



Протокол динамического модуля ввода данных Pay-Logic Server Gate

Руководство разработчика

АННОТАЦИЯ

Содержит описание и техническую спецификацию динамического модуля ввода данных для протокола Pay-Logic Server Gate. Актуален для ПО «Процессинговый центр» 5.1.x и выше

Версия: 1.1.6

2008–2021 ООО «Софт-Лоджик», г. Барнаул, Россия

Данный документ входит в комплект поставки программных продуктов.

Права использования данного документа предусмотрены соответствующим лицензионным договором.

ООО «Софт-Лоджик»

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

Тел: (3852) 72-27-27

© *Soft-logic*

Web: <https://soft-logic.ru/>

Mail: info@soft-logic.ru

ОГЛАВЛЕНИЕ

ИСТОРИЯ ИЗМЕНЕНИЙ.....	5
ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 1.0.0.....	5
ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 1.0.0.....	5
ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 1.0.0.....	5
ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 1.0.1.....	6
ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 1.0.2.....	6
ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 1.0.3.....	7
ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 1.0.7.....	7
ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 1.0.8.....	8
ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 1.0.9.....	8
ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 1.1.0.....	9
ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 1.1.1.....	9
ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 1.1.2.....	10
ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 1.1.3.....	10
ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 1.1.4.....	10
ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 1.1.5.....	11
ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 1.1.6.....	11
НАЗНАЧЕНИЕ ДОКУМЕНТА.....	13
1 ТРАНСПОРТНЫЙ УРОВЕНЬ.....	14
1.1 БАЗОВЫЙ ТРАНСПОРТНЫЙ УРОВЕНЬ — DYNAMICREQUEST.....	14
1.2 РАСШИРЕННЫЙ ТРАНСПОРТНЫЙ УРОВЕНЬ — ADVANCED-ЗАПРОСЫ.....	16
1.3 РЕАЛИЗАЦИЯ НА ТРАНСПОРТНОМ УРОВНЕ. ПРОТОКОЛ «КИБЕРПЛАТ».....	20
2 ПРИКЛАДНОЙ УРОВЕНЬ.....	22
2.1 ЖИЗНЕННЫЙ ЦИКЛ СЕССИИ ПЛАТЕЖА.....	22
2.1.1 ОБЪЕДИНЕННЫЕ ФАЙЛЫ ФОРМ.....	24
2.2 МЕТОДЫ ПРОТОКОЛА.....	25
2.3 ДИАГРАММА ВЗАИМОДЕЙСТВИЯ КЛИЕНТА И ОБРАБОТЧИКА.....	27
2.4 ИНИЦИАЛИЗИРУЮЩИЙ ЗАПРОС — SELECTORVALUE START.....	28
2.5 ЗАПРОС ВЫПОЛНЕНИЯ ДЕЙСТВИЯ — ACTION.....	31

2.6 ЗАПРОСЫ ДЛЯ РАБОТЫ С XFORMS.....	37
2.7 ОБЪЕКТЫ ДАННЫХ ЭКРАНА.....	41
2.8 ОБЪЕКТЫ ДАННЫХ ОТВЕТА.....	53
2.9 ЭЛЕМЕНТ НАВИГАЦИИ — NAVIGATIONITEM.....	57
2.10 ВОЗВРАТ НА ПРОИЗВОЛЬНЫЙ ЭКРАН.....	65
2.11 ОБЩИЕ КОДЫ РЕЗУЛЬТАТОВ.....	74
2.12 ФЛАГИ ЭЛЕМЕНТОВ.....	75
2.13 ИЗМЕНЕНИЕ ПРОИЗВОЛЬНОГО ШАГА.....	77
2.14 ПРИМЕРЫ ЗАПРОСОВ.....	87
2.15 ПРИМЕРЫ РАЗЛИЧНЫХ ТИПОВ ПОЛЕЙ.....	96
2.15.1 ТЕКСТОВОЕ ПОЛЕ.....	96
2.15.2 ЧИСЛОВОЕ ПОЛЕ.....	98
2.15.3 СЕЛЕКТОР.....	100
2.15.4 ВЫБОР ДАТЫ.....	102
2.15.5 ТАБЛИЧНЫЙ СЕЛЕКТОР.....	104
2.15.6 СЦЕНАРИИ. РЕДИРЕКТ.....	107
2.15.7 СЦЕНАРИИ. КОРЗИНА ПЛАТЕЖЕЙ.....	113
2.15.8 ОТПРАВКА ЧАСТИЧНО ЗАПОЛНЕННОЙ SCREENDATA В ОТВЕТЕ.....	136
2.15.9 ПОЛУЧЕНИЕ ПЕРВОГО ЭКРАНА В ОТВЕТ НА ЗАПРОС START.....	140
2.15.10 ПОВТОРНОЕ ПОЛУЧЕНИЕ ПЕРВОГО ЭКРАНА.....	143
2.16 ПРИМЕР РАБОТЫ С МАСКАМИ.....	147
3 НАСТРОЙКА МИКРОСЕРВИСА ДИНАМИЧЕСКИХ ФОРМ.....	152
3.1 ПОРЯДОК НАСТРОЙКИ МИКРОСЕРВИСА ДИНАМИЧЕСКИХ ФОРМ.....	152
3.2 ОПИСАНИЕ КОНФИГУРАЦИОННОГО ФАЙЛА МИКРОСЕРВИСА ДИНАМИЧЕСКИХ ФОРМ.....	153
4 ПРИМЕР РАБОТЫ ДИНАМИЧЕСКОГО МОДУЛЯ ВВОДА ДАННЫХ.....	155
5 ПРИМЕР РАБОТЫ ДИНАМИЧЕСКОГО МОДУЛЯ ВВОДА ДАННЫХ ПО ПРОТОКОЛУ КИБЕРПЛАТ.....	165
5.1 АДРЕСА ДЛЯ НТТР-ЗАПРОСОВ.....	165
5.2 ИСПОЛЬЗОВАНИЕ ПРОТОКОЛА «КИБЕРПЛАТ».....	166
5.3 ПРИМЕР ПРОВЕДЕНИЯ ПЛАТЕЖА С ИСПОЛЬЗОВАНИЕМ ПРОТОКОЛА ДФ.....	168

ИСТОРИЯ ИЗМЕНЕНИЙ**ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 1.0.0**

Дата публикации: 12.01.2018.

Изменение	Раздел
Общие улучшения в документе:	
Документ создан	-

ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 1.0.0

Дата публикации: 06.03.2018.

Изменение	Раздел
Общие улучшения в документе:	
Документ дополнен, задокументированы базовые функции, подготовлены примеры	-

ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 1.0.0

Дата публикации: 20.03.2018.

Изменение	Раздел
Общие улучшения в документе:	
Редакционная правка, дополнение описаний алгоритмов	-

ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 1.0.1

Дата публикации: 06.04.2018.

Изменение	Раздел
Общие улучшения в документе:	
Добавлено описание настройки микросервисов	3

ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 1.0.2

Дата публикации: 20.11.2018.

Изменение	Раздел
Улучшения в версии 4.7.3:	
Обновлено описание взаимодействия	2.1
Добавлено описание стартового параметра redirectData	2.4
Добавлено описание параметра redirectData	2.5
Добавлено описание элемента экрана type	2.6
Добавлено описание элемента экрана actionsExt	2.6
Добавлено описание элемента экрана dialogs	2.6

Изменение	Раздел
Добавлено описание атрибутов поля (titleId, messageOff, messageOffId)	2.6
Добавлен новый общий код результата (7 - Редирект)	2.8
Добавлены примеры работы с редиректом	2.11.6
Добавлены примеры работы с корзиной	2.11.7
Работа с частичным запросом	2.11.8
Добавлено описание новых параметров конфигурации	3.2

ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 1.0.3

Дата публикации: 30.11.2018.

Изменение	Раздел
Улучшения в версии 4.7.3:	
Добавлен пример для работы модуля динамических форм по протоколу Киберплат	5
Добавлены описания расширенного транспортно уровня и соответствующих advanced запросов	1

ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 1.0.7

Дата публикации: 31.05.2019.

Изменение	Раздел
Улучшения в версии 4.7.3:	

Изменение	Раздел
Добавлен параметр code , описывающий код сервиса	2.4
Добавлен атрибут объекта img , описывающий имя изображения для пункта селектора	2.7

ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 1.0.8

Дата публикации: 19.08.2019.

Изменение	Раздел
Улучшения в версии 4.7.3:	
Добавлен атрибут объекта dataFormatter , определяющий форматирование данных	2.7
Добавлен атрибут объекта viewFormatter , определяющий форматирование значения при отображении	2.7
Добавлены объекты FieldFormatter , regexFieldModifierRules	2.7
Общие улучшения в документе:	
Добавлено описание атрибутов объекта valueTitleId , allowChangeCase , layoutNames , alert , messageOn , messageOnId	2.7

ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 1.0.9

Дата публикации: 30.09.2019.

Изменение	Раздел
Улучшения в версии кабинета 4.8.8:	

Изменение	Раздел
Добавлена возможность возврата с экрана оплаты на предыдущий экран	2.5
Добавлен параметр lastScreen , который характеризует последний экран формы/сценария, за которым следует экран оплаты	2.5

ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 1.1.0

Дата публикации: 23.10.2019.

Изменение	Раздел
Общие улучшения в документе:	
Добавлены ссылки на документацию, где приведено описание работы с масками	2.7
Дополнено описание параметра code , который является кодом сервиса	2.4

ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 1.1.1

Дата публикации: 13.11.2019.

Изменение	Раздел
Общие улучшения в документе:	
Добавлена возможность работы с объединенными файлами форм	2.1.1
Изменено описание раздела, добавлен пример	2.12.8

ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 1.1.2

Дата публикации: 29.11.2019.

Изменение	Раздел
Общие улучшения в документе:	
Добавлен пример работы с масками	2.13

ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 1.1.3

Дата публикации: 19.02.2021.

Изменение	Раздел
Общие улучшения в документе:	
Актуализирован раздел «Настройка микросервиса динамических форм»	3.1, 3.2

ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 1.1.4

Дата публикации: 14.05.2021.

Изменение	
Общие улучшения в документе:	
Исправлены ошибки	2.1, 2.5, 2.7, 2.8

ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 1.1.5

Дата публикации: 02.07.2021.

Изменение	
Общие улучшения в документе:	
Актуализирован раздел: исправлена обязательность полей для атрибутов объекта ScreenDate — decor и actionsExt . Исправлена обязательность полей для атрибута объекта ScreenField — type	2.7

ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 1.1.6

Дата публикации: 20.08.2021.

Изменение	
Улучшения в версии 1.3.0:	
В таблицу с параметрами запроса start добавлены параметры firstScreen и locale	2.4
Добавлен раздел «Получение первого экрана в ответ на запрос start »	2.15.9
Добавлен раздел «Повторное получение первого экрана»	2.15.10

Изменение	
<ol style="list-style-type: none">1. Переименован jar-файл.2. Стало необходимым подключение к базе данных.3. В параметре <dynamic-request-url> изменилось значение url	3.1
<p>В структуру конфигурационного файла приложения добавлены параметры для подключения к БД.</p> <p>Удалены параметры verify.host, verify.port, так как приложение стало использовать настройки шлюзов из БД</p>	3.2
<p>В таблицу с параметрами запроса на запрос action добавлены:</p> <ol style="list-style-type: none">1. Поле screen для указания экрана, где выполняется действие. <p>В таблицу с параметрами ответа на запрос action добавлены:</p> <ol style="list-style-type: none">1. Объект FeeSum, содержащий информацию о комиссии провайдера.2. Параметры multiplicitySum, minSum, maxSum.3. Объект navigations, возвращающий историю пройденных экранов	2.5
Добавлен раздел «Элемент навигации — NavigationItem »	2.9
Добавлен раздел «Возврат на произвольный экран»	2.10
В таблицу с атрибутами объекта ScreenData добавлен параметр ID (экрана)	2.7
Добавлен раздел «Изменение произвольного экрана»	2.13

НАЗНАЧЕНИЕ ДОКУМЕНТА

Документ предназначен для разработчиков информационных систем, интегрирующихся с процессингом Pay-Logic. Документ содержит описание протокола динамических форм, позволяющего в рамках своих возможностей абстрагироваться (полностью) как от бизнес-логики работы форм, сценариев процессинга Pay-Logic, так и от бизнес-логики работы конкретных поставщиков услуги. Протокол позволяет единообразно интегрировать любые услуги по любым сценариям оплаты, так как реализация инкапсулирует все вариации бизнес-логики внутри платежного процессинга, взаимодействие внешней системы во всех случаях остается единообразным и унифицированным.

Документ предполагает знакомство разработчика с основным протоколом «Pay-Logic Server Gate», знание концепции работы этого протокола. Протокол описан в документе [«Протокол взаимодействия между внешней платежной системой и Процессинговым Центром Pay-logic. «Руководство разработчика»](#). Технически, модуль динамического ввода данных является расширением к протоколу взаимодействия «Pay-Logic Server Gate».

Документ предполагает знание формата представления объектов JSON, умение работать с такими объектами.

1 ТРАНСПОРТНЫЙ УРОВЕНЬ

1.1 БАЗОВЫЙ ТРАНСПОРТНЫЙ УРОВЕНЬ — DYNAMICREQUEST

Базовый транспортный уровень применяется в случае, когда реализуемые сервисы динамического обработчика применяются для обработки стандартных форм и сценариев (любой сложности и независимо от нее) в связке с поставщиками услуг, работающих по классической схеме взаимодействия через запросы и ответы, и не использующие многошаговые и иные динамические структуры/структуры, требующие сохранения состояния на стороне поставщика. Базовый транспортный уровень применяется для всех тех форм и сценариев, которые реализуются в ТПО и РМА средствами мобильного, универсального и усовершенствованного обработчиков.

На базовом транспортном уровне для обмена пакетами используется протокол «Paylogic», новый тип запроса и ответа — `DynamicRequest` и `DynamicResponse`. Пакет может содержать строго один динамический запрос. Сам запрос и ответ по динамической форме представляет собой JSON-объект, который для целей передачи кодируется в `base64` и помещается в элемент `dynamicRequest` протокола «PayLogic». Аналогичным образом формируется и ответ сервера на запрос (`DynamicResponse`).

Пример запроса:

```
<request point="17235">
  <dynamicRequest>
    Eg0xMDM3NzAwMDg1NDQ0MR0wGAYIKoUDA4EDAQESDDAwNzcXNzEwNzk5MTEvM
    uy4g0KHRg9GJ0ZHqstGB0LrQuNC5INcy0LDQuyDQtC4gMTgxHzAdBgkqhkiG9
    cnlwdG9wcm8ucnUxCzAJBgNVBAYTA1JVMRkwFwYDVQQIDBDQsy4g0JzQvtGB0
    VQQHDAzQnNC+0YHQutCy0LAXJTAjBgNVBAoMHNce0J7QniAi0JrQoNCY0J/Qo
    JTAjBgNVBAMMHNCe0J7QniAi0JrQoNCY0J/QotCeLdCf0KDQniIwHhcNMTcwN
    MTgwNjE5MTAwNTA4WjCCASsxGDAWBgUqhQNkARINMTAyNzcOTIxMDYzMDEaM
  </dynamicRequest>
</request>
```

Пример ответа:

```
<response>
  <result code="0">
    Eg0xMDM3NzAwMDg1NDQ0MR0wGAYIKoUDA4EDAQESDDAwNzcxNzEwNzk5MTEvM
    uy4g0KHRg9GJ0ZHQstGB0LrQuNC5INCY0LDQuyDQtC4gMTgxHzAdBgkqhkiG9
    cnlwdG9wcm8ucnUxCzAJBgNVBAYTA1JVMRkwFwYDVQQIDBDQsy4g0JzQvtGB0
    VQQHDAzQnNC+0YHQutCy0LAXJTAjBgNVBAoMHNCE0J7QniAi0JrQoNCY0J/Qo
    JTAjBgNVBAMMHNCE0J7QniAi0JrQoNCY0J/QotCeLdCf0KDQniIwHhcNMTcwN
    MTgwNjE5MTAwNTA4WjCCASsxGDAWBgUqhQNkARINMTAyNzczOTIxMDYzMDUeAM
  </result>
</response>
```

Все остальные параметры запроса (точка, подпись запроса и подпись ответа) выполняются по правилам протокола «Paylogic2», и выполняются по спецификации [«PayLogicServerGate. Электронная подпись»](#).

1.2 РАСШИРЕННЫЙ ТРАНСПОРТНЫЙ УРОВЕНЬ — ADVANCED-ЗАПРОСЫ

Расширенный транспортный уровень применяется в случае, когда реализуемые сервисы динамического обработчика работают с провайдером услуг, который в свою очередь так же функционирует по некоторому динамическому протоколу, использует многошаговую схему на своей стороне. Технически, такой транспортный уровень применяется, когда для платежного процессинга существует необходимость обратиться к поставщику услуг, чтобы динамически получить от него описание полей (в его формате), и лишь затем можно сформировать очередной шаг. Такая схема применяется для отдельных поставщиков услуг (примером может служить агрегатор А-3), и тех случаев, когда на ТПО и РМА для услуг используются узкоспециализированные обработчики.

Для работы на расширенном транспортном уровне в протоколе Paylogic2 применяются функции advanced-запросов. Синтаксически запросы формируются по правилам и описанию advanced-запросов, согласно документации [«Протокол взаимодействия между внешней платежной системой и Процессинговым центром Pay-logic. Руководство разработчика»](#):

dynamic-request-implementation, start и action

Запрос **dynamic-request-implementation** не имеет параметров. Его назначение — дать внешней системе однозначный ответ о том, можно ли работать по расширенному транспортному уровню либо следует использовать стандартный. Ответом на запрос является статус результата и атрибут **dynamic-request-implementation-version**:

- Если запрос возвращает **serverError=0**, **serviceError=0** и атрибут **dynamic-request-implementation-version** со значением "1" — это сигнализирует о необходимости использовать расширенный транспортный уровень;

- Иной ответ сигнализирует о невозможности работы через расширенный транспортный уровень и необходимости использования базового транспортного уровня — запроса `dynamicRequest`.

Пример проверки:

```
<request point="236">
  <advanced function="dynamic-request-implementation " service="672" />
</request>
```

Ответ:

```
<response>
  <result code="0" service="0">
    <data>
      <input key="dynamic-request-implementation-version"\
        keyTitle="" value="1"
        valueTitle="" />
    </data>
  </result>
</response>
```

Параметры Advanced-запросов **start** и **action** — стандартные, для транспортной обертки протокола используется `input-element` с кодом **dynamic-request**. Ответ всегда содержит единственный атрибут с кодом **dynamic-response**.

Атрибуты содержат динамический запрос и ответ на этот динамический запрос соответственно в виде base-64, кодированного значения самого динамического запроса. Таким образом, расширенный транспортный уровень использует в качестве инкапсуляции запроса и ответа advanced-функции.

Advanced-функция **start** инициирует начало работы с динамическим сценарием и всегда инкапсулирует в себе метод `start` — старт сеанса, описанный в разделе 2.2. настоящего документа.

Пример запроса start:

```
<request point="236">
  <advanced function="start" service="672">
    <attribute name="dynamic-request" value="ewogICJpZFNLcnZpY2UiOiA2NzIsCiAgImlkUG9pbmQiOiAyMzYsCiAgInZlcmlmeSI6IGZhbHNlCn0=" />
  </advanced>
</request>
```

Advanced-функция **step** всегда инкапсулирует в себе метод **action** — выполнение действия по сценарию динамического обработчика, так же описанный в разделе 2.2 настоящего документа.

Пример запроса step:

```
<request point="236">
  <advanced function="step" service="672">
    <attribute name="dynamic-request" value="ewogICJpZFNLcnZpY2UiOiA2NzIsCiAgImlkUG9pbmQiOiAyMzYsCiAgInZlcmlmeSI6IGZhbHNlCn0=" />
  </advanced>
</request>
```

Ответом на обе функции всегда служит advanced-ответ, всегда содержащий единственный атрибут **dynamic-response**, который включает в себя ответ динамического обработчика по настоящему протоколу.

Пример запроса и ответа с использованием advanced-функций:

```
<request point="236">
  <advanced function="step" service="672">
    <attribute name="dynamic-request" value="ewogICJpZFNLcnZpY2UiOiA2NzIsCiAgImlkUG9pbmQiOiAyMzYsCiAgInZlcmlmeSI6IGZhbHNlCn0=" />
  </advanced>
</request>
Ответ

<response>
  <result code="0" service="0">
    <data>
      <input key="dynamic-response" keyTitle="" value="ewogICJjb2R
```

```
lIjogMCwKICAic2Vzc2lvdiiI6ICI4ZDcxZWVmMC1jNjFhLTQxOGUtYWQyYy1mN2RjYmZmNGF1
MTUiCn0="
        valueTitle="" />
    </data>
</result>
</response>
```

1.3 РЕАЛИЗАЦИЯ НА ТРАНСПОРТНОМ УРОВНЕ. ПРОТОКОЛ «КИБЕРПЛАТ»

Для интеграции систем, для которых протокол Киберплат является нативным, есть реализация динамического обработчика, использующего формат запросов Киберплат в качестве транспортного уровня. Принципиальное отличие реализации на протоколе Киберплат состоит в том, что применяется не base64, а URL-кодирование запроса протокола динамических форм, т. к. Киберплат работает с таким кодированием.

Запрос по протоколу динамических форм помещается в поле DYNAMIC_RQ запроса по протоколу Киберплат. Ответ по протоколу в формате динамических форм возвращается в поле DYNAMIC_RS ответа по протоколу Киберплат.

Для формирования значения поля DYNAMIC_RQ необходимо применить к тексту динамического запроса, описываемого далее в настоящем документе, URL-кодирование согласно формату [application/x-www-form-urlencoded](#), используя кодировку Windows-1251.

В случае успешного прохождения запроса в ответе Киберплат возвращаются поля ERROR=0, RESULT=0. При получении этих кодов нужно извлечь значение поля DYNAMIC_RS, декодировав его по тому же алгоритму [application/x-www-form-urlencoded](#) в кодировке Windows-1251. Полученное значение представляет JSON-ответ по протоколу динамических форм.

Пример исходного запроса динамических форм:

```
{"idService":672,"idPoint":4590,"verify":false}
```

В формате транспортного уровня Киберплат такой запрос имеет вид:

```
SESSION=1543224664571  
DYNAMIC_RQ=%7B%22idService%22%3A672%2C%22idPoint%22%3A4590%2C%22verify%22%3Afalse%7D
```

Ответ в формате Киберплат имеет вид:

```
ERROR=0  
RESULT=0  
DYNAMIC_RS=%7B%22code%22%3A0%2C%22session%22%3A%22a12bb89c-b42a-48d3-  
adc2-af35c0a5c985%22%7D
```

Ответ в ДФ:

```
{"code":0,"session":"a12bb89c-b42a-48d3-adc2-af35c0a5c985"}
```

2 ПРИКЛАДНОЙ УРОВЕНЬ

2.1 ЖИЗНЕННЫЙ ЦИКЛ СЕССИИ ПЛАТЕЖА

Все действия по разбору сценария и логике выполняет сервер. Валидация и обработка действий перехода по полям формы, шагам сценария (навигация) также выполняется сервером.

На основе формы или сценария сервер формирует последовательность шагов, которые необходимо пройти, и выдает эти шаги клиенту по порядку.

Концепция обработки сервисов на основе форм заключается в том, что каждое поле на форме представляет из себя отдельный шаг, на котором это поле клиенту возвращается вместе со списком дальнейших шагов «назад, далее» и т. д.

Концепция обработки сервисов на основе сценариев заключается в том, что каждый экран сценария представляет из себя отдельный шаг, на котором этот экран клиенту возвращается вместе со списком дальнейших шагов «назад, далее» и т. д. и при наличии диалоговых окон.

Задача клиента в этой схеме предельно упрощена и прозрачна. Клиент должен:

1. Обеспечить отображение возвращенных к вводу полей в соответствии с их типами и флагами, т. е. отобразить содержимое шага, возвращенного сервером.
2. Обеспечить ввод данных и их валидацию согласно правилам, возвращенным в составе полей. Правила валидации контекстно независимы (это простейшие регулярные выражения и ограничения по диапазонам). Валидация необязательная, так как сервер в любом случае провалидирует каждый шаг на своей стороне.
3. Отправить запрос для выполнения действия (из списка допустимых, возвращенных сервером).

4. Получить ответ с результатом обработки шага, на этом этапе:

- 1) либо цикл обработки замыкается (клиент получает следующий шаг);
- 2) либо обработка формы или сценария завершается и клиент получает соответствующий ответ и объект с результатом (на основе которого может формировать операцию);
- 3) либо возникает ошибка на шаге. В таком случае клиент в зависимости от ошибки остается на текущем шаге и получает информацию (что неверно) либо, если ошибка фатальна, обработка сценария завершается.

В случае нефатальной ошибки сервер возвращает:

1. Этот же шаг.
2. Введённые на нем данные (значения в полях).
3. Признак ошибки (в каких полях не пройдена валидация).
4. Диалоговые окна.
5. А также опционально различные дополнительные сообщения (к полям об ошибках, к шагу в целом).

2.1.1 ОБЪЕДИНЕННЫЕ ФАЙЛЫ ФОРМ

Доступна возможность работы с формами, которые хранятся в объединенных файлах вида *1xx...x.xml*. Чтение форм происходит в следующем порядке:

1. Файлы форм, в названии которых нет символа "x";
2. Файлы форм, в названии которых один символ "x";
3. В соответствие с увеличением количества "x" в названии файла.

2.2 МЕТОДЫ ПРОТОКОЛА

Существуют два метода протокола: `start` и `action`.

Метод `start` — старт сеанса.

Входными параметрами для метода служит идентификатор точки, идентификатор сервиса, признак выполнения в конце принудительной проверки реквизитов на сервере и опционально список стартовых значений контекста.

Выходным параметром служит идентификатор сессии (токен).

Метод `action` — выполнение действия по сценарию.

Обязательным входным параметром выступает идентификатор сессии (токен).

Необязательными входными параметрами являются введённые данные и действие, которое следует выполнить на шаге.

На первом шаге введённые данные и действие не передаются, так как не определены.

Выходными параметрами метода являются:

1. Объект со списком экранных полей к заполнению.
2. Объект с результатом выполнения шага, в том числе — составной валидации, выполненной по итогам шага.
3. Список доступных действий — список типов допустимых действий.

4. Объект со списком заполненных на предыдущих шагах полей (контекст выполнения).

5. Объект, содержащий сумму к оплате, определённую в сценарии или полученную от поставщика услуги.

В версии 4.6.5 добавлены методы `bill_start`, `bill_action`, `withdrawal` – это методы для оплаты счетов по системе подписок, выполняемые по сценарию работы с динамическими формами. Применяются в случае, когда для оплаты сервиса необходимо использовать динамический обработчик и реализуется оплата выставленного поставщиком счета (в системе подписок Pay-logic).

Описание методов:

1. `bill_start` — начало оплаты счета по многошаговой схеме. На вход метод принимает только идентификатор счета. Результатом возвращает клиенту идентификатор рабочей сессии;

2. `bill_action` — выполнение шага для оплаты счета по многошагу. На вход метод принимает идентификатор рабочей сессии и объект шага в представлении динамического модуля ввода. Объект шага — опциональный, на первом шаге он пуст;

3. Дополненный метод `withdrawal`, завершающий выполнение списания в схеме многошага. Этот метод завершает оплату счета и выполняется когда шагов к заполнению система более не вернула. От обычного `Withdrawal` протокольно отличается обязательным параметром "id рабочей сессии".

На каждый шаг `bill_step` система возвращает ответ, содержащий представление результата выполнения шага в модели динамического ввода данных `ScreenData` : набор полей к заполнению и список возможных действий.

Работу с многошагом поставщика в себя инкапсулирует реализация шлюза. При этом формирование объектов `ScreenData`, разбор (конвертацию) объекта поставщика (динамической формы) выполняет не сама реализация шлюза, а для этого она обращается к реализации микросервиса динамических форм.

2.3 ДИАГРАММА ВЗАИМОДЕЙСТВИЯ КЛИЕНТА И ОБРАБОТЧИКА

Диаграмма взаимодействия клиента и обработчика приведена на рисунке 2.3.1.

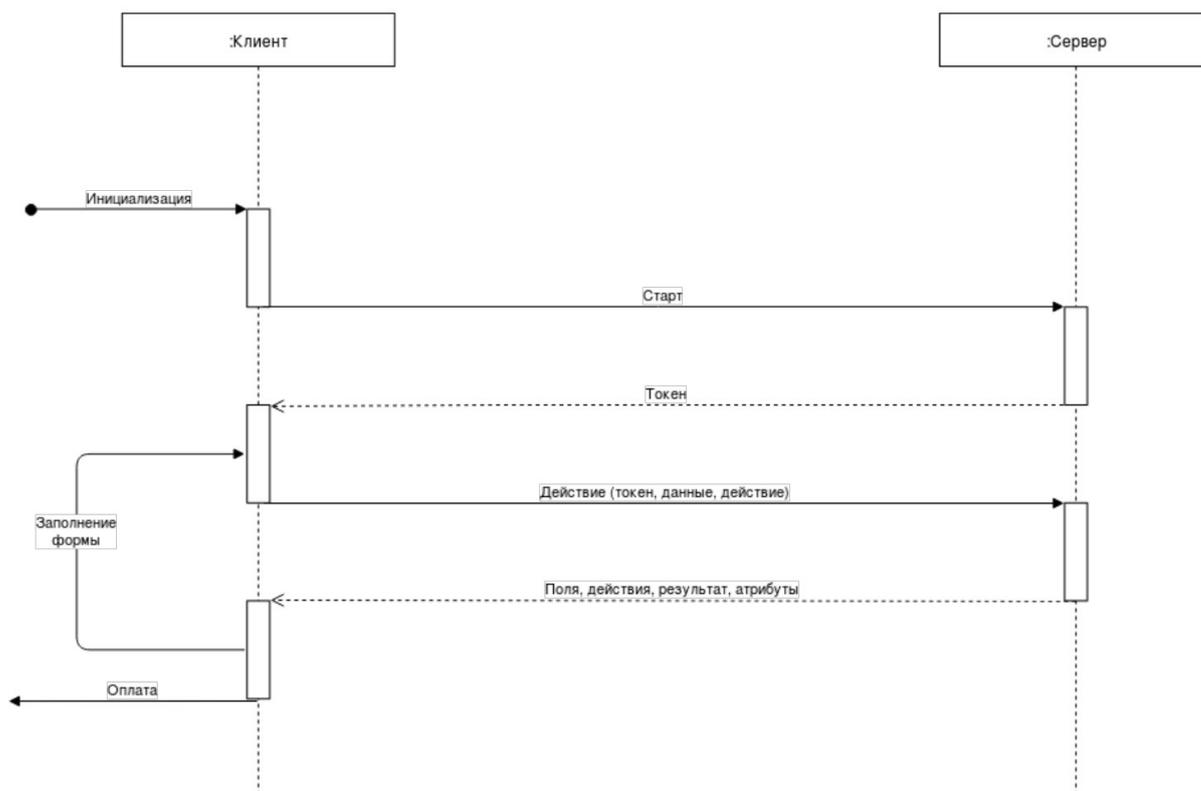


Рисунок 2.3.1 — Диаграмма взаимодействия клиента и обработчика

2.4 ИНИЦИАЛИЗИРУЮЩИЙ ЗАПРОС — SELECTORVALUE START

`start` — запрос, инициализирующий работу с формой (сценарием).

Запрос инициализирует сессию, инициализирует на сервере поиск и разбор формы (сценария) и возвращает в качестве результата идентификатор сессии (токен).

Токен представляет собой уникальную строку. Токен имеет ограниченное время бездействия. При каждом последующем вызове запроса `action` для этого токена, время бездействия сбрасывается. Таким образом, время жизни сессии — потенциально не ограничено, но при отсутствии последующих запросов по токену (независимо от того, был ли токен только создан или уже были выполнены некоторые шаги), токен аннулируется и более невалиден.

Таймаут неактивности задаётся в конфигурационном файле, по умолчанию — пять минут.

Параметры запроса `start` приведены в таблице 2.4.1.

Таблица 2.4.1 — Параметры инициализирующего запроса `start`

Параметр	Тип	Обяз.	Описание
<code>idService</code>	Integer	Да	Идентификатор сервиса
<code>idPoint</code>	Integer	Да	Идентификатор точки
<code>verify</code>	Boolean	Да	Признак выполнения принудительной проверки при достижении последнего шага
<code>redirectData</code>	RedirectData	Нет	Стартовые значения контекста Атрибуты объекта приведены в таблице 2.4.3
<code>code</code>	Integer	Нет	Код сервиса. Если указано значение данного параметра, то поиск формы (сценария)

			выполняется по коду
firstScreen	Boolean	Нет	Если задано значение true , то в ответ на стартовый запрос будет отдан первый экран. По умолчанию false . Пример вызова первого экрана приведен в разделе 2.15.9, повторный вызов первого экрана — в разделе 2.15.10
locale	String	Нет	Региональные настройки для формы/сценария. Если значение указано, то поиск формы/сценария выполняется по <ID или код сервиса>_<локаль>.xml

Параметры ответа приведены в таблице 2.4.2.

Таблица 2.4.2 — Параметры ответа на запрос `start`

Параметр	Тип	Обяз.	Описание
code	int	Да	Код результата. Коды результатов общие для всех типов ответа (раздел 2.11)
session	String	Да	Идентификатор сессии (токен)

Атрибуты объекта `RedirectData` приведены в таблице 2.4.3.

Таблица 2.4.3 — Атрибуты объекта **RedirectData**

Параметр	Тип	Обяз.	Описание
idService	int	Да	ID сервиса для редиректа
params	List<InputElement>	Нет	Стартовые атрибуты редиректа. Список параметров задается в тэге <code>redirect</code> , параметром <code>params</code> (<code><redirect ... params="id1,id2"/></code>). Атрибуты объекта <code>InputElement</code> приведены в таблице 2.8.1

Если задан признак выполнения принудительной проверки, то по достижении последнего шага система выполняет принудительную проверку по сервису, и в зависимости от её результата либо завершает обработку и возвращает набор элементов, либо возвращает на предыдущий шаг с сообщением об ошибке и ошибкой, которую вернула проверка в качестве результата.

2.5 ЗАПРОС ВЫПОЛНЕНИЯ ДЕЙСТВИЯ — ACTION

`action` — запрос, инициализирующий выполнение действия формы (сценария).

Запрос инициализирует выполнение действия, инициализирует на сервере валидацию и сохранение введённого значения (кроме первого выполнения), выполнение запрошенного действия (согласно значению `action`) для формы (сценария) и возвращает в качестве результата следующее поле или информацию о непройденной валидации.

Параметры запроса `action` приведены в таблице 2.5.1.

Таблица 2.5.1 — Параметры запроса `action`

Параметр	Тип	Обяз.	Описание
<code>session</code>	String	Да	Идентификатор сессии (токен)
<code>screenData</code>	ScreenData	Нет	Заполненные данные для текущего экрана. см раздел «Объекты данных экрана»
<code>action</code>	String	Нет	Название действия
<code>xForm</code>	String	Нет	Форма в формате xml для формирования следующего шага
<code>screen</code>	String	Нет	Позволяет указать, на каком экране выполняется действие. Пример: <pre>{ "session": "2c5d9370-c907-4947-ae71-6c756ac9ce88", "screen": "01screen", "screenData": { "fields": [{ "code": "id1",</pre>

			<pre> "value": "3333333333", "originalValue": "3333333333" }] }, "action": "next" } </pre>
--	--	--	---

Параметры ответа на запрос `action` приведены в таблице 2.5.2.

Таблица 2.5.2 — Параметры ответа на запрос `action`

Параметр	Тип	Обяз.	Описание
<code>code</code>	Integer	Да	Код результата. Коды результатов общие для всех типов ответа (раздел 2.11)
<code>screenData</code>	ScreenData	Нет	Данные для отображения очередного экрана (раздел 2.7)
<code>xform</code>	String	Нет	Объект XForm в виде xml (html) представления объекта xforms. Возвращается в виде текстового представления
<code>sumpurchase</code>	Long	Нет	Сумма к оплате (в копейках), полученная во время выполнения формы (сценария) или полученная от поставщика услуг
<code>action</code>	List<String>	Нет	Список допустимых действий на текущем шаге (раздел 2.7)
<code>actionExt</code>	List<ScreenAction>	Нет	Расширенный список допустимых действий на текущем шаге, включает описание действий (раздел 2.7)
<code>actionResult</code>	ActionResult	Нет	Результат выполнения действия. Атрибуты приведены в разделе 2.8
<code>attributes</code>	List<InputElement>	Нет	Список атрибутов, накопленных на

			<p>предыдущих шагах. Не включает элементы текущего возвращённого в ScreenData шага. Они могут также присутствовать, если были проинициализированы ранее, но со значениями с предыдущего шага. По существу — это текущий накопленный контекст (раздел 2.8)</p>
redirectData	RedirectData	Нет	Информация о редиректе (2.4.3)
lastScreen	Boolean	Да	<p>Признак последнего экрана формы/сценария, за которым следует экран оплаты. Имеет значение true в том случае, если:</p> <ul style="list-style-type: none"> • экран формы содержит последнее поле, которое не является селектором со вложенными элементами. Пример селектора со вложенными элементами: <pre> <text-selector name="id1" message="Селектор" title="Селектор"> <item text="Оплата" value="1"/> <item text="Вложенное поле" value="2"> <text-field name="id2" keyboard="Digital" max-len="10" message="Поле 2" title="Поле 2"> <verify> <rule regex="^\d{10}\$"/> </verify> <split default="*** ** ***"/> <help>Введите номер счета</help> </text-field> </pre>

			<pre> </item> </text-selector> </pre> <ul style="list-style-type: none"> экран сценария содержит элемент <code><action>...</action></code>, включающий переход на экран оплаты <code><goto target="pay"/></code>. <p>Пример:</p> <pre> <action type="Next" title="Далее"> <goto target="pay"/> </action> </pre>
feeSum	FeeSum	Нет	<p>Информация о комиссии, полученная либо:</p> <ol style="list-style-type: none"> От провайдера, которую он вернул через онлайн-запрос <code>get-fee</code>. Если провайдер не вернул комиссию, — она берется из профиля меню в процессинге. <p>Атрибуты объекта приведены в разделе 2.8</p>
multiplicitySum	Long	Нет	<p>Платежи должны быть кратны этой сумме (в копейках). Если сумма платежа больше, остаток от деления сохраняется на счете клиента.</p>
minSum	Long	Нет	<p>Минимальная сумма платежа в копейках. Передается после перехода на экран оплаты при условии, что не задана сумма к оплате (параметр <code>sumpurchase</code>)</p>
maxSum	Long	Нет	<p>Максимальная сумма платежа в копейках. Передается после перехода на экран оплаты при условии, что не задана сумма к оплате (параметр</p>

			sumpurchase)
navigations	List<NavigationItem>	Нет	Возвращает историю пройденных экранов. Атрибуты NavigationItem приведены в разделе 2.9

Результатом выполнения команды `action` (кроме последнего финального шага) всегда является получение данных **ScreenData**, составляющих содержимое текущего шага.

Опционально, если это определено платежным сценарием, на любом из шагов может быть установлена требуемая сумма к оплате (`sumpurchase`). Если такая сумма установлена — клиент обязан использовать ее значение в качестве требуемой суммы, зачисленной на счет (без учета комиссии с клиента).

Также на каждом из шагов возвращается полный набор накопленных в контексте выполнения динамического сценария введенных (и полученных от сервера) атрибутов (**attributes**).

Согласно диаграмме взаимодействия для перехода к следующему шагу Партнер должен отправить динамическому обработчику запрос `Step`, передав в нем полностью заполненную `ScreenData` с предыдущего шага, плюс выбранное пользователем действие (`Action`). В текущей версии протокола реализована и поддерживается возможность частичного заполнения `ScreenData` — а именно: возможность заполнять в полях ввода только код и значение поля и не передавать остальные атрибуты, такие как оригинальное значение, выражение проверки и т. д., а также не передавать обратно поля, которые на шаге предназначались только для чтения и показа клиенту, но не для ввода (`read-only` поля). Такой подход значительно экономит трафик и позволяет упростить обработку шага на стороне клиента.

Пример такого сокращенного обмена приведен в разделе 2.15.8 — отправка частично заполненных объектов `ScreenData`.

Возможно вернуться с экрана оплаты на предыдущий экран. Для этого на экране оплаты необходимо отправить запрос `action`, содержащий пустой объект `screenData` и действие `prev`.

Пример вызова `action`:

```
{  
  "session": "4395483223494687960",  
  "screenData": null,  
  "action": "prev"  
}
```

2.6 ЗАПРОСЫ ДЛЯ РАБОТЫ С XFORMS

В ряде случаев (для поставщиков, применяющих технологию xForms, примером — поставщик А-3), существует необходимость получать представление содержимого текущего шага в виде объекта XForms. Для работы с XForms предназначены дополнительные функции динамического обработчика:

- `getNextStepXForm` — получение следующего шага в представлении протокола и представлении Xforms;
- `getXFormStep` — получение следующего шага в представлении только XForm (без `ScreenData`);
- `getLastXForm` `getXFormStep` — получение XForm-представления текущего выполняемого шага (без перехода к следующему).

Параметры запроса `getNextStepXForm` приведены в таблице 2.6.1.

Таблица 2.6.1 — Параметры запроса `getNextStepXForm`

Параметр	Тип	Обяз.	Описание
<code>session</code>	String	Да	Идентификатор сессии (токен)
<code>screenData</code>	ScreenData	Нет	Заполненные данные для текущего экрана. См. раздел «Объекты данных экрана»
<code>action</code>	String	Нет	Название действия, которое нужно выполнить (<code>next</code> , <code>prev</code> , <code>exit</code>)
<code>xform</code>	String	Нет	Объект XForm в виде xml (html) представления объекта xforms. Возвращается в виде текстового представления

Параметры ответа на запрос `getNextStepXForm` приведены в таблице 2.6.2.

Таблица 2.6.2 — Параметры ответа на запрос `getNextStepXForm`

Параметр	Тип	Обяз.	Описание
<code>code</code>	Integer	Да	Код результата. Коды результатов общие для всех типов ответа (раздел 2.11)
<code>screenData</code>	ScreenData	Нет	Данные для отображения очередного экрана (раздел 2.7)
<code>sumpurchase</code>	Long	Нет	Сумма к оплате (в копейках), полученная во время выполнения формы (сценария) или полученная от поставщика услуг
<code>attributes</code>	List<InputElement>	Нет	Список атрибутов, накопленных на предыдущих шагах. Не включает элементы текущего возвращённого в ScreenData шага. Они могут также присутствовать, если были проинициализированы ранее, но со значениями с предыдущего шага. По существу — это текущий накопленный контекст (раздел 2.7)

Параметры запроса `getXFormStep` приведены в таблице 2.6.3.

Таблица 2.6.3 — Параметры запроса `getXFormStep`

Параметр	Тип	Обяз.	Описание
<code>session</code>	String	Да	Идентификатор сессии (токен)
<code>screenData</code>	ScreenData	Нет	Заполненные данные для текущего экрана. см раздел «Объекты данных экрана»

Параметры ответа на запрос `getXFormStep` приведены в таблице 2.6.4.

Таблица 2.6.4 — Параметры ответа на запрос `getXFormStep`

Параметр	Тип	Обяз.	Описание
<code>code</code>	Integer	Да	Код результата. Коды результатов общие для всех типов ответа (раздел 2.11)
<code>sumpurchase</code>	Long	Нет	Сумма к оплате (в копейках) полученная во время выполнения формы (сценария) или полученная от поставщика услуг
<code>attributes</code>	List<InputElement>	Нет	Список атрибутов, накопленных на предыдущих шагах. не включает элементы текущего возвращённого в <code>ScreenData</code> шага. Они могут также присутствовать, если были проинициализированы ранее, но со значениями с предыдущего шага. По существу — это текущий накопленный контекст (раздел 2.7)
<code>xform</code>	String	Нет	Объект <code>XForm</code> в виде <code>xml</code> (<code>html</code>) представления объекта <code>xforms</code> . Возвращается в виде текстового представления

Параметры запроса `getLastXForm` приведены в таблице 2.6.5.

Таблица 2.6.5 — Параметры запроса `getLastXForm`

Параметр	Тип	Обяз.	Описание
<code>session</code>	String	Да	Идентификатор сессии (токен)

Параметры ответа на запрос `getXFormStep` приведены в таблице 2.6.6.

Таблица 2.6.6 — Параметры ответа на запрос `getLastXForm`

Параметр	Тип	Обяз.	Описание
<code>code</code>	Integer	Да	Код результата. Коды результатов общие для всех типов ответа (раздел 2.11)
<code>sumpurchase</code>	Long	Нет	Сумма к оплате (в копейках), полученная во время выполнения формы (сценария) или полученная от поставщика услуг
<code>attributes</code>	List<InputElement>	Нет	Список атрибутов, накопленных на предыдущих шагах. Не включает элементы текущего возвращённого в <code>ScreenData</code> шага. Они могут также присутствовать, если были проинициализированы ранее, но со значениями с предыдущего шага. По существу — это текущий накопленный контекст (раздел 2.7)
<code>xform</code>	String	Нет	Объект <code>XForm</code> в виде <code>xml</code> (<code>html</code>) представления объекта <code>xforms</code> . Возвращается в виде текстового представления

2.7 ОБЪЕКТЫ ДАННЫХ ЭКРАНА

ScreenData — объект, содержащий отображаемый экран с полями ввода и информацией.

Атрибуты объекта приведены в таблице 2.7.1.

Таблица 2.7.1 — Атрибуты объекта **ScreenData**

Атрибут объекта	Тип	Обяз.	Описание
title	String	Да	Заголовок экрана ввода
titleId	String	Нет	Идентификатор локализуемой текстовки экрана
id	String	Да	Идентификатор экрана, используется для форм и сценариев. Добавлен в версии 1.3.0. При передаче id восстанавливается контекст экрана, если имеется (если контекста нет, отобразится ошибка), и выполняется запрошенное действие. Дальнейший контекст очищается. Пример использования атрибута приведен в разделе 2.13
help	String	Нет	Текст подсказки (информации), выводимый на экране вместе с полями
decor	String	Нет	Декор поля
type	String	Нет	Тип экрана (используется только для сценариев)

info	String	Нет	Информация для экранов с типом info (используется только для сценариев)
fields	List<ScreenField>	Да	Список полей, отображаемых на экране для ввода
actions	List<String>	Нет	Список действий, доступных на экране
actionsExt	List<ScreenAction>	Нет	Расширенный список действий, доступных на экране. Содержит описание, тип действий, расшифровку, указывает, является ли действие платежным
dialogs	List<ScreenDialog>	Нет	Список диалогов для отображения на текущем шаге (используется только для сценариев)

Атрибуты объекта, описывающего экранное поле ввода (**ScreenField**), приведены в таблице 2.7.2.

Таблица 2.7.2 — Атрибуты объекта, описывающего экранное поле ввода (**ScreenField**)

Атрибут объекта	Тип	Обяз.	Описание
Базовые атрибуты поля			
code	String	Да	Код поля ввода
type	FieldType	Нет	Тип элемента. На экране подтверждения не задается. Если не задано, то считается не полем ввода, а значением для вывода
title	String	Нет	Заголовок (титл) элемента
titleId	String	Нет	Идентификатор локализуемой текстовки заголовка
value	String	Нет	Значение элемента ввода.

			Возвращается, если данный элемент присутствует в контексте (определён на более ранних шагах, установлен set-ом, возвращён онлайн-запросом и т.д., в контексте текущего шага так или иначе определен)
help	String	Нет	Подсказка для поля
helpId	String	Нет	Идентификатор локализуемой текстовки подсказки для поля
messageOff	String	Нет	Сообщение при снятом чек-боксе (используется только для чек-боксов)
messageOffId	String	Нет	Код текстовки сообщения при снятом чек-боксе (используется только для чек-боксов)
flags	int	Нет	Битовая маска флагов. См. раздел «Флаги элементов»
invalid	Boolean	Да	Поле ввода не прошло валидацию. Может принимать значение true при выполнении очередного шага, если выполненные сервером проверки согласно форме/сценарию выявили невалидность значения поля (несоответствие регулярке и т.д.)
messageOn	String	Нет	Сообщение при установленном чек-боксе (используется только для чек-боксов)
messageOnId	String	Нет	Код текстовки сообщения при установленном чек-боксе (используется только для чек-боксов)

valueTitleId	String	Нет	ID локализованного сообщения для значения
alert	String	Нет	Предупреждение (используется только для XForm)
required	String	Да	Обязательность платежного атрибута. Если true , то атрибут должен быть заполнен и отправлен в платеже. Если значение false , то атрибут носит информационный характер
Атрибуты, описывающие способы ввода значений			
keyboard	KeyboardType	Нет	Клавиатура (для числовых/текстовых полей ввода). Типы клавиатур перечислены ниже
letterCase	KeyboardCase	Нет	Возможность установить первую букву сообщения заглавной. Возможные значения: UpperCase — заглавная буква LowerCase — строчная буква
secure	Boolean	Да	Ввод закрытых значений (***). Если true , то вместо значения клиенту надо визуализировать точки или *
defaultValue	String	Нет	Значение по умолчанию, которое допустимо править / редактировать
example	String	Нет	Пример ввода информации, отображается на экране, исчезает при вводе (отображение на усмотрение клиента)
prefix	String	Нет	Префикс для текстовых полей (не редактируется клиентом)
postfix	String	Нет	Постфикс для текстовых полей (не редактируется клиентом)

readOnly	Boolean	Нет	Поле только для чтения, не допускается изменять
format	String	Нет	Формат ввода для полей выбора даты
formatTitle	String	Нет	Формат отображения введённой даты в поле выбора даты
layoutNames	List<java.lang.String>	Нет	Допустимые раскладки клавиатуры. Возможные значения: en (английский алфавит), ru (русский алфавит)
allowChangeCase	boolean	Нет	Разрешено изменять регистр
Атрибуты, определяющие проверки и ограничения, предписываемые к выполнению на клиенте			
maxLen	int	Нет	Максимальная длина вводимых данных
dateRange	DateRange	Нет	Допустимый диапазон дат (для полей выбора даты)
range	Range	Нет	Диапазон (для числовых значений), которому должно удовлетворять значение поля
regex	List<String>	Нет	Список регулярных выражений, которым должно соответствовать значение поля. Если значение проходит хотя бы по одному регулярному выражению, то значение считается валидным
blockRegex	List<String>	Нет	Список регулярных выражений, которым должно не соответствовать значение поля. Если значение проходит хотя бы по одному регулярному выражению, то значение считается не валидным

Атрибуты, определяющие цену и комиссию сервиса корзины			
basketField	BasketField	Нет	Информация о стоимости и комиссии сервиса корзины
Атрибуты, определяющие значения для селекторов (взаимно исключают)			
values	List<SelectorValue>	Нет	Список допустимых значений селектора
tableItems	List<TableItem>	Нет	Список допустимых значений табличного селектора
Атрибуты, определяющие форматирование данных			
dataFormatter	FieldFormatter	Нет	Форматирование данных
viewFormatter	FieldFormatter	Нет	Форматирование значения при отображении
regexFieldModifierRules	List<RegexFieldModifierRule>	Нет	Правило для изменения значения

Перечисление допустимых типов полей (**FieldType**) приведено в таблице 2.7.3.

Таблица 2.7.3 — Перечисление с допустимыми типами полей (**FieldType**)

Значение	Описание
TEXT_FIELD	Текстовое поле
NUMERIC_FIELD	Числовое поле
DATE_FIELD	Календарь, ввод даты
SELECTOR_FIELD	Селектор, выбор значения из списка
TABLE_FIELD	Табличный селектор, многоколоночный

Перечисление с допустимыми типами клавиатуры (**KeyboardType**) приведено в таблице 2.7.4.

Таблица 2.7.4 — Перечисление с допустимыми типами клавиатура (**KeyboardType**)

Значение	Описание
DIGITAL	Клавиатура, содержащая только цифры
LETTER	Текстовые символы
ANY	Любые символы

Перечисление с допустимыми регистрами клавиатуры (**KeyboardCase**) приведено в таблице 2.7.5.

Таблица 2.7.5 — Перечисление с допустимыми регистрами клавиатуры (**KeyboardCase**)

Значение	Описание
Upper	Верхний регистр
Lower	Нижний регистр

Объект, описывающий допустимый диапазон дат (**DateRange**), приведен в таблице 2.7.6.

Таблица 2.7.6 — Объект, описывающий допустимый диапазон дат (**DateRange**)

Атрибут объекта	Тип	Обяз.	Описание
start	String	Нет	Минимальная дата и время (ISO_8601 dd.MM.yyyy HH:mm)
end	String	Нет	Максимальная дата и время (ISO_8601 dd.MM.yyyy HH:mm)

Объект, описывающий допустимый диапазон чисел (**Range**), приведен в таблице 2.7.7.

Таблица 2.7.7 — Объект, описывающий допустимый диапазон чисел (**Range**)

Атрибут объекта	Тип	Обяз.	Описание
start	Double	Да	Минимально допустимое число
end	Double	Да	Максимально допустимое число

Атрибуты объекта, описывающего доступные значения выпадающего списка (**SelectorValue**), приведен в таблице 2.7.8.

Таблица 2.7.8 — Атрибуты объекта, описывающего доступные значения выпадающего списка (**SelectorValue**)

Атрибут объекта	Тип	Обяз.	Описание
title	String	Да	Заголовок элемента (отображаемое значение)
value	String	Да	Значение элемента (значение, фактически подставляемое в качестве value к полю)
help	String	Нет	Подсказка элемента (отображается на усмотрение клиента)
img	String	Нет	Имя изображения для пункта селектора

TableItem — объект, описывающий доступные значения для табличного выпадающего списка. Содержит значение (которое должно быть передано в качестве **value** для поля типа селектор) и список полей табличной части, отображаемых клиенту в качестве содержимого строки табличного селектора.

Таблица 2.7.9 — Атрибуты объекта, описывающего доступные значения для табличного выпадающего списка (**TableItem**)

Атрибут объекта	Тип	Обяз.	Описание
value	String	Да	Значение элемента
valueTitle	String	Да	Заголовок значения элемента
elements	List<TableItemValue>	Да	Список элементов содержащихся в строке табличного селектора

Атрибуты объекта, описывающего строку табличного селектора (**TableItemValue**), приведены в таблице 2.7.10.

Таблица 2.7.10 — Атрибуты объекта, описывающего строку табличного селектора (**TableItemValue**)

Атрибут объекта	Тип	Обяз.	Описание
code	String	Да	Код элемента
title	String	Нет	Заголовок элемента
titleId	String	Нет	Идентификатор локализуемой текстовой заголовка
value	String	Да	Значение элемента
valueTitle	String	Нет	Заголовок значения элемента
flags	Integer	Да	Флаги элемента

Типовой пример табличного селектора: выбор маршрута, маршруты показываются в виде таблицы с колонками:

1. «Номер маршрута».
2. «Пункт отправления».
3. «Пункт назначения».
4. «Дата отправления».

Выбирая строку в такой таблице, клиент выбирает маршрут с заданным номером.

Атрибуты объекта, описывающего доступные действия для текущего экрана (**ScreenAction**), приведены в таблице 2.7.11.

Таблица 2.7.11 — Атрибуты объекта, описывающего доступные действия для текущего экрана (**ScreenAction**)

Атрибут объекта	Тип	Обяз.	Описание
type	String	Да	Тип действия для передачи в запросе выполнения действия
title	String	Да	Заголовок действия для отображения пользователю

Перечисление с допустимыми типами всплывающих окон (**DialogType**) приведено в таблице 2.7.12.

Таблица 2.7.12 — Перечисление с допустимыми типами всплывающих окон (**DialogType**)

Значение	Описание
INFO	Информационное диалоговое окно
WARNING	Диалоговое окно, содержащее предупреждение
ERROR	Диалоговое окно, содержащее ошибку
QUESTION	Диалоговое окно, содержащее вопрос

Атрибуты объекта, описывающего диалоговое окно для текущего шага (**ScreenDialog**), приведены в таблице 2.7.13.

Таблица 2.7.13 — Атрибуты объекта, описывающего диалоговое окно для текущего шага (**ScreenDialog**)

Атрибут объекта	Тип	Обяз.	Описание
type	DialogType	Да	Тип всплывающего окна
title	String	Да	Выводимый заголовок диалогового окна
titleId	String	Нет	Код текстовой выводимого заголовка диалогового окна
message	String	Да	Сообщение диалогового окна
messageId	String	Нет	Код текстовой сообщения диалогового окна
timeout	String	Нет	Указывается время, на которое выводится диалоговое окно
defaultAction	String	Нет	Действие по умолчанию
params	List<String>	Нет	Выводимые в сообщении значения переменных. Переменные перечисляются через запятую без пробелов
actions	List<ScreenAction>	Да	Список доступных действий в диалоговом окне

Атрибуты объекта, описывающего стоимость и комиссию сервиса (**BasketField**) приведены в таблице 2.7.14.

Таблица 2.7.14 — Атрибуты объекта, описывающего стоимость и комиссию сервиса корзины (**BasketField**)

Атрибут объекта	Тип	Обяз.	Описание
summ	String	Нет	Стоимость сервиса корзины (сумма оплаты)
comm	String	Нет	Сумма комиссии сервиса корзины

Таблица 2.7.15 — Атрибуты объекта, описывающего форматирование данных (FieldFormatter)

Атрибут объекта	Тип	Обяз.	Описание
mask	String	Нет	Маска по умолчанию
masks	String	Нет	Список масок
byStep	boolean	Нет	Маска в текстовом поле будет отображаться по мере ввода символов

Примеры использования масок приведены в разделе 2.16. Описание форматирования данных приведено в документе [«Формы оплаты для универсального модуля ввода данных. Программное обеспечение «Процессинговый центр Pay-logic». Руководство пользователя»](#) для универсального и в руководстве [«Сценарии оплаты для усовершенствованного модуля ввода данных. Программное обеспечение «Процессинговый центр Pay-logic». Руководство пользователя»](#) для усовершенствованного обработчика. Подробное описание правил составления масок приведено на странице <https://docs.oracle.com/javase/7/docs/api/javax/swing/text/MaskFormatter.html>.

Таблица 2.7.16 — Атрибуты объекта, описывающего правила для изменения значения (RegexFieldModifierRule)

Атрибут объекта	Тип	Обяз.	Описание
regex	String	Нет	Регулярное выражение для проверки
target	String	Нет	Правила для разбития значения на группы
value	String	Нет	Ссылка на номер группы

2.8 ОБЪЕКТЫ ДАННЫХ ОТВЕТА

1. Параметр `attributes` содержит массив заполненных полей ввода (**InputElement**).

InputElement — заполненное поле ввода. Атрибуты элемента приведены в таблице 2.8.1.

Таблица 2.8.1 — Атрибуты **InputElement**

Атрибут объекта	Тип	Обяз.	Описание
<code>code</code>	String	Да	Код поля
<code>title</code>	String	Нет	Заголовок (титл) элемента
<code>titleId</code>	String	Нет	Идентификатор локализуемой текстовки заголовка
<code>value</code>	String	Да	Значение элемента ввода
<code>valueTitle</code>	String	Нет	Заголовок значения элемента
<code>originalValue</code>	String	Да	Значение элемента ввода (оригинальное). Отличается от <code>value</code> в том случае, если элемент был модифицирован сценарием, а не через поле ввода
<code>exttype</code>	String	Нет	Кастомизированное свойство, которое позволяет идентифицировать бизнес-сущность представляемых данных
<code>flags</code>	int	Да	Битовая маска флагов (раздел 2.12)
<code>group</code>	Int	Нет	Группа атрибутов. Используется при создании пачки платежей, при наличии атрибутов с одинаковыми именами позволяет понять, к какому платежу относится атрибут

2. Объект **ActionResult** содержит результат выполнения действия (action).
Атрибуты объекта приведены в таблице 2.8.2.

Таблица 2.8.2 — Атрибуты **ActionResult** (результата выполнения действия)

Атрибут объекта	Тип	Обяз.	Описание
code	Integer	Да	Код результата действия. Значения кодов: 0 — шаг успешно выполнен, есть следующий шаг; 1 — шаг успешно выполнен, является завершающим. Дальнейшая навигация невозможна, нужно переходить к платежу; 2 — при выполнении шага произошла ошибка валидации, переход на следующий шаг не произведён
description	String	Да	Краткое описание результата
messages	List<ActionMessage>	Нет	Список сообщений, которые необходимо показать пользователю. Сообщения по результатам валидации, онлайн-проверки и т.д. Атрибуты ActionMessage приведены в таблице 2.8.3.

Таблица 2.8.3 — Атрибуты сообщения о результате выполнения действия (ActionMessage)

Атрибут объекта	Тип	Обяз.	Описание
type	int	Да	Код типа сообщения о результате выполнения шага действия. Значения кодов: 0 — системное сообщение (зарезервировано); 1 — сообщение процессинга; 2 — сообщение поставщика услуги
message	String	Да	Содержание сообщения

3. Объект **FeeSum** содержит информацию о комиссии, которая передана от провайдера через онлайн-запрос, или о комиссии из профиля меню (в случае, если провайдер комиссию не передал). Атрибуты объекта приведены в таблице ниже.

Таблица 2.8.4 — Атрибуты объекта **FeeSum**

Параметр	Тип	Обяз.	Описание
fee	Long	Да	Сумма посчитанной комиссии в копейках
feeName	String	Нет	Название комиссии
feePercent	Double	Нет	Значение процента комиссии
feeCommission	FeeCommission	Нет	Описание комиссии. Атрибуты объекта FeeCommission приведены в таблице 2.8.5.

Таблица 2.8.5 — Атрибуты объекта **FeeCommission**

Параметр	Тип	Обяз.	Описание
name	String	Нет	Название комиссии
currency	String	Да	Валюта

fixed	Long	Да	Фиксированная комиссия
min	Long	Да	Минимальная сумма комиссии
max	Long	Да	Максимальная сумма комиссии
steps	List<FeeCommissionStep>	Нет	Шаги комиссии. Атрибуты объекта FeeCommissionStep приведены в таблице 2.8.6.

Таблица 2.8.6 — Атрибуты объекта **FeeCommissionStep**

Параметр	Тип	Обяз.	Описание
sum	Long	Да	Сумма, начиная с которой, применяется комиссия
percent	Double	Да	Значение комиссии в %
fixed	Long	Да	Фиксированное значение комиссии в копейках
min	Long	Да	Минимальная комиссия, сумма в копейках
max	Long	Да	Максимальная комиссия, сумма в копейках

Данные по комиссии заполняются после получения комиссии от провайдера или, в случае ее отсутствия, значением комиссии из профиля меню (за исключением FeeSum.fee, FeeSum.feeName, FeeSum.feePercent) после перехода на экран оплаты.

2.9 ЭЛЕМЕНТ НАВИГАЦИИ — NAVIGATIONITEM

Объект `navigations` (таблица 2.5.2) содержит массив с пройденными экранами (**NavigationItem**).

Атрибуты элемента **NavigationItem** приведены в таблице 2.9.1.

Таблица 2.9.1 — Атрибуты элемента **NavigationItem**.

Параметр	Тип	Обяз.	Описание
<code>NavigationItem.id</code>	String	Да	Уникальный идентификатор экрана
<code>NavigationItem.title</code>	String	Нет	Заголовок экрана
<code>NavigationItem.titleId</code>	String	Нет	Код локализованного сообщения для заголовка экрана
<code>NavigationItem.current</code>	Boolean	Да	Если значение <code>true</code> , то экран является текущим

Пример для сценария:

```
<scenario begin="01screen">
  <screen type="digital" decor="button" title="title01screen"
id="01screen">
  <fields>
    <text-field id="id1" title="titleId1" max-len="10"
keyboard="Digital">
      <validator type="regex">
        <rules>
          <rule regex="^\d{1,10}$"/>
        </rules>
      </validator>
    </text-field>
  </fields>
```

```
<actions>
  <action type="Next" title="Далее">
    <goto target="02screen"/>
  </action>
  <action type="Exit" title="Выход">
    <goto target="exit"/>
  </action>
</actions>
</screen>
<screen type="digital" decor="button" title="title02screen"
id="02screen">
  <fields>
    <text-field id="id2" title="titleId2" max-len="10"
keyboard="Digital">
      <validator type="regex">
        <rules>
          <rule regex="^\d{1,10}$"/>
        </rules>
      </validator>
    </text-field>
  </fields>
  <actions>
    <action type="Next" title="Далее">
      <goto target="pay"/>
    </action>
    <action type="Prev" title="Назад">
      <goto target="01screen"/>
    </action>
    <action type="Exit" title="Выход">
      <goto target="exit"/>
    </action>
  </actions>
</screen>
</scenario>
```

1. Start — запрос:

```
{"idPoint":"1","idService":"1","verify":false,"locale":"","firstScreen":false}
```

Start — ответ:

```
{"code":0,"session":"2c5d9370-c907-4947-ae71-6c756ac9ce88"}
```

2. Первый экран — запрос:

```
{"session":"2c5d9370-c907-4947-ae71-6c756ac9ce88","screenData":null,"action":"next","screen":null}
```

Первый экран — ответ:

```
{
  "code": 0,
  "session": "2c5d9370-c907-4947-ae71-6c756ac9ce88",
  "screenData": {
    "id": "01screen",
    "title": "title01screen",
    "decor": "button",
    "type": "digital",
    "fields": [
      {
        "code": "id1",
        "type": "TEXT_FIELD",
        "title": "titleId1",
        "flags": 0,
        "keyboard": "DIGITAL",
        "letterCase": "Lower",
        "allowChangeCase": true,
        "secure": false,
        "maxLen": 10,
        "readOnly": false,
        "invalid": false,
        "required": true,
        "regexp": [
          "\\d{1,10}$"
        ]
      }
    ],
    "actions": [
      "next",
      "exit"
    ],
    "actionsExt": [
      {
        "type": "next",
        "title": "Далее",
        "pay": false
      }
    ]
  }
}
```

```
{
  "type": "exit",
  "title": "Выход",
  "pay": false
}
],
"lastScreen": false,
"actionResult": {
  "code": 0,
  "description": "Шаг успешно выполнен, есть следующий шаг"
},
"attributes": [],
"navigations": [
  {
    "id": "01screen",
    "title": "title01screen",
    "current": true
  }
]
}
```

3. Второй экран — запрос:

```
{
  "session": "2c5d9370-c907-4947-ae71-6c756ac9ce88",
  "screenData": {
    "fields": [
      {
        "code": "id1",
        "value": "1111111111",
        "originalValue": "1111111111"
      }
    ]
  },
  "action": "next",
  "screen": null
}
```

Второй экран — ответ:

```
{
  "code": 0,
  "session": "2c5d9370-c907-4947-ae71-6c756ac9ce88",
}
```

```
"screenData": {
  "id": "02screen",
  "title": "title02screen",
  "decor": "button",
  "type": "digital",
  "fields": [
    {
      "code": "id2",
      "type": "TEXT_FIELD",
      "title": "titleId2",
      "flags": 0,
      "keyboard": "DIGITAL",
      "letterCase": "Lower",
      "allowChangeCase": true,
      "secure": false,
      "maxLen": 10,
      "readOnly": false,
      "invalid": false,
      "required": true,
      "regexp": [
        "\\d{1,10}$"
      ]
    }
  ],
  "actions": [
    "next",
    "prev",
    "exit"
  ],
  "actionsExt": [
    {
      "type": "next",
      "title": "Далее",
      "pay": true
    },
    {
      "type": "prev",
      "title": "Назад",
      "pay": false
    },
    {
      "type": "exit",
      "title": "Выход",
    }
  ]
}
```

```
        "pay": false
      }
    ]
  },
  "lastScreen": true,
  "actionResult": {
    "code": 0,
    "description": "Шаг успешно выполнен, есть следующий шаг"
  },
  "attributes": [
    {
      "code": "id1",
      "title": "titleId1",
      "value": "1111111111",
      "valueTitle": "1111111111",
      "originalValue": "1111111111",
      "flags": 0
    }
  ],
  "navigations": [
    {
      "id": "01screen",
      "title": "title01screen",
      "current": false
    },
    {
      "id": "02screen",
      "title": "title02screen",
      "current": true
    }
  ]
}
```

4. Переход на экран оплаты — запрос:

```
{
  "session": "2c5d9370-c907-4947-ae71-6c756ac9ce88",
  "screenData": {
    "fields": [
      {
        "code": "id2",
        "value": "2222222222",
        "originalValue": "2222222222"
      }
    ]
  }
}
```

```
]
},
"action": "next",
"screen": null
}
```

Переход на экран оплаты — ответ:

```
{
  "code": 0,
  "session": "2c5d9370-c907-4947-ae71-6c756ac9ce88",
  "minSum": 100,
  "maxSum": 500000,
  "lastScreen": true,
  "actionResult": {
    "code": 1,
    "description": "Шаг успешно выполнен, является завершающим"
  },
  "attributes": [
    {
      "code": "id1",
      "title": "titleId1",
      "value": "1111111111",
      "valueTitle": "1111111111",
      "originalValue": "1111111111",
      "flags": 0
    },
    {
      "code": "id2",
      "title": "titleId2",
      "value": "2222222222",
      "valueTitle": "2222222222",
      "originalValue": "2222222222",
      "flags": 0
    }
  ],
  "navigations": [
    {
      "id": "01screen",
      "title": "title01screen",
      "current": false
    },
    {
      "id": "02screen",
```

```
"title": "title02screen",  
  "current": false  
}  
]  
}
```

2.10 ВОЗВРАТ НА ПРОИЗВОЛЬНЫЙ ЭКРАН

Для возврата на произвольный экран в запросе `action` необходимо:

1. В параметре **action** передать значение `"_navi"`:

```
"action": "_navi"
```

2. В параметре **screen** — ID экрана, на который необходимо вернуться:

```
"screen": <IDэкрана>
```

При переходе между экранами контекст после выбранного экрана очищается.

Пример для сценария:

```
<scenario begin="01screen">
  <screen type="digital" decor="button" title="title01screen"
id="01screen">
  <fields>
    <text-field id="id1" title="titleId1" max-len="10"
keyboard="Digital">
      <validator type="regex">
        <rules>
          <rule regex="^\d{1,10}$"/>
        </rules>
      </validator>
    </text-field>
  </fields>
  <actions>
    <action type="Next" title="Далее">
      <goto target="02screen"/>
    </action>
    <action type="Exit" title="Выход">
      <goto target="exit"/>
    </action>
  </actions>
</screen>
```

```
<screen type="digital" decor="button" title="title02screen"
id="02screen">
  <fields>
    <text-field id="id2" title="titleId2" max-len="10"
keyboard="Digital">
      <validator type="regex">
        <rules>
          <rule regex="^\d{1,10}$"/>
        </rules>
      </validator>
    </text-field>
  </fields>
  <actions>
    <action type="Next" title="Далее">
      <goto target="pay"/>
    </action>
    <action type="Prev" title="Назад">
      <goto target="01screen"/>
    </action>
    <action type="Exit" title="Выход">
      <goto target="exit"/>
    </action>
  </actions>
</screen>
</scenario>
```

1. Start — запрос:

```
{"idPoint":"1","idService":"1","verify":false,"locale":"","firstScreen":false}
```

Start — ответ:

```
{"code":0,"session":"553cfd37-3b16-4b1e-8a77-1f490236e4a7"}
```

2. Первый экран — запрос:

```
{"session":"553cfd37-3b16-4b1e-8a77-1f490236e4a7","screenData":null,"action":"next","screen":null}
```

Первый экран — ответ:

```
{
  "code": 0,
```

```
"session": "553cfd37-3b16-4b1e-8a77-1f490236e4a7",
"screenData": {
  "id": "01screen",
  "title": "title01screen",
  "decor": "button",
  "type": "digital",
  "fields": [
    {
      "code": "id1",
      "type": "TEXT_FIELD",
      "title": "titleId1",
      "flags": 0,
      "keyboard": "DIGITAL",
      "letterCase": "Lower",
      "allowChangeCase": true,
      "secure": false,
      "maxLen": 10,
      "readOnly": false,
      "invalid": false,
      "required": true,
      "regexp": [
        "\\d{1,10}$"
      ]
    }
  ],
  "actions": [
    "next",
    "exit"
  ],
  "actionsExt": [
    {
      "type": "next",
      "title": "Далее",
      "pay": false
    },
    {
      "type": "exit",
      "title": "Выход",
      "pay": false
    }
  ]
},
"lastScreen": false,
```

```
"actionResult": {
  "code": 0,
  "description": "Шаг успешно выполнен, есть следующий шаг"
},
"attributes": [],
" navigations": [
  {
    "id": "01screen",
    "title": "title01screen",
    "current": true
  }
]
}
```

3. Второй экран — запрос:

```
{
  "session": "553cfd37-3b16-4b1e-8a77-1f490236e4a7",
  "screenData": {
    "fields": [
      {
        "code": "id1",
        "value": "1111111111",
        "originalValue": "1111111111"
      }
    ]
  },
  "action": "next",
  "screen": null
}
```

Второй экран — ответ:

```
{
  "code": 0,
  "session": "553cfd37-3b16-4b1e-8a77-1f490236e4a7",
  "screenData": {
    "id": "02screen",
    "title": "title02screen",
    "decor": "button",
    "type": "digital",
    "fields": [
      {
        "code": "id2",
```

```
"type": "TEXT_FIELD",
"title": "titleId2",
"flags": 0,
"keyboard": "DIGITAL",
"letterCase": "Lower",
"allowChangeCase": true,
"secure": false,
"maxLen": 10,
"readOnly": false,
"invalid": false,
"required": true,
"regexp": [
  "^\\d{1,10}$"
]
},
"actions": [
  "next",
  "prev",
  "exit"
],
"actionsExt": [
  {
    "type": "next",
    "title": "Далее",
    "pay": true
  },
  {
    "type": "prev",
    "title": "Назад",
    "pay": false
  },
  {
    "type": "exit",
    "title": "Выход",
    "pay": false
  }
],
"lastScreen": true,
"actionResult": {
  "code": 0,
  "description": "Шаг успешно выполнен, есть следующий шаг"
```

```
},
"attributes": [
  {
    "code": "id1",
    "title": "titleId1",
    "value": "1111111111",
    "valueTitle": "1111111111",
    "originalValue": "1111111111",
    "flags": 0
  }
],
"navigations": [
  {
    "id": "01screen",
    "title": "title01screen",
    "current": false
  },
  {
    "id": "02screen",
    "title": "title02screen",
    "current": true
  }
]
}
```

4. Переход на экран оплаты — запрос:

```
{
  "session": "553cfd37-3b16-4b1e-8a77-1f490236e4a7",
  "screenData": {
    "fields": [
      {
        "code": "id2",
        "value": "2222222222",
        "originalValue": "2222222222"
      }
    ]
  },
  "action": "next",
  "screen": null
}
```

Переход на экран оплаты — ответ:

```
{
  "code": 0,
  "session": "553cfd37-3b16-4b1e-8a77-1f490236e4a7",
  "minSum": 100,
  "maxSum": 500000,
  "lastScreen": true,
  "actionResult": {
    "code": 1,
    "description": "Шаг успешно выполнен, является завершающим"
  },
  "attributes": [
    {
      "code": "id1",
      "title": "titleId1",
      "value": "1111111111",
      "valueTitle": "1111111111",
      "originalValue": "1111111111",
      "flags": 0
    },
    {
      "code": "id2",
      "title": "titleId2",
      "value": "2222222222",
      "valueTitle": "2222222222",
      "originalValue": "2222222222",
      "flags": 0
    }
  ],
  "navigations": [
    {
      "id": "01screen",
      "title": "title01screen",
      "current": false
    },
    {
      "id": "02screen",
      "title": "title02screen",
      "current": false
    }
  ]
}
```

5. Возврат на первый экран — запрос:

```
{
  "session": "553cfd37-3b16-4b1e-8a77-1f490236e4a7",
  "screenData": null,
  "action": "__navi",
  "screen": "01screen"
}
```

Возврат на первый экран — ответ:

```
{
  "code": 0,
  "session": "553cfd37-3b16-4b1e-8a77-1f490236e4a7",
  "screenData": {
    "id": "01screen",
    "title": "title01screen",
    "decor": "button",
    "type": "digital",
    "fields": [
      {
        "code": "id1",
        "type": "TEXT_FIELD",
        "title": "titleId1",
        "value": "1111111111",
        "valueTitle": "1111111111",
        "originalValue": "1111111111",
        "flags": 0,
        "keyboard": "DIGITAL",
        "letterCase": "Lower",
        "allowChangeCase": true,
        "secure": false,
        "maxLen": 10,
        "readOnly": false,
        "invalid": false,
        "required": true,
        "regex": [
          "\\d{1,10}$"
        ]
      }
    ]
  },
  "actions": [
    "next",
    "exit"
  ]
}
```

```
],
  "actionsExt": [
    {
      "type": "next",
      "title": "Далее",
      "pay": false
    },
    {
      "type": "exit",
      "title": "Выход",
      "pay": false
    }
  ]
},
"lastScreen": false,
"actionResult": {
  "code": 0,
  "description": "Шаг успешно выполнен, есть следующий шаг"
},
"attributes": [
  {
    "code": "id1",
    "title": "titleId1",
    "value": "1111111111",
    "valueTitle": "1111111111",
    "originalValue": "1111111111",
    "flags": 0
  }
],
"navigations": [
  {
    "id": "01screen",
    "title": "title01screen",
    "current": true
  }
]
}
```

2.11 ОБЩИЕ КОДЫ РЕЗУЛЬТАТОВ

Протокол в ответ на оба типа запросов всегда предполагает возврат кода результата в объекте ответа.

Коды результатов приведены в таблице 2.11.1.

Таблица 2.11.1 — Коды результатов

Код	Описание
0	Успешное выполнение запроса
1	Ошибка выполнения, сервис не найден
5	Запрошенное действие выполнить невозможно в текущем контексте
6	Токен (сессия) невалиден, истекло время не активности
7	Редирект
100	Внутренняя ошибка системы, ошибка базы данных
200	Пользователь вызвал EXIT
201	Не корректный формат запроса

2.12 ФЛАГИ ЭЛЕМЕНТОВ

Флаги элементов — битовые значения, каждое из которых отвечает за один пункт, определяющий статус элемента. Расшифровка их приведена ниже. Текстовое значение для целей протокола не используется, и приведено для целей понимания и сопоставления флагов полей в терминах языка сценариев и форм. На уровне протокола флаги возвращаются в виде целочисленного значения, целое число представляет собой битовую маску. Расшифровка и значение отдельных битов приведено в таблице 2.12.1.

Таблица 2.12.1 — Расшифровка и значение отдельных битов

Символьная константа	Числовое значение	Описание
HIDE_ON_CONFIRM	0x01	Не показывать на экране подтверждения
HIDE_ON_PRINT	0x02	Не выводить на печать в чеке. Используется, в том случае когда атрибут платежа, не нужно печатать на чеке
HIDE_ON_SERVER	0x04	Не показывать в атрибутах платежа в кабинете. Администратору ПС атрибут отображается вне зависимости от значения флага
HIDE_FROM_EXTERNAL	0x100	Запрет отдачи атрибута внешним агентам
HIDE_ON_FRONTEND	0x200	Не отображать атрибут плательщику. Используется, в том случае когда для совершения платежа передается служебный атрибут, который не нужно показывать

		пользователю
USE_CONTACTS	0x10	Возможность использования контактов из телефонной книги в качестве значений поля . Используется на мобильных клиентах
VALIDATE_BY_CAPACITY	0x20	Использовать валидацию по номерным емкостям и БДПН. Предназначен для полей, содержащие телефонные номера
RECEIVED_FROM_SERVER	0x300	Атрибут был получен от сервера. Используется при онлайн-запросах. Для использования в коде сценария недоступен. Используется разработчиками
VALUE_IS_UNCHANGED	0x400	Позволяет помечать input-элементы, как не редактируемые
MASKED	0x800	Определяет вывод значений атрибута в логах в маскированном виде. При этом originalValue, valueTitle не передаются, а в value каждый оригинальный символ значения заменяется на символ «*»

2.13 ИЗМЕНЕНИЕ ПРОИЗВОЛЬНОГО ШАГА

Пример для сценария:

```
<scenario begin="01screen">
  <screen type="digital" decor="button" title="title01screen"
id="01screen">
  <fields>
    <text-field id="id1" title="titleId1" max-len="10"
keyboard="Digital">
      <validator type="regex">
        <rules>
          <rule regex="^\d{1,10}$"/>
        </rules>
      </validator>
    </text-field>
  </fields>
  <actions>
    <action type="Next" title="Далее">
      <goto target="02screen"/>
    </action>
    <action type="Exit" title="Выход">
      <goto target="exit"/>
    </action>
  </actions>
</screen>
  <screen type="digital" decor="button" title="title02screen"
id="02screen">
  <fields>
    <text-field id="id2" title="titleId2" max-len="10"
keyboard="Digital">
      <validator type="regex">
        <rules>
          <rule regex="^\d{1,10}$"/>
        </rules>
      </validator>
    </text-field>
  </fields>
  <actions>
```

```
<action type="Next" title="Далее">
  <goto target="pay"/>
</action>
<action type="Prev" title="Назад">
  <goto target="01screen"/>
</action>
<action type="Exit" title="Выход">
  <goto target="exit"/>
</action>
</actions>
</screen>
</scenario>
```

1. Start — запрос:

```
{"idPoint":"1","idService":"1","verify":false,"locale":"","firstScreen":false}
```

Start — ответ:

```
{"code":0,"session":"2c5d9370-c907-4947-ae71-6c756ac9ce88"}
```

2. Первый экран — запрос:

```
{"session":"2c5d9370-c907-4947-ae71-6c756ac9ce88","screenData":null,"action":"next","screen":null}
```

Первый экран — ответ:

```
{
  "code": 0,
  "session": "2c5d9370-c907-4947-ae71-6c756ac9ce88",
  "screenData": {
    "id": "01screen",
    "title": "title01screen",
    "decor": "button",
    "type": "digital",
    "fields": [
      {
        "code": "id1",
        "type": "TEXT_FIELD",
        "title": "titleId1",
        "flags": 0,
        "keyboard": "DIGITAL",
```

```
        "letterCase": "Lower",
        "allowChangeCase": true,
        "secure": false,
        "maxLen": 10,
        "readOnly": false,
        "invalid": false,
        "required": true,
        "regexp": [
            "\\d{1,10}$"
        ]
    },
    ],
    "actions": [
        "next",
        "exit"
    ],
    "actionsExt": [
        {
            "type": "next",
            "title": "Далее",
            "pay": false
        },
        {
            "type": "exit",
            "title": "Выход",
            "pay": false
        }
    ]
},
"lastScreen": false,
"actionResult": {
    "code": 0,
    "description": "Шаг успешно выполнен, есть следующий шаг"
},
"attributes": [],
" navigations": [
    {
        "id": "01screen",
        "title": "title01screen",
        "current": true
    }
]
}
```

3. Второй экран — запрос:

```
{
  "session": "2c5d9370-c907-4947-ae71-6c756ac9ce88",
  "screenData": {
    "fields": [
      {
        "code": "id1",
        "value": "1111111111",
        "originalValue": "1111111111"
      }
    ]
  },
  "action": "next",
  "screen": null
}
```

Второй экран — ответ:

```
{
  "code": 0,
  "session": "2c5d9370-c907-4947-ae71-6c756ac9ce88",
  "screenData": {
    "id": "02screen",
    "title": "title02screen",
    "decor": "button",
    "type": "digital",
    "fields": [
      {
        "code": "id2",
        "type": "TEXT_FIELD",
        "title": "titleId2",
        "flags": 0,
        "keyboard": "DIGITAL",
        "letterCase": "Lower",
        "allowChangeCase": true,
        "secure": false,
        "maxLen": 10,
        "readOnly": false,
        "invalid": false,
        "required": true,
        "regexp": [
          "\\d{1,10}$"
        ]
      }
    ]
  }
}
```

```
    }
  ],
  "actions": [
    "next",
    "prev",
    "exit"
  ],
  "actionsExt": [
    {
      "type": "next",
      "title": "Далее",
      "pay": true
    },
    {
      "type": "prev",
      "title": "Назад",
      "pay": false
    },
    {
      "type": "exit",
      "title": "Выход",
      "pay": false
    }
  ]
},
"lastScreen": true,
"actionResult": {
  "code": 0,
  "description": "Шаг успешно выполнен, есть следующий шаг"
},
"attributes": [
  {
    "code": "id1",
    "title": "titleId1",
    "value": "1111111111",
    "valueTitle": "1111111111",
    "originalValue": "1111111111",
    "flags": 0
  }
],
"navigations": [
  {
    "id": "01screen",
```

```
    "title": "title01screen",
    "current": false
  },
  {
    "id": "02screen",
    "title": "title02screen",
    "current": true
  }
]
```

4. Переход на экран оплаты — запрос:

```
{
  "session": "2c5d9370-c907-4947-ae71-6c756ac9ce88",
  "screenData": {
    "fields": [
      {
        "code": "id2",
        "value": "2222222222",
        "originalValue": "2222222222"
      }
    ]
  },
  "action": "next",
  "screen": null
}
```

Переход на экран оплаты — ответ:

```
{
  "code": 0,
  "session": "2c5d9370-c907-4947-ae71-6c756ac9ce88",
  "minSum": 100,
  "maxSum": 500000,
  "lastScreen": true,
  "actionResult": {
    "code": 1,
    "description": "Шаг успешно выполнен, является завершающим"
  },
  "attributes": [
    {
      "code": "id1",
      "title": "titleId1",
    }
  ]
}
```

```
"value": "1111111111",
"valueTitle": "1111111111",
"originalValue": "1111111111",
"flags": 0
},
{
  "code": "id2",
  "title": "titleId2",
  "value": "2222222222",
  "valueTitle": "2222222222",
  "originalValue": "2222222222",
  "flags": 0
}
],
"navigations": [
  {
    "id": "01screen",
    "title": "title01screen",
    "current": false
  },
  {
    "id": "02screen",
    "title": "title02screen",
    "current": false
  }
]
}
```

5. Изменение данных на первом экране — запрос:

```
{
  "session": "2c5d9370-c907-4947-ae71-6c756ac9ce88",
  "screen": "01screen",
  "screenData": {
    "fields": [
      {
        "code": "id1",
        "value": "3333333333",
        "originalValue": "3333333333"
      }
    ]
  },
  "action": "next"
}
```

Изменение данных на первом экране — ответ:

```
{
  "code": 0,
  "session": "2c5d9370-c907-4947-ae71-6c756ac9ce88",
  "screenData": {
    "id": "02screen",
    "title": "title02screen",
    "decor": "button",
    "type": "digital",
    "fields": [
      {
        "code": "id2",
        "type": "TEXT_FIELD",
        "title": "titleId2",
        "value": "2222222222",
        "valueTitle": "2222222222",
        "originalValue": "2222222222",
        "flags": 0,
        "keyboard": "DIGITAL",
        "letterCase": "Lower",
        "allowChangeCase": true,
        "secure": false,
        "maxLen": 10,
        "readOnly": false,
        "invalid": false,
        "required": true,
        "regexp": [
          "\\d{1,10}$"
        ]
      }
    ],
    "actions": [
      "next",
      "prev",
      "exit"
    ],
    "actionsExt": [
      {
        "type": "next",
        "title": "Далее",
        "pay": true
      }
    ]
  }
}
```

```
{
  "type": "prev",
  "title": "Назад",
  "pay": false
},
{
  "type": "exit",
  "title": "Выход",
  "pay": false
}
],
"lastScreen": true,
"actionResult": {
  "code": 0,
  "description": "Шаг успешно выполнен, есть следующий шаг"
},
"attributes": [
  {
    "code": "id1",
    "title": "titleId1",
    "value": "3333333333",
    "valueTitle": "3333333333",
    "originalValue": "3333333333",
    "flags": 0
  },
  {
    "code": "id2",
    "title": "titleId2",
    "value": "2222222222",
    "valueTitle": "2222222222",
    "originalValue": "2222222222",
    "flags": 0
  }
],
"navigations": [
  {
    "id": "01screen",
    "title": "title01screen",
    "current": false
  },
  {
    "id": "02screen",
```

```
"title": "title02screen",  
  "current": true  
}  
]  
}
```

2.14 ПРИМЕРЫ ЗАПРОСОВ

Для примера возьмём форму:

```
<?xml version="1.0" encoding="UTF-8"?>
<forms>
  <form service-id="915">
    <text-selector name="selector" message="M-selector" title="T-selector"
      flags="HIDE_ON_SERVER" message-id="MID-selector"
      title-id="TID-selector" decor="button">
      <item text="T-item1" value="val1" text-id="TID-item1" sum="50.55">
        <group-field>
          <date-field name="date" title="T-date" message="M-date"
            message-id="MID-date" title-id="TID-date" flags="HIDE_ON_PRINT"
            format="dd.MM.yy" format-title="dd.MM.yy" from="-1d" to="1d"
            default="1m"/>
          </group-field>
          <set-field name="set1" value="54321" title="T-set1"/>
        </item>
        <text-field name="text" keyboard="Digital" max-len="15"
          message="M-text" message-id="MID-text"
          title="T-text" title-id="TID-text"
          flags="5" example="abc" read-only="false"
          secure="false" default="123">
          <verify>
            <rule regex="^\d{3}$"/>
            <rule regex="^\d{3}$"/>
            <block regex="abc"/>
          </verify>
          <help>H-text</help>
        </text-field>
      </item>
      <item text="T-item2" value="val2">
        <numeric-field name="num" title="T-num" title-id="TID-num" message="M-num" message-id="MID-num"
          flags="HIDE_ON_PRINT|HIDE_ON_CONFIRM" min="10" max="15000" change-sumpurchase="true">
          <help>H-num</help>
        </numeric-field>
        <table-field name="table" message="M-table" title="T-table" title-id="TID-table"
          message-id="MID-table" read-only="false" decor="table1">
          <columns>
            <column key="#name" title="Маршрут:" width="0.4"/>
            <column key="#km" title="Расстояние:" width="0.3"/>
            <column key="#price" title="Цена:" width="0.3"/>
          </columns>
          <items>
            <item>
              <cell key="#name" key-title="Маршрут" value="Маршрут 1" value-title="Маршрут 1"/>
              <cell key="#km" key-title="Расстояние" value="25" value-title="25"/>
              <cell key="#price" key-title="Цена" value="256" value-title="256" sum="12"/>
            </item>
            <item>
              <cell key="#name" key-title="Маршрут" value="Маршрут 2" value-title="Маршрут 2"/>
              <cell key="#km" key-title="Расстояние" value="21" value-title="21"/>
              <cell key="#price" key-title="Цена" value="222" value-title="222" sum="12"/>
            </item>
          </items>
        </table-field>
      </item>
    </text-selector>
  </form>
</forms>
```

Инициализирующий запрос:

```
{
  "idPoint": 636,
  "idService": 915,
  "verify": false
}
```

Ответ на инициализирующий запрос:

```
{
  "code": 0,
  "session": "4395483223494687960"
}
```

Первый вызов action:

```
{
  "session": "4395483223494687960",
  "screenData": null,
  "action": "next"
}
```

Ответ на первый вызов action:

```
{
  "code": 0,
  "screenData": {
    "title": "M-selector",
    "titleId": "MID-selector",
    "help": null,
    "decor": "button",
    "fields": [
      {
        "code": "selector",
        "type": "SELECTOR_FIELD",
        "title": "T-selector",
        "titleId": "TID-selector",
        "value": null,
        "originalValue": null,
        "flags": 4,

```

```
"keyboard": null,  
"letterCase": null,  
"secure": false,  
"defaultValue": null,  
"example": null,  
"prefix": null,  
"postfix": null,  
"maxLen": null,  
"readOnly": null,  
"format": null,  
"formatTitle": null,  
"dateRange": null,  
"range": null,  
"invalid": false,  
"help": null,  
"helpId": null,  
"regexp": null,  
"blockRegexp": null,  
"values": [  
  {  
    "title": "T-item1",  
    "value": "val1",  
    "help": null  
  },  
  {  
    "title": "T-item2",  
    "value": "val2",  
    "help": null  
  }  
],  
"tableItems": null  
}  
],  
"actions": [  
  "next",  
  "exit",  
  "prev"  
]  
},  
"sumpurchase": null,  
"actionResult": {  
  "code": 0,  
  "description": "Шаг успешно выполнен, есть следующий шаг",
```

```
"messages": null
},
"attributes": []
}
```

Заполняем поле **value** для поля **select** и отправляем запрос:

```
{
  "session": "4395483223494687960",
  "screenData": {
    "title": "M-selector",
    "titleId": "MID-selector",
    "help": null,
    "decor": "button",
    "fields": [
      {
        "code": "selector",
        "type": "SELECTOR_FIELD",
        "title": "T-selector",
        "titleId": "TID-selector",
        "value": "V-item1",
        "originalValue": "V-item1",
        "flags": 4,
        "keyboard": null,
        "letterCase": null,
        "secure": false,
        "defaultValue": null,
        "example": null,
        "prefix": null,
        "postfix": null,
        "maxLen": null,
        "readOnly": null,
        "format": null,
        "formatTitle": null,
        "dateRange": null,
        "range": null,
        "invalid": false,
        "help": null,
        "helpId": null,
        "regexp": null,
        "blockRegexp": null,
        "values": [
          {
            "title": "T-item1",
```

```
        "value": "val1",
        "help": null
    },
    {
        "title": "T-item2",
        "value": "val2",
        "help": null
    }
],
"tableItems": null
}
],
"actions": [
    "next",
    "exit",
    "prev"
]
},
"action": "next"
}
```

При правильном заполнении получим следующее поле из формы:

```
{
  "code": 0,
  "screenData": {
    "title": "M-date",
    "titleId": "MID-date",
    "help": null,
    "decor": null,
    "fields": [
      {
        "code": "date",
        "type": "DATE_FIELD",
        "title": "T-date",
        "titleId": "TID-date",
        "value": null,
        "originalValue": null,
        "flags": 2,
        "keyboard": null,
        "letterCase": null,
        "secure": false,
        "defaultValue": "16.04.18",
        "example": null,

```

```
"prefix": null,
"postfix": null,
"maxLen": null,
"readOnly": null,
"format": "dd.MM.yy",
"formatTitle": "dd.MM.yy",
"dateRange": {
  "start": "15.03.2018 00:00:00",
  "end": "17.03.2018 00:00:00"
},
"range": null,
"invalid": false,
"help": null,
"helpId": null,
"regexp": null,
"blockRegexp": null,
"values": null,
"tableItems": null
}
],
"actions": [
  "next",
  "exit",
  "prev"
]
},
"sumpurchase": 5055,
"actionResult": {
  "code": 0,
  "description": "Шаг успешно выполнен, есть следующий шаг",
  "messages": null
},
"attributes": [
  {
    "code": "selector",
    "title": "T-selector",
    "titleId": "TID-selector",
    "value": "val1",
    "valueTitle": "T-item1",
    "originalValue": "val1",
    "flags": 4
  },
  {
```

```
"code": "#sum",
"title": "Сумма зачисленная",
"titleId": null,
"value": "5055",
"valueTitle": null,
"originalValue": "5055",
"flags": 0
},
{
  "code": "set1",
  "title": "T-set1",
  "titleId": null,
  "value": "54321",
  "valueTitle": null,
  "originalValue": "54321",
  "flags": 0
}
]
}
```

Рассмотрим случай с данными, не проходящими валидацию. В таком случае нам возвращается то же поле, но в объекте **actionResult** проставляется код ошибки валидации и добавляется соответствующее сообщение:

```
{
  "code": 0,
  "screenData": {
    "title": "M-text",
    "titleId": "MID-text",
    "help": null,
    "decor": "button",
    "fields": [
      {
        "code": "text",
        "type": "TEXT_FIELD",
        "title": "T-text",
        "titleId": "TID-text",
        "value": "asdasd",
        "originalValue": "asdasd",
        "flags": 5,
        "keyboard": "DIGITAL",
        "letterCase": "Lower",
        "secure": false,

```

```
"defaultValue": "123",
"example": "abc",
"prefix": null,
"postfix": null,
"maxLen": 15,
"readOnly": false,
"format": null,
"formatTitle": null,
"dateRange": null,
"range": null,
"invalid": true,
"help": "H-text",
"helpId": null,
"regexp": [
  "\\d{3}$",
  "^123$"
],
"blockRegexp": [
  "abc"
],
"values": null,
"tableItems": null
}
],
"actions": [
  "next",
  "exit",
  "prev"
]
},
"sumpurchase": 5055,
"actionResult": {
  "code": 2,
  "description": "При выполнении шага произошла ошибка валидации,
переход на следующий шаг не произведен",
  "messages": null
},
"attributes": [
  {
    "code": "selector",
    "title": "T-selector",
    "titleId": "TID-selector",
    "value": "V-item1",
```

```
"valueTitle": "T-item1",
"originalValue": "V-item1",
"flags": 4
},
{
"code": "#sum",
"title": "Сумма зачисленная",
"titleId": null,
"value": "5055",
"valueTitle": null,
"originalValue": "5055",
"flags": 0
},
{
"code": "set1",
"title": "T-set1",
"titleId": null,
"value": "54321",
"valueTitle": null,
"originalValue": "54321",
"flags": 0
},
{
"code": "date",
"title": "T-date",
"titleId": "TID-date",
"value": "01.March.2018",
"valueTitle": null,
"originalValue": "01.March.2018",
"flags": 2
}
]
}
```

2.15 ПРИМЕРЫ РАЗЛИЧНЫХ ТИПОВ ПОЛЕЙ

2.15.1 ТЕКСТОВОЕ ПОЛЕ

В форме добавлено текстовое поле:

```
<text-field name="text" keyboard="Digital"
  max-len="15" message="M-text"
  message-id="MID-text"
  title="T-text" title-id="TID-text"
  flags="5" example="abc" read-only="false"
  secure="false" default="123">
  <verify>
    <rule regex="^\d{3}$"/>
    <rule regex="^123$"/>
    <block regex="abc"/>
  </verify>
  <help>H-text</help>
</text-field>
```

В ответе получаем:

```
"screenData": {
  "title": "M-text",
  "titleId": "MID-text",
  "help": null,
  "decor": "button",
  "fields": [
    {
      "code": "text",
      "type": "TEXT_FIELD",
      "title": "T-text",
      "titleId": "TID-text",
      "value": null,
      "originalValue": null,
      "flags": 5,
      "keyboard": "DIGITAL",
```

```
    "letterCase": "Lower",
    "secure": false,
    "defaultValue": "123",
    "example": "abc",
    "prefix": null,
    "postfix": null,
    "maxLen": 15,
    "readOnly": false,
    "format": null,
    "formatTitle": null,
    "dateRange": null,
    "range": null,
    "invalid": false,
    "help": "H-text",
    "helpId": null,
    "regexp": [
      "\\d{3}$",
      "123$"
    ],
    "blockRegexp": [
      "abc"
    ],
    "values": null,
    "tableItems": null
  }
],
"actions": [
  "next",
  "exit",
  "prev"
]
}
```

2.15.2 ЧИСЛОВОЕ ПОЛЕ

В форме добавлено числовое поле:

```
<numeric-field name="num" title="Т-num" title-id="TID-num"
              message="M-num" message-id="MID-num"
              flags="HIDE_ON_PRINT|HIDE_ON_CONFIRM"
              min="10" max="15000" change-sumpurchase="true">
  <help>H-num</help>
</numeric-field>
```

В ответе получаем:

```
"screenData": {
  "title": "M-num",
  "titleId": "MID-num",
  "help": null,
  "decor": null,
  "fields": [
    {
      "code": "#sum",
      "type": "NUMERIC_FIELD",
      "title": "Т-num",
      "titleId": "TID-num",
      "value": null,
      "originalValue": null,
      "flags": 3,
      "keyboard": "DIGITAL",
      "letterCase": "Lower",
      "secure": false,
      "defaultValue": null,
      "example": null,
      "prefix": null,
      "postfix": null,
      "maxLen": null,
      "readOnly": false,
      "format": null,
      "formatTitle": null,
      "dateRange": null,
```

```
        "range": null,  
        "invalid": false,  
        "help": "H-num",  
        "helpId": null,  
        "regexp": [],  
        "blockRegexp": [],  
        "values": null,  
        "tableItems": null  
    }  
],  
"actions": [  
    "next",  
    "exit",  
    "prev"  
]  
}
```

2.15.3 СЕЛЕКТОР

В форме добавлен селектор с двумя элементами для выбора:

```
<text-selector name="selector0" message="M-selector"
              title="T-selector" flags="HIDE_ON_SERVER"
              message-id="MID-selector" title-id="TID-selector"
              decor="button">
  <item text="T-item1" value="val1" text-id="TID-item1"/>
  <item text="T-item2" value="val2" text-id="TID-item2"/>
</text-selector>
```

В ответе получаем:

```
"screenData": {
  "title": "M-selector",
  "titleId": "MID-selector",
  "help": null,
  "decor": "button",
  "fields": [
    {
      "code": "selector0",
      "type": "SELECTOR_FIELD",
      "title": "T-selector",
      "titleId": "TID-selector",
      "value": null,
      "originalValue": null,
      "flags": 4,
      "keyboard": null,
      "letterCase": null,
      "secure": false,
      "defaultValue": null,
      "example": null,
      "prefix": null,
      "postfix": null,
      "maxLen": null,
      "readOnly": null,
      "format": null,
      "formatTitle": null,
```

```
"dateRange": null,  
"range": null,  
"invalid": false,  
"help": null,  
"helpId": null,  
"regexp": null,  
"blockRegexp": null,  
"values": [  
  {  
    "title": "T-item1",  
    "value": "val1",  
    "help": null  
  },  
  {  
    "title": "T-item2",  
    "value": "val2",  
    "help": null  
  }  
],  
"tableItems": null  
}  
],  
"actions": [  
  "next",  
  "exit",  
  "prev"  
]  
}
```

2.15.4 ВЫБОР ДАТЫ

В форме добавлено поле для выбора даты:

```
<date-field name="date" title="T-date" message="M-date"
            message-id="MID-date" title-id="TID-date"
            flags="HIDE_ON_PRINT"
            format="dd.MM.yy" format-title="dd.MM.yy"
            from="-1d" to="1d" default="1m">
</date-field>
```

В ответе получаем:

```
"screenData": {
  "title": "M-date",
  "titleId": "MID-date",
  "help": null,
  "decor": null,
  "fields": [
    {
      "code": "date",
      "type": "DATE_FIELD",
      "title": "T-date",
      "titleId": "TID-date",
      "value": null,
      "originalValue": null,
      "flags": 2,
      "keyboard": null,
      "letterCase": null,
      "secure": false,
      "defaultValue": "16.04.18",
      "example": null,
      "prefix": null,
      "postfix": null,
      "maxLen": null,
      "readOnly": null,
      "format": "dd.MM.yy",
      "formatTitle": "dd.MM.yy",
      "dateRange": {
```

```
        "start": "15.03.2018 00:00:00",
        "end": "17.03.2018 00:00:00"
    },
    "range": null,
    "invalid": false,
    "help": null,
    "helpId": null,
    "regexp": null,
    "blockRegexp": null,
    "values": null,
    "tableItems": null
}
],
"actions": [
    "next",
    "exit",
    "prev"
]
}
```

2.15.5 ТАБЛИЧНЫЙ СЕЛЕКТОР

В форме добавлено числовое поле:

```
<table-field name="table" message="M-table" title="T-table"
            title-id="TID-table" message-id="MID-table"
            read-only="false" decor="table1">
  <columns>
    <column key="#name" title="Маршрут:" width="0.4"/>
    <column key="#km" title="Расстояние:" width="0.3"/>
    <column key="#price" title="Цена:" width="0.3"/>
  </columns>
  <items>
    <item sum="256">
      <cell key="#name" key-title="Маршрут"
            value="Маршрут 1" value-title="Маршрут 1"/>
      <cell key="#km" key-title="Расстояние"
            value="25" value-title="25"/>
      <cell key="#price" key-title="Цена"
            value="256" value-title="256"/>
    </item>
    <item sum="222">
      <cell key="#name" key-title="Маршрут"
            value="Маршрут 2" value-title="Маршрут 2"/>
      <cell key="#km" key-title="Расстояние"
            value="21" value-title="21"/>
      <cell key="#price" key-title="Цена"
            value="222" value-title="222" sum="12"/>
    </item>
  </items>
</table-field>
```

В ответе получаем:

```
"screenData": {
  "title": "M-table",
  "titleId": "MID-table",
  "help": null,
  "decor": "table1",
```

```
"fields": [
  {
    "code": "table",
    "type": "TABLE_FIELD",
    "title": "T-table",
    "titleId": "TID-table",
    "value": null,
    "originalValue": null,
    "flags": 0,
    "keyboard": null,
    "letterCase": null,
    "secure": false,
    "defaultValue": null,
    "example": null,
    "prefix": null,
    "postfix": null,
    "maxLen": null,
    "readOnly": null,
    "format": null,
    "formatTitle": null,
    "dateRange": null,
    "range": null,
    "invalid": false,
    "help": null,
    "helpId": null,
    "regexp": null,
    "blockRegexp": null,
    "values": null,
    "tableItems": [
      {
        "value": " #name: Маршрут 1, #km: 25, #price: 256",
        "valueTitle": " Маршрут: Маршрут 1, Расстояние: 25, Цена:
256",
        "elements": [
          {
            "code": "#name",
            "title": "Маршрут",
            "titleId": null,
            "value": "Маршрут 1",
            "valueTitle": "Маршрут 1",
            "flags": 0
          },
          {

```

```
        "code": "#km",
        "title": "Расстояние",
        "titleId": null,
        "value": "25",
        "valueTitle": "25",
        "flags": 0
    },
    {
        "code": "#price",
        "title": "Цена",
        "titleId": null,
        "value": "256",
        "valueTitle": "256",
        "flags": 0
    }
]
},
{
    "value": " #name: Маршрут 2, #km: 21, #price: 222",
    "valueTitle": " Маршрут: Маршрут 2, Расстояние: 21, Цена:
222",
    "elements": [
        {
            "code": "#name",
            "title": "Маршрут",
            "titleId": null,
            "value": "Маршрут 2",
            "valueTitle": "Маршрут 2",
            "flags": 0
        },
        {
            "code": "#km",
            "title": "Расстояние",
            "titleId": null,
            "value": "21",
            "valueTitle": "21",
            "flags": 0
        },
        {
            "code": "#price",
            "title": "Цена",
            "titleId": null,
```

```
        "value": "222",
        "valueTitle": "222",
        "flags": 0
      }
    ]
  }
]
}
],
"actions": [
  "next",
  "exit",
  "prev"
]
}
```

2.15.6 СЦЕНАРИИ. РЕДИРЕКТ

Если во время выполнения встречается редирект, то возвращается общий код «7», который сигнализирует о том, что нужно совершить редирект и данные для редиректа (redirectData). Для того чтобы корректно обработать редирект, необходимо создать новую сессию по сервису, полученному в redirectData, и передать стартовые данные.

В сценарии добавлен редирект на другой сервис:

```
<scenario begin="01screen">
  <screen type="digital" decor="button" title="title01" id="01screen">
    <fields>
      <text-field id="id1" title="titleId1" max-len="10"
        keyboard="Digital">
        <validator type="regex">
          <rules>
            <rule regex="^\d{1,20}$"/>
          </rules>
        </validator>
      </text-field>
    </fields>
    <actions>
```

```
<action type="Next" title="Далее">
  <redirect target="7002" params="id1"/>
</action>
<action type="Exit" title="Выход">
  <goto target="exit"/>
</action>
</actions>
</screen>
</scenario>
```

Получаем первый экран:

```
{
  "code": 0,
  "screenData": {
    "title": "title01",
    "decor": "button",
    "type": "digital",
    "fields": [
      {
        "code": "id1",
        "type": "TEXT_FIELD",
        "title": "titleId1",
        "flags": 0,
        "keyboard": "DIGITAL",
        "letterCase": "Lower",
        "secure": false,
        "maxLen": 10,
        "readOnly": false,
        "invalid": false,
        "required": true,
        "regexp": [
          "\\d{1,20}$"
        ],
        "blockRegexp": []
      }
    ],
    "actions": [
      "next",
      "exit"
    ],
    "actionsExt": [
      {
        "type": "next",
```

```
        "title": "Далее"
      },
      {
        "type": "exit",
        "title": "Выход"
      }
    ]
  },
  "actionResult": {
    "code": 0,
    "description": "Шаг успешно выполнен, есть следующий шаг"
  },
  "attributes": []
}
```

Заполняем текстовое поле (id1):

```
{
  "session": "c99dc9fe-d6b5-4a13-8d3f-5c01cbf5dbe8",
  "screenData": {
    "fields": [
      {
        "code": "id1",
        "value": "11111",
        "originalValue": "11111"
      }
    ],
  },
  "action": "next"
}
```

В ответ получаем экран с информацией о редиректе:

```
{
  "code": 7,
  "redirectData": {
    "idService": 7002,
    "params": [
      {
        "code": "id1",
        "title": "titleId1",
        "value": "11111",

```

```
        "valueTitle": "11111",
        "originalValue": "11111",
        "flags": 0
    }
]
},
"actionResult": {
    "code": 1,
    "description": "Шаг успешно выполнен, является завершающим"
},
"attributes": [
    {
        "code": "id1",
        "title": "titleId1",
        "value": "11111",
        "valueTitle": "11111",
        "originalValue": "11111",
        "flags": 0
    }
]
}
```

Код ответа говорит о том что нужно выполнить редирект, также мы получили информацию о сервисе (redirectData.idService) и стартовых данных для редиректа.

Для того, чтобы продолжить работу с сервисом, необходимо создать новую сессию с полученными данными из последнего шага:

```
{
    "idPoint": 636,
    "idService": "7002",
    "verify": false,
    "redirectData": {
        "params": [
            {
                "code": "id1",
                "title": "titleId1",
                "value": "11111",
                "valueTitle": "11111",
                "originalValue": "11111",
                "flags": 0
            }
        ]
    }
}
```

```
}  
]  
}  
}
```

В ответ получим токен сессии:

```
{  
  "code": 0,  
  "session": "0ed80e50-ec41-4a35-9059-5df09170ae36"  
}
```

Отправим запрос на получение первого экрана

```
{  
  "session": "0ed80e50-ec41-4a35-9059-5df09170ae36",  
  "action": "next"  
}
```

Получим информацию о первом экране:

```
{  
  "code": 0,  
  "screenData": {  
    "title": "title01",  
    "decor": "",  
    "type": "digital",  
    "fields": [  
      {  
        "code": "id2",  
        "type": "TEXT_FIELD",  
        "title": "titleId2",  
        "flags": 0,  
        "keyboard": "DIGITAL",  
        "letterCase": "Lower",  
        "secure": false,  
        "maxLen": 10,  
        "readOnly": false,  
        "invalid": false,  
        "required": true,  
        "regexp": [  
          "^\\d{1,20}$"  
        ],  
      },  
    ],  
  },  
}
```

```
        "blockRegexp": []
    },
    ],
    "actions": [
        "next",
        "exit"
    ],
    "actionsExt": [
        {
            "type": "next",
            "title": "Далее"
        },
        {
            "type": "exit",
            "title": "Выход"
        }
    ]
},
"actionResult": {
    "code": 0,
    "description": "Шаг успешно выполнен, есть следующий шаг"
},
"attributes": [
    {
        "code": "id1",
        "title": "titleId1",
        "value": "11111",
        "valueTitle": "11111",
        "originalValue": "11111",
        "flags": 0
    }
]
}
```

Видим, что в атрибутах уже содержится переданный нами при старте атрибут **id1** со значением «**11111**».

2.15.7 СЦЕНАРИИ. КОРЗИНА ПЛАТЕЖЕЙ

Используем сценарий с корзиной платежей.

Сценарий состоит из трех экранов:

1. Ввод номера лицевого счета.
2. Экран корзины.
3. Экран изменения суммы платежа по сервису.

```
<scenario begin="input">
  <payment-params key="#services" distribution-type="consistent"
    single-check="true" />
  <!-- Номер квартиры -->
  <screen type="numeric" title="Введите номер счета" id="input">
    <fields>
      <text-field id="id1" title="Номер счета" max-len="5"
        keyboard="Digital" >
        <validator type="regex">
          <rules>
            <rule regex="^\d{5}$"/>
          </rules>
        </validator>
      </text-field>
    </fields>
    <actions>
      <action type="Next" title="ДАЛЕЕ">
        <online-request type="default" function="find-all"
          params="id1">
          <actions>
            <action type="success">
              <goto target="com_screen"/>
            </action>
            <action type="error">
              <dialog type="Error" title="Error"
                message="Please try again later" timeout="10" default="okay">
                <actions>
                  <action type="okay" title="OK">
```

```
                <cancel/>
            </action>
        </actions>
    </dialog>
    </action>
</actions>
</online-request>
</action>
<action type="Prev" title="НАЗАД">
    <goto target="previous"/>
</action>
<action type="Exit" title="ВЫХОД">
    <goto target="exit"/>
</action>
</actions>
</screen>
<!-- СТОИМОСТЬ УСЛУГ -->
<screen type="Communal" title="Выбор услуг" id="com_screen">
    <fields key="#services"/>
    <actions>
        <action type="Next" title="ДАЛЕЕ">
            <sum key="summ" result-key="ssum" input-data="#services"
                filter="selected=1"/>
            <set-sum from="ssum" type="units"/>
            <goto target="pay"/>
        </action>
        <action type="Prev" title="НАЗАД">
            <goto target="input"/>
        </action>
        <action type="Exit" title="ВЫХОД">
            <goto target="exit"/>
        </action>
        <action type="Edit" title="Удалить">
            <if condition="is-not-null #selected">
                <then>
                    <unpack type="data" key="#selected"
                        elements="summ,name"/>
                    <goto target="input_sum"/>
                </then>
                <else>
                    <if condition="is-not-null #removed">
                        <then>
```

```
<unpack type="data" key="#removed"
        elements="name"/>
<dialog type="Info" title="Запрос на
удаление" message="Вы действительно хотите удалить {0} из корзины?"
timeout="10" default="cancel" mess-params="name">
  <actions>
    <action type="okay" title="ДА">
      <set key="selected" value="0"
        value-title="0"/>
      <pack type="data"
        key="#removed"
        elements="selected"/>
      <clear keys="#removed"/>
      <goto target="com_screen"/>
    </action>
    <action type="cancel"
      title="НЕТ">
      <clear keys="#removed"/>
      <cancel/>
    </action>
  </actions>
</dialog>
</then>
<else/>
</if>
</else>
</if>
</action>
</actions>
</screen>
<screen type="Digital" decor="info" title="Укажите сумму услуги"
  id="input_sum">
  <fields>
    <numeric-field id="summ" title="Сумма платежа" format="5.2">
      <verify>
        <range begin="1" end="15000"/>
      </verify>
      <help>
        <![CDATA[<html>Услуга: ${name}<br>
        Минимальная сумма: 1 руб.<br>
        Максимальная сумма: 15000 руб.<br>]]>
      </help>
    </numeric-field>
```

```
</fields>
<actions>
  <action type="Next" title="ДАЛЕЕ">
    <set key="selected" value="1" value-title="1"/>
    <pack type="data" key="#selected"
      elements="summ,selected"/>
    <clear keys="#selected,summ,selected,name"/>
    <goto target="com_screen"/>
  </action>
  <action type="Prev" title="НАЗАД">
    <clear keys="#selected,name,summ"/>
    <goto target="com_screen"/>
  </action>
  <action type="Exit" title="ВЫХОД">
    <goto target="exit"/>
  </action>
</actions>
</screen>
</scenario>
```

Отправляем стартовый запрос:

```
{
  "idPoint": 636,
  "idService": "4351",
  "verify": false
}
```

В ответ получаем токен сессии:

```
{
  "code": 0,
  "session": "e20c9243-13b4-482d-8f9e-599bcab30791"
}
```

Запрашиваем первый экран:

```
{
  "session": "e20c9243-13b4-482d-8f9e-599bcab30791",
  "screenData": null,
  "action": "next"
}
```

В ответ получаем первый экран с полем ввода лицевого счета:

```
{
  "code": 0,
  "screenData": {
    "title": "Введите номер счета",
    "decor": "",
    "type": "numeric",
    "fields": [
      {
        "code": "idl",
        "type": "TEXT_FIELD",
        "title": "Номер счета",
        "flags": 0,
        "keyboard": "DIGITAL",
        "letterCase": "Lower",
        "secure": false,
        "maxLen": 5,
        "readOnly": false,
        "invalid": false,
        "required": true,
        "regexp": [
          "\\d{5}$"
        ],
        "blockRegexp": []
      }
    ],
    "actions": [
      "next",
      "prev",
      "exit"
    ],
    "actionsExt": [
      {
        "type": "next",
        "title": "ДАЛЕЕ"
      },
      {
        "type": "prev",
        "title": "НАЗАД"
      },
      {
        "type": "exit",

```

```
        "title": "ВЫХОД"
      }
    ]
  },
  "actionResult": {
    "code": 0,
    "description": "Шаг успешно выполнен, есть следующий шаг"
  },
  "attributes": []
}
```

Заполняем номер счета:

```
{
  "session": "e20c9243-13b4-482d-8f9e-599bcab30791",
  "screenData": {
    "fields": [
      {
        "code": "id1",
        "value": "11111",
      }
    ]
  },
  "action": "next"
}
```

В ответ получаем экран корзины:

```
{
  "code": 0,
  "screenData": {
    "title": "Выбор услуг",
    "decor": "",
    "type": "communal",
    "fields": [
      {
        "code": "#selected",
        "type": "BASKET_FIELD",
        "title": "Консьерж",
        "value": "Консьерж",
        "secure": false,
        "readOnly": false,
      }
    ]
  }
}
```

```
"invalid": false,  
"required": true,  
"basketField": {  
  "summ": "450"  
}  
},  
{  
  "code": "#selected",  
  "type": "BASKET_FIELD",  
  "title": "Обслуживание и ремонт лифта",  
  "value": "Обслуживание и ремонт лифта",  
  "secure": false,  
  "readOnly": false,  
  "invalid": false,  
  "required": true,  
  "basketField": {  
    "summ": "200"  
  }  
},  
{  
  "code": "#selected",  
  "type": "BASKET_FIELD",  
  "title": "Сбор и вывод ТБО",  
  "value": "Сбор и вывод ТБО",  
  "secure": false,  
  "readOnly": false,  
  "invalid": false,  
  "required": true,  
  "basketField": {  
    "summ": "300"  
  }  
},  
{  
  "code": "#selected",  
  "type": "BASKET_FIELD",  
  "title": "Содержание и ремонт",  
  "value": "Содержание и ремонт",  
  "secure": false,  
  "readOnly": false,  
  "invalid": false,  
  "required": true,  
  "basketField": {  
    "summ": "500"  
  }  
}
```

```
    }
  },
  {
    "code": "#selected",
    "type": "BASKET_FIELD",
    "title": "Содержание и ремонт мусоропровода",
    "value": "Содержание и ремонт мусоропровода",
    "secure": false,
    "readOnly": false,
    "invalid": false,
    "required": true,
    "basketField": {
      "summ": "150"
    }
  }
],
"actions": [
  "next",
  "prev",
  "exit",
  "edit"
],
"actionsExt": [
  {
    "type": "next",
    "title": "ДАЛЕЕ"
  },
  {
    "type": "prev",
    "title": "НАЗАД"
  },
  {
    "type": "exit",
    "title": "ВЫХОД"
  },
  {
    "type": "edit",
    "title": "УДАЛИТЬ"
  }
]
},
"actionResult": {
  "code": 0,
```

```
"description": "Шаг успешно выполнен, есть следующий шаг",
},
"attributes": [
  {
    "code": "idl",
    "title": "Номер счета",
    "value": "11111",
    "valueTitle": "11111",
    "originalValue": "11111",
    "flags": 0
  }
]
}
```

В поле `fields` мы получаем список доступных сервисов с информацией о них. Если код поля `#selected`, то сервис еще не добавлен в корзину, если код поля `#removed`, то сервис уже добавлен в корзину.

Для того, чтобы выбрать сервис из корзины нужно отправить частично заполненную информацию о экране.

Добавим в корзину сервис «Сбор и вывоз ТБО»:

```
{
  "session": "e20c9243-13b4-482d-8f9e-599bcab30791",
  "screenData": {
    "fields": [
      {
        "code": "#selected",
        "value": "Сбор и ВЫВОЗ ТБО"
      }
    ]
  },
  "action": "edit"
}
```

В ответ получим экран с заполнением суммы платежа по сервису (для сервиса уже есть предустановленная сумма платежа 300р):

```
{
  "code": 0,
  "screenData": {
    "title": "Укажите сумму услуги",
    "decor": "info",
    "type": "digital",
    "fields": [
      {
        "code": "summ",
        "type": "NUMERIC_FIELD",
        "title": "Сумма платежа",
        "value": "300",
        "valueTitle": "300",
        "originalValue": "300",
        "flags": 0,
        "keyboard": "DIGITAL",
        "letterCase": "Lower",
        "secure": false,
        "readOnly": false,
        "range": {
          "start": 1,
          "end": 15000
        },
        "invalid": false,
        "help": "Услуга: ${name}\nМинимальная сумма: 1 руб.\n\nМаксимальная сумма: 15000 руб.\n",
        "required": true,
        "regexp": [],
        "blockRegexp": []
      }
    ],
    "actions": [
      "next",
      "prev",
      "exit"
    ],
    "actionsExt": [
      {
        "type": "next",
        "title": "ДАЛЕЕ"
      }
    ]
  }
}
```

```
    },
    {
      "type": "prev",
      "title": "НАЗАД"
    },
    {
      "type": "exit",
      "title": "ВЫХОД"
    }
  ]
},
"actionResult": {
  "code": 0,
  "description": "Шаг успешно выполнен, есть следующий шаг"
},
"attributes": [
  {
    "code": "idl",
    "title": "Номер счета",
    "value": "11111",
    "valueTitle": "11111",
    "originalValue": "11111",
    "flags": 0
  },
  {
    "code": "name",
    "title": "name",
    "value": "Сбор и ВЫВОЗ ТВО",
    "valueTitle": "Сбор и ВЫВОЗ ТВО",
    "originalValue": "Сбор и ВЫВОЗ ТВО",
    "flags": 0
  },
  {
    "code": "summ",
    "title": "summ",
    "value": "300",
    "valueTitle": "300",
    "originalValue": "300",
    "flags": 0
  }
]
}
```

Заполним сумму платежа по сервису:

```
{
  "session": "e20c9243-13b4-482d-8f9e-599bcab30791",
  "screenData": {
    "fields": [
      {
        "code": "summ",
        "value": "350",
      }
    ],
  },
  "action": "next"
}
```

В ответ снова получим экран с корзиной, но сервис «Сбор и вывоз ТБО» уже будет добавлен в корзину с суммой платежа 350 рублей, также код поля изменился с #selected на #removed, что говорит о том, что сервис уже добавлен в корзину:

```
{
  "code": 0,
  "screenData": {
    "title": "Выбор услуг",
    "decor": "",
    "type": "communal",
    "fields": [
      {
        "code": "#selected",
        "type": "BASKET_FIELD",
        "title": "Консьерж",
        "value": "Консьерж",
        "secure": false,
        "readOnly": false,
        "invalid": false,
        "required": true,
        "basketField": {
          "summ": "450"
        }
      },
      {
        "code": "#selected",
        "type": "BASKET_FIELD",
        "title": "Обслуживание и ремонт лифта",

```

```
"value": "Обслуживание и ремонт лифта",
"secure": false,
"readOnly": false,
"invalid": false,
"required": true,
"basketField": {
  "summ": "200"
}
},
{
  "code": "#removed",
  "type": "BASKET_FIELD",
  "title": "Сбор и вывод ТВО",
  "value": "Сбор и вывод ТВО",
  "secure": false,
  "readOnly": false,
  "invalid": false,
  "required": true,
  "basketField": {
    "summ": "350"
  }
},
{
  "code": "#selected",
  "type": "BASKET_FIELD",
  "title": "Содержание и ремонт",
  "value": "Содержание и ремонт",
  "secure": false,
  "readOnly": false,
  "invalid": false,
  "required": true,
  "basketField": {
    "summ": "500"
  }
},
{
  "code": "#selected",
  "type": "BASKET_FIELD",
  "title": "Содержание и ремонт мусоропровода",
  "value": "Содержание и ремонт мусоропровода",
  "secure": false,
  "readOnly": false,
  "invalid": false,
```

```
        "required": true,
        "basketField": {
            "summ": "150"
        }
    },
    ],
    "actions": [
        "next",
        "prev",
        "exit",
        "edit"
    ],
    "actionsExt": [
        {
            "type": "next",
            "title": "ДАЛЕЕ"
        },
        {
            "type": "prev",
            "title": "НАЗАД"
        },
        {
            "type": "exit",
            "title": "ВЫХОД"
        },
        {
            "type": "edit",
            "title": "Удалить"
        }
    ]
},
"actionResult": {
    "code": 0,
    "description": "Шаг успешно выполнен, есть следующий шаг"
},
"attributes": [
    {
        "code": "id1",
        "title": "Номер счета",
        "value": "11111",
        "valueTitle": "11111",
        "originalValue": "11111",
        "flags": 0
    }
]
```

```
},
{
  "code": "title",
  "title": "title",
  "value": "Сбор и вывод ТБО",
  "valueTitle": "Сбор и вывод ТБО",
  "originalValue": "Сбор и вывод ТБО",
  "flags": 0,
  "group": 0
},
{
  "code": "id1",
  "title": "id1",
  "value": "11111",
  "valueTitle": "11111",
  "originalValue": "11111",
  "flags": 0,
  "group": 0
},
{
  "code": "name",
  "title": "name",
  "value": "Сбор и вывод ТБО",
  "valueTitle": "Сбор и вывод ТБО",
  "originalValue": "Сбор и вывод ТБО",
  "flags": 0,
  "group": 0
},
{
  "code": "record-id",
  "title": "record-id",
  "value": "49299285",
  "valueTitle": "49299285",
  "originalValue": "49299285",
  "flags": 0,
  "group": 0
},
{
  "code": "summ",
  "title": "Сумма платежа",
  "value": "350",
  "valueTitle": "350",
  "originalValue": "350",
```

```
    "flags": 0,  
    "group": 0  
  }  
]  
}
```

Для удаления сервиса из корзины необходимо отправить частично заполненный экран с полем #removed и названием удаляемого сервиса:

```
{  
  "session": "e20c9243-13b4-482d-8f9e-599bcab30791",  
  "screenData": {  
    "fields": [  
      {  
        "code": "#removed",  
        "value": "Сбор и ВЫВОЗ ТБО"  
      }  
    ]  
  },  
  "action": "edit"  
}
```

В ответ получим текущий экран с диалогом подтверждения удаления сервиса из корзины:

```
{  
  "code": 0,  
  "screenData": {  
    "title": "Выбор услуг",  
    "decor": "",  
    "type": "communal",  
    "fields": [  
      {  
        "code": "#selected",  
        "type": "BASKET_FIELD",  
        "title": "Консьерж",  
        "value": "Консьерж",  
        "secure": false,  
        "readOnly": false,  
        "invalid": false,  
        "required": true,  
        "basketField": {  
          "summ": "450"  
        }  
      }  
    ]  
  }  
}
```

```
    }
  },
  {
    "code": "#selected",
    "type": "BASKET_FIELD",
    "title": "Обслуживание и ремонт лифта",
    "value": "Обслуживание и ремонт лифта",
    "secure": false,
    "readOnly": false,
    "invalid": false,
    "required": true,
    "basketField": {
      "summ": "200"
    }
  },
  {
    "code": "#removed",
    "type": "BASKET_FIELD",
    "title": "Сбор и вывод ТВО",
    "value": "Сбор и вывод ТВО",
    "secure": false,
    "readOnly": false,
    "invalid": false,
    "required": true,
    "basketField": {
      "summ": "350"
    }
  },
  {
    "code": "#selected",
    "type": "BASKET_FIELD",
    "title": "Содержание и ремонт",
    "value": "Содержание и ремонт",
    "secure": false,
    "readOnly": false,
    "invalid": false,
    "required": true,
    "basketField": {
      "summ": "500"
    }
  },
  {
    "code": "#selected",
```

```
"type": "BASKET_FIELD",
"title": "Содержание и ремонт мусоропровода",
"value": "Содержание и ремонт мусоропровода",
"secure": false,
"readOnly": false,
"invalid": false,
"required": true,
"basketField": {
  "summ": "150"
}
},
"dialogs": [
  {
    "type": "INFO",
    "title": "Запрос на удаление",
    "titleId": "",
    "message": "Вы действительно хотите удалить Сбор и вывоз ТБО из корзины?",
    "timeout": 10,
    "defaultAction": "cancel",
    "params": [
      "name"
    ],
    "actions": [
      {
        "type": "dialog__okay",
        "title": "ДА"
      },
      {
        "type": "dialog__cancel",
        "title": "НЕТ"
      }
    ]
  }
],
"actions": [
  "next",
  "prev",
  "exit",
  "edit"
],
"actionsExt": [
```

```
{
  "type": "next",
  "title": "ДАЛЕЕ"
},
{
  "type": "prev",
  "title": "НАЗАД"
},
{
  "type": "exit",
  "title": "ВЫХОД"
},
{
  "type": "edit",
  "title": "Удалить"
}
]
},
"actionResult": {
  "code": 0,
  "description": "Шаг успешно выполнен, есть следующий шаг"
},
"attributes": [
  {
    "code": "id1",
    "title": "Номер счета",
    "value": "11111",
    "valueTitle": "11111",
    "originalValue": "11111",
    "flags": 0
  },
  {
    "code": "name",
    "title": "name",
    "value": "Сбор и вывод ТВО",
    "valueTitle": "Сбор и вывод ТВО",
    "originalValue": "Сбор и вывод ТВО",
    "flags": 0
  },
  {
    "code": "title",
    "title": "title",
    "value": "Сбор и вывод ТВО",
```

```
"valueTitle": "Сбор и ВЫВОЗ ТВО",
"originalValue": "Сбор и ВЫВОЗ ТВО",
"flags": 0,
"group": 0
},
{
  "code": "id1",
  "title": "id1",
  "value": "11111",
  "valueTitle": "11111",
  "originalValue": "11111",
  "flags": 0,
  "group": 0
},
{
  "code": "name",
  "title": "name",
  "value": "Сбор и ВЫВОЗ ТВО",
  "valueTitle": "Сбор и ВЫВОЗ ТВО",
  "originalValue": "Сбор и ВЫВОЗ ТВО",
  "flags": 0,
  "group": 0
},
{
  "code": "record-id",
  "title": "record-id",
  "value": "49299285",
  "valueTitle": "49299285",
  "originalValue": "49299285",
  "flags": 0,
  "group": 0
},
{
  "code": "summ",
  "title": "Сумма платежа",
  "value": "350",
  "valueTitle": "350",
  "originalValue": "350",
  "flags": 0,
  "group": 0
}
]
}
```

Для подтверждения удаления необходимо отправить пустую ScreenData и указать действие из диалога:

```
{
  "session": "e20c9243-13b4-482d-8f9e-599bcab30791",
  "screenData": null,
  "action": "dialog__okay"
}
```

В ответ получим экран корзины без выбранных сервисов, т.к. был удален единственный сервис:

```
{
  "code": 0,
  "screenData": {
    "title": "Выбор услуг",
    "decor": "",
    "type": "communal",
    "fields": [
      {
        "code": "#selected",
        "type": "BASKET_FIELD",
        "title": "Консьерж",
        "value": "Консьерж",
        "secure": false,
        "readOnly": false,
        "invalid": false,
        "required": true,
        "basketField": {
          "summ": "450"
        }
      },
      {
        "code": "#selected",
        "type": "BASKET_FIELD",
        "title": "Обслуживание и ремонт лифта",
        "value": "Обслуживание и ремонт лифта",
        "secure": false,
        "readOnly": false,
        "invalid": false,
        "required": true,
        "basketField": {
          "summ": "200"
        }
      }
    ]
  }
}
```

```
    }
  },
  {
    "code": "#selected",
    "type": "BASKET_FIELD",
    "title": "Сбор и вывод ТБО",
    "value": "Сбор и вывод ТБО",
    "secure": false,
    "readOnly": false,
    "invalid": false,
    "required": true,
    "basketField": {
      "summ": "350"
    }
  },
  {
    "code": "#selected",
    "type": "BASKET_FIELD",
    "title": "Содержание и ремонт",
    "value": "Содержание и ремонт",
    "secure": false,
    "readOnly": false,
    "invalid": false,
    "required": true,
    "basketField": {
      "summ": "500"
    }
  },
  {
    "code": "#selected",
    "type": "BASKET_FIELD",
    "title": "Содержание и ремонт мусоропровода",
    "value": "Содержание и ремонт мусоропровода",
    "secure": false,
    "readOnly": false,
    "invalid": false,
    "required": true,
    "basketField": {
      "summ": "150"
    }
  }
],
"actions": [
```

```
"next",
"prev",
"exit",
"edit"
],
"actionsExt": [
  {
    "type": "next",
    "title": "ДАЛЕЕ"
  },
  {
    "type": "prev",
    "title": "НАЗАД"
  },
  {
    "type": "exit",
    "title": "ВЫХОД"
  },
  {
    "type": "edit",
    "title": "Удалить"
  }
]
},
"actionResult": {
  "code": 0,
  "description": "Шаг успешно выполнен, есть следующий шаг"
},
"attributes": [
  {
    "code": "selected",
    "title": "selected",
    "value": "0",
    "valueTitle": "0",
    "originalValue": "0",
    "flags": 0
  },
  {
    "code": "id1",
    "title": "Номер счета",
    "value": "11111",
    "valueTitle": "11111",
    "originalValue": "11111",
```

```
"flags": 0
},
{
  "code": "name",
  "title": "name",
  "value": "Сбор и вывоз ТБО",
  "valueTitle": "Сбор и вывоз ТБО",
  "originalValue": "Сбор и вывоз ТБО",
  "flags": 0
}
]
}
```

2.15.8 ОТПРАВКА ЧАСТИЧНО ЗАПОЛНЕННОЙ SCREENDATA В ОТВЕТЕ

С версии обработчика 1.0.3 в запросе с информацией о пользовательском вводе возможно отправить часть атрибутов поля — указать только код и значение элемента ввода (значения атрибутов **code**, **value**). Остальные атрибуты не влияют на результат выполнения запроса.

Допустим, что существует сценарий с одним полем:

```
<scenario begin="01screen">
  <screen type="digital" title="title01" id="01screen">
    <fields>
      <text-field id="id2" title="titleId2" max-len="10"
        keyboard="Digital">
        <validator type="regex">
          <rules>
            <rule regex="^\d{1,20}$"/>
          </rules>
        </validator>
      </text-field>
    </fields>
    <actions>
      <action type="Next" title="Далее">
        <goto target="pay"/>
      </action>
    </actions>
  </screen>
</scenario>
```

```
<action type="Exit" title="Выход">
  <goto target="exit"/>
</action>
</actions>
</screen>
</scenario>
```

При запросе первого экрана получаем информацию об экране:

```
{
  "code": 0,
  "screenData": {
    "title": "Введите номер счета",
    "decor": "",
    "type": "numeric",
    "fields": [
      {
        "code": "id1",
        "type": "TEXT_FIELD",
        "title": "Номер счета",
        "flags": 0,
        "keyboard": "DIGITAL",
        "letterCase": "Lower",
        "secure": false,
        "maxLen": 5,
        "readOnly": false,
        "invalid": false,
        "required": true,
        "regexp": [
          "\\d{5}$"
        ],
        "blockRegexp": []
      }
    ],
    "actions": [
      "next",
      "prev",
      "exit"
    ],
    "actionsExt": [
      {
        "type": "next",
        "title": "ДАЛЕЕ"
      }
    ]
  }
}
```

```
{
  {
    "type": "prev",
    "title": "НАЗАД"
  },
  {
    "type": "exit",
    "title": "ВЫХОД"
  }
]
},
"actionResult": {
  "code": 0,
  "description": "Шаг успешно выполнен, есть следующий шаг"
},
"attributes": []
}
```

Для того, чтобы отправить информацию о пользовательском вводе в частично заполненном виде, можно воспользоваться конструкцией:

```
{
  "session": "3115889b-5b57-4a9b-b346-d5b497d075fb",
  "screenData": {
    "fields": [
      {
        "code": "id1",
        "value": "11111"
      }
    ]
  },
  "action": "next"
}
```

Соответствующий полный запрос может содержать дополнительные атрибуты поля и выглядеть следующим образом:

```
{
  "session": "3115889b-5b57-4a9b-b346-d5b497d075fb",
  "screenData": {
    "fields": [
      {
        "code": "id1",
        "value": "11111",
        "type": "TEXT_FIELD"
      }
    ]
  }
}
```

```
        "title": "Номер счета"  
    }  
  ]  
},  
"action": "next"  
}
```

Получим ответ:

```
{  
  "code": 0,  
  "actionResult": {  
    "code": 1,  
    "description": "Шаг успешно выполнен, является завершающим"  
  },  
  "attributes": [  
    {  
      "code": "id2",  
      "title": "titleId2",  
      "value": "11111",  
      "valueTitle": "11111",  
      "originalValue": "11111",  
      "flags": 0  
    }  
  ]  
}
```

2.15.9 ПОЛУЧЕНИЕ ПЕРВОГО ЭКРАНА В ОТВЕТ НА ЗАПРОС START

Сценарий:

```
<scenario begin="screen0">
  <screen type="letter" title="T-screen0" id="screen0">
    <fields>
      <text-field id="text0" title="T-text0" max-len="3"
keyboard="any:[en]:lower:true">
        <validator type="regex">
          <rules>
            <rule regex="^[a-z]{3}$"/>
          </rules>
        </validator>
      </text-field>
    </fields>

    <actions>
      <action type="Next" title="Далее">
        <goto target="screen1"/>
      </action>

      <action type="Exit" title="Выход">
        <goto target="exit"/>
      </action>
    </actions>
  </screen>

  <screen type="letter" title="T-screen1" id="screen1">
    <fields>
      <text-field id="text1" title="T-text1" max-len="3"
keyboard="any:[en]:lower:true">
        <validator type="regex">
          <rules>
            <rule regex="^[a-z]{3}$"/>
          </rules>
        </validator>
      </text-field>
    </fields>
```

```
<actions>
  <action type="Next" title="Далее">
    <goto target="pay"/>
  </action>

  <action type="Prev" title="Назад">
    <goto target="screen0"/>
  </action>

  <action type="Exit" title="Выход">
    <goto target="exit"/>
  </action>
</actions>
</screen>
</scenario>
```

Start — запрос:

```
{
  "idPoint": "636",
  "idService": "4000",
  "firstScreen": true
}
```

Start — ответ:

```
{
  "code": 0,
  "session": "5b8f649c-15cc-4929-ad3e-18e0252910b9",
  "screenData": {
    "title": "T-screen0",
    "decor": "",
    "type": "letter",
    "fields": [
      {
        "code": "text0",
        "type": "TEXT_FIELD",
        "title": "T-text0",
        "flags": 0,
        "keyboard": "ANY",
        "letterCase": "Lower",
        "allowChangeCase": true,
        "layoutNames": [
```

```
        "en"
      ],
      "secure": false,
      "maxLen": 3,
      "readOnly": false,
      "invalid": false,
      "required": true,
      "regexp": [
        "[a-z]{3}$"
      ]
    }
  ],
  "actions": [
    "next",
    "exit"
  ],
  "actionsExt": [
    {
      "type": "next",
      "title": "Далее",
      "pay": false
    },
    {
      "type": "exit",
      "title": "Выход",
      "pay": false
    }
  ]
},
"lastScreen": false,
"actionResult": {
  "code": 0,
  "description": "Шаг успешно выполнен, есть следующий шаг"
},
"attributes": []
}
```

2.15.10 ПОВТОРНОЕ ПОЛУЧЕНИЕ ПЕРВОГО ЭКРАНА

Start — запрос:

```
{
  "idPoint": "636",
  "idService": "4000",
}
```

Start — ответ:

```
{
  "code": "SUCCESS",
  "session": "698efa95-e2da-4f15-9d13-6ee941d4fa40"
}
```

Первый экран — запрос:

```
{
  "session": "698efa95-e2da-4f15-9d13-6ee941d4fa40",
  "action": "next"
}
```

Первый экран — ответ:

```
{
  "code": 0,
  "session": "698efa95-e2da-4f15-9d13-6ee941d4fa40",
  "screenData": {
    "title": "T-screen0",
    "decor": "",
    "type": "letter",
    "fields": [
      {
        "code": "text0",
        "type": "TEXT_FIELD",
        "title": "T-text0",
        "flags": 0,
        "keyboard": "ANY",
        "letterCase": "Lower",
      }
    ]
  }
}
```

```
"allowChangeCase": true,
"layoutNames": [
  "en"
],
"secure": false,
"maxLen": 3,
"readOnly": false,
"invalid": false,
"required": true,
"regexp": [
  "[a-z]{3}$"
]
}
],
"actions": [
  "next",
  "exit"
],
"actionsExt": [
  {
    "type": "next",
    "title": "Далее",
    "pay": false
  },
  {
    "type": "exit",
    "title": "Выход",
    "pay": false
  }
]
},
"lastScreen": false,
"actionResult": {
  "code": 0,
  "description": "Шаг успешно выполнен, есть следующий шаг"
},
"attributes": []
}
```

Повторный запрос первого экрана — запрос:

```
{
  "session": "698efa95-e2da-4f15-9d13-6ee941d4fa40",
  "screenData": null,
}
```

```
"action": "__last"  
}
```

Повторный запрос первого экрана — ответ:

```
{  
  "code": 0,  
  "session": "698efa95-e2da-4f15-9d13-6ee941d4fa40",  
  "screenData": {  
    "title": "T-screen0",  
    "decor": "",  
    "type": "letter",  
    "fields": [  
      {  
        "code": "text0",  
        "type": "TEXT_FIELD",  
        "title": "T-text0",  
        "flags": 0,  
        "keyboard": "ANY",  
        "letterCase": "Lower",  
        "allowChangeCase": true,  
        "layoutNames": [  
          "en"  
        ],  
        "secure": false,  
        "maxLen": 3,  
        "readOnly": false,  
        "invalid": false,  
        "required": true,  
        "regexp": [  
          "[a-z]{3}$"  
        ]  
      }  
    ],  
    "actions": [  
      "next",  
      "exit"  
    ],  
    "actionsExt": [  
      {  
        "type": "next",  
        "title": "Далее",  
        "pay": false  
      }  
    ],  
  }  
}
```

```
{
  {
    "type": "exit",
    "title": "Выход",
    "pay": false
  }
]
},
"lastScreen": false,
"actionResult": {
  "code": 0,
  "description": "Шаг успешно выполнен, есть следующий шаг"
},
"attributes": []
}
```

2.16 ПРИМЕР РАБОТЫ С МАСКАМИ

Пусть на первом экране пользователь выбрал тип оплачиваемой услуги. Тогда на следующем экране ему необходимо ввести номер телефона (к номеру прибавляются префикс, постфикс и получается 10-значный номер счета).

В форме, приведенной ниже, для форматирования вводимого номера используется следующее правило разбивки:

```
<split default="*-****-****-*"/>
```

В результате работы маски вводимые символы будут разбиты на четыре части с дефисами между ними. Подробное описание элемента приведено в документе [«Формы оплаты для универсального модуля ввода данных. Программное обеспечение «Процессинговый центр Pay-logic». Руководство пользователя»](#) (раздел 6.1.3).

Пример формы оплаты:

```
<?xml version="1.0" encoding="UTF-8"?>
<forms>
  <form service-id="1">
<!--Создание поля ввода, со следующими параметрами: активна цифровая
клавиатура, максимальное количество вводимых символов 8. Название поля
ввода: "Номер телефона"-->
    <text-field name="id1" keyboard="Digital" max-len="8"
      message="Номер телефона"
      title="Номер телефона"
      prefix="0" postfix="1">
<!--Регулярное выражение для валидации вводимого номера-->
    <verify>
      <rule regex="^\d{8}$"/>
    </verify>
<!--Разбивка вводимого значения на 4 группы для последующей модификации-->
    <replaces>
      <rule regex="^\d{10}$" target="^(\\d{1}) (\\d{4}) (\\d{4}) (\\d{1})$"
        value="$1.$2.$3.$4"/>
    </replaces>
  </form>
</forms>
```

```
</replaces>
<!--Маска для форматирования вводимого номера, применяется по
умолчанию-->
    <split default="*-****-****-*"/>
<!--Подсказка, отображается на терминале-->
    <help>Введите номер счета</help>
    </text-field>
</form>
</forms>
```

На экране, приведенном ниже, задана маска, которая визуально разделяет вводимое значение на части, и атрибут **bystep=false**, в результате действия которого структура маски будет видна сразу, а не по мере ввода символов:

```
"viewFormatter": {
    "mask": "*-****-****-*",
    "byStep": false
},
```

Пример экрана, возвращенного сервером:

```
{
  "code": "SUCCESS",
  "screenData": {
    "title": "Номер телефона",
    "decor": "button",
    "type": "digital",
    "fields": [
      {
        "code": "id1",
        "type": "TEXT_FIELD",
        "title": "Номер телефона",
        "flags": 0,
        "keyboard": "DIGITAL",
        "letterCase": "Lower",
        "allowChangeCase": true,
        "secure": false,
        "prefix": "0",
        "postfix": "1",
        "maxLen": 8,
        "readOnly": false,
        "invalid": false,
        "help": "Введите номер счета",
```

```
"required": true,
"dataFormatter": {
  "mask": "*-****-****-*",
  "byStep": false
},
"viewFormatter": {
  "mask": "*-****-****-*",
  "byStep": false
},
"regexFieldModifierRules": [
  {
    "regex": "^\\d{10}$",
    "target": "^((\\d{1}) (\\d{4}) (\\d{4}) (\\d{1}))$",
    "value": "$1.$2.$3.$4"
  }
],
"regexp": [
  "^\\d{8}$"
]
},
"actions": [
  "next",
  "exit",
  "prev"
],
"actionsExt": [
  {
    "type": "next",
    "title": "Далее",
    "pay": false
  },
  {
    "type": "exit",
    "title": "Выход",
    "pay": false
  },
  {
    "type": "prev",
    "title": "Назад",
    "pay": false
  }
]
```

```
},
"action": "next",
"lastScreen": true,
"actionResult": {
  "code": "SUCCESS_EXIST_NEXT",
  "description": "Шаг успешно выполнен, есть следующий шаг"
},
"attributes": []
}
```

Пример отправки запроса, содержащего заполненные поля формы и действие «next»:

```
{
  "session": "3e666bea-24ce-4e7e-95d3-f4d282405616",
  "screenData": {
    "fields": [
      {
        "code": "id1",
        "value": "12345678",
      }
    ],
  },
  "action": "next"
}
```

В примере ответа, приведенном ниже, отображаются атрибуты:

1. **value** — значение элемента, возвращенное в результате запроса (модифицированное значение);
2. **valueTitle** — заголовок значения элемента (был изменен в результате форматирования);
3. **originalValue** — значение элемента, введенное пользователем.

Пример ответа на выполнение действия «next»:

```
{
  "code": "SUCCESS",
  "action": "pay",
  "lastScreen": false,
}
```

```
"actionResult": {
  "code": "SUCCESS_FINAL",
  "description": "Шаг успешно выполнен, является завершающим"
},
"attributes": [
  {
    "code": "id1",
    "title": "Номер телефона",
    "value": "0.1234.5678.1",
    "valueTitle": "0-1234-5678-1",
    "originalValue": "12345678",
    "flags": 0
  }
]
}
```

3 НАСТРОЙКА МИКРОСЕРВИСА ДИНАМИЧЕСКИХ ФОРМ

3.1 ПОРЯДОК НАСТРОЙКИ МИКРОСЕРВИСА ДИНАМИЧЕСКИХ ФОРМ

Для настройки микросервиса динамических форм выполните следующие действия:

1. Положите рядом с файлом `dynamic-input-server.jar` конфигурационный файл `application.yml` (рисунок 3.1.1). Для запуска приложения используйте JDK 11 версии и выше, jar-файл стал исполняемым, его можно запускать без сервера приложений.

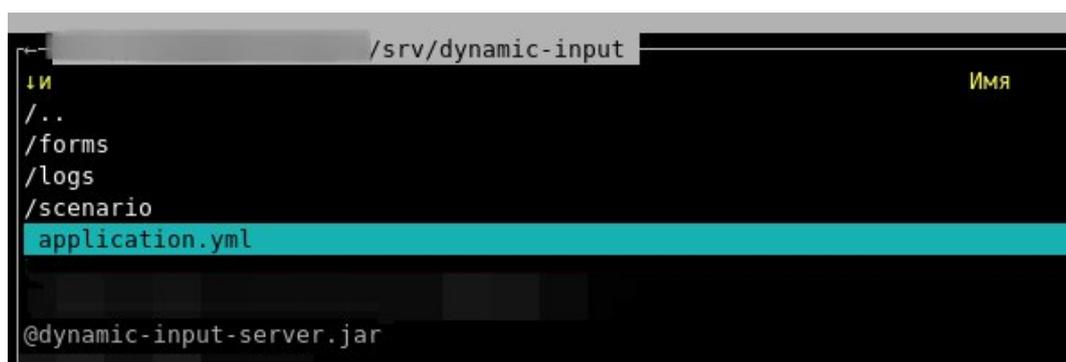


Рисунок 3.1.1 — Конфигурационный файл `application.yml`

2. Настройте параметры конфигурационного файла микросервиса (см. Раздел [3.2](#)). С версии приложения 1.3.0 требуется подключение к БД.
3. Настройте внешний коннектор для работы с микросервисом. В конфигурационном файле сервера приложений `/srv/glassfish3/config.xml` пропишите параметры:

```
<dynamic-request-url>URL динамического обработчика</dynamic-request-url>  
<dynamic-request-protocols>TLSv1.2</dynamic-request-protocols>  
<dynamic-request-timeout>120</dynamic-request-timeout>
```

3.2 ОПИСАНИЕ КОНФИГУРАЦИОННОГО ФАЙЛА МИКРОСЕРВИСА ДИНАМИЧЕСКИХ ФОРМ

Для работы приложения требуется конфигурационный файл `application.yml`. С версии 1.3.0 приложение использует настройки шлюзов из БД.

Структура конфигурационного файла приложения приведена в таблице 3.2.1.

Таблица 3.2.1 — Структура конфигурационного файла приложения

Параметр	Описание
<code>im.form7.path</code>	Путь к формам на сервере (без конечного слэша)
<code>im.form7.period</code>	Период жизни формы в кэше (минут)
<code>im.scenario7.path</code>	Путь к сценариям на сервере (без конечного слэша)
<code>im.scenario7.period</code>	Период жизни формы в кэше (минут)
<code>session.period</code>	Время жизни сессии
<code>server.port</code>	Порт приложения
<code>spring.datasource.url</code>	JDBC URL для подключения к БД. Например: <code>jdbc:postgresql://localhost:5432/procdb</code>
<code>spring.datasource.username</code>	Логин пользователя БД
<code>spring.datasource.password</code>	Пароль пользователя БД
<code>server.servlet.context-path</code>	Контекст приложения. По умолчанию не задан

Параметры задаются в следующем формате:

```
ru:
  softlogic:
    dynamicinput:
      im:
        form7:
          path: './forms'
          period: 0
        scenario7:
          path: './scenarios'
          period: 0
      session:
        period: 5
      server:
        port: 4443
    spring.datasource.url:
    spring.datasource.username:
    spring.datasource.password:
    server.servlet.context-path:
```

4 ПРИМЕР РАБОТЫ ДИНАМИЧЕСКОГО МОДУЛЯ ВВОДА ДАННЫХ

Рассмотрим пример работы динамического модуля ввода данных. Последовательность действий приведена на рисунке 4.1.

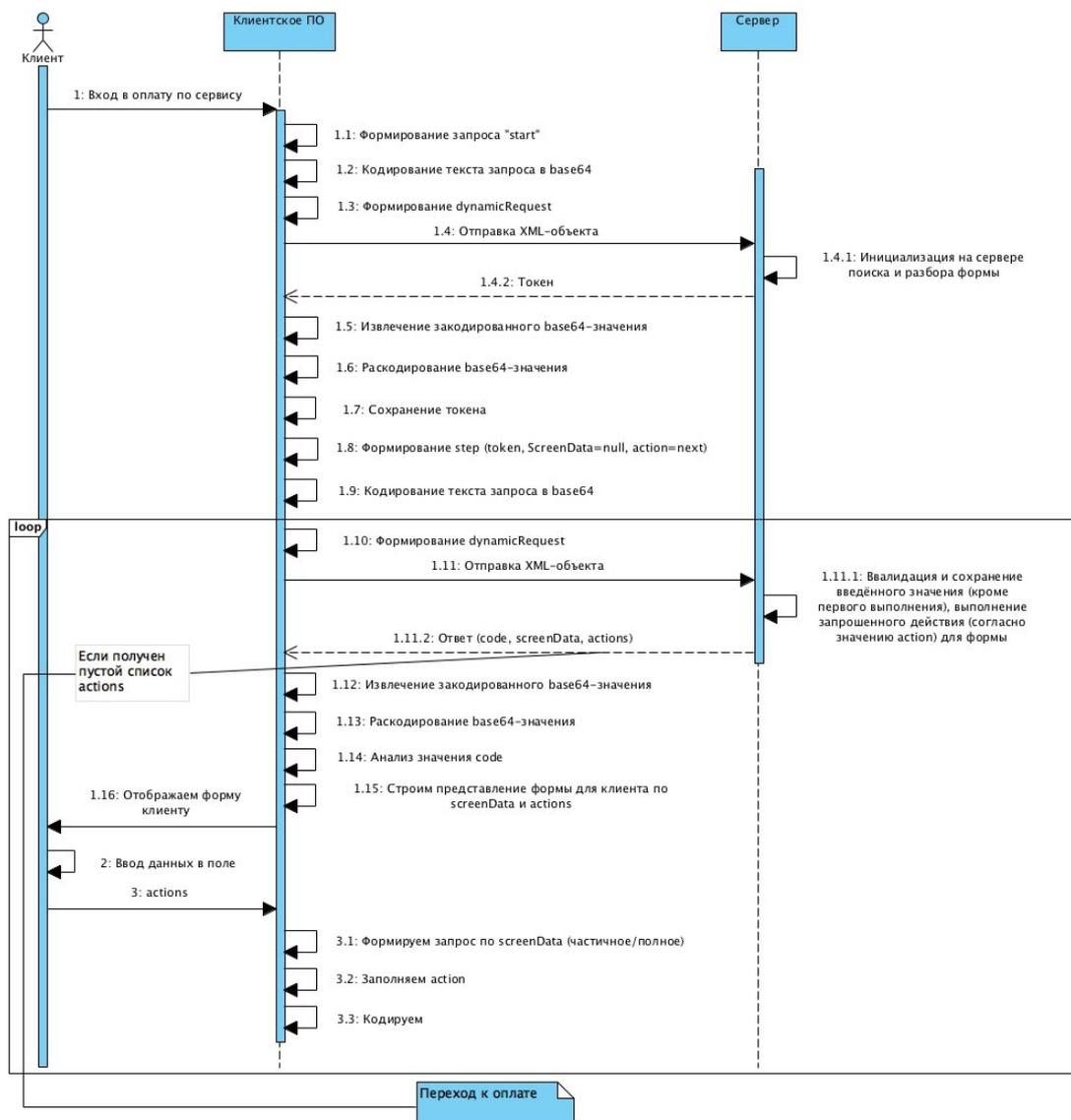


Рисунок 4.1 — Последовательность действий для примера работы динамического модуля ввода данных

Для этого используем сервис на форме с несколькими полями. Форма приведена в этом разделе ниже. Для работы понадобится любая шлюзовая точка, работающая по протоколу Paylogic.

1. Необходимо сформировать запрос start (раздел 2.4):

```
{
  "idPoint": 1145,
  "idService": 1006,
  "verify": false
}
```

В примере номер точки условный, необходимо использовать фактический номер точки и номер сервиса.

2. Необходимо закодировать текст запроса в base64. Для примера запроса start получится:

```
ew0KICJpZFBvaW50IjogMTE0NSwNCiAiaWRTZXJ2aWNlIjogMTAwNiwNCiAidmVyaWZ5IjogZmFsc2UNCn0=
```

3. Необходимо сформировать dynamicRequest, добавив в него секцию dynamicRequest с закодированным JSONом:

```
<?xml version="1.0" encoding="UTF-8"?>
<request point="1145">
<dynamicRequest>ew0KICJpZFBvaW50IjogMTE0NSwNCiAiaWRTZXJ2aWNlIjogMTAwNiwNCiAidmVyaWZ5IjogZmFsc2UNCn0=</dynamicRequest>
</request>
```

4. Необходимо отправить XML-объект на шлюз. Будет возвращен ответ:

```
<response><result>
code="0">ewogICJjb2RlIjogMkwKICAic2Vzc2lvbiI6ICI0NTU0MTIyMTk1NjcxCjEwNzQ0IjogZmFsc2UNCn0=</result></response>
```

5. Из секции Result полученного ответа необходимо извлечь закодированное base64 значение и раскодировать его. Пример:

```
{
  "code": 0,
  "session": "4554122195671610734"
}
```

6. Данные из ответа необходимо использовать согласно спецификации: сформировать запрос следующего шага. Так как необходимо выполнить первый шаг, то ScreenData не заполняется:

```
{
  "session": "4554122195671610734",
  "screenData": null,
  "action": "next"
}
```

7. Для отправки запроса необходимо закодировать его в base64 (в точности, как в пункте 2) и отправить в теге dynamicRequest. Ответ необходимо раскодировать и обработать содержимое полей ответа в соответствии с документацией (в точности как в пункте 5).

Для предлагаемой формы оплаты по свободным реквизитам в ответ на первый шаг будет передано первое поле формы: **name="id1" keyboard="Digital" maxlen="20" message="Введите счёта получателя" title="Номер счета"** в виде ScreenData.

В примере ответ на шаг 7 будет таким:

```
ew0KICAIy29kZSI6IDAsDQogICJzY3JlZW5EYXRhIjogew0KICAgICJ0aXRzZSI6ICLQktCy0LXQQtNC40YlQtSDRgdGH0ZHRgtCwINC/
0L7Qu9GD0YfQsNGC0LXQu9GPIiwNCiAgICAidG10bGVJZCI6IG51bGwsDQogICAgImh1bHAI0iBudWxsLA0KICAgICJkZWNvciI6ICJidXR0b24iLA0KICAgICJmaWVsZHMiOiBbdDQogICAgICB7DQogICAgICAgICJjb2RlIjogIm1kMSIsDQogICAgICAgICJ0eXB1IjogIlRFRFRrklFTEQiLA0KICAgICAgICAidG10bGUiOiAi0J3QvtC80LXRgCDRgdGH0LXRgtCwIiwNCiAgICAgICAgInRpdGx1SWQiOiBudWxsLA0KICAgICAgICAidmFsdWUiOiBudWxsLA0KICAgICAgICAib3JpZ2luYWxWYX1ZSI6IG51bGwsDQogICAgICAgICJmbGFncyI6IDAsDQogICAgICAgICJrZXlib2FyZCI6ICJESUdJVEFMIiwNCiAgICAgICAgImxldHRlckNhc2UiOiAiTG93ZXIiLA0KICAgICAgICAic2VjdXJlIjogZmFsc2UsDQogICAgICAgICJkZWZhdWx0VmFsdWUiOiBudWxsLA0KICAgICAgICAIzXhhbXBsZSI6IG51bGwsDQogICAgICAgICJwcmVmaXgiOiBudWxsLA0KICAgICAgICAIcG9zdGZpeCI6IG51bGwsDQogICAgICAgICJtYXhMZW4iOiAyMwNCiAgICAgICAgInJlYWRRPmX5IjogZmFsc2UsDQogICAgICAgICJmb3JtYXQiOiBudWxsLA0KICAgICAgICAIz9ybWF0VG10bGUiOiBudWxsLA0KICAgICAgICAIzGF0ZVJhbmdlIjogbnVsbCwNCiAgICAgICAgInJhbmdlIjogbnVsbCwNCiAgICAgICAgIm1udmFsaWQiOiBmYWxzZSwNCiAgICAgICAgImh1bHAI0iAi0J/
QviDQvtC60L7QvdGH0LDQvdC40Lgg0LLQstC+0LTQsCDQvdCw0LbQvNC40YlQtSDQutC70LDQstC40YjRgyBcItCU0LDQu9C10LVcIiIsDQogICAgICAgICJ0ZWxwSWQiOiBudWxsLA0KICAgICAgICAIcmVnZXhwIjogWw0KICAgICAgICAgICJ0eKCGzfdQpWzAtOV17NH04MTBbMC05XXsxMn0pJCINCiAgICAgICAgXSwnCiAgICAgICAgImJsbnVnVnZlIjogW10sDQogICAgICAgICJk
```

```
2YWx1ZXMiOiBudWxsLA0KICAgICAgICAidGFibGVJdGVtcyI6IG51bGwNCiAgICAgIH0NCiAg  
ICBdLA0KICAgICJhY3Rpb25zIjogWw0KICAgICAgIm5leHQiLA0KICAgICAgImV4aXQiLA0KI  
CAgICAgInByZXYiDQogICAgXQ0KICB9LA0KICAic3VtcHVyY2hhc2UiOiBudWxsLA0KICAIYW  
N0aW9uUmVzdWx0Ijogew0KICAgICJjb2RlIjogMCwNCiAgICAgICAgICAgICAgICAgICAgIC  
QsNCzINGD0YHQv9C10YjQvdC+INCy0YvQv9C+0LvQvdC10L0sINC10YHRgtGMINGB0LvQtdC0  
0YPRjtGJ0LjQuSDRiNCw0LMI0KICAgICJtZXNzYWdlcyI6IG51bGwNCiAgfSwNCiAgImF0d  
HJpYnV0ZXMiOiBbXQ0KfQ==
```

8. Ответ необходимо раскодировать из base64. Необходимо отобразить поле клиенту, дать ввести значение и выполнить на стороне клиента проверки по длине и регулярным выражениям, вызвать действие Action = next:

```
{  
  "code": 0,  
  "screenData": {  
    "title": "Введите счёта получателя",  
    "titleId": null,  
    "help": null,  
    "decor": "button",  
    "fields": [  
      {  
        "code": "idl",  
        "type": "TEXT_FIELD",  
        "title": "Номер счёта",  
        "titleId": null,  
        "value": null,  
        "originalValue": null,  
        "flags": 0,  
        "keyboard": "DIGITAL",  
        "letterCase": "Lower",  
        "secure": false,  
        "defaultValue": null,  
        "example": null,  
        "prefix": null,  
        "postfix": null,  
        "maxLen": 20,  
        "readOnly": false,  
        "format": null,  
        "formatTitle": null,  
        "dateRange": null,  
        "range": null,  
        "invalid": false,  
        "help": "По окончании ввода нажмите клавишу \"Далее\"",  
        "helpId": null,  
      }  
    ]  
  }  
}
```

```
    "regexp": [
      "^( (3|4) [0-9]{4} 810 [0-9]{12} ) $"
    ],
    "blockRegexp": [],
    "values": null,
    "tableItems": null
  }
],
"actions": [
  "next",
  "exit",
  "prev"
]
},
"sumpurchase": null,
"actionResult": {
  "code": 0,
  "description": "Шаг успешно выполнен, есть следующий шаг",
  "messages": null
},
"attributes": []
}
```

9. В ответе содержится только первое поле — расчетный счет:

```
"code": "id1",
"value": null,
```

10. Клиентское ПО отобразило и получило от клиента введенное значение. Например, клиент ввел **40702810500000000001**. Необходимо сформировать запрос на следующий шаг: для этого объект ScreenData (всё что в "screenData": {}) необходимо дополнить введенным значением:

```
"value": "40702810500000000001"
{
  "session": "4554122195671610734",
  "screenData": "screenData": {
    "title": "Введите счёта получателя",
    "titleId": null,
    "help": null,
    "decor": "button",
    "fields": [
      {
        "code": "id1",
        "type": "TEXT_FIELD",
```



```
<help>
<![CDATA[
    По окончании ввода нажмите клавишу <b>"Далее"</b>
]]>
</help>
</text-field>
<text-field name="bik" keyboard="Digital" max-len="9"
    message="Введите БИК получателя" title="БИК">
    <verify>
        <rule regex="^\d{9}$"/>
    </verify>
    <help>
<![CDATA[
    По окончании ввода нажмите клавишу <b>"Далее"</b>
]]>
</help>
</text-field>
<text-field name="inn" keyboard="Digital" max-len="12"
    message="Введите ИНН получателя" title="ИНН">
    <verify>
        <rule regex="^(\\d{10}|\\d{12})$"/>
    </verify>
    <help>
<![CDATA[
    По окончании ввода нажмите клавишу <b>"Далее"</b>
]]>
</help>
</text-field>
<text-field name="reason" keyboard="any:[ru,en,symb]:lower:true"
    max-len="255" message="Введите назначение платежа"
    title="Назначение платежа">
    <verify>
        <rule regex="^.{10,255}$"/>
    </verify>
    <help>
<![CDATA[
    По окончании ввода нажмите клавишу <b>"Далее"</b>
]]>
</help>
</text-field>
<text-field name="fio" keyboard="any:[ru,symb]:lower:true"
    max-len="255" message="Введите ФИО полностью"
    title="ФИО">
```

```
<verify>
  <rule regex="^.{2,255}$"/>
</verify>
  <help>
<![CDATA[
  По окончании ввода нажмите клавишу <b>"Далее"</b>
]]>
</help>
</text-field>
<text-field name="address" keyboard="any:[ru,symb]:lower:true"
  max-len="255" message="Введите адрес места регистрации"
  title="Адрес места регистрации">
  <verify>
    <rule regex="^.*$"/>
  </verify>
  <help>
<![CDATA[
  По окончании ввода нажмите клавишу <b>"Далее"</b>
]]>
</help>
</text-field>
<text-field name="phone" keyboard="Digital" max-len="10"
  message="Введите контактный телефон плательщика"
  title="Телефон" type="mobile">
  <verify>
    <rule regex="^\d{10}$"/>
  </verify>
  <help>
<![CDATA[
  По окончании ввода нажмите клавишу <b>"Далее"</b>
]]>
</help>
  <split default="8-(***)-***-****"/>
</text-field>
<text-selector name="vidnds" message="Укажите НДС" title="НДС"
  type="static" decor="button">
  <item text="Без НДС" value="НДС не облагается">
    <set-field name="nds" title="НДС" value="0.00"/>
  </item>
  <item text="Ввести сумму НДС" value="в том числе НДС">
    <text-field name="nds" keyboard="any:[ru,symb]:lower:true"
      max-len="9" message="Введите сумму НДС"
      title="НДС">
```

```
<verify>
  <rule regex="^\d{1,6}\.\d{1,2}$"/>
</verify>
  <help>
<![CDATA[
  По окончании ввода нажмите клавишу <b>"Далее"</b>
]]>
</help>
  </text-field>
</item>
</text-selector>
<numeric-field name="sum" show-on-confirm="true"
  message="Введите сумму платежа, включая НДС"
  title="Сумма платежа, включая НДС"
  allow-decimal="true" min="10" max="15000"
  change-sumpurchase="true"/>
</form>
</forms>
```

5 ПРИМЕР РАБОТЫ ДИНАМИЧЕСКОГО МОДУЛЯ ВВОДА ДАННЫХ ПО ПРОТОКОЛУ КИБЕРПЛАТ

5.1 АДРЕСА ДЛЯ HTTP-ЗАПРОСОВ

Адреса для HTTP-запросов в протоколе Киберплат зависят от типа выполняемого запроса и сервиса, по которому выполняется операция.

В реализации протокола Киберплат во внешнем коннекторе Paylogic для запросов используются следующие адреса:

- 1) [/external/cgi-bin/ge_pay_check.cgi/12345](#) — адрес для запроса разрешения на платёж, где /external — context root, в котором запущено приложение коннектора, 12345 — ID сервиса в процессинге Paylogic;
- 2) [/external/cgi-bin/ge_pay.cgi/12345](#) — адрес для запроса проведения платежа;
- 3) [/external/cgi-bin/ge_pay_status.cgi](#) — адрес для запроса статуса платежа.

Для работы с динамическими формами добавлен адрес [/external/cgi-bin/ge_dynamic_forms.cgi](#).

5.2 ИСПОЛЬЗОВАНИЕ ПРОТОКОЛА «КИБЕРПЛАТ»

Протокол динамических форм (далее — ДФ) имеет свою логику и свои запросы, описанные в отдельной документации.

Протокол Киберплат при работе с ДФ используется как транспортный.

Запрос по протоколу ДФ помещается в поле DYNAMIC_RQ запроса по протоколу Киберплат.

Ответ по протоколу ДФ возвращается в поле DYNAMIC_RS ответа по протоколу Киберплат.

Сообщение протокола ДФ представляет из себя текст в формате JSON. Для формирования значения поля DYNAMIC_RQ необходимо применить к тексту JSON кодирование согласно формату [application/x-www-form-urlencoded](#), используя кодировку Windows-1251.

В случае успешного прохождения запроса в ответе Киберплат возвращаются поля ERROR=0, RESULT=0. При получении этих кодов нужно извлечь значение поле DYNAMIC_RS, декодировав его по тому же алгоритму [application/x-www-form-urlencoded](#) в кодировке Windows-1251. Полученное значение представляет JSON-ответ по протоколу ДФ.

Пример запроса ДФ:

```
{"idService":672,"idPoint":4590,"verify":false}
```

Пример этого же запроса в Киберплат:

```
SESSION=1543224664571  
DYNAMIC_RQ=%7B%22idService%22%3A672%2C%22idPoint%22%3A4590%2C%22verify%22%3Afalse%7D
```

Ответ в Киберплат:

```
ERROR=0  
RESULT=0  
DYNAMIC_RS=%7B%22code%22%3A0%2C%22session%22%3A%22a12bb89c-b42a-48d3-  
adc2-af35c0a5c985%22%7D
```

Ответ в ДФ:

```
{"code":0,"session":"a12bb89c-b42a-48d3-adc2-af35c0a5c985"}
```

5.3 ПРИМЕР ПРОВЕДЕНИЯ ПЛАТЕЖА С ИСПОЛЬЗОВАНИЕМ ПРОТОКОЛА ДФ

Рассмотрим пример работы динамического модуля ввода данных по протоколу Киберплат. Последовательность действий приведена на рисунке 5.3.1.

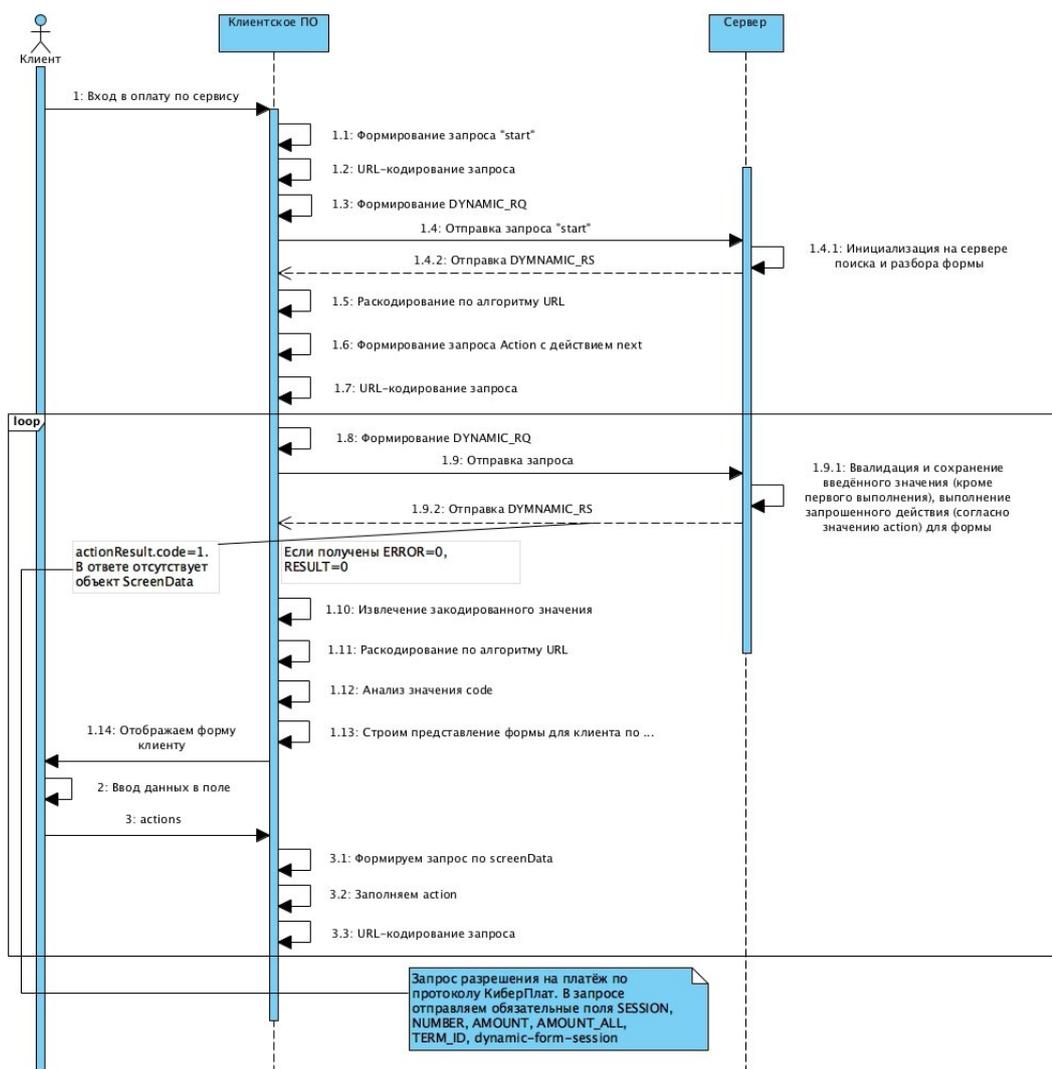


Рисунок 5.3.1 — Последовательность действий для примера работы динамического модуля ввода данных

1. Инициализируем взаимодействие по протоколу ДФ, отправляя запрос Start;

Запрос ДФ:

```
{"idService": 672, "idPoint": 4590, "verify": false}
```

Запроса Киберплат:

```
SESSION=1543224664571  
DYNAMIC_RQ=%7B%22idService%22%3A672%2C%22idPoint%22%3A4590%2C%22verify%22%3Afalse%7D
```

Ответ Киберплат:

```
ERROR=0  
RESULT=0  
DYNAMIC_RS=%7B%22code%22%3A0%2C%22session%22%3A%22a12bb89c-b42a-48d3-adc2-af35c0a5c985%22%7D
```

Ответ ДФ:

```
{"code": 0, "session": "a12bb89c-b42a-48d3-adc2-af35c0a5c985"}
```

2. Выполняем первый запрос Action с действием next, не отправляя никаких данных (ScreenData). Таким образом, мы получаем первый экран;

Запрос ДФ:

```
{"session": "a12bb89c-b42a-48d3-adc2-af35c0a5c985", "action": "next"}
```

Запрос Киберплат:

```
SESSION=1543224665154  
DYNAMIC_RQ=%7B%22session%22%3A%22a12bb89c-b42a-48d3-adc2-af35c0a5c985%22%2C%22action%22%3A%22next%22%7D
```

Ответ Киберплат:

```
ERROR=0  
RESULT=0  
DYNAMIC_RS=%7B%22attributesData%22%3A%5B%7B%22attributes%22%3A%5B%7B%22code%22%3A%22МЕМО_ИФНО%22%2C%22value%22%3A%22%2C%22%F1%EE%EE%F2%E2%E5%F2%F1%F2%E2%E8%E8+%F1+%D4%E5%E4%E5%F0%E0%EB%FC%ED%FB%EC+%E7%E0%EA%EE%ED%EE%EC+%B9115-%D4%C7+%E2%FB+%ED%E5%F1%E5%F2%E5+%EE%F2%E2%E5%F2%F1%F2%E2%E5%ED%ED%EE%F1%F2%FC+%E7%E0+%E2%E2%EE%E4+%E7%E0%E2%E5%E4%EE%EC%EE+%ED%E5%EA%EE%F0%F0%E5%EA%F2%ED%FB%F5+%E4%E0%ED%ED%FB%F5+%EF%F0%E8+%EE%F1%F3%F9%E5%F1%F2%E2%EB%E5%ED%E8%E8+%EF%EB%E0%F2%E5%E6%E5%E9%22%7D%2C%7B
```

```
%22code%22%3A%22orderSum%22%2C%22title%22%3A%22%D1%F3%EC%EC%E0+%EF%EB
%E0%F2%E5%E6%E0%3A%22%2C%22value%22%3A%220.00%22%7D%2C%7B%22code%22%3A%22commissionSum
%22%2C%22title%22%3A%22%D1%F3%EC%EC%E0+%C8%D2%CE%3A%22%2C%22value%22%3A%220.00%22%7D
%2C%7B%22code%22%3A%22totalSum%22%2C%22title%22%3A%22CA+%EE%EF%EB%E0%F2%E5%3A%22%2C
%22value%22%3A%220.00%22%7D%5D%7D%5D%2C%22code%22%3A0%2C%22screenData%22%3A%7B
%22fields%22%3A%5B%7B%22code%22%3A%22PAYER_INN%22%2C%22type%22%3A%22TEXT_FIELD%22%2C
%22title%22%3A%22%C8%CD%CD+%EF%EB%E0%F2%E5%EB%FC%F9%E8%EA%E0%22%2C%22secure%22%3Afalse
%2C%22maxLen%22%3A12%2C%22readOnly%22%3Afalse%2C%22invalid%22%3Afalse%2C%22help%22%3A
%22%C2%E2%E5%E4%E8%F2%E5+%C8%CD%CD%22%2C%22required%22%3Atrue%2C%22regexp%22%3A%5B
%22%28%5E%28%5B1-9%5D.%7C.%5B1-9%5D%7C%5B1-9%5D%7B2%7D%29%5C%5Cd%7B10%7D%24%29%22%5D
%7D%5D%7D%2C%22action%22%3A%22next%22%2C%22actionResult%22%3A%7B%22code%22%3A0%2C
%22description%22%3A%22%D8%E0%E3+%F3%F1%EF%E5%F8%ED%EE+%E2%FB%EF%EE%EB%ED%E5%ED%2C+
%E5%F1%F2%FC+%F1%EB%E5%E4%F3%FE%F9%E8%E9+%F8%E0%E3%22%7D%7D
```

Ответ ДФ (здесь и далее JSON отформатирован для наглядности):

```
{
  "attributesData": [
    {
      "attributes": [
        {
          "code": "MEMO_INFO",
          "value": "В соответствии с Федеральным законом №115-ФЗ вы несете ответственность
за ввод заведомо некорректных данных при осуществлении платежей"
        },
        {
          "code": "orderSum", "title": "Сумма платежа:", "value": "0.00"
        },
        {
          "code": "commissionSum", "title": "Сумма ИТО:", "value": "0.00"
        },
        {
          "code": "totalSum", "title": "К оплате:", "value": "0.00"
        }
      ]
    }
  ],
  "code": 0,
  "screenData": {
    "fields": [
      {
```

```
"code": "PAYER_INN", "type": "TEXT_FIELD", "title": "ИНН плательщика",  
"secure": false, "maxLen": 12, "readOnly": false, "invalid": false, "help": "Введите ИНН",  
"required": true, "regex": [  
    "^[1-9].[1-9][1-9]{2}\\d{10}$"  
]  
}  
]  
,  
"action": "next",  
"actionResult": {  
    "code": 0,  
    "description": "Шаг успешно выполнен, есть следующий шаг"  
}  
}
```

Видим, что получили описание экрана с одним полем — PAYER_INN.

3. Клиент вводит значение поля поле PAYER_INN, отправляем заполненный ScreenData;

Запрос ДФ:

```
{  
    "session": "a12bb89c-b42a-48d3-adc2-af35c0a5c985",  
    "screenData": {  
        "fields": [  
            {  
                "code": "PAYER_INN", "type": "TEXT_FIELD", "title": "ИНН плательщика",  
                "value": "999999999950", "secure": false, "maxLen": 12, "readOnly": false,  
                "invalid": false, "help": "Введите ИНН", "required": true, "regex": [  
                    "^[1-9].[1-9][1-9]{2}\\d{10}$"  
                ]  
            }  
        ]  
    }  
},  
    "action": "next"  
}
```



```
%C4%D0%C0%5C%22%29+%D8%D2%D0%C0%D4+%CF%CE+%C0%C4%CC%C8%CD%C8%D1%D2%D0%C0%D2%C8%2%CD
%CE%CC%D3+%CF%D0%C0%2%CE%CD%C0%D0%D3%D8%C5%CD%C8%DE+%CF%CE%D1%D2%C0%CD%CE%2%CB%5%CD
%C8%C5+%B9%231881006916020918005550603%22%7D%2C%7B%22code%22%3A%22SSRV_SUM_SUB%22%2C
%22title%22%3A%22%D1%F3%EC%EC%E0%22%2C%22value%22%3A%2248%22%7D%2C%7B%22code%22%3A
%22PAY_SUB%22%2C%22title%22%3A%22%D1%F3%EC%EC%E0+%EF%EB%E0%F2%E5%E6%E0%22%2C%22value
%22%3A%221%22%7D%5D%7D%5D%7D%7D%7D%7D%7D%7D%2C%7B%22attributes%22%3A%5B%7B%22code%22%3A
%22MEMO_INFO%22%2C%22value%22%3A%22C2+%F1%EE%EE%F2%E2%E5%F2%F1%F2%E2%E8%E8+%F1+
%D4%E5%E4%E5%F0%E0%EB%FC%ED%FB%EC+%E7%E0%EA%EE%ED%EE%EC+%B9115-%D4%C7+%E2%FB+%ED
%E5%F1%E5%F2%E5+%EE%F2%E2%E5%F2%F1%F2%E2%E5%ED%ED%EE%F1%F2%FC+%E7%E0+%E2%E2%EE%E4+
%E7%E0%E2%E5%E4%EE%EC%EE+%ED%E5%EA%EE%F0%F0%E5%EA%F2%ED%FB%F5+%E4%E0%ED%ED%FB%F5+%EF
%F0%E8+%EE%F1%F3%F9%E5%F1%F2%E2%EB%E5%ED%E8%E8+%EF%EB%E0%F2%E5%E6%E5%E9%22%7D%2C%7B
%22code%22%3A%22orderSum%22%2C%22title%22%3A%22%D1%F3%EC%EC%E0+%EF%EB
%E0%F2%E5%E6%E0%3A%22%2C%22value%22%3A%22144.00%22%7D%2C%7B%22code%22%3A
%22commissionSum%22%2C%22title%22%3A%22%D1%F3%EC%EC%E0+%C8%D2%CE%3A%22%2C%22value
%22%3A%221.44%22%7D%2C%7B%22code%22%3A%22totalSum%22%2C%22title%22%3A%22CA+%EE%EF%EB
%E0%F2%E5%3A%22%2C%22value%22%3A%22145.44%22%7D%5D%7D%5D%2C%22code%22%3A%22%2C
%22screenData%22%3A%7B%22fields%22%3A%5B%7B%22code%22%3A%22%40Table%40SSRV_TABLE%40Row
%401%40Cell%40NAME_SUB%22%2C%22type%22%3A%22TEXT_FIELD%22%2C%22title%22%3A%22%CF%EE%EB
%F3%F7%E0%F2%E5%EB%FC%22%2C%22value%22%3A%22%D3%D4%CA+%CF%CE+%D1%CC%CE%CB%5%CD%D1%CA
%CE%C9+%CE%C1%CB%0%D1%D2%C8%28%C3%8%C1%C4%4+%D3%C2%4+%CF%CE+%DF%CC%0%CB%CE-%CD
%C5%CD%C5%D6%CA%CE%CC%D3+%C0%2D2%CE%CD%CE%CC%CD%CE%CC%D3+%CE%CA%D0%D3%3D3+%D1%CF
%C5%D6%CF%CE%4%D0%0%7%4%4%5%CB%5%CD%8%8%5%5%22%D2%D3%CD%4%D0%0%5C%22%29+
%D8%D2%D0%C0%D4+%CF%CE+%C0%C4%CC%C8%CD%C8%D1%D2%D0%C0%D2%C8%2%CD%CE%CC%D3+%CF
%D0%C0%2%CE%CD%0%D0%D3%D8%C5%CD%C8%DE+%CF%CE%D1%D2%C0%CD%CE%2%CB%5%CD%8%8%5+
%B9%231881006916020918005550601%22%2C%22secure%22%3Afalse%2C%22readOnly%22%3Atrue%2C
%22invalid%22%3Afalse%2C%22required%22%3Afalse%7D%2C%7B%22code%22%3A%22%40Table
%40SSRV_TABLE%40Row%401%40Cell%40SSRV_SUM_SUB%22%2C%22type%22%3A%22NUMERIC_FIELD%22%2C
%22title%22%3A%22%D1%F3%EC%EC%E0%22%2C%22value%22%3A%2248%22%2C%22secure%22%3Afalse%2C
%22readOnly%22%3Atrue%2C%22invalid%22%3Afalse%2C%22required%22%3Afalse%7D%2C%7B%22code
%22%3A%22%40Table%40SSRV_TABLE%40Row%401%40Cell%40PAY_SUB%22%2C%22type%22%3A
%22SELECTOR_FIELD%22%2C%22title%22%3A%22%D1%F3%EC%EC%E0+%EF%EB%E0%F2%E5%E6%E0%22%2C
%22value%22%3A%221%22%2C%22secure%22%3Afalse%2C%22readOnly%22%3Afalse%2C%22invalid
%22%3Afalse%2C%22required%22%3Afalse%2C%22values%22%3A%5B%7B%22title%22%3A%22CD
%E5%F2%22%2C%22value%22%3A%220%22%7D%2C%7B%22title%22%3A%22C4%E0%22%2C%22value%22%3A
%221%22%7D%5D%7D%5D%7D%2C%22action%22%3A%22next%22%2C%22actionResult%22%3A%7B%22code
%22%3A%22%22description%22%3A%22%D8%E0%E3+%F3%F1%EF%E5%F8%ED%EE+%E2%FB%EF%EE%EB%ED
%E5%ED%2C+%E5%F1%F2%FC+%F1%EB%E5%E4%F3%FE%F9%E8%E9+%F8%E0%E3%22%7D%7D
```

Ответ ДФ:

```
{
  "attributesData": [
    {
      "attributes": [
        {
          "code": "PAYER_INN", "title": "ИНН плательщика", "value": "999999999950"
        }
      ]
    }
  ]
}
```

```
}
]
},
{
  "nestedData": {
    "id": "@Table@SSRV_TABLE",
    "data": [
      {
        "nestedData": {
          "id": "@Row@1",
          "data": [
            {
              "attributes": [
                {
                  "code": "NAME_SUB", "title": "Получатель", "value": "УФК ПО СМОЛЕНСКОЙ  
ОБЛАСТИ ШТРАФ ПО АДМИНИСТРАТИВНОМУ ПРАВОНАРУШЕНИЮ ПОСТАНОВЛЕНИЕ  
№#1881006916020918005550601"
                },
                {"code": "SSRV_SUM_SUB", "title": "Сумма", "value": "48"},
                {"code": "PAY_SUB", "title": "Сумма платежа", "value": "1"}
              ]
            }
          ]
        }
      }
    ]
  },
  {
    "nestedData": {
      "id": "@Row@2",
      "data": [
        {
          "attributes": [
            {
              "code": "NAME_SUB", "title": "Получатель", "value": "УФК ПО  
СМОЛЕНСКОЙ ОБЛАСТИ ШТРАФ ПО АДМИНИСТРАТИВНОМУ ПРАВОНАРУШЕНИЮ  
ПОСТАНОВЛЕНИЕ №#1881006916020918005550602"
            },
            {"code": "SSRV_SUM_SUB", "title": "Сумма", "value": "48"},

```

```
        {"code": "PAY_SUB", "title": "Сумма платежа", "value": "1"}
    ]
  }
]
},
{
  "nestedData": {
    "id": "@Row@3",
    "data": [
      {
        "attributes": [
          {
            "code": "NAME_SUB", "title": "Получатель", "value": "УФК ПО
СМОЛЕНСКОЙ ОБЛАСТИ ШТРАФ ПО АДМИНИСТРАТИВНОМУ ПРАВОНАРУШЕНИЮ
ПОСТАНОВЛЕНИЕ №#1881006916020918005550603"
          },
          {"code": "SSRV_SUM_SUB", "title": "Сумма", "value": "48"},
          {"code": "PAY_SUB", "title": "Сумма платежа", "value": "1"}
        ]
      }
    ]
  }
}
]
},
{
  "attributes": [
    {
      "code": "MEMO_INFO", "value": "В соответствии с Федеральным законом №115-ФЗ вы
несете ответственность за ввод заведомо некорректных данных при осуществлении платежей"
    },
    {"code": "orderSum", "title": "Сумма платежа:", "value": "144.00"},
    {"code": "commissionSum", "title": "Сумма ИТО:", "value": "1.44"},
    {"code": "totalSum", "title": "К оплате:", "value": "145.44"}
  ]
}
```

```
}
],
"code": 0,
"screenData": {
  "fields": [
    {
      "code": "@Table@SSRV_TABLE@Row@1@Cell@NAME_SUB",
      "type": "TEXT_FIELD", "title": "Получатель", "value": "УФК ПО СМОЛЕНСКОЙ ОБЛАСТИ
ШТРАФ ПО АДМИНИСТРАТИВНОМУ ПРАВОНАРУШЕНИЮ ПОСТАНОВЛЕНИЕ
№#1881006916020918005550601",
      "secure": false, "readOnly": true, "invalid": false, "required": false
    },
    {
      "code": "@Table@SSRV_TABLE@Row@1@Cell@SSRV_SUM_SUB",
      "type": "NUMERIC_FIELD", "title": "Сумма", "value": "48", "secure": false,
      "readOnly": true, "invalid": false, "required": false
    },
    {
      "code": "@Table@SSRV_TABLE@Row@1@Cell@PAY_SUB",
      "type": "SELECTOR_FIELD", "title": "Сумма платежа", "value": "1",
      "secure": false, "readOnly": false, "invalid": false, "required": false,
      "values": [
        {"title": "Нет", "value": "0"},
        {"title": "Да", "value": "1"}
      ]
    }
  ]
},
"action": "next",
"actionResult": {
  "code": 0,
  "description": "Шаг успешно выполнен, есть следующий шаг"
}
}
```

В объекте `attributesData` видим, что провайдер вернул таблицу с именем `SSRV_TABLE`, содержащую 3 строки. Каждая строка содержит 3 поля (колонки): `NAME_SUB`, `SSRV_SUM_SUB`, `PAY_SUB`. Значения полей установлены провайдером.

Имя таблицы формируется по шаблону `@Table@КОД_ТАБЛИЦЫ`

Имя строки формируется по шаблону `@Row@КОД_СТРОКИ`

Объект `screenData` содержит описание экрана. Видно, что данный экран — это одна строка таблицы. При этом имя поля сформировано по шаблону `@Table@КОД_ТАБЛИЦЫ@Row@КОД_СТРОКИ@Cell@КОД_КОЛОНКИ`

Тип поля `@Table@SSRV_TABLE@Row@1@Cell@PAY_SUB` — селектор. Клиент должен выбрать значение «Да» или «Нет». В поле попадёт соответственно 1 или 0. Поле подписано «Сумма платежа», эту подпись возвращает провайдер. Однако в данном случае смысл выбора — оплачивать или не оплачивать данный штраф (первый из трёх в таблице).

Это единственное редактируемое поле на форме.

4. Отправляем заполненную форму. Запрос и ответ по протоколу Киберплат не приводим;

Запрос ДФ:

```
{
  "session": "a12bb89c-b42a-48d3-adc2-af35c0a5c985",
  "screenData": {
    "fields": [
      {
        "code": "@Table@SSRV_TABLE@Row@1@Cell@NAME_SUB",
        "type": "TEXT_FIELD", "title": "Получатель", "value": "УФК ПО СМОЛЕНСКОЙ ОБЛАСТИ  
ШТРАФ ПО АДМИНИСТРАТИВНОМУ ПРАВОНАРУШЕНИЮ ПОСТАНОВЛЕНИЕ  
№#1881006916020918005550601",
        "secure": false, "readOnly": true, "invalid": false, "required": false
      }
    ]
  }
}
```

```
    },
    {
      "code": "@Table@SSRV_TABLE@Row@1@Cell@SSRV_SUM_SUB",
      "type": "NUMERIC_FIELD", "title": "Сумма", "value": "48", "secure": false,
      "readOnly": true, "invalid": false, "required": false
    },
    {
      "code": "@Table@SSRV_TABLE@Row@1@Cell@PAY_SUB", "type": "SELECTOR_FIELD",
      "title": "Сумма платежа", "value": "1", "secure": false, "readOnly": false, "invalid": false,
      "required": false, "values": [
        {"title": "Нет", "value": "0"},
        {"title": "Да", "value": "1"}
      ]
    }
  ]
},
"action": "next"
}
```

Ответ ДФ:

```
{
  "attributesData": [
    {
      "attributes": [
        {"code": "PAYER_INN", "title": "ИНН плательщика", "value": "999999999950"}
      ]
    },
    {
      "nestedData": {
        "id": "@Table@SSRV_TABLE",
        "data": [
          {
            "nestedData": {
              "id": "@Row@1",
              "data": [
                {
                  "attributes": [
```

```
        {
            "code": "NAME_SUB", "title": "Получатель",
            "value": "УФК ПО СМОЛЕНСКОЙ ОБЛАСТИ ШТРАФ ПО
АДМИНИСТРАТИВНОМУ ПРАВОНАРУШЕНИЮ ПОСТАНОВЛЕНИЕ
№#1881006916020918005550601"
        },
        {"code": "SSRV_SUM_SUB", "title": "Сумма", "value": "48"},
        {"code": "PAY_SUB", "title": "Сумма платежа", "value": "1"}
    ]
}
]
},
{
    "nestedData": {
        "id": "@Row@2",
        "data": [
            {
                "attributes": [
                    {
                        "code": "NAME_SUB", "title": "Получатель", "value": "УФК ПО
СМОЛЕНСКОЙ ОБЛАСТИ ШТРАФ ПО АДМИНИСТРАТИВНОМУ ПРАВОНАРУШЕНИЮ
ПОСТАНОВЛЕНИЕ №#1881006916020918005550602"
                    },
                    {"code": "SSRV_SUM_SUB", "title": "Сумма", "value": "48"},
                    {"code": "PAY_SUB", "title": "Сумма платежа", "value": "1"}
                ]
            }
        ]
    }
},
{
    "nestedData": {
        "id": "@Row@3",
        "data": [
            {
                "attributes": [
```

```
{
  {
    "code": "NAME_SUB", "title": "Получатель", "value": "УФК ПО
СМОЛЕНСКОЙ ОБЛАСТИ ШТРАФ ПО АДМИНИСТРАТИВНОМУ ПРАВОНАРУШЕНИЮ
ПОСТАНОВЛЕНИЕ №#1881006916020918005550603"
  },
  {"code": "SSRV_SUM_SUB", "title": "Сумма", "value": "48"},
  {"code": "PAY_SUB", "title": "Сумма платежа", "value": "1"}
]
}
}
}
}
},
{
  "attributes": [
    {
      "code": "MEMO_INFO", "value": "В соответствии с Федеральным законом №115-ФЗ вы
несете ответственность за ввод заведомо некорректных данных при осуществлении платежей"
    },
    {"code": "orderSum", "title": "Сумма платежа:", "value": "144.00"},
    {"code": "commissionSum", "title": "Сумма ИТО:", "value": "1.44"},
    {"code": "totalSum", "title": "К оплате:", "value": "145.44"}
  ]
}
],
"code": 0,
"screenData": {
  "fields": [
    {
      "code": "@Table@SSRV_TABLE@Row@2@Cell@NAME_SUB",
      "type": "TEXT_FIELD", "title": "Получатель", "value": "УФК ПО СМОЛЕНСКОЙ ОБЛАСТИ
ШТРАФ ПО АДМИНИСТРАТИВНОМУ ПРАВОНАРУШЕНИЮ ПОСТАНОВЛЕНИЕ
№#1881006916020918005550602",
      "secure": false, "readOnly": true, "invalid": false, "required": false
    }
  ]
}
}
```

```
    },
    {
      "code": "@Table@SSRV_TABLE@Row@2@Cell@SSRV_SUM_SUB",
      "type": "NUMERIC_FIELD", "title": "Сумма", "value": "48", "secure": false,
      "readOnly": true, "invalid": false, "required": false
    },
    {
      "code": "@Table@SSRV_TABLE@Row@2@Cell@PAY_SUB",
      "type": "SELECTOR_FIELD", "title": "Сумма платежа", "value": "1",
      "secure": false, "readOnly": false, "invalid": false, "required": false, "values": [
        {"title": "Нет", "value": "0"},
        {"title": "Да", "value": "1"}
      ]
    }
  ]
},
"action": "next",
"actionResult": {
  "code": 0, "description": "Шаг успешно выполнен, есть следующий шаг"
}
}
```

Ответ содержит форму, основанную на второй строке таблицы. Обрабатываем эту форму (для второй строки таблицы) и следующую (для третьей строки таблицы) аналогично.

5. После запроса с заполненной формой третьей строки, получаем ответ ДФ:

```
{
  "attributesData": [
    {
      "attributes": [
        {"code": "PAYER_INN", "title": "ИНН плательщика", "value": "999999999950"}
      ]
    },
    {
      "nestedData": {
        "id": "@Table@SSRV_TABLE",
```

```
"data": [
  {
    "nestedData": {
      "id": "@Row@1",
      "data": [
        {
          "attributes": [
            {
              "code": "NAME_SUB", "title": "Получатель", "value": "УФК ПО СМОЛЕНСКОЙ
ОБЛАСТИ ШТРАФ ПО АДМИНИСТРАТИВНОМУ ПРАВОНАРУШЕНИЮ ПОСТАНОВЛЕНИЕ
№#1881006916020918005550601"
            },
            {"code": "SSRV_SUM_SUB", "title": "Сумма", "value": "48"},
            {"code": "PAY_SUB", "title": "Сумма платежа", "value": "1"}
          ]
        }
      ]
    }
  },
  {
    "nestedData": {
      "id": "@Row@2",
      "data": [
        {
          "attributes": [
            {
              "code": "NAME_SUB", "title": "Получатель", "value": "УФК ПО СМОЛЕНСКОЙ
ОБЛАСТИ ШТРАФ ПО АДМИНИСТРАТИВНОМУ ПРАВОНАРУШЕНИЮ ПОСТАНОВЛЕНИЕ
№#1881006916020918005550602"
            },
            {"code": "SSRV_SUM_SUB", "title": "Сумма", "value": "48"},
            {"code": "PAY_SUB", "title": "Сумма платежа", "value": "0"}
          ]
        }
      ]
    }
  }
],
```

```
{
  "nestedData": {
    "id": "@Row@3",
    "data": [
      {
        "attributes": [
          {
            "code": "NAME_SUB", "title": "Получатель", "value": "УФК ПО СМОЛЕНСКОЙ
ОБЛАСТИ ШТРАФ ПО АДМИНИСТРАТИВНОМУ ПРАВОНАРУШЕНИЮ ПОСТАНОВЛЕНИЕ
№#1881006916020918005550603"
          },
          {"code": "SSRV_SUM_SUB", "title": "Сумма", "value": "48"},
          {"code": "PAY_SUB", "title": "Сумма платежа", "value": "0"}
        ]
      }
    ]
  },
  {
    "attributes": [
      {
        "code": "MEMO_INFO", "value": "В соответствии с Федеральным законом №115-ФЗ вы
несете ответственность за ввод заведомо некорректных данных при осуществлении платежей"
      },
      {"code": "orderSum", "title": "Сумма платежа:", "value": "48.00"},
      {"code": "commissionSum", "title": "Сумма ИТО:", "value": "0.48"},
      {"code": "totalSum", "title": "К оплате:", "value": "48.48"}
    ]
  }
],
"code": 0,
"screenData": {
  "fields": [
    {
```

```
"code": "UIP_CITY", "type": "TEXT_FIELD", "title": "Населенный пункт",
"secure": false, "maxLen": 70, "readOnly": false, "invalid": false, "required": false
},
{
  "code": "UIP_STREET", "type": "TEXT_FIELD", "title": "Улица",
  "secure": false, "maxLen": 70, "readOnly": false, "invalid": false, "required": false
},
{
  "code": "UIP_HOUSE", "type": "TEXT_FIELD", "title": "Дом", "secure": false,
  "maxLen": 10, "readOnly": false, "invalid": false, "required": false
},
{
  "code": "UIP_FLAT", "type": "TEXT_FIELD", "title": "Квартира",
  "secure": false, "maxLen": 7, "readOnly": false, "invalid": false, "required": false
},
{
  "code": "MEMO_INFO", "type": "TEXT_FIELD", "value": "В соответствии с Федеральным
законом №115-ФЗ вы несете ответственность за ввод заведомо некорректных данных при
осуществлении платежей", "secure": false,
  "defaultValue": "В соответствии с Федеральным законом №115-ФЗ вы несете
ответственность за ввод заведомо некорректных данных при осуществлении платежей",
  "maxLen": 3128, "readOnly": true, "invalid": false, "required": false
},
{
  "code": "FIO", "type": "TEXT_FIELD", "title": "ФИО (физ. Лицо)", "secure": false, "maxLen": 70,
  "readOnly": false, "invalid": false, "required": true, "regexp": [
    "^[А-ЯЁа-яёА-Za-z\\.\u0027\\-]*[А-ЯЁа-яёА-Za-z] [А-ЯЁа-яёА-Za-z\\.\u0027\\-]+[А-ЯЁа-яёА-
Za-z]([А-ЯЁа-яёА-Za-z\\.\u0027\\-]+[А-ЯЁа-яёА-Za-z]){0,3}$"
  ]
},
{
  "code": "IDENT_DOCTYPE_VIS", "type": "SELECTOR_FIELD", "title": "Тип документа",
  "secure": false, "readOnly": false, "invalid": false, "required": false, "values": [
    {"title": "ПАСПОРТ РФ", "value": "ПАСПОРТ РФ"},
    {
      "title": "ПАСПОРТ ИНОСТРАННОГО ГРАЖДАНИНА",
```

```
        "value": "ПАСПОРТ ИНОСТРАННОГО ГРАЖДАНИНА"
    }
  ]
},
{
  "code": "IDENT_DOCNUM", "type": "TEXT_FIELD", "title": "Номер документа", "secure": false,
  "maxLen": 12, "readOnly": false, "invalid": false, "required": false
},
{
  "code": "UIP_CITIZEN_VISIBLE", "type": "SELECTOR_FIELD", "title": "Гражданство",
  "secure": false, "readOnly": false, "invalid": false, "required": false,
  "values": [
    {
      "title": "РОССИЙСКАЯ ФЕДЕРАЦИЯ",
      "value": "РОССИЙСКАЯ ФЕДЕРАЦИЯ"
    },
    ...
    {"title": "БЕЗ ГРАЖДАНСТВА", "value": "БЕЗ ГРАЖДАНСТВА" }
  ]
},
{
  "code": "orderSum", "type": "NUMERIC_FIELD", "title": "Сумма платежа:", "value": "48.00",
  "secure": false, "defaultValue": "48.00", "readOnly": true, "invalid": false, "required": false
},
{
  "code": "commissionSum", "type": "NUMERIC_FIELD", "title": "Сумма ИТО:", "value": "0.48",
  "secure": false, "defaultValue": "0.48", "readOnly": true, "invalid": false, "required": false
},
{
  "code": "totalSum", "type": "NUMERIC_FIELD", "title": "К оплате:", "value": "48.48",
  "secure": false, "defaultValue": "48.48", "readOnly": true, "invalid": false, "required": false
}
]
},
"action": "next",
"actionResult": {
  "code": 0, "description": "Шаг успешно выполнен, есть следующий шаг"
```

```
}  
}
```

6. Клиент вводит требуемые данные, отправляет заполненную форму;

Запрос ДФ:

```
{  
  "session": "a12bb89c-b42a-48d3-adc2-af35c0a5c985",  
  "screenData": {  
    "fields": [  
      {  
        "code": "UIP_CITY", "type": "TEXT_FIELD", "title": "Населенный пункт", "value": "Москва",  
        "secure": false, "maxLen": 70, "readOnly": false, "invalid": false, "required": false  
      },  
      {  
        "code": "UIP_STREET", "type": "TEXT_FIELD", "title": "Улица", "value": "Ленина",  
        "secure": false, "maxLen": 70, "readOnly": false, "invalid": false, "required": false  
      },  
      {  
        "code": "UIP_HOUSE", "type": "TEXT_FIELD", "title": "Дом", "value": "1",  
        "secure": false, "maxLen": 10, "readOnly": false, "invalid": false, "required": false  
      },  
      {  
        "code": "UIP_FLAT", "type": "TEXT_FIELD", "title": "Квартира", "value": "2",  
        "secure": false, "maxLen": 7, "readOnly": false, "invalid": false, "required": false  
      },  
      {  
        "code": "MEMO_INFO", "type": "TEXT_FIELD", "value": "В соответствии с Федеральным  
законом №115-ФЗ вы несете ответственность за ввод заведомо некорректных данных при  
осуществлении платежей", "secure": false,  
        "defaultValue": "В соответствии с Федеральным законом №115-ФЗ вы несете  
ответственность за ввод заведомо некорректных данных при осуществлении платежей", "maxLen":  
3128, "readOnly": true, "invalid": false, "required": false  
      },  
      {  
        "code": "FIO", "type": "TEXT_FIELD", "title": "ФИО (физ. лицо)",  
        "value": "Иванов Иван Иванович", "secure": false, "maxLen": 70,  

```

```
"readOnly": false, "invalid": false, "required": true, "regexp": [
  "[А-ЯЁа-яёА-Za-z\\.\u0027\\-]*[А-ЯЁа-яёА-Za-z] [А-ЯЁа-яёА-Za-z\\.\u0027\\-]+[А-ЯЁа-яёА-
Za-z]( [А-ЯЁа-яёА-Za-z\\.\u0027\\-]+[А-ЯЁа-яёА-Za-z]){0,3}$"
]
},
{
  "code": "IDENT_DOCTYPE_VIS", "type": "SELECTOR_FIELD",
  "title": "Тип документа", "secure": false, "readOnly": false, "invalid": false, "required": false,
  "values": [
    { "title": "ПАСПОРТ РФ", "value": "ПАСПОРТ РФ" },
    {
      "title": "ПАСПОРТ ИНОСТРАННОГО ГРАЖДАНИНА",
      "value": "ПАСПОРТ ИНОСТРАННОГО ГРАЖДАНИНА"
    }
  ]
},
{
  "code": "IDENT_DOCNUM", "type": "TEXT_FIELD", "title": "Номер документа",
  "secure": false, "maxLen": 12, "readOnly": false, "invalid": false, "required": false
},
{
  "code": "UIP_CITIZEN_VISIBLE", "type": "SELECTOR_FIELD", "title": "Гражданство",
  "value": "РОССИЙСКАЯ ФЕДЕРАЦИЯ", "secure": false, "readOnly": false, "invalid": false,
  "required": false, "values": [
    { "title": "РОССИЙСКАЯ ФЕДЕРАЦИЯ", "value": "РОССИЙСКАЯ ФЕДЕРАЦИЯ" },
    . . .
    { "title": "БЕЗ ГРАЖДАНСТВА", "value": "БЕЗ ГРАЖДАНСТВА" }
  ]
},
{
  "code": "orderSum", "type": "NUMERIC_FIELD", "title": "Сумма платежа:", "value": "48.00",
  "secure": false, "defaultValue": "48.00", "readOnly": true, "invalid": false, "required": false
},
{
  "code": "commissionSum", "type": "NUMERIC_FIELD", "title": "Сумма ИТО:", "value": "0.48",
  "secure": false, "defaultValue": "0.48", "readOnly": true, "invalid": false, "required": false
},
},
```

```
{
  "code": "totalSum", "type": "NUMERIC_FIELD", "title": "К оплате:", "value": "48.48",
  "secure": false, "defaultValue": "48.48", "readOnly": true, "invalid": false, "required": false
}
],
"action": "next"
}
```

7. Получаем ответ, содержащий `actionResult.code=1`, что означает завершение ввода данных. В ответе также отсутствует форма для ввода (объект `ScreenData`);

Ответ ДФ:

```
{
  "attributesData": [
    {
      "attributes": [
        {"code": "PAYER_INN", "title": "ИНН плательщика", "value": "999999999950"}
      ]
    },
    {
      "nestedData": {
        "id": "@Table@SSRV_TABLE",
        "data": [
          {
            "nestedData": {
              "id": "@Row@1",
              "data": [
                {
                  "attributes": [
                    {
                      "code": "NAME_SUB", "title": "Получатель",
                      "value": "УФК ПО СМОЛЕНСКОЙ ОБЛАСТИ ШТРАФ ПО
АДМИНИСТРАТИВНОМУ ПРАВОНАРУШЕНИЮ ПОСТАНОВЛЕНИЕ
№#1881006916020918005550601"
                    }
                  ]
                },
                {"code": "SSRV_SUM_SUB", "title": "Сумма", "value": "48"},

```

```
        {"code": "PAY_SUB", "title": "Сумма платежа", "value": "1"}
    ]
}
]
},
{
    "nestedData": {
        "id": "@Row@2",
        "data": [
            {
                "attributes": [
                    {
                        "code": "NAME_SUB", "title": "Получатель",
                        "value": "УФК ПО СМОЛЕНСКОЙ ОБЛАСТИ ШТРАФ ПО
АДМИНИСТРАТИВНОМУ ПРАВОНАРУШЕНИЮ ПОСТАНОВЛЕНИЕ
№#1881006916020918005550602"
                    },
                    {"code": "SSRV_SUM_SUB", "title": "Сумма", "value": "48"},
                    {"code": "PAY_SUB", "title": "Сумма платежа", "value": "0"}
                ]
            }
        ]
    }
},
{
    "nestedData": {
        "id": "@Row@3",
        "data": [
            {
                "attributes": [
                    {
                        "code": "NAME_SUB", "title": "Получатель",
                        "value": "УФК ПО СМОЛЕНСКОЙ ОБЛАСТИ ШТРАФ ПО
АДМИНИСТРАТИВНОМУ ПРАВОНАРУШЕНИЮ ПОСТАНОВЛЕНИЕ
№#1881006916020918005550603"
                    },

```

```
        {"code": "SSRV_SUM_SUB", "title": "Сумма", "value": "48"},
        {"code": "PAY_SUB", "title": "Сумма платежа", "value": "0"}
    ]
}
]
}
]
},
{
    "attributes": [
        {"code": "UIP_CITY", "title": "Населенный пункт", "value": "МОСКВА"},
        {"code": "UIP_STREET", "title": "Улица", "value": "ЛЕНИНА"},
        {"code": "UIP_HOUSE", "title": "Дом", "value": "1"},
        {"code": "UIP_FLAT", "title": "Квартира", "value": "2"},
        {
            "code": "MEMO_INFO", "value": "В соответствии с Федеральным законом №115-ФЗ вы
несете ответственность за ввод заведомо некорректных данных при осуществлении платежей"
        },
        {"code": "FIO", "title": "ФИО (физ. лицо)", "value": "ИВАНОВ ИВАН ИВАНОВИЧ"},
        {
            "code": "UIP_CITIZEN_VISIBLE", "title": "Гражданство",
            "value": "РОССИЙСКАЯ ФЕДЕРАЦИЯ", "valueTitle": "РОССИЙСКАЯ ФЕДЕРАЦИЯ"
        },
        {"code": "orderSum", "title": "Сумма платежа:", "value": "48.00"},
        {"code": "commissionSum", "title": "Сумма ИТО:", "value": "0.48"},
        {"code": "totalSum", "title": "К оплате:", "value": "48.48"}
    ]
}
],
"code": 0, "action": "next",
"actionResult": {
    "code": 1, "description": "Шаг успешно выполнен, является завершающим"
}
}
```

8. Выполняем запрос разрешения на платёж по протоколу Киберплат. В запросе отправляем обязательные поля SESSION, NUMBER, AMOUNT, AMOUNT_ALL, TERM_ID, dynamic-form-session.

Поле dynamic-form-session должно содержать идентификатор сессии из протокола ДФ. Других данных из сообщений протокола ДФ передавать не требуется.

Запрос Киберплат:

```
SESSION=1543224684164  
NUMBER=999999999950  
AMOUNT=48.48  
AMOUNT_ALL=48.48  
TERM_ID=rtkl  
dynamic-form-session=a12bb89c-b42a-48d3-adc2-af35c0a5c985
```

Ответ Киберплат:

```
DATE=26.11.2018 16:31:24  
SESSION=1543224684164  
ERROR=0  
RESULT=0
```

Результат ERROR=0, RESULT=3, согласно протоколу Киберплат означает, что платёж был успешно принят на проведение.

Далее нужно выполнить стандартный запрос проверки статуса платежа по протоколу Киберплат.