

*) IPayment

- Name:
IPayment
- Interface description:
Payment interface orchestrates the interaction between users, merchants, and financial institutions, ensuring the smooth processing of payments across various channels and payment methods as a gateway. It encapsulates the method required for authorizing , and completing payment transactions.
- Operation definition:
 1. `payOrder(amount:double, orderInfo: String)`: Set request to pay order with the amount and order info given by client
- Operation description
 1. `payOrder(amount:double, orderInfo: String)`:
+) @param: amount (double), orderInfo (String)
- Interface documentation:

The IPayment interface offers a collection of operations to streamline payment processing. Implementing classes or components must adhere to this interface to ensure a unified approach to handling payments across the system. The `payOrder` operation manages the payment process for a designated order, considering the payment amount, additional order details, and client information as inputs. The implementation of this operation encompasses necessary steps to finalize the payment transaction, such as interfacing with payment gateways, updating order statuses, and issuing payment receipts. The return value of the `payOrder` operation serves to communicate the success or failure status of the payment transaction, enabling the calling code to manage exceptional scenarios or provide relevant user feedback. By conforming to the IPayment interface, different payment implementations can be seamlessly interchanged or expanded

without disrupting other system components reliant on payment functionality.

***) Request**

- Name:
Request
- Class description:
This class appeared after a client requested a payment for an order. This class will handle some requests from PayOrderManager and return proper results.
- Operation definition
 1. request(amount:double, orderInfo: String): this operation is used to save created request from client
 2. String buildQueryURL(): This operation used to get the query string after built
 - @return: return query URL
- Class documentation:

***) PayOrderManager**

- Name:
PayOrderManager
- Class description:
It is used for controlling the way that clients interact with the payment system. It permits clients to request and handle that request with a behind system.
- Operation definition
 1. payOrder(amount: double, orderInfo: String, Client: client): this operation is used to request a request with amount, orderInfo and client 's info)
 2. processResponse(queryString: String): this operation is used to process a response after receiving a query.
- Class documentation:

This class uses a set of operations to help users to interact with the system. They will receive input from clients and request behind classes to complete all the things clients need.

***) Response**

- Name:
Response
- Class description:
This class is created in the aim of verifying a received query string and handling errors from this query.
- Operation definition:
 1. createResponse(queryString: String): this operation to set the response after receiving a query.
 2. parseQueryString(): this operation will check whether the input query has any errors or not.
 - @Return: this will return an integer as a code for checking
 3. handleErrorCode(): this will receive the result of above operation and choose the appropriate operation next.
- Class documentation:
This class is created after the PayOrderManager class requested to process a query. This class will parse the query and handle the problem via the code received after parsing the query.

***) TransactionInfo**

- Name:
TransactionInfo
- Class description:
This will be used to store transaction data once the Pay Order Manager responds.
- Operation definition:

1. transactionInfo(response): This will used to set transaction information after receiving response from PayOrderManager
- Class documentation:
This class has only 1 operation which is saving the transaction info based on the response.

***) VNPayDisplay**

- Name:
VNPayDisplay
- Class description:
This is used to display all the necessary information from the system after receiving input value.
- Operation definition:
 1. displayURL(url: String): This operation used to display received url
 2. displayURL(vnp_ReturnURL_With_Response)
 3. checkURLChange(): This is used to check if url is change or not and then, go to close operation
 4. close(): Close the display.
- Class documentation:
This class has a collection of operations that it will send a pay order and then display url and check . Whenever the url change, we will have operation to handle.

***) Client**

- Name:
Client
- Class description:
User
- Operation definition:
 1. updateTransactionOnFailure(exception): This is used to set the transaction in case error occurred.

2. `updateTransactionOnFailure(trans)`: This is used to set the transaction in case no error occurred

- Class documentation:

Set of operations of the client related to behavior with transaction based on the success or failure of transaction.