# 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 payments for the service.

## 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 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, amount to charge, and the token address in which amount should be transferred.

Method signature:

- function createBillingModel(
    address _payee,
    bytes32 _name,
    uint256 _amount,
    address _token
  ) public override returns (uint256 billingModelID)

Method parameters:

1. **_payee:**
   Indicates the address of the receiver(merchant) who will receive the pull payments.
2. **_name:**
   Indicates the name of the billing model. Name should be given in bytes32 format
3. **_amount:**
   Indicates the amount of settlement tokens that customer needs to pay to the merchant.
4. **_token:**
   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 given billing model Id and payment token.
- The user must approve the payment tokens to the Executor contract before subscribing the billing model.
- The unique subscription id and pullPayment id is created on each subscription.
- 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.

Method signature:

- function subscribeToBillingModel(
    uint256 _billingModelID,
    address _paymentToken
  ) 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.

Returned data:

- This method returns the unique subscription id after subscription.

# editBillingModel()

- editBillingModel

Method detail:

- This method allows the billing model creator i.e payee to edit the details of the billing model.
- It allows the payee to update all the details of the billing model i.e name, payee address, amount and settlement token address.
- Only the owner of the billing model can edit the billing model details.

Method signature:

- function editBillingModel(
      uint256 _billingModelID,
      address _newPayee,
      bytes32 _newName
      uint256 _newAmount,
      address _newSettlementToken
    ) 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.
- **_newName:**
  - Indicates the new name for the billing model.
- **_newAmount:**
  - Indicates the new amount for the billing model.
- **_newSettlementToken:**
  - Indicates the new settlement token address.

Returned data:

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

# getBillingModel()

- getBillingModel

- 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
    - Amount of billing model
    - Settlement token address
    - Subscription Ids (only payee can retrieve)

# getBillingModel()

- 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.
- If the payment token and settlement token is different, It tells you the optimal number of payment tokens needed to pay to settle the payment for the billing model.
- It uses the uniswap 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
  - Settlement token amount
  - Settlement token address
  - Optimal amount of payment tokens
  - Payment token address

# getSubscription()

- 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 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)

# getPullPayment()

- getPullPayment

- 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.

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

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

- 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.

# 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
);
```

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,
    address indexed subscriber
);

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.
- **subscriber::**
    - Indicates the subscriber address who subscribed to the billing model.

# BillingModelEdited

- 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,
    bytes32 indexed newName,
    uint256 amount,
    address settlementToken,
    address oldPayee
);
```

Event data:

- **billingModelID:**
  - Indicates the edited billing model id.
- **newPayee:**
  - Indicates the updated payee address.
- **newName:**
  - Indicates the updated billing model name.
- **Amount:**
  - Indicates the updated amount.
- **settlementToken:**
  - Indicates the settlement token 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.
2. Subscriber should approve the unlimited payment tokens to the Executor contract before subscribing billing model.
3. 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.
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 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..