

Recurring Dynamic PullPayment

Introduction

The Recurring Dynamic PullPayment simplifies the recurring payments that customers need to do on a regular interval to avail the services. This contract allows customers to subscribe to the billing model and start the recurring payments. It also provides the free trial and paid trial of the billing model.

Using this contract merchant can create the billing model by providing the payee address and recurring pullPayment type i.e paid trial, free trial or normal recurring pullPayment. The rest of the details are provided dynamically when the customer subscribes to the billing model.

Contract version

pragma solidity 0.8.0

Contract constructor

In this contract the initialize() method is used in place of the constructor to allow the contract to be upgradable via proxy.

Method Name:

- initialize()

Method Usage:

- The initialize method is used to initialize the Recurring PullPayment contract and set the registry address.

Method Signature:

- function initialize(address registryAddress) external;

Method Parameters:

1. **registryAddress:**
 - indicates the address of the registry contract which keeps the record of all the contract addresses in the ecosystem.

Contract Methods

createBillingModel()

Method name:

- createBillingModel

Method detail:

- This method allows merchants to create a new billing model for their business.
- The unique Id is created for each billing model.
- Merchant can create as many billing models as he wants.
- In this contract, merchants can specify the trial period and the type of the recurring pullPayment.
- The rest of the details are provided dynamically to the customer through the code-snippet.

Method signature:

- ```
function createBillingModel(
 address _payee,
 uint256 _recurringPPType
) public override returns (uint256 billingModelID)
```

Method parameters:

- 1. \_payee:**  
Indicates the address of the receiver(merchant) who will receive the pull payments.
- 2. \_recurringPPType:**  
Indicates the type of the recurring pullPayment. Here are the ids of recurring pullPayments-
  - 1- Normal Recurring pullPayment
  - 2- Free Trial Recurring PullPayment
  - 3- Paid Trial Recurring PullPayment

Returned data:

- This method returns the unique billing model Id.

# subscribeToBillingModel()

Method name:

- subscribeToBillingmodel

Method detail:

- This method allows any customer to subscribe to the billing model with given billing model Id and payment token.
- The remaining details of the billing model are provided dynamically through the code-snippet.
- The remaining detail includes the billing model name, settlement token, payment amount, frequency of payments, total number of payments, the trial period and the initial amount for the trial period.
- Here, trial period value needs to be specified in case of a free trial and paid trial contract and the initial amount needs to be specified only in case of paid trial pullPayment.
- In this, the type of the recurring pull payment is identified with the help of billing model id.
- The user must approve the payment tokens to the Executor contract before subscribing to the billing model.
- The unique subscription id is created on each subscription.

Method signature:

- ```
function subscribeToBillingModel(
    uint256 _billingModelID,
    bytes32 _bmName,
    address _settlementToken,
    address _paymentToken,
    uint256 _paymentAmount,
    uint256 _frequency,
    uint256 _totalPayments,
    uint256 _trialPeriod,
    uint256 _initialAMount
)
    external override returns (uint256 subscriptionID)
```

Method parameters:

- **_billingModelID:**
 - Indicates the unique billing model id that the customer wants to subscribe.
- **_bmName:**
 - Indicates the name of the billing model.
- **_settlementToken:**
 - Indicates the settlement token address in which the payee wants to receive the payment.
- **_paymentToken:**

- Indicates the payment token address in which the customer wants to pay for the subscription after trial period.
- **_paymentAmount:**
 - Indicates the billing model amount which will be charged during each pullPayment
- **_frequency:**
 - Indicates the time interval in seconds after which the pullPayment is executed.
- **_totalPayments:**
 - Indicates the total number of pullPayments that customer needs to complete after the fixed interval of time.
- **_trialPeriod:**
 - Indicates the trial period in seconds for the subscription. Trial period is specified only for free trial and paid trial of recurring pullPayments.
- **_initialAMount:**
 - Indicates the initial amount that will be charged for the paid trial type of recurring pullPayment.

Returned data:

- This method returns the unique subscription id after subscription.

executePullPayment()

Method name:

- executePullPayment

Method detail:

- This method allows anyone to execute the pullPayment for the given subscription id.
- It ensures pullPayment execution time is correct i.e trial period has ended, subscription is not canceled and no. of payments are not exceeded.
- Internally this method calls the execute method of the Executor contract to transfer the tokens from customer to merchant.
- It updates the next payment timestamp, number of payments remaining, pullPayment execution timestamp and last payment timestamp.
- The unique pullPayment id is created for each pullPayment.

Method signature:

- function executePullPayment(uint256 _subscriptionID)
public override returns (uint256 pullPaymentID)

Method parameters:

- **_subscriptionID:**
 - Indicates the unique subscription id.

Returned data:

- This method returns the unique pullPayment id after pullPayment execution.

cancelSubscription()

Method name:

- `cancelSubscription`

Method detail:

- This method allows subscribers to cancel the subscription.
- it updates the cancel timestamp to the current time and makes the subscription inactive.

Method signature:

- `function cancelSubscription(uint256 _subscriptionID)`
public override returns (uint256 subscriptionID)

Method parameters:

- **`_subscriptionID:`**
 - Indicates the unique subscription id.

Returned data:

- This method returns the id of the canceled subscription.

editBillingModel()

Method name:

- editBillingModel

Method detail:

- This method allows the billing model creator i.e payee to edit the details of the billing model.
- It allows payee to update the billing model payee address only.
- Only the owner of the billing model can edit the billing model details.

Method signature:

- ```
function editBillingModel(
 uint256 _billingModelID,
 address _newPayee
) public override returns (uint256 billingModelID)
```

Method parameters:

- **\_billingModelID:**
  - Indicates the unique valid billing model id which merchant wants to edit.
- **\_newPayee:**
  - Indicates the new payee address for the billing model.
  - Merchant cannot use Zero address as payee address.

Returned data:

- This method returns the id of the billing model whose details are updated.

# getBillingModel()

Method name:

- `getBillingModel`

Method detail:

- This method retrieves all the details of the specified billing model that is stored on blockchain while creating the billing model. i.e payee address and recurring pullPayment type.
- Only the payee of a particular billing model can get the subscription ids of the billing model.

Method signature:

- `function getBillingModel(uint256 _billingModelID)`  
external  
view  
override  
returns (BillingModelData memory bm)

Method parameters:

- **`_billingModelID:`**
  - Indicates the valid billing model id.

Returned data:

- This method returns the details of the billing model which includes the-
  - Payee address
  - Type of recurring pullPayment. i.e
    - 1- normal recurring pullPayment
    - 2- free trial recurring pullPayment
    - 3- paid trial recurring pullPayment



# getSubscription()

Method name:

- getSubscription

Method detail:

- This method returns the subscription data for given subscription id.
- Only the payee and subscriber of the billing model can see the pullPayment ids of the subscription.

Method Signature:

- function getSubscription(uint256 \_subscriptionID)  
external  
view  
override  
returns (SubscriptionData memory sb)

Method parameters:

- **\_subscriptionID:**
  - Indicates the valid subscription id

Returned data:

- This method returns the subscription data which includes the following data-
  - Subscriber address
  - Billing model name
  - Payment amount
  - Settlement token address
  - Payment token address
  - Frequency of the payments
  - Total number of payments
  - Remaining number of payments
  - Start timestamp of subscription
  - Canceled timestamp of subscription
  - NextPayment timestamp
  - LastPayment timestamp
  - PullPayment IDs(only payee or subscriber gets the ids)

## getPullPayment()

Method name:

- getPullPayment

Method detail:

- This method retrieves the pullPayment details for given pullPayment id.
- It gives the payment amount and the pullPayment timestamp.
- Only the payee and the subscriber of pullPayment can see the payment amount and execution timestamp.

Method signature:

- function getPullPayment(uint256 \_pullPaymentID)  
    external  
    view  
returns (PullPaymentData memory pullPayment)

Method parameters:

- **\_pullPaymentID:**
  - Indicates the valid pullPayment id.

Returned data:

- This method returns the pullPayment data for given pullPayment id which includes-
  - Payment amount
  - Timestamp when pullPayment executed.

## getBillingModelIdsByAddress()

Method name:

- `getBillingModelIdsByAddress`

Method detail:

- This method retrieves the billing model ids for a given creator address.

Method signature:

- `function getBillingModelIdsByAddress(address _creator)`  
external  
view  
returns (uint256[] memory billingModelIDs)

Method parameters:

- **\_creator:**
  - Indicates the billing model creator address whose billing model id is to retrieve.

Returned data:

- This method returns the lists of billing model ids that a creator has created.

## getSubscriptionIdsByAddress()

Method name:

- `getSubscriptionIdsByAddress`

Method detail:

- This method retrieves the subscription ids for a given subscriber address.

Method signature:

- `function getSubscriptionIdsByAddress(address _subscriber)`  
external  
view  
returns (uint256[] memory subscriptionIDs)

Method parameters:

- **\_subscriber:**
  - Indicates the subscriber address whose subscription id is to retrieve.

Returned data:

- This method returns the lists of subscription ids that a subscriber has subscribed.

## getCanceledSubscriptionIdsByAddress()

Method name:

- `getCanceledSubscriptionIdsByAddress`

Method detail:

- This method retrieves the subscription ids which subscriber has canceled.

Method signature:

- `function getCanceledSubscriptionIdsByAddress(address _subscriber)`  
external  
view  
returns (uint256[] memory subscriptionIDs)

Method parameters:

- **\_subscriber:**
  - Indicates the subscriber address whose canceled subscription ids to be retrieved.

Returned data:

- This method returns the list of subscription ids which the subscriber has canceled.

## getPullPaymentsIdsByAddress()

Method name:

- `getPullPaymentsIdsByAddress`

Method detail:

- This method retrieves the list of pullPayment ids for a given subscriber.

Method signature:

- `function getPullPaymentsIdsByAddress(address _subscriber)`  
external  
view  
returns (uint256[] memory pullPaymentIDs)

Method parameters:

- **\_subscriber:**
  - Indicates the subscriber address whose pullPayment ids to be retrieved.

Returned data:

- This method returns a list of pullPayment ids for a given subscriber.

# Events

## BillingModelCreated

Event Name:

- BillingModelCreated

Event detail:

- This event is emitted when the creator creates a new billing model using **createBillingModel()** method.

Event signature:

- event BillingModelCreated(  
    uint256 indexed billingModelID,  
    address indexed payee,  
    uint8 indexed recurringPPType  
);

Event data:

- **billingModelID:**
  - Indicates the newly created billing model id.
- **Payee:**
  - Indicates the payee address who created the billing model.
- **recurringPPType:**
  - Indicates the type of recurring pullPayment i.e free trial, paid trial and normal recurring pullPayment.

# NewSubscription

Event name:

- NewSubscription

Event detail:

- This event is emitted when a subscriber subscribes to the billing model using **subscribeBillingModel()** method.

Event signature:

```
event NewSubscription(
 uint256 indexed billingModelID,
 uint256 indexed subscriptionID,
 address indexed payer
);
```

Event data:

- **billingModelID:**
  - Indicates the valid billing model id that subscriber has subscribed to.
- **subscriptionID:**
  - Indicates the subscription id when a subscriber subscribes to a given billing model.
- **Payer:**
  - Indicates the subscriber address who subscribed to the billing model.

# PullPaymentExecuted

Event name:

- PullPaymentExecuted

Event detail:

- This event is emitted when the pullPayment is executed for a given subscription.
- This event is emitted inside the **executePullPayment()** method.

Event signature:

```
event PullPaymentExecuted(
 uint256 indexed subscriptionID,
 uint256 indexed pullPaymentID,
 uint256 indexed billingModelID,
 address payer
);
```

Event data:

- **subscriptionID:**
  - Indicates the subscription id whose pullPayment is executed.
- **pullPaymentID:**
  - Indicates the id of executed pullPayment.
- **billingModelID:**
  - Indicates the billing model id for which pullPayment is executed.
- **Payer:**
  - Indicates the subscriber address who paid for the subscription.

# SubscriptionCancelled

Event name:

- SubscriptionCancelled

Event detail:

- This event is emitted when a subscriber cancels the subscription using **cancelSubscription()** method.

Event signature:

```
event SubscriptionCancelled(
 uint256 indexed billingModelID,
 uint256 indexed subscriptionID
);
```

Event data:

- **billingModelID:**
  - Indicates the billing model id to which canceled subscription belongs.
- **subscriptionID:**
  - Indicates the canceled subscription id.



# BillingModelEdited

Event name:

- BillingModelEdited

Event detail:

- This event is emitted when the billing model is edited using **editBillingModel()** method.

Event signature:

```
event BillingModelEdited(
 uint256 indexed billingModelID,
 address indexed newPayee,
 address oldPayee
);
```

Event data:

- **billingModelID:**
  - Indicates the edited billing model id.
- **newPayee:**
  - Indicates the updated payee address.
- **oldPayee:**
  - Indicates the payee address who edits the billing model.

## Flow of Execution:

1. First merchant creates the billing model with required configuration for his business. Merchant needs to specify the payee address and the type of the recurring pullPayment only. Rest of the details are provided dynamically through the code-snippet.
2. When a subscriber subscribes the billing model, the remaining details of the billing model are provided dynamically through the code snippet. This detail includes the name of the billing model, settlement token address, payment amount, frequency of pullPayments, total number of pullPayments, trial period if required and the initial amount for trial period in case of paid trial recurring pullPayment.
3. Subscriber should approve the unlimited payment tokens to the Executor contract before subscribing billing model.
4. Executor is the contract which gets the tokens from the subscriber, swaps the payment token to the settlement token and then transfers the settlement tokens to the payee.
5. When a customer subscribes to the Normal type of recurring pullPayment, The first pullPayment is executed immediately and rest of the pullPayments are executed after a fixed interval of time.
6. When a customer subscribes to the Free trial type of recurring pullPayment, The customers free trial gets started. No amount is charged to the customer during the free trial period. After the completion of a free trial, the user is charged for the services via pullPayments.
7. When a customer subscribes to the Paid trial type of recurring pullPayment, The initial amount is charged to the customer and paid trial is activated. After the completion of the paid trial, the customer is charged for the services using the pullPayments.
8. The pullPayments are executed until the subscriber cancels the subscription or the total number of pullPayments gets completed.
9. Users can cancel the subscription if they don't want to continue the service offered by the merchant.
10. Merchants can edit the billing model name and payee address if they want.