SOFT-LOGIC

Paylogic

Шлюз создания шаблонов платежей keeper-payment-template. Программное обеспечение «Процессинговый центр Pay-logic».

Руководство администратора

АННОТАЦИЯ

Документ описывает настройку шлюза keeper-payment-template, предназначенного для создания шаблонов платежей

Версия руководства: 1.1

Руководство актуально для «Процессингового центра Pay-logic» версий 5.4.x

ООО «Софт-Лоджик»

656006, г. Барнаул, Малахова ул., дом 146в

Тел: (3852) 72-27-27

ОГЛАВЛЕНИЕ

ИСТОРИЯ ИЗМЕНЕНИЙ

ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 1.0

Дата публикации: 06.11.2020.

Изменение	Раздел
Общие улучшения:	
Документ создан	-

ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 1.1

Дата публикации: 24.04.2023.

Изменение	Раздел
Общие улучшения:	
Добавлена возможность передачи входного параметра ***multiservice-mode***	3.1, 3.2

1 ВВЕДЕНИЕ

Данное руководство описывает процесс настройки шлюза создания шаблонов платежей ***keeper-payment-template***. Документ адресован специалистам технической поддержки компании «Soft-logic».

Шлюз создания шаблонов платежей ***keeper-payment-template*** предназначен для регистрации шаблонов платежей в системе процессинга и работы с ними.

Шлюз создания шаблонов платежей работает по протоколу paylogic2 и позволяет:

- сохранять шаблоны платежей в процессинге;
- возвращать сохраненные шаблоны посредством advanced-запроса.

Описание структуры элемента `<advanced>`, позволяющего выполнить сложную проверку реквизитов платежа, приведено в документе «Протокол взаимодействия между внешней платежной системой и Процессинговым центром Pay-logic. Руководство разработчика».

Внимание!
Шлюз шаблонизации может быть запущен только для внешних агентов, подключаемых по протоколу PayLogic2.

2 НАСТРОЙКА ШЛЮЗА KEEPER-PAYMENT-TEMPLATE

2.1 КОНФИГУРАЦИОННЫЙ ФАЙЛ ШЛЮЗА

Пример настройки конфигурационного файла шлюза:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Корневая секция конфигурационного файла -->
<config>
 <servers>
<!-- Настройка адреса сервера -->
  <server host="192.168.0.28" port="8080" scheme="http"> </server>
 </servers>
<!-- Класс API шлюза -->
 <api class="ru.softlogic.processing.gates.keeper.payment.template.ProcessingTemplateApi"/>
<!-- Название логгера шлюза -->
 <logger name="template-processing"/>
<!-- Коды, при которых статус платежа будет считаться успешным, ошибочным
и неизвестным -->
 <codes success="0" error="1,300" process="" unknown="-10000"/>
<!-- Параметры работы шлюза -->
 <gate-config>
 <params>
<!-- Параметр позволяет маскировать значение атрибута id1. По умолчанию
false -->
  <param name="mask" param="false"/>
<!-- Параметр позволяет возвращать в атрибуте #sumpurchase сумму к
оплате, взятую из шаблона. По умолчанию true -->
  <param name="return-sumpurchase" param="true"/>
<!-- Параметр позволяет генерировать название шаблона из параметра name-
template, заданного в конфигурационном файле шлюза. Если значение
параметра false, то используется шаблон названия "Оплата <Название
сервиса>". По умолчанию false -->
  <param name="generate-template-name-by-config" param="false"/>
<!-- Параметр позволяет задать шаблон названия для шаблона платежа. Если
не задан, то используется шаблон названия "Оплата <Название сервиса>" -->
  <param name="name-template"
         param="Платеж на номер #id1# сервис #service_name#"/>
```

```
<!-- Параметр позволяет проверять PIN при запросе шаблонов. По умолчанию true -->
  <param name="validate-pin" param="true"/>
<!-- Параметр позволяет генерировать PIN при регистрации нового клиента. По умолчанию true -->
  <param name="generate-pin" param="true"/>
<!-- Параметр позволяет отправлять сгенерированный PIN по SMS. По умолчанию true -->
  <param name="send-sms-pin" param="true"/>
<!-- Параметр позволяет задать текст SMS для отправки сгенерированного PIN. Обязательный -->
  <param name="sms-message-template" param="Сгенерировал пин #pin#"/>
<!-- Параметр позволяет задать максимальное количество попыток восстановления пароля в течение часа -->
  <param name="max-recover-pin-attempts-count" param="2"/>
<!-- Параметр позволяет установить мультисервисный режим (запрос по функции login, возвращает шаблоны для всех сервисов) -->
  <param name="multiservice-mode" param="false"/>
<!-- Параметры позволяют удалять шаблоны, которые не использовались более чем количество дней, указанное в параметре amount-days-delete-old -->
  <param name="cron-delete-old" param="0 0 0 1 * ?"/>
  <param name="amount-days-delete-old" param="2000"/>
<!-- Параметры позволяют задать время ожидания, формат даты / времени и кодировку ответа -->
  <param name="timeout" param="20"/>
  <param name="date-format" param="yyyyMMddHHmmss"/>
  <param name="response-charset" param="utf-8"/>
 </params>
 </gate-config>
</config>
```

2.2 НАСТРОЙКА ШАБЛОНИЗАЦИИ В БЭК-ОФИСЕ

Для настройки шаблонизации в кабинете применяется произвольное свойство сервиса **keeper.template.save**. Свойство может иметь значение «Да» или «Нет», и выбор соответствующего значения в настройках сервиса будет определять, используется ли шаблонизация для данного сервиса.

Чтобы настроить шаблонизацию для сервиса в бэк-офисе, необходимо выполнить следующие действия:

1. Перейдите во вкладку «Справочники — Сервисы — Сервисы».
2. Перейдите в раздел редактирования сервиса, для которого необходимо использовать шаблонизацию.
3. В настройках сервиса перейдите во вкладку «Свойства сервиса» и выберите значение «Да» для произвольного свойства модуля шаблонизации **keeper.template.save**.

Шаблон сохраняется при успешном проведении платежа, если он не был сохранен ранее.

2.3 НАСТРОЙКА ШАБЛОНИЗАЦИИ НА ТПО И РМА

Для работы шаблонизации в интерфейсе ТПО должны быть реализованы экраны шаблонизации. Для РМА доработки не требуются.

3 ADVANCED-ЗАПРОСЫ

3.1 ФУНКЦИЯ LOGIN

Функция **login** позволяет выполнить поиск шаблонов по логину и паролю пользователя, сохранившему шаблоны.

Входные параметры функции:

1. ***keeper.login*** — логин пользователя для сохранения шаблона;
2. ***keeper.password*** — пароль пользователя.
3. ***multiservice-mode*** — необязательный параметр, позволяющий управлять мультисервисным режимом для поиска шаблонов. Возможные значения:
   1) ***true*** — мультисервисный режим включен. В ответе возвращаются шаблоны по всем сервисам, оплата которых возможна на точке, то есть сервис есть в профиле меню и в профиле вознаграждения Агента;
   2) ***false*** — мультисервисный режим выключен.

Если шаблоны найдены, то в ответе вернется селектор с идентификатором «templates».

Каждый элемент селектора содержит:

1. ***template-id*** — идентификатор шаблона;
2. ***#sumpurchase*** — сумма к оплате;
3. ***#dlast*** — дата последней оплаты в формате dd.MM.yy;
4. ***#service-id*** — идентификатор сервиса (обязательный в мультисервисном режиме);
5. Набор атрибутов — атрибуты, необходимые для оплаты. Например, ФИО клиента.

Пример запроса для функции **login**:

```
<request point="ID_точки">
 <advanced function="login" service="ID_сервиса">
  <attribute name="keeper.login" value="Логин_пользователя"/>
  <attribute name="keeper.password" value="Пароль_пользователя"/>
   <attribute name="multiservice.mode" value="true"/>
 </advanced>
</request>
```

Пример успешного ответа для функции **login**:

```
<response>
 <result code="0" service="0">
  <data>
   <nested id="services">
    <data>
     <input key="template-id" keyTitle="template-id" value="1"
            valueTitle="1" flags="0"/>
     <input key="#sumpurchase"
            keyTitle="#sumpurchase" value="100" valueTitle="100"
            flags="0"/>
     <input key="#dlast" keyTitle="#dlast" value="26.08.20"
            valueTitle="26.08.20" flags="0"/>
     <input key="#service-id" keyTitle="#service-id" value="1308"
            valueTitle="1308" flags="0"/>
     <input key="id1" keyTitle="id1" value="9609609600"
            valueTitle="9609609600" flags="0"/>
     <input key="fio" keyTitle="fio" value="Иванов Иван Иванович"
            valueTitle="Иванов Иван Иванович" flags="0"/>
    </data>
    <data>
     <input key="template-id" keyTitle="template-id" value="2"
            valueTitle="2" flags="0"/>
     <input key="#sumpurchase" keyTitle="#sumpurchase" value="200"
            valueTitle="200" flags="0"/>
     <input key="#dlast" keyTitle="#dlast" value="25.08.20"
            valueTitle="25.08.20" flags="0"/>
     <input key="#service-id" keyTitle="#service-id" value="1308"
            valueTitle="1308" flags="0"/>
     <input key="id1" keyTitle="id1" value="9609609696"
            valueTitle="9609609696" flags="0"/>
     <input key="fio" keyTitle="fio" value="Петров Петр Петрович"
            valueTitle="Петров Петр Петрович" flags="0"/>
```

```
    </data>
   </nested>
  </data>
 </result>
</response>
```

В случае неуспешного выполнения функции, в ответе могут быть получены ошибки:

1. Если имя пользователя или пароль неверен — «Not found»:

```
<result code="0" service="1">
```

2. Если перехвачено исключение внутри функции — «Provider answer error»:

```
<result code="0" service="3">
```

3. Если задано ошибочное имя функции — «Online not supported»:

```
<result code="0" service="4">
```

4. Если не найдены шаблоны — «No debts»:

```
<result code="0" service="5">
```

Пример ответа в случае ошибки:

```
<response>
  <result code="0" service="5">
</response>
```

3.2 ФУНКЦИЯ LOGIN-WITHOUT-AUTHOR

Функция **login-without-author** позволяет выполнить поиск шаблонов по логину пользователя, сохранившему шаблоны. Она работает аналогично функции **login**, но проверяет и возвращает шаблоны только по логину, не требуя пароль.

Входные параметры функции:

1. ***keeper.login*** — логин пользователя для сохранения шаблона.
2. ***multiservice-mode*** — необязательный параметр, позволяющий управлять мультисервисным режимом для поиска шаблонов. Возможные значения:
   1) ***true*** — мультисервисный режим включен. В ответе возвращаются шаблоны по всем сервисам, оплата которых возможна на точке, то есть сервис есть в профиле меню и в профиле вознаграждения Агента;
   2) ***false*** — мультисервисный режим выключен.

Если шаблоны найдены, то в ответе вернется селектор с идентификатором «templates». Каждый элемент селектора содержит:

1. ***template-id*** — идентификатор шаблона.
2. ***#sumpurchase*** — сумма к оплате.
3. ***#dlast*** — дата последней оплаты в формате dd.MM.yy.
4. ***#service-id*** — идентификатор сервиса (обязательный в мультисервисном режиме).
5. Набор атрибутов — необходимых для оплаты. Например, ФИО клиента.

Клиент может выбрать шаблон из найденных. После выбора шаблона все его атрибуты автоматически попадают в платеж. При необходимости клиент может вернуться назад по сценарию оплаты (стандартному) и изменить данные, полученные из шаблона.

Пример запроса для функции **login-without-author**:

```
<request point="id_точки">
 <advanced function="login-without-author" service="id_сервиса">
   <attribute name="keeper.login" value="Логин_пользователя"/>
   <attribute name="multiservice.mode" value="true"/>
 </advanced>
</request>
```

Пример успешного ответа для функции **login-without-author**:

```
<response>
 <result code="0" service="0">
  <data>
   <nested id="services">
    <data>
     <input key="template-id" keyTitle="template-id" value="3"
            valueTitle="3" flags="0"/>
     <input key="#sumpurchase" keyTitle="#sumpurchase" value="300"
            valueTitle="300" flags="0"/>
     <input key="#dlast" keyTitle="#dlast" value="27.08.20"
       valueTitle="27.08.20" flags="0"/>
     <input key="#service-id" keyTitle="#service-id" value="1591"
       valueTitle="1591" flags="0"/>
     <input key="id1" keyTitle="id1" value="9529529522"
       valueTitle="9529529522" flags="0"/>
     <input key="fio" keyTitle="fio" value="Сидоров Сидр Сидорович"
       valueTitle="Сидоров Сидр Сидорович" flags="0"/>
    </data>
    <data>
     <input key="template-id" keyTitle="template-id" value="4"
            valueTitle="4" flags="0"/>
     <input key="#sumpurchase" keyTitle="#sumpurchase" value="400"
            valueTitle="400" flags="0"/>
     <input key="#dlast" keyTitle="#dlast" value="27.08.20"
            valueTitle="27.08.20" flags="0"/>
     <input key="#service-id" keyTitle="#service-id" value="1591"
            valueTitle="1591" flags="0"/>
     <input key="id1" keyTitle="id1" value="9529525555"
            valueTitle="9529525555" flags="0"/>
     <input key="fio" keyTitle="fio" value="Семенов Семен Семенович"
            valueTitle="Семенов Семен Семенович" flags="0"/>
```

```
    </data>
   </nested>
  </data>
 </result>
</response>
```

В случае неуспешного выполнения функции, в ответе могут быть получены ошибки:

1. Если имя пользователя неверно — «Not found»:

```
<result code="0" service="1">
```

2. Если внутри функции перехвачено исключение — «Provider answer error»:

```
<result code="0" service="3">
```

3. Если задано ошибочное имя функции — «Online not supported»:

```
<result code="0" service="4">
```

4. Если не найдены шаблоны — «No debts»:

```
<result code="0" service="5">
```

Пример ответа в случае ошибки:

```
<response>
  <result code="0" service="5">
</response>
```

3.3 ФУНКЦИЯ PIN-RECOVER

В случае утери пароля пользователя, который создал шаблон платежа, возможно восстановить его с помощью функции **pin-recover**.

Входные параметры функции:

1. ***keeper.login*** — логин пользователя для сохранения шаблона. Восстановленный пароль придет в СМС-сообщении на номер, указанный в качестве значения параметра.

Пример запроса для функции **pin-recover**:

```
<request point="ID_точки">
 <advanced function="pin-recover" service="ID_сервиса">
  <attribute name="keeper.login" value="Логин_пользователя"/>
 </advanced>
</request>
```

Пример успешного ответа для функции **pin-recover**:

```
<result code="0" service="0">
```

В случае неуспешного выполнения функции, в ответе могут быть получены ошибки:

1. Если внутри функции было перехвачено исключение — «Provider answer error»:

```
<result code="0" service="3">
```

2. Если задано ошибочное имя функции — «Online not supported»:

```
<result code="0" service="4">
```

3. Если возникла ошибка при обновлении пароля в БД — «Illegal argument»:

```
<result code="0" service="6">
```

Пример ответа в случае ошибки:

```
<response>
  <result code="0" service="4">
</response>
```

3.4 ПРОВЕДЕНИЕ ПЛАТЕЖА

Запрос офлайн-проведения платежа должен содержать следующие атрибуты:

1. ***template-id*** — идентификатор шаблона. Атрибут проверяется на существование. Если он не задан, то создается новый шаблон. Если задан, то обновляется шаблон платежа;
2. ***keeper.template.name*** — наименование шаблона. Атрибут не является обязательным. Если не задан, то генерируется из настроек конфигурационного файла шлюза;
3. ***keeper.login*** — логин клиента;
4. Набор атрибутов — атрибуты, необходимые для оплаты. Например, ФИО клиента.

Пример запроса:

```
<payment id="14546" sum="1000" check="17235" service="2"
         account="9609609696" date="2007-10-12T12:00:00+0300">
  <attribute name="template-id" value="Идентификатор_шаблона"/>
  <attribute name="keeper.template.name" value="Наименование_шаблона"/>
  <attribute name="keeper.login" value="Логин_пользователя"/>
  <attribute name="fio" value="Сидоров Сидр Сидорович"/>
</payment>
```

Пример ответа в случае успешного проведения:

```
<result id="123" state="60" substate="0" code="0" ps_code="0" final="1"
        trans="123"/>
```

Пример ответа в случае ошибки:

```
<result id="123" state="80" substate="7" code="-2" ps_code="4" final="1"
        trans="123"/>
```