EasyPay-Provider V3.1

История изменений.

Дата/Версия	Изменения	Стр.
01.10.2019	Добавлено информационное поле Info.	5, 10, 15
22.04.2020	Добавлен блок финансового мониторинга FinMon.	4, 13, 18

Содержание

1 Описание

Протокол взаимодействия реализован на основе интернет протокола HTTPS. Данные передаются в xmlформате, используя метод POST. Для защиты канала передачи данных используется SSL-сертификат на стороне сервера (провайдера). Все запросы и ответы подписываются по алгоритму RSA (SHA1) с длиной ключа 1024, соответственно стороны (Провайдер, EasyPay) обмениваются сертификатами в формате *.PEM. На данный момент подпись является необязательной!

1.1 Общая структура

Запрос

```
<Request>
        <DateTime>yyyy-MM-ddTHH:mm:ss</DateTime>
        <Sign></Sign>
        <Operation>
                <Parameter1/>
                 ...
                <ParameterN/>
        </Operation>
</Request>
```

Ответ

```
<Response>
        <StatusCode>int</StatusCode>
        <StatusDetail>string</StatusDetail>
        <DateTime>yyyy-MM-ddTHH:mm:ss</DateTime>
        <Sign></Sign>
        <Parameter1/>
        ...
        <ParameterN/>
</Response>
```

Где:

DateTime – время создания закпроса(ответа);
Sign – цифровая подпись в виде HEX-строки(AF4ED0…);
Operation – тип операции (список доступных операций приведен ниже);
StatusCode – код состояния (если запрос выполнен успешно то код 0, если ошибка то меньше нуля);
StatusDetail – описание результата операции.

2 Типы запросов

- Check (проверка пользователя, счета);
- Payment (создание заказа);
- Confirm (подтверждение заказа);
- Cancel (отмена платежа);

2.1 Проверка пользователя

Запрос

```
<Request>
        <DateTime>yyyy-MM-ddTHH:mm:ss</DateTime>
    <Sign></Sign>
    <Check>
            <ServiceId>int</ServiceId>
            <Account>string</Account>
    </Check>
</Request>
```

Ответ

```
<Response>
    <StatusCode>int</StatusCode>
    <StatusDetail>string</StatusDetail>
    <DateTime>yyyy-MM-ddTHH:mm:ss</DateTime>
    <Sign></Sign>
    <AccountInfo>
            <Parameter1>string</Parameter1>
            …
            <ParameterN>string</ParameterN>
    </AccountInfo>
    <FinMon>…</FinMon>
</Response>
```

ServiceId – номер вашего сервиса в нашей системе;
Account – идентификатор пользователя (номер телефона, номер договора, …);
AccountInfo – пользовательская информация, которая отображается на экране. Значение параметров – должны быть на английском языке (Эти параметры нам сообщаются при создании сервиса). Ввиду ограничения отображаемой информации, количество параметров не должно быть больше 4-х;
FinMon – блок финансового мониторинга (не обязательный, см п.8.4).

2.2 Создание заказа

Запрос

```
<Request>
    <DateTime>yyyy-MM-ddTHH:mm:ss</DateTime>
    <Sign></Sign>
    <Payment>
            <ServiceId>int</ServiceId>
            <OrderId>long</OrderId>
```

```
            <Account>string</Account>
            <Amount>decimal</Amount>
        </Payment>
</Request>
```

Ответ

```
<Response>
        <StatusCode>int</StatusCode>
        <StatusDetail>string</StatusDetail>
        <DateTime>yyyy-MM-ddTHH:mm:ss</DateTime>
        <Sign></Sign>
        <PaymentId>string</PaymentId>
</Response>
```

OrderId – это наш уникальный идентификатор транзакции (тип long 8-байт);
Amount – сумма платежа(в формате NN.NN, пример 10.50);
PaymentId – номер платежа в вашей системе.

2.3 Подтверждение заказа

Запрос

```
<Request>
        <DateTime>yyyy-MM-ddTHH:mm:ss</DateTime>
        <Sign></Sign>
        <Confirm>
            <ServiceId>int</ServiceId>
            <PaymentId>string</PaymentId>
        </Confirm>
</Request>
```

Ответ

```
<Response>
        <StatusCode>int</StatusCode>
        <StatusDetail>string</StatusDetail>
        <DateTime>yyyy-MM-ddTHH:mm:ss</DateTime>
        <Sign></Sign>
        <OrderDate>yyyy-MM-ddTHH:mm:ss</OrderDate>
        <Info>string</Info>
</Response>
```

OrderDate – дата, которая фиксируется у вас как дата совершения платежа (списание денег), она же проходит по бухгалтерии и финансовым документам.
Info – текстовая информация для печати в чеке (не обязательный).

2.4 Отмена заказа

Запрос

```
<Request>
        <DateTime>yyyy-MM-ddTHH:mm:ss</DateTime>
        <Sign></Sign>
        <Cancel>
                <ServiceId>int</ServiceId>
                <PaymentId>string</PaymentId>
        </Cancel>
</Request>
```

Ответ

```
<Response>
        <StatusCode>int</StatusCode>
        <StatusDetail>string</StatusDetail>
        <DateTime>yyyy-MM-ddTHH:mm:ss</DateTime>
        <Sign></Sign>
        <CancelDate>yyyy-MM-ddTHH:mm:ss</CancelDate>
</Response>
```

CancelDate – дата отмены заказа.

3 Сверка платежей

Поставщику услуг раз в сутки отправляется на email(указанный при подключении) реестр платежей в формате .csv.
Формат файла:

OrderId;PaymentId;ServiceId;Account;Amount;OrderDate;
11;7891123;223;4589687;45.50;2010-05-02T14:05:30; 12;4139874;497;3257879;5.00;2010-05-02T20:05:30;
14;5478877;544;1121458;15.00;2010-05-02T21:08:30;

4 Цифровая подпись

Для подписи запроса вы должны использовать все тело запроса с пустым тегом <Sign></Sign>:

Запрос:

```
<Request>
        <DateTime>yyyy-MM-ddTHH:mm:ss</DateTime>
    <Sign></Sign>
    <Check>
            <ServiceId>int</ServiceId>
            <Account>string</Account>
    </Check>
</Request>
```

Подпись: RSA1024(SHA1(Запрос))

Для проверки подписи используется аналогичная процедура:

Ответ:

```
<Response>
    <StatusCode>int</StatusCode>
    <StatusDetail>string</StatusDetail>
    <DateTime>yyyy-MM-ddTHH:mm:ss</DateTime>
    <Sign></Sign>
</Response>
```

5 Схемы оплаты

5.1 Прямое пополнение

1 Проверка возможности платежа(проверка абонента)

```
<Request>
        <DateTime>2010-09-01T12:00:00</DateTime>
    <Sign>94C8682CABCF1D…</Sign>
    <Check>
            <ServiceId>100</ServiceId>
            <Account>12345678</Account>
    </Check>
</Request>

<Response>
    <StatusCode>0</StatusCode>
    <StatusDetail>OK</StatusDetail>
    <DateTime>2010-09-01T12:00:05</DateTime>
    <Sign>47EDE8324D05CC1…</Sign>
    <AccountInfo>
            <Name>Иванов А.А.</Name>
          <Address>ул. Садовая 5, кв. 16</Address>
            <Balance>125.00</Balance>
    </AccountInfo>
</Response>
```

Если проверка выполнена успешно переходим на шаг 2

2 Создание заказа

```
<Request>
        <DateTime>2010-09-01T12:00:10</DateTime>
    <Sign>2EE9485D8747963E…</Sign>
    <Payment>
            <ServiceId>100</ServiceId>
          <OrderId>11</OrderId>
            <Account>12345678</Account>
            <Amount>25.00</Amount>
    </Payment>
</Request>

<Response>
    <StatusCode>0</StatusCode>
    <StatusDetail>Order Created</StatusDetail>
    <DateTime>2010-09-01T12:00:15</DateTime>
    <Sign>D3479BB615EFF…</Sign>
    <PaymentId>145</PaymentId>
</Response>
```

В случае успеха переходим на подтверждение платежа.

3 Подтверждение заказа

```
<Request>
        <DateTime>2010-09-01T12:00:20</DateTime>
    <Sign>615EFFD1D9E02BD…</Sign>
```

```
        <Confirm>
                <PaymentId>145</PaymentId>
        </Confirm>
</Request>

<Response>
        <StatusCode>0</StatusCode>
        <StatusDetail>Payment Confirmed</StatusDetail>
        <DateTime>2010-09-01T12:00:25</DateTime>
        <Sign>E4E54DD77C157F…</Sign>
          <OrderDate>2010-09-01T12:00:25</OrderDate>
        <Info>25 UAH credited to your account. Balance 150 UAH.</Info> </Response>
```

При подтверждении происходит списание средств со счета партнера. Если во время подтверждения произошол timeout – подтверждение может повторится, при этом должно возвращатся то же, что и в первый раз(OrderDate – при этом не изменяется).

Если добавить не обязательный тег Info, то текстовую строку из него можно распечатать в чеке.

6 Офлайновые платежи

6.1 Формат данных

Информацию о своих абонентах (если такая имеется) провайдер должен передавать в следующем формате:

```
<Clients>
        <Client>
                <Account>string</Account>
                <AccountInfo>
                        <Name>string</Name>
                        <Address>string</ Address >
                        …
                </AccountInfo>
        </Client>
        …
</ Clients>
```

Account – параметр, по которому осуществляется поиск в базе абонентов;
AccountInfo – любая информация о пользователе, необходимая, например, для вывода на терминале

Файл именуется как clients-304.xml, где 304 – номер сервиса (услуги), который сообщается провайдеру при заведении сервиса в системе.

7 Комунальные платежи

Оплата коммунальных платежей основывается на получении платёжной информации, по лицевому счёту или другому идентификатору. Заполнение счётчиков, вычисление льгот, ну и создание платежа на основании введённых данных.

Существует два подхода по оплате счетов:

- Набор услуг оплачиваются как один сервис;
- Каждая услуга оплачивается как отдельный сервис.

Соответственно, если при проверке в Products пришёл один тег Product – то оплачивается все как одна услуга. Если больше одного, то создаётся отдельно столько платежей, сколько услуг пришло с разными идентификаторами ServiceId.

7.1 Получение информации о платеже

Для получения информации о платеже, нужно передать Account(жек, лицевой счет, баркод, и т.д.), значение которого будет зависеть от конкретной услуги. Если идентификаторов два или более, то все они передаются в операции Check – отдельными тегами. Например: <Bill>, <Jek>.

Запрос

```
<Request>
	<DateTime>yyyy-MM-ddTHH:mm:ss</DateTime>
	<Sign>Sign</Sign>
	<Check>
		<ServiceId>int</ServiceId>
		<Account>string</Account>
	</Check>
</Request>
```

Ответ

```
<Response>
	<StatusCode>int</StatusCode>
	<StatusDetail>string</StatusDetail>
	<DateTime>yyyy-MM-ddTHH:mm:ss</DateTime>
	<Sign>Sign</Sign>
	<Products>
		<Product>
			<ProductId>string</ProductId>
			<Address>string</Address>
			<Account>string</Account>
			<UserName>string</UserName>
			<Name lang="ru">string</Name>
			<Name lang="uk">string</Name>
			<Date>yyyy-MM-dd</Date>
			<Amount>decimal</Amount>
			<PaymentAmount>decimal</ PaymentAmount>
			<Amount.Min>decimal</Amount.Min>
			<Debt>decimal</Debt>
			<Parameters/>
			<BankingDetails/>
```

```
                    <Meters>
                        <Meter>
                            <Id>string</Id>
                             <Name>string</Name>
                            <Tariff>decimal</Tariff>
                            <Unit>string</Unit>
                             <OldValue>decimal</OldValue>
                            <Privileges>
                                <Privilege>
                                    <Units>decimal</Units>
                                     <Percent>decimal</Percent>
                                </Privilege>
                            </Privileges>
                        </Meter>
                    </Meters>
                </Product>
                …
            </Products>
            <FinMon>…</FinMon>
</Response>
```

где:

Address - желтым цветом помечены необязательные теги
Products – список возможных услуг;
Product – услуга (содержит дополнительную информацию о платеже), если приходит несколько услуг, то для каждой услуги будет создавать новый платёж;
BankingDetails – банковские реквизиты(содержит информацию об получателе платежа), если этот параметр не пустой, то он передаётся в платеже. Этот параметр также может передаваться в не тега Product, но у тега больше приоритет. Детальней см. пункт 8.1
ProductId – динамический номер услуги (нужен если в пределах одного сервиса есть несколько вложенных услуг), если этот параметр не пустой, то передаётся в платеже;
Address – почтовый адрес, плательщика;
Account – значение параметра Account, которое нужно указывать при платеже;
UserName – имя плательщика;
Name – название услуги, может быть на разных языках;
Date – дата начисления;
Amount.Min – минимальная сумма к оплате;
Amount.Max – максимальная сумма к оплате;
Amount – сумма начислено, без учёта счётчиков;
Debt – долг, если значение отрицательное то переплата;
PaymentAmount – Сумма которую предлагать оплатить клиенту, используется только на сайте. По умолчанию используется Amount.
Parameters – дополнительная информация об услуге;

Meters – счётчики;
Meter.Id – идентификатор счётчика;
Meter.Name – название;
Meter.Tariff – тариф;
Meter.Unit – единицы измерения (кВт., куб…);
Meter.OldValue – последнее значение счётчика, значение может быть пустым;

Privileges – льготы по услуге;
Privilege.Units – количество единиц, на которые действует льгота; Privilege.Percent – процент, который нужно оплатить;
FinMon – блок финансового мониторинга (не обязательный, см п.8.4), один для всех услуг.

7.2 Создание платежа

При создании платежа, передаётся SessionId а также параметр PaymentInfo, с заполненными значениями счётчиков (если такие присутствуют) или пустой тег <PaymentInfo/>. Дополнительно передаются ProductId и BankingDetails, если они были указаны при проверке платежа. Свойство Amount – в счётчиках указывается с учётом льгот.

Запрос

```
<Request>
        <DateTime>yyyy-MM-ddTHH:mm:ss</DateTime>
        <Sign>Sign</Sign>
        <Payment>
                <ServiceId>int</ServiceId>
                <ProductId>string</ProductId>
                <OrderId>long</OrderId>
                <Account>string</Account>
                <Amount>decimal</Amount>
                <PaymentInfo>
                        <Meters>
                                <Meter>
                                        <Id>string</Id>
                                        <OldValue>decimal</OldValue>
                                        <NewValue>decimal</NewValue>
                                        <Amount>decimal</Amount>
                                </Meter>
                                <Meter>
                                        <Id>string</Id>
                                        <OldValue>decimal</OldValue>
                                        <NewValue>decimal</NewValue>
                                        <Amount>decimal</Amount>
                                </Meter>
                        </Meters>
                </PaymentInfo>
        </Payment>
</Request>
```

Ответ

```
<Response>
        <StatusCode>int</StatusCode>
        <StatusDetail>string</StatusDetail>
        <DateTime>yyyy-MM-ddTHH:mm:ss</DateTime>
        <Sign>Sign</Sign>
        <PaymentId>long</PaymentId> </Response>
```

NewValue – новое значение счетчика.

7.3 Подтверждение платежа

Подтверждение платежа происходит по стандартной схеме

Запрос

```
<Request>
```

```
        <DateTime>yyyy-MM-ddTHH:mm:ss</DateTime>
        <Sign></Sign>
        <Confirm>
                <ServiceId>int</ServiceId>
               <PaymentId>string</PaymentId>
        </Confirm>
</Request>
```

Ответ

```
<Response>
        <StatusCode>int</StatusCode>
        <StatusDetail>string</StatusDetail>
        <DateTime>yyyy-MM-ddTHH:mm:ss</DateTime>
        <Sign></Sign>
          <OrderDate>yyyy-MM-ddTHH:mm:ss</OrderDate>
         <Info>string</Info>
</Response>
```

8 Дополнительные параметры

8.1 Банковские реквизиты

Дополнительный параметр в операции Check. Возвращается провайдером, когда платежи на один сервис имеют разные банковские реквизиты.

```
<BankingDetails>
        <Payee>
                <Id>ЕГРПОУ или ИНН получателя</Id>
                <Name>Название или имя получателя</Name>
                <Bank>
                        <Name>Название банка получателя</Name>
                        <Mfo>МФО получателя</Mfo>
                        <Account>счет получателя</Account> </Bank>
        </Payee>
        <Payer/>
        <Narrative>
                <Name>Назначение платежа в формате точно из договора</Name>
                <Vat>20</Vat> <!--НДС, если не берется то 0--> </Narrative>
</BankingDetails>
```

Чтобы принимать оплату на нескольких получателей и распределить платеж на несколько получателей, в дополнение к основным банковским реквизитам на операцию Check нужно передать в ответе структуру.

```
<AdditionalPayments>
    <AdditionalPayment>
        <BankingDetails>
            ---------- стандартная структура ----------
</BankingDetails>
        <Rule>
          <Unit>
            ------------------- может быть одно из двух значений "Amount" или "Percent" ----------------------  </Unit>
          <Value>
            --------------------- число формата 0.00 ------------------
</Value>
        </Rule>
    </AdditionalPayment>
    <AdditionalPayment>
        .....................
    </AdditionalPayment>
    ......................
</AdditionalPayments>
```

Выполняем расщепление платежа на вложенные банковские реквизиты согласно правилу для каждого AdditionalPayment.
Правило- это набор из двух элементов:

- Unit - признак того как будет делиться основная сумма. "Amount" - означает что будет вычитаться значение. "Percent" - браться процент от суммы транзакции.
- Value - величина либо в процентах, либо в деньгах.

8.2 Оригинальный номер сервиса

Дополнительный параметр в операции Check, возвращается, если для данного пользователь сервис пополнения отличный от того, который указан в запросе(можно указывать только те сервиса, которые заведены у нас в системе). Это бывает необходимо в ситуациях когда например в терминале сервис один, а у нас в системе для каждого города или филиала заведен свой(разные договора, банковские реквизиты). В этом случае операция покупки пройдет с новым ServiceId.

<OriginalServiceId>int</OriginalServiceId>

Пример:

```
<Request>
        <DateTime>2010-09-01T12:00:00</DateTime>
    <Sign>94C8682CABCF1D…</Sign>
    <Check>
            <ServiceId>100</ServiceId>
            <Account>12345678</Account>
    </Check>
</Request>

<Response>
    <StatusCode>0</StatusCode>
    <StatusDetail>OK</StatusDetail>
    <DateTime>2010-09-01T12:00:05</DateTime>
    <Sign>47EDE8324D05CC1…</Sign>
    <AccountInfo>
            <Name>Иванов А.А.</Name>
            <Address>ул. Садовая 5, кв. 16</Address>
            <Balance>125.00</Balance>
    </AccountInfo>
    <OriginalServiceId>101</OriginalServiceId>
</Response>
```

8.3 Передача суммы оплаты

Дополнительные параметры в операции Check, возвращаются, если для данного пользователя фиксированная сумма к оплате или есть нижняя граница и верхняя, в рамках которой должна быть принята оплата. Это бывает необходимо в ситуациях когда у пользователей например разные кридитные задолженности и он должен погасить сумму не менее указанной Y и не имеет права погасить больше чем X(или если у пользователей разные тарифы, и каждый должен платить по своему). В этом случае поставщик нам должен вернуть элементы Amount.Min и Amount.Max.

<Amount.Min>decimal</Amount.Min>

<Amount.Max>decimal</Amount.Max >

Пример:

```
<Request>
        <DateTime>2010-09-01T12:00:00</DateTime>
    <Sign>94C8682CABCF1D…</Sign>
    <Check>
            <ServiceId>100</ServiceId>
            <Account>12345678</Account>
```

```
        </Check>
</Request>

<Response>
        <StatusCode>0</StatusCode>
        <StatusDetail>OK</StatusDetail>
        <DateTime>2010-09-01T12:00:05</DateTime>
        <Sign>47EDE8324D05CC1…</Sign>
        <AccountInfo>
                <Name>Иванов А.А.</Name>
                <Address>ул. Садовая 5, кв. 16</Address>
                <Balance>125.00</Balance>
        </AccountInfo>
        <Amount.Min>125.00</Amount.Min>
        <Amount.Max>125.00</Amount.Max>
</Response>
```

*Если значения Amount.Min и Amount.Max равны клиент сможет оплатить только указаную сумму в значениях этих параметров, если нет тога в диапазоне от Amount.Min до Amount.Max

8.4 Блок финансового мониторинга

В соответствии со ст. 14 Закон Украины О предотвращении и противодействии легализации (отмыванию) доходов, полученных преступным путем, финансированию терроризма и финансированию распространения оружия массового уничтожения (проект 2179) все переводы должны сопровождаться информацией о плательщике (инициатора перевода).

Требования статьи 14 не распространяются на операции по переводу средств на сумму, меньше чем 5000 гривен.

Перевод на сумму от 5000 до 30000 грн (29 999,99 грн), при предоставлении нужных параметров распространяется на платежи с целью оплаты стоимости товаров, работ, услуг, погашения задолженности по кредиту для зачисления на счет получателя.

Исключение:

Перевод средств с целью уплаты налогов, сборов, платежей, сборов на обязательное государственное пенсионное и социальное страхование, штрафных санкций и пени за нарушение законодательства в государственный и местные бюджеты, Пенсионный фонд, на счета органов государственной власти, органов местного самоуправления или перевода средств за жилищно-коммунальные услуги (до 29 999,99 грн без проведения доп. идентификации)

Если поставщик хочет, чтобы по сервису у клиента была возможность совершать платежи суммарно более чем 5000 гр-н в день (до 29 999,99 грн.), то поставщик должен нам вернуть на операции Check следующий блок:

```
<FinMon>
  <FullName>Иванов Иван Иванович</FullName> - поле обязательно,
```

дальше один из параметров на выбор, но обязательно один должен быть заполнен:

<Id>1234567891</Id> - реєстраційний номер облікової картки платника податків

<Residence>Бровары, Киевская, 243, кв.14</Residence> - місце проживання

<Passport>АН 123456</Passport> - номер (та за наявності - серію) паспорта громадянина України

<BirthDate>10.01.1993, Житомир</BirthDate> - дата і місце народження

</FinMon>

Пример Check:

```
<Request>
  <DateTime>2020-04-22T09:07:18</DateTime>
  <Sign>…</Sign>
  <Check>
    <ServiceId>int</ServiceId>
    <Account>string</Account>
  </Check>
</Request>
```

```
<Response>
  <StatusCode>0</StatusCode>
  <StatusDetail>OK</StatusDetail>
  <DateTime>2020-04-22T09:07:18</DateTime>
  <Sign>…</Sign>
  <FinMon>
    <FullName>Иванов Иван Иванович</FullName>
    <Id>1234567891</Id>
    <Residence></Residence>
    <Passport>АН 123456</Passport>
    <BirthDate></BirthDate>
  </FinMon>
</Response>
```

9 Дополнение А

9.1 Создание сертификата

Для создания сертификата подписи, можно воспользоваться бесплатной утилитой openssl.

```
openssl genrsa -out provider.ppk 1024
openssl req -new -key provider.ppk -out provider.req
openssl x509 -req -days 730 -in provider.req -signkey provider.ppk -out provider.cer
```

При создании сертификата подписи используется CN=ProviderSign-ProviderName, где ProviderName – название провайдера латинскими буквами (например, CN=ProviderSign-TopNet)

9.2 Подпись данных

Пример подписи на C#:

```
string path = "Provider-Test.pfx";
X509Certificate2 cert = new X509Certificate2(path,"test");
RSACryptoServiceProvider rsa = (RSACryptoServiceProvider)cert.PrivateKey;

byte[] signature = rsa.SignData(Encoding.UTF8.GetBytes(data), new SHA1CryptoServiceProvider()); string result =
Utilities.BytesToHex(signature);
```

9.3 Тестовый хост

Перед боевым запуском, Вы должны пройти тестирование на странице:

http://provider.easysoft.com.ua

9.4 Авторизационные данные на стороне провайдера

Провайдер должен заранее проинформировать о том, какого типа авторизация предусмотрена на их сервере (сертификаты, логин и пароль и т.д.)