

Single PullPayment Contract

Introduction

The Single PullPayment contract simplifies the one time payment for the businesses. Using the Single PullPayment contract any merchant can create the billing model with required configuration for their business to simplify the payment. Customers can subscribe to the billing model to make the one time payments for the service.

Contract version

pragma solidity 0.8.0

Contract constructor

The Single Pullpayment contract's constructor takes one argument i.e the main registry contract address which is used to access the registry methods.

Constructor Signature:

- constructor(address registryAddress) RegistryHelper(registryAddress);

Constructor Parameters:

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

Contract Methods

createBillingModel()

Method name:

- createBillingModel

Method detail:

- This method allows merchants to create a new billing model for their business with required configurations.
- The unique Id is created for each billing model.
- Merchant can create as many billing models as he wants.

- Creator needs to specify the payee address, name of billing model, merchant name, billing model reference, merchant url, amount to charge and the token address in which amount should be transferred.
- The unique reference is generated if external reference is not given.
- This method emits **BillingModelCreated** event after successfully creating billing model

Method signature:

- function createBillingModel(
 address _payee,
 string memory _name,
 string memory _merchantName,
 string memory _reference,
 string memory _merchantURL,
 uint256 _amount,
 address _token
) external virtual override returns (uint256 billingModelID)

Method parameters:

1. **_payee:**
 - a. Indicates the address of the receiver(merchant) who will receive the pull payments.
2. **_name:**
 - a. Indicates the name for a billing model that can be injected from the creator of the billing model for any future reference. It can be empty.
3. **_merchantName:**
 - a. Indicates the name of the merchant. It can be empty.
4. **_reference:**
 - a. Indicates the unique reference for the billing model.
 - b. The Unique reference is generated if external reference is not given.
5. **_merchantURL:**
 - a. Indicates the merchant's personal url.
6. **_amount:**
 - a. Indicates the amount of settlement tokens that customer needs to pay to the merchant.
7. **_token:**
 - a. Indicates the settlement token address in which the merchant wants to get paid in.

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 a given billing model Id, supported payment token address and optional unique reference for subscription.
- The user must approve the payment tokens to the Executor contract before subscribing to the billing model if not approved.
- The unique subscription id and pullPayment id is created on each subscription.
- The unique subscription reference is created if external reference is not given.
- When a customer subscribes to the billing model with a given payment token, Immediately the direct pullPayment is executed with the help of the Executor contract.
- This method emits the **NewSubscription** event after subscription.

Method signature:

- function subscribeToBillingModel(
uint256 _billingModelID,
address _paymentToken,
string memory _reference
) public override returns (uint256 subscriptionID)

Method parameters:

- **_billingModelID:**
 - Indicates the unique billing model Id that customer wants to subscribe.
- **_paymentToken:**
 - Indicates the payment token address in which the customer wants to pay for the subscription.
- **_reference:**
 - Indicates the unique reference for the subscription.
 - Contract generates the unique reference for the subscription if external reference is not given.

Returned data:

- This method returns the unique subscription id after subscription.

editBillingModel()

Method name:

- editBillingModel

Method detail:

- This method Allows merchants to edit their billing models.
- Editing a billing model allows the creator of the billing model to update only attributes that do not affect the billing cycle of the customer, i.e. the name and the payee address etc.
- It allows the payee to update some of the details of the billing model i.e name, payee address, amount, settlement token address, merchant name and merchant url.
- This method emits the **BillingModelEdited** event after editing the billing model.

Method signature:

- ```
function editBillingModel(
 uint256 _billingModelID,
 address _newPayee,
 string memory _newName,
 uint256 _newAmount,
 address _newSettlementToken,
 string memory _newMerchantName,
 string memory _newMerchantURL
) external virtual override returns (uint256 billingModelID);
```

Method parameters:

- **\_billingModelID:**
  - Indicates the unique valid billing model id which the merchant wants to edit.
- **\_newPayee:**
  - Indicates the new payee address for the billing model.
  - Merchants cannot set Zero address as payee address.
- **\_newName:**
  - Indicates the new name for the billing model. Can be kept empty.
- **\_newAmount:**
  - Indicates the new amount for the billing model.
- **\_newSettlementToken:**
  - Indicates the new supported settlement token address.
- **\_newMerchantName:**
  - Indicates the new merchant name. Can be kept empty.
- **\_newMerchantURL:**
  - Indicates the new merchant url. Can be kept empty.

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.
- 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
  - Name of billing model
  - Name of the merchant
  - Then unique reference of billing model
  - Amount of billing model
  - Settlement token address
  - Subscription Ids (only payee can retrieve)
  - Billing model creation time.

# getBillingModel()

Method name:

- `getBillingModel`

Method detail:

- This method returns the details of the specified billing model.
- User specifies the payment token in which he wants to pay for subscription.
- It tells you the optimal number of payment tokens needed to pay to settle the payment for the billing model.
- It uses the quickswap router to calculate the optimal payment amount for a given payment token.

Method signature:

- `function getBillingModel(uint256 _billingModelID, address _token)`  
external  
view  
override  
returns (SwappableBillingModel 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
  - Name of the billing model
  - Merchant name
  - Unique reference of billing model
  - Merchant URL
  - Settlement token amount
  - Settlement token address
  - Optimal amount of payment tokens
  - Payment token address
  - Billing model creation time

# getSubscription()

Method name:

- getSubscription

Method detail:

- This method returns the subscription data for a given subscription id.
- Only the payee and subscriber of the billing model can see the pullPayment ids of 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
  - Payment amount
  - Settlement token address
  - Payment token address
  - PullPayment IDs(only payee or subscriber gets the ids)
  - Billing model id
  - Unique reference of subscription.

## getPullPayment()

Method name:

- getPullPayment

Method detail:

- This method retrieves the pullPayment details for given pullPayment id.
- 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.
  - Billing model id
  - Subscription id



## 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.

## 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.

## getCurrentBillingModelId()

Method name:

- getCurrentBillingModelId

Method detail:

- This method returns the current billing model id of the pullpayment

Method signature:

- function getCurrentBillingModelId() external view virtual returns (uint256);

Returned data:

- This method returns the current billing model id.

## getCurrentSubscriptionId()

Method name:

- `getCurrentSubscriptionId`

Method detail:

- This method returns the current subscription id of the pullpayment

Method signature:

- `function getCurrentSubscriptionId() external view virtual returns (uint256);`

Returned data:

- This method returns the current subscription id.

## getCurrentPullPaymentId()

Method name:

- `getCurrentPullPaymentId`

Method detail:

- This method returns the current id of the pullpayment

Method signature:

- `function getCurrentPullPaymentId() external view virtual returns (uint256);`

Returned data:

- This method returns the current pullpayment id.

# 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
);
```

Event data:

- **billingModelID:**
  - Indicates the newly created billing model id.
- **Payee:**
  - Indicates the payee address who created the billing model.

# 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,  
    uint256 indexed pullPaymentID,  
    address payee,  
    address payer,  
    uint256 executionFee,  
    uint256 userAmount,  
    uint256 receiverAmount  
);

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.
- **pullPaymentID:**
  - Indicates the pullpayment id for the subscription.
- **payee:**
  - Indicates the payee address of the billing model.
- **payer:**
  - Indicates the subscriber address.
- **executionFee:**
  - Indicates the execution fee charged for executing the pullpayment
- **userAmount:**
  - Indicates the amount of tokens that the user paid for subscription.
- **receiverAmount:**
  - Indicates the amount of tokens that payee receives from the pullpayment.

# 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,  
    string indexed newName,  
    string newMerchantName,  
    uint256 amount,  
    address settlementToken,  
    address oldPayee,  
    string newMerchantUrl  
);

Event data:

- **billingModelID:**
  - Indicates the edited billing model id.
- **newPayee:**
  - Indicates the updated payee address.
- **newName:**
  - Indicates the updated billing model name.
- **newMerchantName:**
  - Indicates the new merchant name address.
- **amount:**
  - Indicates the updated amount.
- **settlementToken:**
  - Indicates the settlement token address.
- **oldPayee:**
  - Indicates the payee address who edits the billing model.
- **newMerchantUrl:**
  - Indicates the updated merchant url.

## Flow of Execution:

1. First merchant creates the billing model with required configuration for his business.
2. Subscriber should approve the unlimited payment tokens to the Executor contract before subscribing to the billing model.
3. Executor is the contract which gets the tokens from the subscriber, swaps the payment token to the settlement token, transfers the execution fee to the execution fee receiver and then transfers the settlement tokens to the payee.
4. Subscriber subscribes the required billing model with payment token that he wants to pay in to avail the services offered by the merchant.
5. When a Subscriber subscribes to the billing model, immediately a single pullPayment is executed.
6. Whenever you want to avail the service you subscribe to the billing model and make the single pullPayment.
7. Merchants can edit the billing model details i.e name, settlement token address, amount and payee address etc.