

PAYTURE

Payture.js

Руководство пользователя

Версия 3.2.1

19.08.2019



Оглавление

1	Описание	3
1.1	Требования к шаблонам	3
1.2	Названия шаблонов.....	4
2	Настройки	5
2.1	Основные параметры.....	5
2.2	Дополнительные опции.....	8
2.3	Функции	10
3	Подключение	12
4	Шаблоны для интерфейса Payture InPay	13
4.1	Стандартное подключение	13
4.2	Расширение опций	13
4.3	Настройка полей для ввода.....	13
4.4	Изменение функции отображения ошибок.....	14
4.5	JSON шаблоны	14
5	Шаблоны для интерфейса Payture eWallet	16
5.1	Шаблон добавления карты	16
5.2	Шаблон оплаты	16
5.3	Удаление карт	16
5.4	JSON шаблоны	17
6	Методы.....	18
6.1	Pay.....	18
6.2	Add	20
6.3	Validate	21
6.4	RemoveCardBySession	21
6.5	GetList	22
	ПРИЛОЖЕНИЕ «Коды и тексты ошибок, результатов выполнения операций и текстов шаблонов» ...	23

1 Описание

Использование библиотеки `payture-3.2.js` позволяет Продавцу упростить процесс работы с шаблонами при проведении оплаты на стороне платежного шлюза.

Библиотека содержит всю необходимую функциональность для корректного сбора и отправки данных банковской карты Покупателя при оплате и привязке карты.

`Payture.js` используется в шаблонах оплаты, привязки карт и странице возврата на стороне шлюза в интерфейсах `Payture InPay` и `Payture eWallet`.

Библиотека позволяет как использовать стандартный сценарий оплаты, так и настраивать работу шаблона согласно вашим личным требованиям и предпочтениям. Вы также можете реализовать собственный, уникальный сценарий оплаты с помощью готовых методов, описанных в соответствующем разделе данного документа.

1.1 Требования к шаблонам

- На странице оплаты должно присутствовать скрытое поле `Key` со значением `{key}`, в котором передаются параметры платежа
- На странице шаблона не должно быть ссылок, перенаправляющих пользователей на сторонние вебсайты
- На странице шаблона запрещено использование внешних ресурсов, т.е. все изображения, файлы стилей и скриптов должны передаваться вместе со страницами шаблонов
- Дополнительные параметры платежа могут быть переданы при инициализации сессии в параметре `DATA`. Для вывода их на странице оплаты следует использовать запись `{param}`, где `{param}` – это имя передаваемого параметра в нижнем регистре
- Ссылки на используемые ресурсы (стили, изображения) должны иметь вид `/Templates/{Key}/{File}`, где `{Key}` – название терминала.
- В случае использования нескольких шаблонов необходимо хранить все ресурсы в общей папке
- Возможно использование нескольких языков. В этих случаях вы передаете параметр `{Language}` в составе параметров `DATA` при инициализации сессии



- HTML-код страницы должен содержать блок с текстом {error}, который выводится при возникновении ошибки. Блок с плейсхолдером {error} используется для вывода ошибок при не JSON типе ответа сервера. Блок с id errorTop используется для вывода ошибок при JSON типе ответа сервера

1.2 Названия шаблонов

Название шаблона должно формироваться следующим образом:

`[{API}]{Key}-{Type}-{TemplateTag}-{Language}.template`

Например:

`[MERCHANT]Merchant-Pay-Default-Default.template`

`[CUSTOMERS]VWMerchant-Add-MyCustom-Spanish.template`

Параметр	Описание	Значение
<code>{API}</code>	Тип используемого интерфейса	MERCHANT — для интерфейса InPay CUSTOMERS — для интерфейса eWallet
<code>{Key}</code>	Наименование терминала	По умолчанию Merchant
<code>{Type}</code>	Тип шаблона	Pay — для шаблона оплаты Add — для шаблона привязки карты Return — для страницы возврата. Данный тип шаблона открывается после 3DS-проверки.
<code>{TemplateTag}</code>	Название шаблона, если используется несколько шаблонов	По умолчанию Default
<code>{Language}</code>	Язык шаблона, если используются шаблоны на разных языках	По умолчанию Default

2 Настройки

2.1 Основные параметры

В данной таблице перечислены основные параметры сбора данных, влияющие на работу платежной формы. Вы можете использовать стандартный набор параметров или же передавать свои, что может быть актуально при внесении изменений в HTML-верстку шаблона или при написании собственных сценариев оплаты.

Название	Описание	Формат	По умолчанию
Type	Тип шаблона	Строка, Default - стандартный JSON	JSON
TypeAPI	Тип интерфейса	Строка, InPay или eWallet	InPay
Action	Тип действия	Строка, PaySubmit или AddSubmit	PaySubmit
Event	Событие, по которому происходит сбор и отправка формы	Строка, первое слово – событие, второе – элемент, к которому оно делегируется Внимание! Название элемента не должны быть «form»	submit #payForm

Название	Описание	Формат	По умолчанию
Data	Набор полей для ввода данных или значения.	Объект, в качестве ключей используются названия необходимых полей, а в качестве значений могут использоваться: <ul style="list-style-type: none"> ■ Соответствующее поле для ввода ■ Строка с конкретным значением ■ Функция, возвращающая строку 	CardNumber: <code>\$("#input[name=CardNumber]:visible")</code> EMonth: <code>\$("#input[name=EMonth]")</code> , EYear: <code>\$("#input[name=EYear]")</code> CardHolder: <code>\$("#input[name=CardHolder]:visible")</code> SecureCode: <code>\$("#input[name=SecureCode]:visible")</code>
TopErrorContainer	Контейнер для вывода ошибок	HTML элемент	<code>#errorTop</code>
TemplateErrors	Текст ошибок, обрабатываемых на стороне клиента	Объект, в качестве ключа используется код ошибки, в качестве значения – текст ошибки	См. в Приложении
ServerErrors	Коды ошибок, возвращаемых сервером	Объект, в качестве ключа используется код ошибки, в качестве значения – текст ошибки	См. в Приложении
AnswerText	Тексты ответов о результатах операций	Объект, в качестве ключа используется код ответа, в качестве значения – текст ответа	См. в Приложении
TemplateText	Тексты, используемые в шаблонах	Объект, в качестве ключа используется код текста, в качестве значения – текст шаблона	См. в Приложении



Название	Описание	Формат	По умолчанию
RemoveEvent	Событие, по которому происходит удаление карты. Внимание! Доступно только для интерфейса eWallet	Строка, первое слово – событие, второе – элемент, к которому оно делегируется	click #removeCard
PaymentKey	Наименование платежного терминала. Необходимо для интерфейса eWallet для удаления карт	Строка	
MasterPass	Использование шаблонов MasterPass	Boolean	false
MasterPassUse CVV	Ввод CVV на шаблонах MasterPass	Boolean	false
Inbox	Использование шаблонов для оплаты в интернет-банках	Boolean	false

2.2 Дополнительные опции

В данном разделе перечислены дополнительные опции взаимодействия с интерфейсом формы. Большинство из них влияют только на визуальное отображение и работу с элементами формы.

Внимание! Возможность использования некоторых опций зависит от HTML-кода страницы, наличия или отсутствия определенных блоков, их идентификаторов и классов.

Название	Описание	Формат	Значение
CheckNumbers	Проверка ввода цифр в соответствующие поля – номер карты, дата и CVV Внимание! Поле для ввода цифр должно иметь класс «onlyNum»	Boolean	true
CheckLetters	Проверка ввода символов в текстовое поле – держатель карты – и автоматический перевод русских символов в латинские Внимание! Поле для ввода латинских символов должно иметь класс «onlyLat»	Boolean	true
DetectCardType	Подсветка логотипы МПС при вводе номера карты – Visa или MasterCard. Внимание! Блок для логотипа должен иметь идентификатор «CardType» и стили для классов «visa» и «master»	Boolean	true
DisableButton	Деактивация кнопки до того момента, пока необходимые поля не будут корректно заполнены	Boolean	true
GroupCardNumber	Автоматическая смена фокуса в полях ввода номера карты или группировка номера по 4 цифры, если это мобильное устройство Внимание! Для автоматической смены фокуса необходимо, чтобы у поля имелись атрибуты «prev» и «next», со значением, соответствующим атрибуту «name» поля для фокуса	Boolean	true



Название	Описание	Формат	Значение
ResizeCardNumberInput	Изменение размера последнего поля для ввода карты при вводе 19-значного номера, в случае если номер карты состоит из нескольких полей.	Boolean	false
HideCardList	Для интерфейса eWallet скрывать поле для выбора карты, если нет привязанных карт	Boolean	true
BINs	Определение и отображение логотипа/названия банка, выпустившего карту	Boolean	true
Language	Язык интерфейса оплаты: RU или EN. Обратите внимание, что будут измене	String	RU
CheckLuna	Проверять номер карты с помощью алгоритма Луна	Boolean	true
IsAmex	Возможность использования карт Amex	Boolean	false
ApplePay	Использование ApplePay для оплаты	Boolean	false
GoogleWebPay	Использование GooglePay для оплаты	Boolean	false
SamsungPay	Использование SamsungPay для оплаты	Boolean	false

2.3 Функции

Вы можете задать собственный сценарий поведения страницы перед отправкой формы, при выборе привязанной карты (для интерфейса eWallet) или при удалении карты, при показе и скрывании ошибок, обрабатываемых на клиентской стороне.

Кроме того, если вы используете JSON шаблоны, вы должны описать поведение страницы в случае успеха или неуспеха операции.

Название	Описание	Параметры
onBeforeSubmit	Дополнительная функция, выполняется перед отправкой формы	без параметров
onSelectCard	Функция, выполняемая при выборе карты из списка привязанных карт, заменяет функцию по умолчанию	cardId – идентификатор выбранной карты cardMask – маска выбранной карты в формате «123456xxxxxx1234»
onShowErrors	Показ сообщений об ошибках, заменяет функцию по умолчанию	Success - true/false, Element - строка, с названием поля, соответствует ключам в параметре «Data», ErrorMessage - строка с текстом сообщения
onHideErrors	Скрытие сообщений об ошибках, заменяет функцию по умолчанию	element – название поля, соответствует ключам в параметре «Data»
onRemoveCard	Функция выполняемая при удалении карты, заменяет функцию по умолчанию	cardId – идентификатор удаленной карты
onSuccess	Для JSON шаблонов Функция, выполняемая в случае успеха операции	redirectUrl – адрес перенаправления пользователя
onError	Для JSON шаблонов Функция, выполняемая в случае ошибки	errorCode – код ошибки canRetry – возможность повторной оплаты redirectUrl – адрес возврата key – новый ключ



Название	Описание	Параметры
onSuccessAdd	Для JSON шаблонов Функция, выполняемая в случае успеха операции добавления	redirectUrl – адрес перенаправления пользователя
onErrorAdd	Для JSON шаблонов Функция, выполняемая в случае ошибки операции добавления	errorCode – код ошибки canRetry – возможность повторной оплаты redirectUrl – адрес возврата key – новый ключ

3 Подключение

Для работы с библиотекой `payture-3.2.js` вам понадобятся несколько дополнительных библиотек. Все необходимые ресурсы расположены в папке `_Resources` вместе с библиотекой `payture-3.2.js`; от вас потребуется разместить на своей странице оплаты следующие скрипты:

```
<script type="text/javascript"
src="/Templates/_Resources/3DPartyUtils/jquery-1.9.1.min.js"></script>
<script type="text/javascript"
src="/Templates/_Resources/3DPartyUtils/jquery.maskedinput.min.js"></sc
ript>
<script type="text/javascript"
src="/Templates/_Resources/3DPartyUtils/jquery.payment.js"></script>
<script type="text/javascript"
src="/Templates/_Resources/3DPartyUtils/fingerprint2.js"></script>
<script type="text/javascript"
src="/Templates/_Resources/CommonJSUtils/bins.js"></script>
<script type="text/javascript"
src="/Templates/_Resources/errors.js"></script>
<script type="text/javascript" src="/Templates/_Resources/payture-
3.2.js"></script>
```



4 Шаблоны для интерфейса Payture InPay

4.1 Стандартное подключение

Для простого подключения скрипта со стандартными параметрами используйте функцию:

```
Payture.InPay()
```

4.2 Расширение опций

Вы можете включить или отключить любую опцию при подключении скрипта, например, отключить блокировку кнопки до того момента, пока не будут заполнены необходимые поля, и изменение размеров поля ввода номера карты для 19-значного номера.

```
Payture.InPay({
    DisableButton : false,
    ResizeCardNumberInput : true
})
```

4.3 Настройка полей для ввода

Обратите внимание, что при изменении параметра «**Data**» в нем должны быть перечислены все необходимые поля. Это касается и изменения параметра «**TemplateErrors**».

```
Payture.InPay({
    Data : {
        CardNumber : $("input[name=CN]"),
        CardHolder : "Cardholder Name",
        EMonth : $("[name=EMonth]"),
        EYear : function () { return $("#Year").val(); },
        SecureCode : $("[name=CVV]")
    }
})
```

4.4 Изменение функции отображения ошибок

Вы можете написать собственную функцию отображения ошибок на шаблоне, которая заменит стандартную. Функция принимает на вход объект «**error**», состоящий из трех элементов: «**Element**» – имя элемента в объекте «**Data**», «**Success**» – верно или неверно значение данного поля, и «**ErrorMessage**» – сообщение об ошибке.

```
Payture.InPay({
  onShowError : function (error) { if
    (!error.Success) {
      alert("Ошибка в поле" + error.Element + ": " +
        error.ErrorMessage);
    }
  }
})
```

4.5 JSON шаблоны

Для работы с JSON шаблонами необходимо передать параметр «**Type**» и описать поведение в случае успеха или неуспеха операции – для этого используются функции «**OnSuccess**» и «**OnError**» соответственно.

В случае успеха операции в функцию «**OnSuccess**» передается «**redirectUrl**» – адрес перенаправления пользователя, указанный вами при подключении к платежному шлюзу.

В случае ошибки в функцию «**OnError**» передаются параметры:

- **errorCode** – код ошибки от платежного шлюза
- **canRetry** – возможность повторного платежа
- **redirectUrl** – адрес возврата пользователя, указанный при подключении к платежному шлюзу
- **key** – новый ключ, в соответствующее поле подставляется автоматически



```
Payture.InPay({
  Type : "JSON",
  OnSuccess : function (redirectUrl) {
    // Тело функции
  },
  OnError : function (errorCode, canRetry, redirectUrl, key) {
    // Тело функции
  }
})
```

Для корректной обработки ответа при оплате картой, защищенной системой аутентификации 3-D Secure, в тело шаблона оплаты и в тело шаблона возврата необходимо вставить HTML код, в который будет вставляться ошибка после перенаправления пользователя со страницы ACS банка.

```
<span class="response_json">{response_json}</span>
```

Если карта защищена 3-D Secure, то в случае ошибки покупатель возвращается на шаблон оплаты, при этом выполнится функция «**onError**».

Если операция по 3-D Secure была успешна, покупатель будет перенаправлен на шаблон страницы возврата, где функция «**Payture.getJSONResponse()**» вернет вам ответ об успехе операции, аналогичный ответу в случае успеха операции с картой, не защищенной аутентификацией 3-D Secure.

5 Шаблоны для интерфейса Payture eWallet

5.1 Шаблон добавления карты

Для подключения скрипта для шаблона добавления карты со стандартными параметрами используйте функцию:

```
Payture.eWalletAdd();
```

Все параметры, опции и собственные функции настраиваются аналогично интерфейсу InPay.

5.2 Шаблон оплаты

Для подключения скрипта для шаблона оплаты по карте со стандартными параметрами используйте функцию:

```
Payture.eWalletPay();
```

Также вы можете включить или отключить любую опцию при подключении скрипта.

5.3 Удаление карт

Если вы хотите добавить возможность покупателю удалять привязанные карты на странице оплаты, вы должны передавать в настройках параметр «PaymentKey». Так же вы можете описать поведение страницы в случае успешного удаления карты.

```
Payture.eWalletPay({
    PaymentKey : "Merchant",
    OnRemoveCard: function (cardId) {
        // Тело функции
    }
});
```


5.4 JSON шаблоны

Для работы с JSON шаблонами необходимо передать параметр **«Type»** и описать поведение в случае успеха или неуспеха операции – функции **«OnSuccess»**, **«onSuccessAdd»** и **«OnError»**, **«onErrorAdd»** соответственно.

В случае успеха операции/добавления в функцию **«OnSuccess»/ «OnSuccess»** передается параметр **«redirectUrl»** - адрес перенаправления пользователя, указанный вами при подключении к платежному шлюзу.

В случае ошибки в функцию **«OnError»/ «onErrorAdd»** передаются параметры:

- **errorCode** – код ошибки от платежного шлюза
- **canRetry** – возможность повторного платежа
- **redirectUrl** – адрес возврата пользователя, указанный при подключении к платежному шлюзу
- **key** – новый ключ, в соответствующее поле подставляется автоматически

```
Payture.eWalletAdd({
    // Необходимые и дополнительные параметры
})
Payture.eWalletPay({
    // Необходимые и дополнительные параметры
})
```

Для корректной обработки ответа при оплате картой, защищенной 3-D Secure, в тело шаблона оплаты и в тело шаблона возврата необходимо вставить HTML код, в который будет вставляться ошибка после перенаправления пользователя со страницы ACS банка.

```
<span class="response_json">{response_json}</span>
```

Если карта защищена 3-D Secure, то в случае ошибки покупатель возвращается на шаблон оплаты, при этом выполнится функция **«onError»**.

Если авторизация карты в системе 3-D Secure была успешна, покупатель будет перенаправлен на шаблон страницы возврата, где функция **«Payture.getJSONResponse()»** вернет вам ответ об успехе операции, аналогичный ответу в случае успеха операций с картами, не защищенными 3-D Secure.

6 Методы

Мы предлагаем набор готовых методов для работы с картами, для использования нестандартных сценариев оплаты, для привязки карты и других действий.

6.1 Pay

Данный метод позволяет напрямую производить оплату по карте. Для этого необходимо передать в метод тип шаблона – «Default» (стандартный шаблон оплаты) или «JSON», тип используемых API – «InPay» или «eWallet», платежный ключ – обычно значение скрытого поля «Key», и объект «Data», представляющий собой набор параметров карты.

Если вы используете JSON шаблоны, необходимо так же передать в качестве параметров описание функций «OnSuccess», «OnSuccessAdd», «OnError» и «OnErrorAdd».

```
Payture.Pay({
  TypeTemplate : "JSON",
  TypeAPI : "InPay",
  Key : $("[name=Key]").val(),
  Data : {
    CardNumber : "1234567890123456",
    CardHolder : "TEST",
    EMonth : 12,
    EYear : 16,
    SecureCode : 123
  },
  OnSuccess : function (redirectUrl) {
    // Тело функции
  },
  OnError : function (errorCode, canRetry, redirectUrl, key) {
    // Тело функции
  }
});
```

Если вы используете интерфейс eWallet, в параметре «Data» должен присутствовать идентификатор карты «CardId» и флаг привязки карты «AddCard».

При использовании интерфейса eWallet, состав параметра «Data» отличается



для оплаты по привязанной и непривязанной карте: в случае привязанной карты вы должны передать только идентификатор карты «**CardId**» и «**SecureCode**».

```
Payture.Pay({  
  ...  
  Data : {  
    CardId : "1234",  
    SecureCode : 123  
  }  
});
```

В случае непривязанной карты – необходимо передать идентификатор карты с значением «**FreePay**», номер карты, срок действия карты – месяц и год в соответствующих полях, имя держателя карты, CVV, а также флаг привязки карты «**AddCard**» с значением «**true**», если вы хотите привязать карты или «**false**», если нет.

```
Payture.Pay({  
  ...  
  Data : {  
    CardId : "FreePay",  
    CardNumber : "1234567890123456",  
    CardHolder : "TEST",  
    EMonth : 12,  
    EYear : 16,  
    SecureCode : 123,  
    AddCard : true  
  }  
});
```

Результат выполнения данного метода будет зависеть от типа шаблона. Если вы используете стандартный шаблон, вы будете либо возвращены на шаблон оплаты с ошибкой, либо перенаправлены на шаблон возврата пользователя. Если же вы используете JSON шаблон, вы остаетесь на странице оплаты и будут выполнены функции, описанные вами в параметрах «**OnSuccess**» и «**OnError**», в случае успеха или неуспеха операции соответственно.



6.2 Add

Данный метод позволяет производить оплату по карте. Для этого необходимо передать в метод тип шаблона – «Default» (стандартный) или «JSON», ключ – обычно значение скрытого поля «Key», и объект «Data», состоящий из параметров карты.

Если вы используете JSON шаблоны, необходимо так же передать в качестве параметров описание функций «OnSuccess», «OnSuccessAdd», «OnError» и «OnErrorAdd».

```
Payture.Add({
    TypeTemplate : "JSON",
    Key : $("[name=Key]").val(),
    Data : {
        CardNumber : "1234567890123456",
        CardHolder : "TEST",
        EMonth : 12,
        EYear : 16,
        SecureCode : 123
    },
    OnSuccess : function (redirectUrl) {
        // Тело функции
    },
    OnError : function (errorCode, canRetry, redirectUrl, key) {
        // Тело функции
    }
});
```

В результате выполнения данного метода, вы будете либо возвращены на шаблон оплаты с ошибкой, либо перенаправлены на шаблон возврата пользователя (если вы используете стандартный шаблон), или, если вы используете JSON шаблон, вы остаетесь на странице оплаты и будут выполнены функции, описанные вами в параметрах «OnSuccess»/«OnSuccessAdd» и «OnError»/«OnErrorAdd», в случае успеха или неуспеха операции соответственно.

6.3 Validate

Данный метод используется для проверки корректности вводимых значений. На вход принимает объект, состоящий из поля «Name», содержащего имя проверяемого поля (соответствует именам полей из набора «Data»), и поля «Value», содержащего значение проверяемого поля.

```
Payture.Validate({
    Name : "CardNumber",
    Value : "1234567890"
});
```

В результате метод возвращает объект с полями «Success» - успешно или не успешно пройдена валидация, «ErrorMessage» - текст ошибки (соответствует передаваемым ошибкам в параметре «TemplateErrors»), и «Element» - название элемента, согласно именам параметра «Data», к которому относится ошибка.

```
{
    Success : false,
    ErrorMessage : "Введите от 16 до 19 знаков номера карты",
    Element : "CardNumber"
}
```

6.4 RemoveCardBySession

Внимание! Данный метод используется только для интерфейса eWallet.

С помощью данного метода вы можете удалить привязанную карту покупателя из списка.

Для работы необходимо передать методу объект из двух параметров:

- **CardId** – идентификатор карты, которую необходимо удалить;

```
Payture.RemoveCardBySession
({ CardId: "1234",
  onRemove : function (CardId) {
    // Тело функции
  }
});
```



В результате успешного удаления карты будет вызвана функция, переданная в качестве параметра «**onRemove**». Если такой параметр не был передан, будет выполнена функция обновления списка карт по умолчанию.

6.5 GetList

Внимание! Данный метод доступен только для интерфейса eWallet.

Вы можете получить расширенный список карт с дополнительными параметрами, чтобы создать собственный сценарий построения списка карт. Для это в шаблоне оплаты в HTML код вставить следующий скрипт:

```
<script>
    {CardConfigurationJS}
</script>
```

После этого вызвать метод **GetList**.

```
Payture.GetList();
```

Метод обрабатывает данные карт и возвращает коллекцию карт. Ключом каждой карты является маскированный номер карты формата «123456xxxxxx1234», значением является объект, содержащий в себе «**Id**» - идентификатор карты, строка, флаг «**NoCVV**» - возможность оплаты по данной карте без ввода CVV, флаг «**Expired**» - является ли карта просроченной и переменная типа Date «**LastPay**» - дата последней успешной оплаты по карте.

```
{
    "123456xxxxxx1234" : {
        Id: "1234",
        NoCVV : true,
        Expired : False,
        LastPay : Date {Fri Dec 05 2014 12:01:10 GMT+0300 (MSK)}
    }
}
```



ПРИЛОЖЕНИЕ «Коды и тексты ошибок, результатов выполнения операций и текстов шаблонов»

ServerErrors

RU:

"default": "К сожалению, в настоящее время платеж с данной карты невозможен. Попробуйте оплатить другой картой",
"ACCESS_DENIED": "Запрещены операции с данным набором параметров для терминала",
"AMOUNT_ERROR": "Ошибка суммы операции. Превышена сумма либо сумма операции не проверена в биллинге",
"AMOUNT_EXCEED": "Сумма транзакции превышает доступный остаток средств на выбранном счете",
"API_NOT_ALLOWED": "Запрет использования API с данного IP проверьте реквизиты доступа",
"CARD_EXPIRED": "Истек срок действия карты",
"CARD_NOT_FOUND": "Не найдена карта по данному идентификатору",
"COMMUNICATE_ERROR": "Ошибка связи в физических каналах",
"CURRENCY_NOT_ALLOWED": "Валюта не разрешена для предприятия",
"CUSTOMER_NOT_FOUND": "Пользователь не найден",
"DUPLICATE_CARD": "Карта уже существует",
"DUPLICATE_ORDER_ID": "Заказ существует в Payture с данным идентификатором",
"DUPLICATE_PROCESSING_ORDER_ID": "Заказ существует в процессинге с данным идентификатором",
"DUPLICATE_USER": "Пользователь уже существует",
"EMAIL_ERROR": "Ошибка при обработке сообщения электронной почты ошибка отправки сообщения",
"EMPTY_RESPONSE": "Пустой ответ",
"FRAUD_ERROR": "Ошибка связанная с мошенничеством",
"FRAUD_ERROR_BIN_LIMIT": "Превышение оплат с 1 банка",
"FRAUD_ERROR_BLACKLIST_AEROPORT": "Аэропорт в черном списке мошеннических",
"FRAUD_ERROR_BLACKLIST_BANKCOUNTRY": "Страна банка в черном списке мошеннических",
"FRAUD_ERROR_BLACKLIST_USERCOUNTRY": "Страна пользователя в черном списке мошеннических",
"FRAUD_ERROR_CRITICAL_CARD": "Карта помечена как мошенническая",
"FRAUD_ERROR_CRITICAL_CUSTOMER": "Пользователь помечен как мошенник",
"FRAUD_ERROR_IP": "IP помечен как мошеннический",
"FRAUD_INTERNAL_ERROR": "Ошибка антифрод фильтра при обработке транзакции",
"ILLEGAL_ORDER_STATE": "Неверное состояние заказа",
"INTERNAL_ERROR": "Внутренняя ошибка шлюза",
"INVALID_PAYTUREID": "Неверный fingerprint устройства",
"INVALID_SIGNATURE": "Неверная подпись запроса",
"ISSUER_BLOCKED_CARD": "Карта заблокирована у эмитента",
"ISSUER_CARD_FAIL": "Эмитент не смог разрешить списание средств с карты",
"ISSUER_FAIL": "Эмитент не ответил на запрос",
"ISSUER_LIMIT_AMOUNT_FAIL": "Превышение лимитов на сумму у эмитента",
"ISSUER_LIMIT_COUNT_FAIL": "Превышение лимитов на количество операций у эмитента",
"ISSUER_LIMIT_FAIL": "Превышение лимитов у эмитента(без расшифровки)",
"ISSUER_TIMEOUT": "Эмитент не ответил в установленное время",
"MERCHANT_FORWARD": "Перенаправление на другой терминал",
"MERCHANT_RESTRICTION": "Запрет МПС или экваера на проведение операции мерчанту",
"MPI_CERTIFICATE_ERROR": "Ошибка сервиса MPI(шлюз)",
"MPI_RESPONSE_ERROR": "Ошибка сервиса MPI(МПС)",
"ORDER_NOT_FOUND": "Заказ не найден",
"ORDER_TIME_OUT": "Заказ просрочен",
"PAYMENT_ENGINE_ERROR": "Ошибка взаимодействия в ядре процессинга",
"PROCESSING_ACCESS_DENIED": "Доступ к процессингу запрещен",
"PROCESSING_ERROR": "Ошибка при процессинге платежа",
"PROCESSING_FRAUD_ERROR": "Процессинг отклонил мошенническую транзакцию",
"PROCESSING_TIME_OUT": "Не получен ответ от процессинга в установленное время",
"REFUSAL_BY_GATE": "Операция отклонена шлюзом",
"THREE_DS_ATTEMPTS_FAIL": "Попытка 3DS авторизации неудачна",
"THREE_DS_AUTH_ERROR": "Ошибка авторизации 3DS",
"THREE_DS_ERROR": "Ошибка оплаты 3DS",
"THREE_DS_FAIL": "Ошибка сервиса 3DS",
"THREE_DS_NOT_ATTEMPTED": "3DS не вводился",
"THREE_DS_NOTENROLLED": "Карта не вовлечена в систему 3DS",
"THREE_DS_TIME_OUT": "Превышено время ожидания 3DS",
"THREE_DS_USER_AUTH_FAIL": "Пользователь ввел неверный код 3DS",
"UNKNOWN_STATE": "Неизвестный статус транзакции",
"USER_NOT_FOUND": "Пользователь не найден",
"WRONG_AUTHORIZATION_CODE": "Неверный код авторизации",
"WRONG_CARD_INFO": "Неверная информация на карте",
"WRONG_CARDHOLDER": "Недопустимый кардхолдер",



"WRONG_CONFIRM_CODE": "Недопустимый код подтверждения",
"WRONG_CVV": "Недопустимый CVV",
"WRONG_EXPIRE_DATE": "Неправильная дата окончания срока действия",
"WRONG_PAN": "Неверный номер карты",
"WRONG_PARAMS": "Неверные параметры",
"WRONG_PAY_INFO": "Неверные карточные данные в шлюзе",
"WRONG_PHONE": "Неверный телефон",
"WRONG_USER_PARAMS": "Неверные параметры пользователя",
"OTHER_ERROR": "Ошибка которая произошла при невозможном стечении обстоятельств",
"AMOUNT_EXCEED_BALANCE": "К сожалению, в настоящее время платеж с данной карты невозможен. Пожалуйста, свяжитесь со своим банком и попробуйте еще раз, либо воспользуйтесь другой картой",
"AUTHENTICATION_ERROR": "К сожалению, в настоящее время платеж с данной карты невозможен. Пожалуйста, свяжитесь со своим банком и попробуйте еще раз, либо воспользуйтесь другой картой",
"AUTHORIZATION_TIMEOUT": "К сожалению, в настоящее время платеж с данной карты невозможен. Пожалуйста, свяжитесь со своим банком и попробуйте еще раз, либо воспользуйтесь другой картой",
"LIMIT_EXCHAUST": "Время, отведенное для ввода данных, исчерпано. Пожалуйста, оформите новый заказ",
"WRONG_CARD_PAN": "Неверный номер карты. Пожалуйста, повторите ввод данных",

EN:

"default": "Sorry, currently the payment with the card is impossible. Try another card",
"ACCESS_DENIED": "Operations with selected parameters are forbidden via terminal",
"AMOUNT_ERROR": "Transaction amount error. Amount exceeded or transaction amount not verified",
"AMOUNT_EXCEED": "Transaction amount exceeds available balance on selected account",
"API_NOT_ALLOWED": "Use of payment API forbidden from this IP. Please check access details",
"CARD_EXPIRED": "Card expired",
"CARD_NOT_FOUND": "Card with given ID not found",
"COMMUNICATE_ERROR": "Communication error in physical channels",
"CURRENCY_NOT_ALLOWED": "Currency forbidden for enterprise",
"CUSTOMER_NOT_FOUND": "User not found",
"DUPLICATE_CARD": "Card already exists",
"DUPLICATE_ORDER_ID": "Order exists in Payture with the following ID",
"DUPLICATE_PROCESSING_ORDER_ID": "Order exists in processing with the following ID",
"DUPLICATE_USER": "User already exists",
"EMAIL_ERROR": "Error processing email. Error sending message",
"EMPTY_RESPONSE": "Empty processing response",
"FRAUD_ERROR": "Anti-fraud error",
"FRAUD_ERROR_BIN_LIMIT": "Limit of payments via 1 bank exceeded",
"FRAUD_ERROR_BLACKLIST_AEROPORT": "Airport blacklisted for fraud",
"FRAUD_ERROR_BLACKLIST_BANKCOUNTRY": "Bank's country blacklisted for fraud",
"FRAUD_ERROR_BLACKLIST_USERCOUNTRY": "User's country blacklisted for fraud",
"FRAUD_ERROR_CRITICAL_CARD": "Card flagged for fraud",
"FRAUD_ERROR_CRITICAL_CUSTOMER": "User flagged for fraud",
"FRAUD_ERROR_IP": "IP flagged for fraud",
"FRAUD_INTERNAL_ERROR": "Anti-fraud filter error during payment processing",
"ILLEGAL_ORDER_STATE": "Invalid order status",
"INTERNAL_ERROR": "Internal gateway error",
"INVALID_PAYTUREID": "Invalid device fingerprint",
"INVALID_SIGNATURE": "Invalid request signature",
"ISSUER_BLOCKED_CARD": "Card blocked by issuing bank",
"ISSUER_CARD_FAIL": "Issuing bank could not allow debiting of card",
"ISSUER_FAIL": "No response from issuing bank",
"ISSUER_LIMIT_AMOUNT_FAIL": "Exceeded transaction limits with issuing bank",
"ISSUER_LIMIT_COUNT_FAIL": "Exceeded limits on number of transactions with issuing bank",
"ISSUER_LIMIT_FAIL": "Exceeded limits with issuing bank",
"ISSUER_TIMEOUT": "Issuing bank response timeout",
"MERCHANT_FORWARD": "Redirecting to another terminal",
"MERCHANT_RESTRICTION": "Merchant banned by acquiring bank or MPS",
"MPI_CERTIFICATE_ERROR": "MPI (gateway) service error",
"MPI_RESPONSE_ERROR": "MPI (MPS) service error",
"ORDER_NOT_FOUND": "Order not found",
"ORDER_TIME_OUT": "Order expired",
"PAYMENT_ENGINE_ERROR": "Interoperability error in processing core",
"PROCESSING_ACCESS_DENIED": "Processing access denied",
"PROCESSING_ERROR": "Error during transaction processing",
"PROCESSING_FRAUD_ERROR": "Processing rejected fraudulent transaction",
"PROCESSING_TIME_OUT": "Processing timeout error",
"REFUSAL_BY_GATE": "Transaction rejected by gateway",
"THREE_DS_ATTEMPTS_FAIL": "3DS authorisation unsuccessful",
"THREE_DS_AUTH_ERROR": "3DS authorisation error",
"THREE_DS_ERROR": "3DS payment error",
"THREE_DS_FAIL": "3DS service error",
"THREE_DS_NOT_ATTEMPTED": "3DS not entered",



"THREE_DS_NOTENROLLED": "Card does not use 3DS system",
"THREE_DS_TIME_OUT": "3DS timeout exceeded",
"THREE_DS_USER_AUTH_FAIL": "Incorrect 3DS code entered",
"UNKNOWN_STATE": "Unknown transaction status",
"USER_NOT_FOUND": "User not found",
"WRONG_AUTHORIZATION_CODE": "Incorrect authorisation code",
"WRONG_CARD_INFO": "Incorrect card details",
"WRONG_CARDHOLDER": "Invalid card holder",
"WRONG_CONFIRM_CODE": "Invalid authorisation code",
"WRONG_CVV": "Invalid CVV",
"WRONG_EXPIRE_DATE": "Incorrect valid through date",
"WRONG_PAN": "Incorrect card number",
"WRONG_PARAMS": "Incorrect parameters",
"WRONG_PAY_INFO": "Incorrect card data in gateway",
"WRONG_PHONE": "Incorrect telephone number",
"WRONG_USER_PARAMS": "Incorrect user parameters",
"OTHER_ERROR": "Error due to impossible combination of circumstances",
"AMOUNT_EXCEED_BALANCE": "Unfortunately, at present, the payment from the card is impossible. Please contact your Bank and try again, or use another card",
"AUTHENTICATION_ERROR": "Unfortunately, at present, the payment from the card is impossible. Please contact your Bank and try again, or use another card",
"AUTHORIZATION_TIMEOUT": "Unfortunately, at present, the payment from the card is impossible. Please contact your Bank and try again, or use another card",
"LIMIT_EXCHAUST": "Time allocated for input data is exhausted. Please place a new order",
"WRONG_CARD_PAN": "Invalid card number. Please repeat input",

AnswerText

RU:

"PAY_SUCCESS_TITLE": "Оплата прошла успешно!",
"PAY_ERROR_TITLE": "Оплата не прошла!",
"REDIRECT_TEXT": "Через несколько секунд вы будете перемещены на страницу магазина или нажмите
сюда чтобы перейти без ожидания",
"ADD_SUCCESS_TITLE": "Карта успешно привязана!",
"ADD_ERROR_TITLE": "Карта не привязана!"

EN:

"PAY_SUCCESS_TITLE": "The payment is successful",
"PAY_ERROR_TITLE": "The payment is unsuccessful",
"REDIRECT_TEXT": "Please wait a few seconds OR press here",
"ADD_SUCCESS_TITLE": "This card is successfully added to your account!",
"ADD_ERROR_TITLE": "Sorry, this card was not added to your account!"

TemplateText

RU:

"CARDHOLDER": "ДЕРЖАТЕЛЬ КАРТЫ",
"CARDNUMBER": "НОМЕР КАРТЫ",
"CARDHOLDERP": "IVAN IVANOV",
"CARDNUMBERP": "4242 4242 4242 4242",
"CARDNUMBERP_AMEX": "3434 343434 34343",
"UNDERCVV": "Последние три цифры на обратной стороне",
"UNDERCVVAMEX": "Последние цифры на обратной стороне",
"MONTH": "МЕСЯЦ",
"MONTHP": "ММ",
"YEAR": "ГОД",
"YEARP": "ГГ",
"PHONE": "Номер телефона",
"VALID": "VALID
THRU",
"ADDCARD": "Привязать",
"DELET CARD": "Удалить карту",
"PAYCARD": "Оплатить",
"ADDCHECKBOX": "Запомнить карту",
"DELETEALERT": "Вы действительно хотите удалить карту",
"ERRORALERT": "Произошла ошибка",
"SELECTCARD": "Выберите карту",
"DEFAULTCARD": "другая",
"NEEDEDFIELDS": "Все поля обязательны для заполнения",
"SECURITYTEXT": "Безопасность платежей гарантируется использованием
TLS v1.2 протокола для передачи конфиденциальной информации.",
"PAYMENTDESCRIPTION": "Покупка {product} на сумму {total} руб.",
"PAYMENTDESCRIPTIONNPF": "Пополнение договора ИПП № {contract} на сумму {total} руб.",
"RETURN_TITLE_SUCCESS": "Завершение",



```
"RETURN_TITLE_ERROR": "Ошибка",
"RETURNTEXT": "<br>Через несколько секунд вы&nbsp;&nbsp;&nbsp;будете перемещены на&nbsp;&nbsp;&nbsp;страницу магазина или&nbsp;&nbsp;&nbsp;нажмите
<a href=>сюда</a> чтобы перейти без ожидания",
"GIFT": "Получить подарок",
"WELCOME": "Введите реквизиты карты:",
"MORE": "Узнать больше",
"NPF": "Подключить регулярные платежи"
```

EN:

```
"CARDHOLDER": "CARDHOLDER",
"CARDNUMBER": "CARD NUMBER",
"CARDHOLDERP": "IVAN IVANOV",
"CARDNUMBERP": "4242 4242 4242 4242",
"CARDNUMBERP_AAMEX": "3434 343434 34343",
"UNDERCVV": "The last three digits on the reverse side",
"UNDERCVVAMEX": "The last digits on the reverse side",
"MONTH": "Month",
"MONTHP": "MM",
"YEAR": "Year",
"YEARP": "YY",
"PHONE": "Phone number",
"VALID": "VALID<br>THRU",
"ADDCARD": "Add",
"DELETECARD": "Delete this card",
"PAYCARD": "Pay",
"ADDCHECKBOX": " Remember this card",
"DELETEALERT": "Are you sure you want to remove this card",
"ERRORALERT": "An error occurred",
"SELECTCARD": "Select card",
"DEFAULTCARD": "other",
"SECURITYTEXT": "Safety is ensured by using TLS v1.2 for private data transfer",
"NEEDEDFIELDS": "All fields are mandatory",
"PAYMENTDESCRIPTION": "Purchase {product} for the sum {total} rubles",
"PAYMENTDESCRIPTIONNPF": "Пополнение договора ИПП № <b>{contract}</b> на сумму <b>{total}</b> руб.</b>",
"PAYMENTDESCRIPTIONNPF": "",
"RETURN_TITLE_SUCCESS": "Success",
"RETURN_TITLE_ERROR": "Error",
"RETURNTEXT": "<br>You will be redirected to the merchant page in a few seconds or click <a href=>here</a>",
"GIFT": "Get a gift",
"WELCOME": "Enter card details",
"MORE": "Read more",
"NPF": "Connect regular payments"
```

TemplateErrors

RU:

```
"PanErrors": "В номере карты допущены ошибки",
"EmptyPan": "Введите от 16 до 19 знаков номера карты",
"EmptyDate": "Укажите дату, до которой действительна карта",
"EmptyMonth": "Укажите месяц, до которого действительна карта",
"EmptyYear": "Укажите год, до которого действительна карта",
"WrongDate": "Неверно указана дата",
"WrongEmail": "Неверно указан E-mail",
"WrongPhone": "Неверно указан Телефон",
"WrongFIO": "Неверно заполнено Ф.И.О.",
"EmptyCardHolder": "Укажите имя владельца карты",
"EmptyCVV": "Укажите код CVV"
```

EN:

```
"PanErrors": "Incorrect card number",
"EmptyPan": "Input from 16 to 19 digits of the card number",
"EmptyDate": "Select card expiration date",
"EmptyMonth": "Select card expiration month",
"EmptyYear": "Select card expiration year",
"WrongDate": "The date is wrong",
"WrongEmail": "The E-mail is wrong",
"WrongPhone": "The Phone is wrong",
"WrongFIO": "The FIO is wrong",
"EmptyCardHolder": "Select cardholder name",
"EmptyCVV": "Select CVV"
```