

Green Deal - Demo

Video tutorial [found here](#). (Abridged version of example below.)

Contract hash on NEO and CoZ testnet.

```
7f1fd78c94bf8509947fce77eb275c99360e9756
```

Example Usage

Below is a demo scenario of how buyers and sellers would use Green Deal. Further documentation for each function and parameters can be found in [README](#).

Step 1: As a seller

Create the promo for people to purchase.

1. Open my seller wallet

```
open wallet demo_seller.wallet
```

2. Get wallet public key using `wallet`

```
<creator_public_key>
```

3. Create promo called `promo_code`. Once created, `promo_code` phrase cannot be used by other promos unless deleted by creator. Here we have an ice cream sale, each ticket costs 3 gas, expires on Jan 1, 2018 (unix time of 1546300800), need to sell minimum of 5 tickets, and maximum of 8.

```
testinvoke 7f1fd78c94bf8509947fce77eb275c99360e9756 create ['<creator_public_key>', 'promo_code', 'Opening-day-sale-for-ice-cream!', 'Discount-for-any-flavor', 3, 1546300800, 5, 8]
```

Step 1.a (Optional) : Get details

If you only know the `promo_code` but want to find out more like how many tickets have already been sold or the price.

```
> testinvoke 7f1fd78c94bf8509947fce77eb275c99360e9756 details ['promo_code']
```

Step 1.b (Optional): Delete promo

For this tutorial, don't actually delete the promo because then we won't be able to showcase the `buy` function. But this function does exist.

```
> testinvoke 7f1fd78c94bf8509947fce77eb275c99360e9756 delete ['promo_code']
```

Step 2: As buyer_A

Say we're a person called `buyer_A` and want to purchase 2 tickets for `promo_code`.

1. Open `buyer_A` wallet

```
open wallet demo_buyer_a.wallet
```

2. Get buyer_A public key using `wallet`

```
<buyer_a_public_key>
```

3. Buy 2 tickets, each costs 3 gas so I attach a total of 6 gas

```
testinvoke 7f1fd78c94bf8509947fce77eb275c99360e9756 buy ['<buyer_a_public_key>','promo_code',2] --attach-gas=6
```

Step 3: As buyer_B

2 tickets still not enough for the promo to take effect, we need at least 5 tickets sold. So now let's say buyer_B and wants to purchase 4 tickets for `promo_code`.

1. Open buyer_B wallet

```
open wallet demo_buyer_b.wallet
```

2. Get buyer_B public key using `wallet`

```
<buyer_b_public_key>
```

3. Buy 4 tickets, each costs 3 gas so I attach a total of 12 gas

```
testinvoke 7f1fd78c94bf8509947fce77eb275c99360e9756 buy ['<buyer_b_public_key>','promo_code',4] --attach-gas=12
```

Step 3.a (Optional) Get refund

You can claim a refund on your purchase if you change your mind. We won't actually get a refund for this demo because we wouldn't have enough purchases to showcase the `claim` feature.

```
> `testinvoke 7f1fd78c94bf8509947fce77eb275c99360e9756 refund ['<buyer_b_public_key>','promo_code']`
```

Step 4: Promo has expired by now

To try out the `claim` feature as a seller, you should the expiration timestamp in `create` function to something short like 5 minutes. (The timestamp used in Step 1 is unix time for Jan 1, 2018.)

Step 5: As a seller

1. Claim my funds since minimum number of tickets has been sold.

```
testinvoke 7f1fd78c94bf8509947fce77eb275c99360e9756 claim ['promo_code']
```