# Документация к CardPay API

Для разработки

Если в настройках проекта включить Построение -> Вывод -> XML-файл документации, то вместе с библиотекой будет генерироваться файл CardPay.xml, в нем техническая документация для разработчиков.

Использование

Общий принцип работы такой:

1. Создается объект CardPayAPI
2. Вызываются его необходимые методы
3. Получаем ответы или сообщения об ошибке в своем обработчике (асинхронные запросы)

Пример использования:

1. Создание обработчика

//Создаем простейший обработчик вызовов API для примера

var listenerExample = new CardPayListenerExample();

Он представляет собой простейшую реализацию интерфейса ICardPayListener для примера.

/// <summary>

/// API listener

/// </summary>

public interface ICardPayListener {

/// <summary>

/// Called when order request completed.

/// </summary>

/// <param name="orderResult">The order result.</param>

void OnPayCompleted(OrderResponse orderResult);

/// <summary>

/// Called when print form ready.

/// </summary>

/// <param name="form">The print form.</param>

void OnPrintFormCompleted(PrintForm form);

/// <summary>

/// Called when error occured.

/// </summary>

/// <param name="errorDescription">The error description.</param>

void OnError(string errorDescription);

/// <summary>

/// Called when report request completed.

/// </summary>

/// <param name="reportResponse">The report response.</param>

void OnReportRequestCompleted(ReportResponse reportResponse);

/// <summary>

/// Called when change request completed.

/// </summary>

/// <param name="changeResponse">The change response.</param>

void OnChangeRequestCompleted(ChangeResponse changeResponse);

}

public class CardPayListenerExample : ICardPayListener

{

public void OnPayCompleted(OrderResponse result) {

var payResult = result;

Console.WriteLine(payResult);

}

public void OnPrintFormCompleted(PrintForm form) {

var printForm = form;

Console.WriteLine("OrderXML : {0}; SHA512 : {1}", printForm.OrderXML, printForm.SHA512);

}

public void OnError(string errorDescription) {

Console.WriteLine("CardPayAPI error: {0}", errorDescription);

}

public void OnReportRequestCompleted(ReportResponse report) {

var reportResponse = report;

Console.WriteLine(reportResponse);

}

public void OnChangeRequestCompleted(ChangeResponse change) {

var changeResponse = change;

Console.WriteLine(changeResponse);

}

}

1. Инициализация CardPayAPI

//Инициализация CardPay API

var api = new CardPayAPI("RxzX58Z0p1Hg", 1598, listenerExample);

где "RxzX58Z0p1Hg" – секретный ключ,

1598 – wallet\_id

listenerExample – наш обработчик вызовов API

1. Формируем параметры запросов

Для примера простейший заказ:

//Формирование заказа

var order = new OrderRequest();

order.Number = "CardPay.NET Number";

order.Amount = 0.01m;

order.EMail = "f1restarter@rocketmail.com";

1. Получение печатной формы

//Получение печатной формы

api.PrintForm(order);

Мы получим в методе OnPrintFormCompleted нашего обработчика экземпляр PrintForm

public class PrintForm

{

public string OrderXML { get; set; }

public string SHA512 { get; set; }

}

1. Отправка заказа

//Оплата

api.Pay(order);

Мы получим в методе OnPayCompleted класс с данными ответа.

1. Получение информации о транзакциях

В данном запрос есть два обязательных параметра это логин и пароль и остальные необязательные, которые указываются как на примерах внизу.

//Получение транзакции #621640

api.GetTransactionsInfo("dev.cardpay.com", "6Zl1JDo3Gx0w", orderNumber: "621640");

//Получение последних 10 транзакций

api.GetTransactionsInfo("dev.cardpay.com", "6Zl1JDo3Gx0w");

//Получение транзакций с 1 января 2014 года

api.GetTransactionsInfo("dev.cardpay.com", "6Zl1JDo3Gx0w", dateBegin : new DateTime(2014, 01, 01).ToString("yyyy-MM-dd"));

//Получение транзакций с 1 января 2014 года по 5 января 2014 года

api.GetTransactionsInfo("dev.cardpay.com", "6Zl1JDo3Gx0w", dateBegin: new DateTime(2014, 01, 01).ToString("yyyy-MM-dd"), dateEnd: new DateTime(2014, 01, 05).ToString("yyyy-MM-dd"));

Ответ получим в методе OnReportRequestCompleted.

1. Смена статуса транзакции

В данном запросе 4 обязательных параметра: логин, пароль, номер транзакции и статус.

Статусы бывают:

/// <summary>

/// Order status

/// </summary>

public enum TransactionStatus

{

[XmlEnum(Name = "void")]

VOID,

[XmlEnum(Name = "capture")]

CAPTURE,

[XmlEnum(Name = "refund")]

REFUND,

}

Примеры вызова:

//Смена статуса транзакции #621640

api.ChangeTransactionStatus("dev.cardpay.com", "6Zl1JDo3Gx0w", 621640, TransactionStatus.VOID);

//Смена статуса транзакции #621640 с необязательными параметрами

api.ChangeTransactionStatus("dev.cardpay.com", "6Zl1JDo3Gx0w", 621640, TransactionStatus.VOID, amount : 0.01m, reason : "Test change status");

Ответ получим в методе OnChangeRequestCompleted.

1. Обработка ошибок

В случае возникновения ошибки при вызове API данные об ошибке попадают в метод OnError установленного обработчика.

Также в код добавлен пример для проверки, что обозначение валюты состоит только из 3-х символов.

/// <summary>

/// Gets or sets the currency code.

/// </summary>

/// <value>

/// ISO 4217 currency code.

/// </value>

/// <exception cref="CardPayException"></exception>

[XmlAttribute("currency")]

public string Currency {

get { return \_currency; }

set {

if (value.Length != 3)

throw new CardPayException(string.Format("Wrong currency format: {0}", value));

\_currency = value;

}

}

Эксепшн выбрасывается при присваивании невалидного значения, поэтому для отлова данной ситуации можно обернуть инициализацию параметров запроса.

try {

//Формирование заказа

var order = new OrderRequest();

order.Number = "CardPay.NET Number";

order.Amount = 0.01m;

order.EMail = "f1restarter@rocketmail.com";

} catch (CardPayException e) {

Console.WriteLine("CardPay error: {0}", e.Message);

return;

}