_count() > 1000:
         raise ServiceUnavailable('order matcher overloaded, please try again later')

+    if order.market.future and not order.account.can_trade_futures:
+        raise BadRequest('Account cannot trade futures')
```



```diff
+
     sess = current_session()
 
     order.account.free_collateral  # Fetch this now to save time later
```

5 🟩🟩🟩🟩🟩 hkg/models/account.py

```diff
@@ -226,6 +226,11 @@ def etf_fee(self) -> Decimal:
     # Maximum amount the backstop liquidity provider will trade per hour.
     backstop_capacity_per_hour = Column(Numeric(48, 16), server_default='0', nullable=False)
 
+    # If this account can have negative balances
+    allow_negative = Column(Boolean, server_default='false', nullable=False)
+
+    can_trade_futures = Column(Boolean, server_default='true', nullable=False)
+
     def to_json(self) -> dict:
         from hkg.futures.risk_checks import get_position_limit_used
 
```

3 🟩🟩🟥⬜⬜ hkg/otc/collateral_converter.py

```diff
@@ -20,7 +20,8 @@
 @always_run_in_transaction
 def process_negative_collaterals() -> None:
     sess = current_session()
-    accounts = sess.query(Account).join(Account.usd_balance).filter(Balance.size < 0).all()
+    accounts = sess.query(Account).filter(~Account.allow_negative).join(
+        Account.usd_balance).filter(Balance.size < 0).all()
     random.shuffle(accounts)
     for account in accounts:
         _process_account(account)
```

1 🟩⬜⬜⬜⬜ hkg/otc/quoter.py

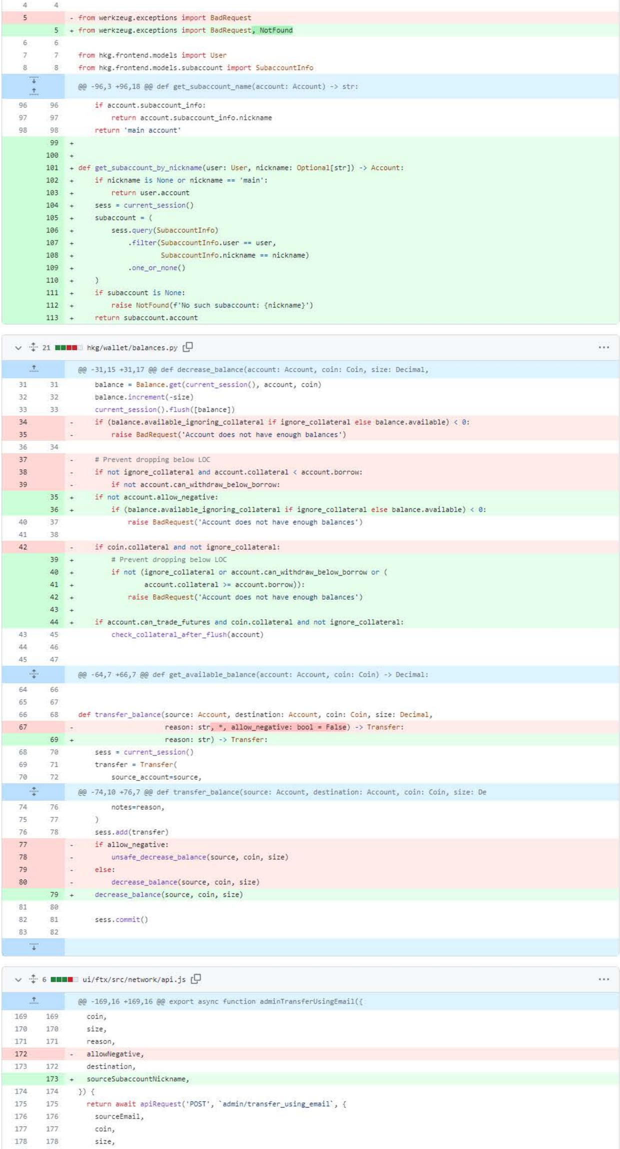
```diff
@@ -229,6 +229,7 @@ def _validate_can_accept_quote(account: Account, quote: OtcQuote) -> Tuple[Decim
 
 
 def create_fills_from_quote(account: Account, quote: OtcQuote,
+                            *,
                             ignore_collateral: bool = False,
                             counterparty: Optional[Account] = None) -> None:
     sess = current_session()
```

17 🟩🟥🟥🟥⬜ hkg/subaccounts/api.py

```diff
@@ -6,11 +6,12 @@
 from werkzeug.exceptions import NotFound
 
 from hkg.frontend.api_utils import api_endpoint, json_param, parse_decimal
-from hkg.frontend.models import SubaccountInfo
 from hkg.markets import order_placer
 from hkg.markets.models import Market, Order, OrderType
 from hkg.models import Account, Side
-from hkg.subaccounts.subaccount_utils import create_subaccount, transfer_between_subaccounts
+from hkg.subaccounts.subaccount_utils import (
+    create_subaccount, get_subaccount_by_nickname, transfer_between_subaccounts,
+)
 from hkg.utils.db import current_session
 from hkg.wallet.balances import get_available_balance
 from hkg.wallet.models.coin import Coin
@@ -89,14 +90,4 @@ def _get_coin(ticker: str) -> Coin:
 
 
 def _get_subaccount(nickname: Optional[str]) -> Account:
-    if nickname is None or nickname == 'main':
-        return g.user.account
-    subaccount = (
-        g.sess.query(SubaccountInfo)
-            .filter(SubaccountInfo.user == g.user,
-                    SubaccountInfo.nickname == nickname)
-            .one_or_none()
-    )
-    if subaccount is None:
-        raise NotFound(f'No such subaccount: {nickname}')
-    return subaccount.account
+    return get_subaccount_by_nickname(g.user, nickname)
```

17 🟩🟩🟩🟩⬜ hkg/subaccounts/subaccount_utils.py

```diff
@@ -2,7 +2,7 @@
 from decimal import Decimal
 from typing import Optional
```



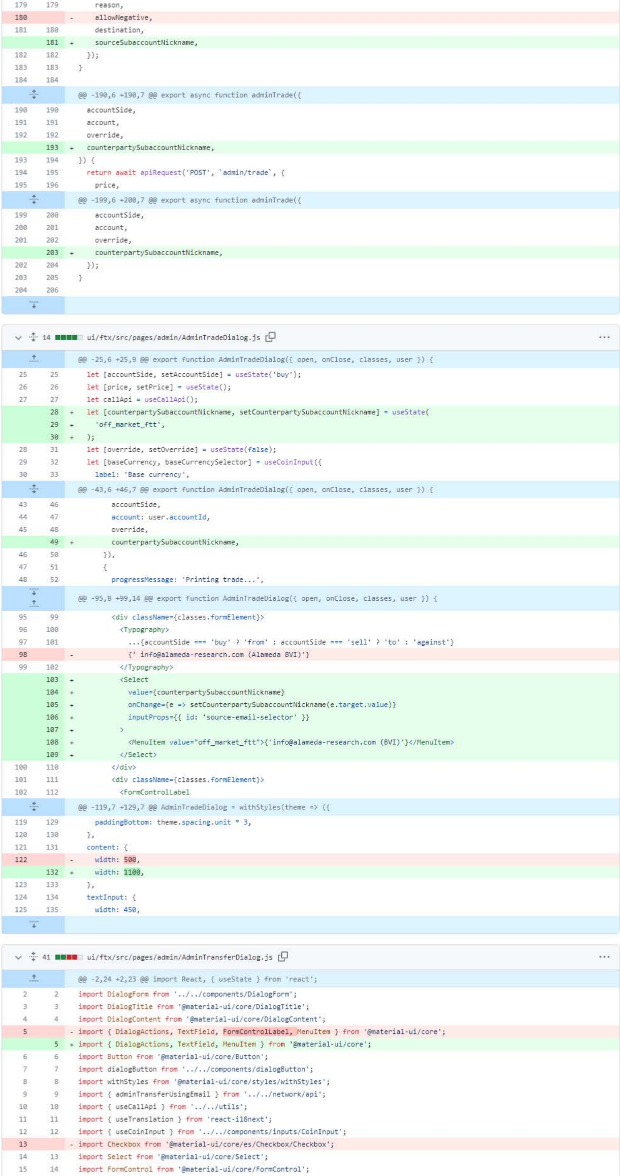
```diff
 
-from werkzeug.exceptions import BadRequest
+from werkzeug.exceptions import BadRequest, NotFound
 
 from hkg.frontend.models import User
 from hkg.frontend.models.subaccount import SubaccountInfo
@@ -96,3 +96,18 @@ def get_subaccount_name(account: Account) -> str:
     if account.subaccount_info:
         return account.subaccount_info.nickname
     return 'main account'
+
+
+def get_subaccount_by_nickname(user: User, nickname: Optional[str]) -> Account:
+    if nickname is None or nickname == 'main':
+        return user.account
+    sess = current_session()
+    subaccount = (
+        sess.query(SubaccountInfo)
+            .filter(SubaccountInfo.user == user,
+                    SubaccountInfo.nickname == nickname)
+            .one_or_none()
+    )
+    if subaccount is None:
+        raise NotFound(f'No such subaccount: {nickname}')
+    return subaccount.account
```

21 hkg/wallet/balances.py

```diff
@@ -31,15 +31,17 @@ def decrease_balance(account: Account, coin: Coin, size: Decimal,
     balance = Balance.get(current_session(), account, coin)
     balance.increment(-size)
     current_session().flush([balance])
-    if (balance.available_ignoring_collateral if ignore_collateral else balance.available) < 0:
-        raise BadRequest('Account does not have enough balances')
 
-    # Prevent dropping below LOC
-    if not ignore_collateral and account.collateral < account.borrow:
-        if not account.can_withdraw_below_borrow:
+    if not account.allow_negative:
+        if (balance.available_ignoring_collateral if ignore_collateral else balance.available) < 0:
             raise BadRequest('Account does not have enough balances')
 
-    if coin.collateral and not ignore_collateral:
+        # Prevent dropping below LOC
+        if not (ignore_collateral or account.can_withdraw_below_borrow or (
+                account.collateral >= account.borrow)):
+            raise BadRequest('Account does not have enough balances')
+
+    if account.can_trade_futures and coin.collateral and not ignore_collateral:
         check_collateral_after_flush(account)
 
 
@@ -64,7 +66,7 @@ def get_available_balance(account: Account, coin: Coin) -> Decimal:
 
 
 def transfer_balance(source: Account, destination: Account, coin: Coin, size: Decimal,
-                     reason: str, *, allow_negative: bool = False) -> Transfer:
+                     reason: str) -> Transfer:
     sess = current_session()
     transfer = Transfer(
         source_account=source,
@@ -74,10 +76,7 @@ def transfer_balance(source: Account, destination: Account, coin: Coin, size: De
         notes=reason,
     )
     sess.add(transfer)
-    if allow_negative:
-        unsafe_decrease_balance(source, coin, size)
-    else:
-        decrease_balance(source, coin, size)
+    decrease_balance(source, coin, size)
 
     sess.commit()
 
```

6 ui/ftx/src/network/api.js

```diff
@@ -169,16 +169,16 @@ export async function adminTransferUsingEmail({
   coin,
   size,
   reason,
-  allowNegative,
   destination,
+  sourceSubaccountNickname,
 }) {
   return await apiRequest('POST', `admin/transfer_using_email`, {
     sourceEmail,
     coin,
     size,
```



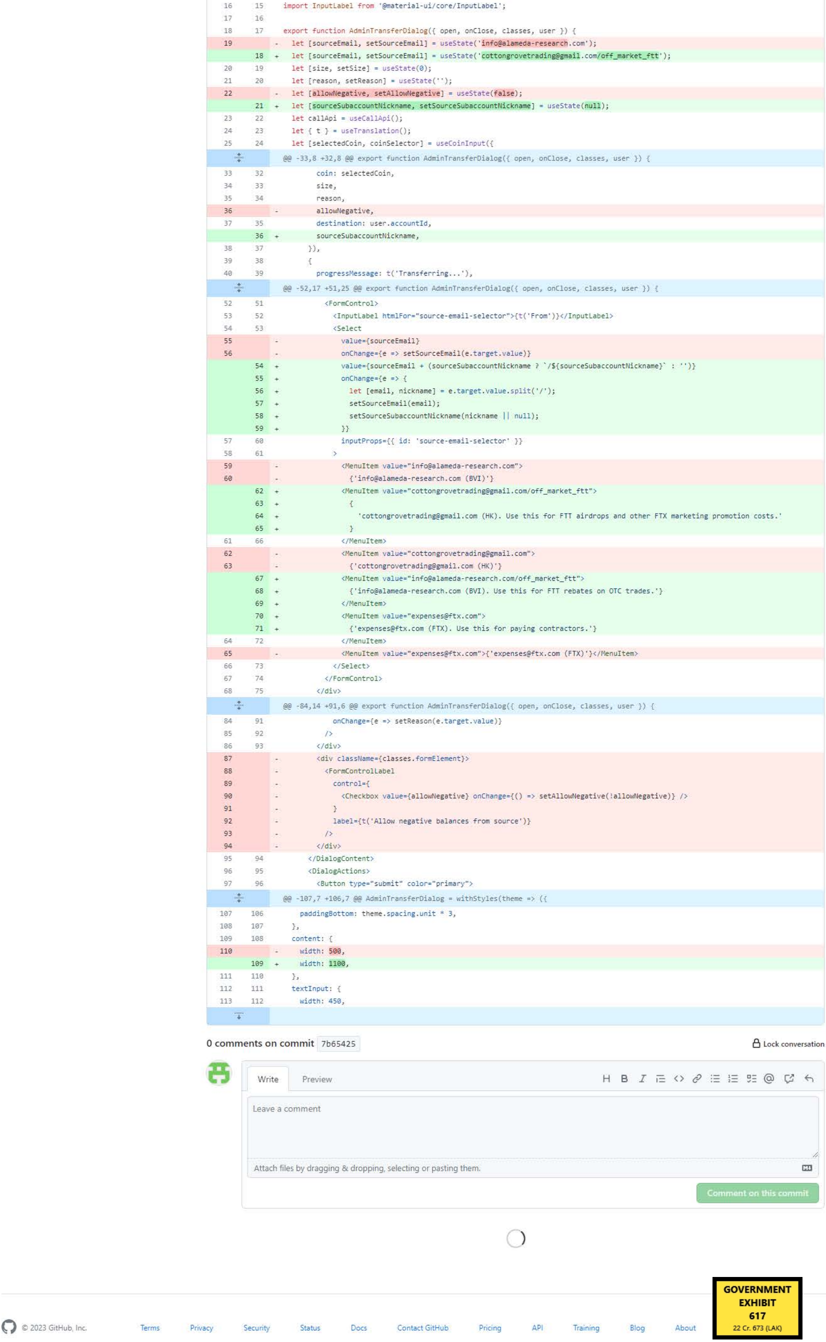
```diff
     reason,
-    allowNegative,
     destination,
+    sourceSubaccountNickname,
   });
 }

@@ -190,6 +190,7 @@ export async function adminTrade({
   accountSide,
   account,
   override,
+  counterpartySubaccountNickname,
 }) {
   return await apiRequest('POST', `admin/trade`, {
     price,
@@ -199,6 +200,7 @@ export async function adminTrade({
     accountSide,
     account,
     override,
+    counterpartySubaccountNickname,
   });
 }

```

14 ui/ftx/src/pages/admin/AdminTradeDialog.js

```diff
@@ -25,6 +25,9 @@ export function AdminTradeDialog({ open, onClose, classes, user }) {
   let [accountSide, setAccountSide] = useState('buy');
   let [price, setPrice] = useState();
   let callApi = useCallApi();
+  let [counterpartySubaccountNickname, setCounterpartySubaccountNickname] = useState(
+    'off_market_ftt',
+  );
   let [override, setOverride] = useState(false);
   let [baseCurrency, baseCurrencySelector] = useCoinInput({
     label: 'Base currency',
@@ -43,6 +46,7 @@ export function AdminTradeDialog({ open, onClose, classes, user }) {
         accountSide,
         account: user.accountId,
         override,
+        counterpartySubaccountNickname,
       }),
       {
         progressMessage: 'Printing trade...',
@@ -95,8 +99,14 @@ export function AdminTradeDialog({ open, onClose, classes, user }) {
         <div className={classes.formElement}>
           <Typography>
             ...{accountSide === 'buy' ? 'from' : accountSide === 'sell' ? 'to' : 'against'}
-            {' info@alameda-research.com (Alameda BVI)'}
           </Typography>
+          <Select
+            value={counterpartySubaccountNickname}
+            onChange={e => setCounterpartySubaccountNickname(e.target.value)}
+            inputProps={{ id: 'source-email-selector' }}
+          >
+            <MenuItem value="off_market_ftt">{'info@alameda-research.com (BVI)'}</MenuItem>
+          </Select>
         </div>
         <div className={classes.formElement}>
           <FormControlLabel
@@ -119,7 +129,7 @@ AdminTradeDialog = withStyles(theme => ({
     paddingBottom: theme.spacing.unit * 3,
   },
   content: {
-    width: 500,
+    width: 1100,
   },
   textInput: {
     width: 450,
```

41 ui/ftx/src/pages/admin/AdminTransferDialog.js

```diff
@@ -2,24 +2,23 @@ import React, { useState } from 'react';
 import DialogForm from '../../components/DialogForm';
 import DialogTitle from '@material-ui/core/DialogTitle';
 import DialogContent from '@material-ui/core/DialogContent';
-import { DialogActions, TextField, FormControlLabel, MenuItem } from '@material-ui/core';
+import { DialogActions, TextField, MenuItem } from '@material-ui/core';
 import Button from '@material-ui/core/Button';
 import dialogButton from '../../components/dialogButton';
 import withStyles from '@material-ui/core/styles/withStyles';
 import { adminTransferUsingEmail } from '../../network/api';
 import { useCallApi } from '../../utils';
 import { useTranslation } from 'react-i18next';
 import { useCoinInput } from '../../components/inputs/CoinInput';
-import Checkbox from '@material-ui/core/es/Checkbox/Checkbox';
 import Select from '@material-ui/core/Select';
 import FormControl from '@material-ui/core/FormControl';
```



```diff
  import InputLabel from '@material-ui/core/InputLabel';

  export function AdminTransferDialog({ open, onClose, classes, user }) {
-   let [sourceEmail, setSourceEmail] = useState('info@alameda-research.com');
+   let [sourceEmail, setSourceEmail] = useState('cottongrovetrading@gmail.com/off_market_ftt');
    let [size, setSize] = useState(0);
    let [reason, setReason] = useState('');
-   let [allowNegative, setAllowNegative] = useState(false);
+   let [sourceSubaccountNickname, setSourceSubaccountNickname] = useState(null);
    let callApi = useCallApi();
    let { t } = useTranslation();
    let [selectedCoin, coinSelector] = useCoinInput({
@@ -33,8 +32,8 @@ export function AdminTransferDialog({ open, onClose, classes, user }) {
          coin: selectedCoin,
          size,
          reason,
-         allowNegative,
          destination: user.accountId,
+         sourceSubaccountNickname,
        }),
        {
          progressMessage: t('Transferring...'),
@@ -52,17 +51,25 @@ export function AdminTransferDialog({ open, onClose, classes, user }) {
            <FormControl>
              <InputLabel htmlFor="source-email-selector">{t('From')}</InputLabel>
              <Select
-               value={sourceEmail}
-               onChange={e => setSourceEmail(e.target.value)}
+               value={sourceEmail + (sourceSubaccountNickname ? `/${sourceSubaccountNickname}` : '')}
+               onChange={e => {
+                 let [email, nickname] = e.target.value.split('/');
+                 setSourceEmail(email);
+                 setSourceSubaccountNickname(nickname || null);
+               }}
                inputProps={{ id: 'source-email-selector' }}
              >
-               <MenuItem value="info@alameda-research.com">
-                 {'info@alameda-research.com (BVI)'}
+               <MenuItem value="cottongrovetrading@gmail.com/off_market_ftt">
+                 {
+                   'cottongrovetrading@gmail.com (HK). Use this for FTT airdrops and other FTX marketing promotion costs.'
+                 }
               </MenuItem>
-               <MenuItem value="cottongrovetrading@gmail.com">
-                 {'cottongrovetrading@gmail.com (HK)'}
+               <MenuItem value="info@alameda-research.com/off_market_ftt">
+                 {'info@alameda-research.com (BVI). Use this for FTT rebates on OTC trades.'}
+               </MenuItem>
+               <MenuItem value="expenses@ftx.com">
+                 {'expenses@ftx.com (FTX). Use this for paying contractors.'}
               </MenuItem>
-               <MenuItem value="expenses@ftx.com">{'expenses@ftx.com (FTX)'}</MenuItem>
             </Select>
           </FormControl>
         </div>
@@ -84,14 +91,6 @@ export function AdminTransferDialog({ open, onClose, classes, user }) {
             onChange={e => setReason(e.target.value)}
           />
         </div>
-         <div className={classes.formElement}>
-           <FormControlLabel
-             control={
-               <Checkbox value={allowNegative} onChange={() => setAllowNegative(!allowNegative)} />
-             }
-             label={t('Allow negative balances from source')}
-           />
-         </div>
       </DialogContent>
       <DialogActions>
         <Button type="submit" color="primary">
@@ -107,7 +106,7 @@ AdminTransferDialog = withStyles(theme => ({
      paddingBottom: theme.spacing.unit * 3,
    },
    content: {
-     width: 500,
+     width: 1100,
    },
    textInput: {
      width: 450,
```

0 comments on commit 7b65425

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

© 2023 GitHub, Inc.

GOVERNMENT EXHIBIT 617
22 Cr. 673 (LAK)