

WebSSO Login API

Оглавление

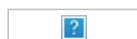
- # 1. Схема аутентификации
- # 2. Схема работы модуля отображения формы логина
- # 3. Схема работы модуля отображения формы логина со вводом captcha
- # 4. Схема работы модуля проверки введенных логина и пароля с запросом OTP
- # 5. Схема работы модуля проверки введенных логина и пароля
- # 6. Схема работы модуля входа через социальные сети
- # 7. Схема работы модуля привязки аккаунта социальной сети
- # 8. Схема работы модуля проверки данных входа через социальную сеть
- # 9. Схема работы модуля проверки OTP при входе через социальную сеть
- # 10. Схема работы модуля подтверждения операции одноразовым паролем
- # 11. Точка входа
- # 12. Блокировки
- # 13. API для виджета логина.
 - # 13.1. Общий механизм взаимодействия
 - # 13.2. Вход в ЛК
 - # 13.3. Форматы запросов и ответов
 - # 13.4. Возврат к защищаемому сервису
 - # 13.5. Возможные ограничения на поля формы
 - # 13.6. Возможные коды ошибок
 - # 13.7. Дополнительные условия
 - # 13.8. Алгоритм формирования socialData

ВАЖНО

Документ предназначен для внутреннего использования

1. Схема аутентификации

Цветом выделены состояния которые передают управление виджету.



ЗАМЕТКА

Из всех визуальных состояний есть переходы "cancel" и "fail", завершающие цепочку. Не представлено на схеме для сокращения. Из-за невозможности редактора создавать блоки на схеме с одним заголовком, состояние social разделено на social1.1, social1.2, social1.3. В действительности все они являются реализацией одного функционала. Архитектура состояния social рассмотрено в 6 пункте. Архитектура состояния login_captcha рассмотрено в 3 пункте. Архитектура состояния login_otp рассмотрено в 4 пункте. Архитектура состояния login_router рассмотрено в 2 пункте.

2. Схема работы модуля отображения формы логина

Цветом выделены состояния которые передают управление виджету.



ЗАМЕТКА

Из всех визуальных состояний есть переходы "cancel" и "fail", завершающие цепочку. Не представлено на схеме для сокращения.

3. Схема работы модуля отображения формы логина со вводом captcha

Цветом выделены состояния которые передают управление виджету.



ЗАМЕТКА

Из всех визуальных состояний есть переходы "cancel" и "fail", завершающие цепочку. Не представлено на схеме для сокращения.

4. Схема работы модуля проверки введенных логина и пароля с запросом ОТП

Цветом выделены состояния которые передают управление виджету.



ЗАМЕТКА

Из всех визуальных состояний есть переходы "cancel" и "fail", завершающие цепочку. Не представлено на схеме для сокращения. Архитектура состояния widget рассмотрено в 5 пункте. Архитектура состояния otp рассмотрено в 10 пункте.

5. Схема работы модуля проверки введенных логина и пароля

Цветом выделены состояния которые передают управление виджету.



ЗАМЕТКА

Из всех визуальных состояний есть переходы "cancel" и "fail", завершающие цепочку. Не представлено на схеме для сокращения.

6. Схема работы модуля входа через социальные сети

Цветом выделены состояния которые передают управление виджету.



ЗАМЕТКА

Из всех визуальных состояний есть переходы "cancel" и "fail", завершающие цепочку. Не представлено на схеме для сокращения. Из-за невозможности редактора создавать блоки на схеме с одним заголовком, состояние login разделено на login2.1, login1.1. В действительности login1.1 представляет собой вход через соцсети, а login2.1 проверяет введенные логин и пароль пользователя. Архитектура состояния login1.1 рассмотрено в 8 пункте. Архитектура состояния otp рассмотрено в 9 пункте. Архитектура состояния login_router рассмотрено во 2 пункте. Архитектура состояния login_captcha рассмотрено в 3 пункте. Архитектура состояния login2.1 рассмотрено в 5 пункте. Архитектура состояния attach рассмотрено в 7 пункте.

7. Схема работы модуля привязки аккаунта социальной сети

Цветом выделены состояния которые передают управление виджету.



ЗАМЕТКА

Из всех визуальных состояний есть переходы "cancel" и "fail", завершающие цепочку. Не представлено на схеме для сокращения. Из-за невозможности редактора создавать блоки на схеме с одним заголовком, состояние otp разделено на otp1.1, otp1.2. В действительности все они являются реализацией одного функционала. Архитектура состояния otp рассмотрено в 10 пункте.

8. Схема работы модуля проверки данных входа через социальную сеть

Цветом выделены состояния которые передают управление виджету.

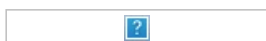


ЗАМЕТКА

Из всех визуальных состояний есть переходы "cancel" и "fail", завершающие цепочку. Не представлено на схеме для сокращения.

9. Схема работы модуля проверки OTP при входе через социальную сеть

Цветом выделены состояния которые передают управление виджету.



ЗАМЕТКА

Из всех визуальных состояний есть переходы "cancel" и "fail", завершающие цепочку. Не представлено на схеме для сокращения. Архитектура состояния otp рассмотрено во 10 пункте.

10. Схема работы модуля подтверждения операции одноразовым паролем

Цветом выделены состояния которые передают управление виджету.



11. Точка входа

Все цепочки аутентификации должны начинаться с OAuth 2.0 endpoint-a.

Например:

- OAuth 2.0 `http://<sso_host>/sso/oauth2/authorize?response_type=code&realm=/customer&client_id=ocb_lk&redirect_uri=http://<sso_host>/sso/secure/lk.jsp&scope=sso/oauth2/authotioType`

12. Блокировки

После `com.rooxteam.uidm.captcha.limit` (default 2) неверных попыток ввода пароля для одного и того же логина требуется вводить CAPTCHA. После `com.rooxteam.uidm.block.limit` (default 10) неверных попыток ввода пароля или CAPTCHA для одного и того же логина происходит временная блокировка по логину на `com.rooxteam.uidm.block.seconds` секунд (default 3600 с = 1 час). `com.rooxteam.uidm.block.limit` включает в себя `com.rooxteam.uidm.captcha.limit`, то есть при значениях по умолчанию достаточно 2 неверных вводов пароля и 8 неверных ввода CAPTCHA чтобы заблокироваться. После `com.rooxteam.uidm.auth.fail.ip.block.limit` (default 240) неверных попыток ввода логина/пароля/CAPTCHA с одного и того же IP-адреса в течение `com.rooxteam.uidm.block.ip.count.period.seconds` секунд (default 86400 с = 24 часа) происходит блокировка этого адреса на `com.rooxteam.uidm.block.ip.seconds` секунд (default 86400 с = 24 часа)

13. API для виджета логина.

13.1. Общий механизм взаимодействия

Для аутентификации пользователя по протоколу OAuth будет происходить запрос со стороннего ресурса (например Личный Кабинет) на WebSSO, где взаимодействие WebSSO сервера с виджетом происходит путем встраивания кода виджета (HTML код) в WebSSO страницу через запрос на WRS с передачей параметров встраивания. А так же начальных данных (параметр `up_inputData`) в виде JSON объекта для отрисовки виджета.

Пример начальных данных:

```
{
  "form": {
    "errors": [],
    "name": "loginForm",
    "fields": {
      "username": {
        "constraints": [
          {
            "name": "NotNull"
          },
          {
            "name": "Size",
            "attributes": {
              "min": 10,
              "max": 25
            }
          },
          {
            "name": "FilteredSize",
            "attributes": {
              "skip": "([^9]+)|([^0-9])",
              "message": "symbols {skip} should be filtered out, and resulting string should have length between {min} and {max}",
              "min": 10,
              "max": 10
            }
          }
        ]
      }
    }
  }
}
```

```
    }
  ],
  {
    "password": {
      "constraints": [
        {
          "name": "Size",
          "attributes": {
            "min": 4,
            "max": 1024
          }
        }
      ],
      "name": "NotNull"
    }
  ]
},
{
  "view": {
    "logoutReason": "idle_timeout",
    "isBlocked": false,
    "utm": {}
  },
  "serverUrl": "https://sso.rooxteam.com/sso/auth/websso",
  "step": "auth_form",
  "execution": "dec7dacf-as2d-4345-aae5-d08410cb8f95_H4sIAAAAAAAAAAN1WTWwcRRYu27GdxE5wEnYF..."
}
```

имя	обязательный	описание	ограничения (constraints)
serverUrl	да	URL для следующего запроса	валидный абсолютный URL
execution	да	Идентификатор предсессии аутентификации	
step	да	Код состояния	всегда auth_form
view	да	Дополнительные данные текущего шага	
view.msisdn	да	Номер телефона, на который будет отправлен OTP SMS	11 цифр, первая всегда 7
view.otpCodeAvailableAttempts	да	Кол-во попыток ввода OTP кода	2 цифры
view.nextOtpCodePeriod	да	Время в секундах до наступления возможности отправки нового OTP кода	секунды
view.isBlocked	да	Признак блокировки пользователя	boolean
view.blockedFor	нет	Время до разблокировки в секундах, не передается если	long

view.isBlocked == false

view.logoutReason	нет	Причина логаута пользователя (см.ниже)	long
view.utm	нет	Содержит GET-параметры вида "utm_", переданные в исходном запросе. Используется для передачи UTM-контекста в цепочке редиректов	
form	нет	Если есть параметр form, значит надо будет отрисовать форму с полями (fields) внутри объекта	
form.fields	нет	Список заполняемых данных пользователем	
fields.username	да	Номер телефона	
fields.password	да	Пароль	
form.constraints	нет	Список ограничений на поле	
constraints.name	нет	Название ограничения поля формы, обязательно для каждого ограничения	
constraints.attributes	нет	Атрибуты ограничения формы	
constraints.attributes.regexp	нет	Регулярное выражение (обязательно для ограничения Pattern)	
constraints.attributes.flags	нет	Флаги регулярного выражения (обязательно для ограничения Pattern)	
constraints.attributes.min	нет	Минимальное значение длины поля (обязательно для ограничения Size)	
constraints.attributes.max	нет	Максимальное значение длины поля (обязательно для ограничения Size)	
form.errors[].message	нет	Сообщение об ошибке валидации	
form.errors[].field	нет	Поле в котором найдена ошибка	

13.1.1. Logout Reason

В первоначальных данных для виджета может быть указан параметр `logoutReason`. В этом случае нужно отобразить пользователю сообщение о причине логута и попадания на виджет логина.

Возможные значения: 1. `session_timeout` - Истечение времени сессии 2. `idle_timeout` - Бездействие пользователя в течение определенного времени

13.2. Вход в ЛК

Для аутентификации по OAuth протоколу пользователь будет перенаправлен на страницу входа, где будет отрисован виджет с формой для ввода логина и пароля.

13.3. Форматы запросов и ответов

13.3.1. Формат запроса на смену состояния

```
POST <sso_host>/sso/auth/webssso
Accept: application/json
Content-Type: application/x-www-form-urlencoded
```

ВАЖНО

Параметры обязательные для всех запросов

```
execution: <executionId>
_eventId: <eventId>
```

- `<sso_host>` - базовый адрес сервера SSO, например `sso.rooxteam.com`
- `<executionId>` - брать из переданного JSON объекта, параметр `execution`
- `<eventId>` - идентификатор действия, определяется отдельно для каждого состояния (для `captcha_auth_form` нужно передавать `next`)
 - для запроса новой "капчи" на `captcha_auth_form` - `<eventId>` должен быть равен `updateCaptcha`

13.3.2. Состояние `auth_form`

Состояние используется для отображения формы входа через логин пароль. Форма отображается в двух сценариях:

- как стартовый экран всего виджета, тогда она используется для входа через логин-пароль
- как шаг сценария создания привязки с соцсетями, тогда она используется для подтверждения пользователя

Принято решение для этих двух целей использовать одну и ту же форму. Признаком, отличающим сценарии является view-переменная `socialNetworkId`, которая имеет значение - id соцсети (`vkontakte`, `odnoklassniki`) если форма отображается в сценарии создания привязки, и отсутствует (или принимает значение `null`, что одно и то же), если форма отображается в сценарии входа через логин-пароль.

Входные данные (от сервера к виджету)

Таблица 1. Данные, передающиеся во всех сценариях

имя	обязательный	описание	констрейнты
vkontakteAppld	да	Идентификатор приложения Вконтакте	строка
odnoklassnikiAppld	да	Идентификатор приложения в Одноклассниках	строка
odnoklassnikiRedirectUri	да	Один из зарегистрированных redirect uri для приложения в сети Одноклассники.	строка
vkontakteRequestScopesAsArray	нет	Скоупы для приложения Вконтакте	массив строк
odnoklassnikiRequestScopesAsArray	нет	Скоупы для приложения в Одноклассниках	массив строк
avatarUrl	нет	Адрес фотографии аккаунта соц. сети для привязки	
firstName	нет	Имя пользователя аккаунта соц. сети для привязки	
fullName	нет	Полное имя пользователя аккаунта соц. сети для привязки	
socialNetworkId	да	Идентификатор соц. сети (vkontakte, odnoklassniki) для привязки	

Таблица 2. Данные, передающиеся только в сценарии создания привязки

имя	обязательный	описание	констрейнты
avatarUrl	нет	Адрес фотографии аккаунта соц. сети для привязки	
firstName	нет	Имя пользователя аккаунта соц. сети для привязки	
fullName	нет	Полное имя пользователя аккаунта соц. сети для привязки	
socialNetworkId	да	Идентификатор соц. сети (vkontakte, odnoklassniki) для привязки	

```

{
  "form": {
    "errors": [
      {
        "message": "size must be not less than 10",
        "field": "username"
      }
    ],
    "name": "loginForm",
    "fields": {
      "username": {
        "constraints": [
          {
            "name": "NotNull"
          },
          {
            "name": "Size",
            "attributes": {
              "min": 10,
              "max": 25
            }
          },
          {
            "name": "FilteredSize",
            "attributes": {
              "skip": "([^9]*)|([^\d])",
              "min": 10,
              "max": 10
            }
          }
        ]
      },
      "password": {
        "constraints": [
          {
            "name": "Size",
            "attributes": {
              "min": 4,
              "max": 1024
            }
          },
          {
            "name": "NotNull"
          }
        ]
      }
    }
  },
  "view": {
    "isBlocked": false,
    "vkontakteAppId": "1000121",
    "vkontakteRequestScopesAsArray": [
      "1",
      "2"
    ]
  },
  "serverUrl": "https://sso.rooxteam.com/sso/auth/websso",
  "step": "auth_form",
  "execution": "dec7dacf-as2d-4345-aae5-d08410cb8f95_H4sIAAAAAAAAAAN1WTWwcRRYu27GdxE5wEnYF..."
}

```

```
}
```

Пример с ошибкой аутентификации

```
{
  "form": {
    "errors": [
      {
        "message": "invalid_credentials"
      }
    ],
    "name": "loginForm",
    "fields": {
      "username": {
        "constraints": [
          {
            "name": "NotNull"
          },
          {
            "name": "Size",
            "attributes": {
              "min": 10,
              "max": 25
            }
          },
          {
            "name": "FilteredSize",
            "attributes": {
              "skip": "([^9]*)|([^\d])",
              "min": 10,
              "max": 10
            }
          }
        ]
      },
      "password": {
        "constraints": [
          {
            "name": "Size",
            "attributes": {
              "min": 4,
              "max": 1024
            }
          },
          {
            "name": "NotNull"
          }
        ]
      }
    }
  },
  "view": {
    "isBlocked": false
  },
  "serverUrl": "https://sso.rooxteam.com/sso/auth/websso",
  "step": "auth_form",
  "execution": "dec7dacf-as2d-4345-aae5-d08410cb8f95_H4sIAAAAAAAAAAN1WTWwcRRYu27GdxE5wEnYF..."
}
```

```
{
  "form": {
    "errors": [
      {
        "message": "user_blocked"
      }
    ],
    "name": "loginForm",
    "fields": {
      "username": {
        "constraints": [
          {
            "name": "NotNull"
          },
          {
            "name": "Size",
            "attributes": {
              "min": 10,
              "max": 25
            }
          },
          {
            "name": "FilteredSize",
            "attributes": {
              "skip": "([^9]*)|([^\d])",
              "min": 10,
              "max": 10
            }
          }
        ]
      },
      "password": {
        "constraints": [
          {
            "name": "Size",
            "attributes": {
              "min": 4,
              "max": 1024
            }
          },
          {
            "name": "NotNull"
          }
        ]
      }
    }
  },
  "view": {
    "blockedFor": 123456,
    "isBlocked": true
  },
  "serverUrl": "https://sso.rooxteam.com/sso/auth/websso",
  "step": "auth_form",
  "execution": "dec7dacf-as2d-4345-aae5-d08410cb8f95_H4sIAAAAAAAAAAN1WTWwcRRYu27GdxE5wEnYF..."
}
```

имя	обязательный	описание	констрейнты
_eventId	да	Следующий сценарий: vkontakte, odnoklassniki - вход через соответствующую соцсеть; next - для входа через логин-пароль;	
username	да для сценария входа через логин-пароль	Логин пользователя	набор из символов, минимум 1, максимум 1024
password	да для сценария входа через логин-пароль	Пароль пользователя	1-1024 символа
socialData	да для сценария входа через соцсети	Информация от API кнопки входа через соцсети. Описание алгоритма формирования socialData	
loginType	да для сценария входа через соцсети	Способ входа через соцсеть	всегда строка "WEBSITE"

13.3.3. Состояние srp_auth

Состояние проводит аутентификации по протоколу SRP.

Клнфигурация:

И для клиента и для сервера устанавливаются общие константы:

Таблица 3. Конфигурация SRP для клиента (приведены параметыр виджета логина, в других клиентах могут использоваться свои названия)

Параметр	Описание	Пример
com.rooxteam.widgets.srp.n	"безопасное простое" = $2 \cdot q + 1$ простое, где q тоже простое число. Число в 10-чной системе.	11144252...13345603
com.rooxteam.widgets.srp.g	генератор по модулю $N = \varphi(N)$ для любого $0 < X < N$ существует и единственный x такой, что $g^x \% N = X$.	2
com.rooxteam.widgets.srp.h	односторонняя функция хеширования, достаточно безопасная на текущий момент	SHA-1
com.rooxteam.widgets.srp.salt_bytes	размер соли в байтах	16

Таблица 4. Конфигурация SRP для сервера

Параметр	Описание	Пример
com.rooxteam.sso.srp.n	"безопасное простое" = $2 \cdot q + 1$ простое, где q тоже простое число. Число в 10-чной системе.	11144252...13345603
com.rooxteam.sso.srp.g	генератор по модулю $N = \varphi(N)$ для любого $0 < X < N$ существует и единственный x такой, что $g^x \% N = X$.	2
com.rooxteam.sso.srp.h	односторонняя функция хеширования, достаточно безопасная на текущий момент	SHA-1
com.rooxteam.sso.auth_with_challenge	(boolean) включить аутентификацию через SRP-Challenge	true

Сценарий аутентификации:

Таблица 5. Сценарий работы SRP

Шаг	Клиент	Сервер
1	- Получает от пользователя логин login и пароль password - Отправляет серверу login	- Вытаскивает из базы соль salt и верификатор verifier по логину login - Вычисляет переменную $B = \text{step1}(\text{login}, \text{salt}, \text{verifier})$ - Возвращает клиенту salt и B

2	- Вычисляет и передаёт серверу A и M1	- Проверяет, можно ли доверять A и M1 - Успешно завершает аутентификацию - Вычисляет и отправляет клиенту M2
3	- Проверяет, можно ли доверять M2	

Шаг 1

Выходные данные (от клиента к серверу)

имя	обязательный	описание	констрейнты
_eventId	да	next	
username	да	Логин пользователя	набор из символов, минимум 1, максимум 1024
password	да	Требуется любое непустое значение	набор из символов, минимум 1, максимум 1024

Входные данные (от сервера к клиенту)

имя	обязательный	описание	констрейнты
execution	да	Идентификатор предсессии аутентификации	
view.salt	да	Соль	Число в 16-чном формате
view.B	да	Переменная B	Число в 16-чном формате

Шаг 2

Выходные данные (от клиента к серверу)

имя	обязательный	описание	констрейнты
_eventId	да	next	
A	да	Переменная A	Число в 16-чном формате
M1	да	Переменная M1	Число в 16-чном формате

Входные данные (от сервера к клиенту)

имя	обязательный	описание	констрейнты
execution	да	Идентификатор предсессии	

execution	да	идентификатор сессии аутентификации	
view.M2	да	Переменная M2	Число в 16-чном формате

13.3.4. Состояние captcha_auth_form

Состояние может отрисовываться как в сценарии входа по логину-паролю, так и в сценарии создания привязки с соцсетями.

Входные данные (от сервера к виджету)

имя	обязательный	описание	констрейнты
serverUrl	да	URL для следующего запроса	
execution	да	Идентификатор предсессии аутентификации	
step	да	Код состояния	всегда captcha_auth_form
view.recaptchaSiteKey	да	Ключ для формирования response для последующей валидации ReCaptcha	
form.errors[].message	нет	Сообщение об ошибке валидации	
form.errors[].field	нет	Поле в котором найдена ошибка	

Таблица 6. Данные, передающиеся только в сценарии создания привязки

имя	обязательный	описание	констрейнты
avatarUrl	нет	Адрес фотографии аккаунта соц. сети для привязки	
firstName	нет	Имя пользователя аккаунта соц. сети для привязки	
fullName	нет	Полное имя пользователя аккаунта соц. сети для привязки	
socialNetworkId	да	Идентификатор соц. сети (vkontakte, odnoklassniki) для привязки	

Пример

```
{
  "form": {
    "errors": [
      {
        "field": "captchaCode",
```

```
    "message": "invalid_captcha"
  },
  {
    "name": "captchaLoginForm",
    "fields": {
      "username": {
        "constraints": [
          {
            "name": "NotNull"
          },
          {
            "name": "Size",
            "attributes": {
              "min": 10,
              "max": 25
            }
          },
          {
            "name": "FilteredSize",
            "attributes": {
              "skip": "([^9]*)|([^\d])",
              "min": 10,
              "max": 10
            }
          }
        ]
      },
      "password": {
        "constraints": [
          {
            "name": "Size",
            "attributes": {
              "min": 4,
              "max": 1024
            }
          },
          {
            "name": "NotNull"
          }
        ]
      }
    }
  },
  {
    "view": {
      "recaptchaSiteKey": "6Lf5CQkUAAAAA3shbFPdp0b7HEWJivKgKFuAcNtR"
    },
    "serverUrl": "https://sso.rooxteam.com/sso/auth/websso",
    "step": "captcha_auth_form",
    "execution": "dec7dacf-as2d-4345-aae5-d08410cb8f95_H4sIAAAAAAAAAAN1WTWwcRRYu27GdxE5wEnYF..."
  }
}
```

Выходные данные (от виджета к серверу)

имя	обязательный	описание	констрейнты
username	да	Номер телефона	набор из символов, минимум 10, максимум 25
password	да	Пароль	4-1024 символа

gRecaptchaResponse	да	Response сформированный рекапчей для валидации
--------------------	----	---

13.3.5. Состояние show_attach_form

Входные данные (от сервера к виджету)

имя	обязательный	описание	констрейнты
step	да	Код состояния	всегда attach_form
avatarUrl	нет	Адрес фотографии аккаунта соц. сети для привязки	
firstName	нет	Имя пользователя аккаунта соц. сети	
fullName	нет	Полное имя пользователя аккаунта соц. сети	
socialNetworkId	да	Идентификатор соц. сети (vkontakte, odnoklassniki)	

Выходные данные (от виджета к серверу)

имя	обязательный	описание	констрейнты
_eventId	да	Следующий сценарий	"next" для продолжения или "cancel" для отмены

13.3.6. Состояние show_reattach_form

Входные данные (от сервера к виджету)

имя	обязательный	описание	констрейнты
step	да	Код состояния	всегда reattach_form
avatarUrl	нет	Адрес фотографии аккаунта соц. сети для привязки	
firstName	нет	Имя пользователя аккаунта соц. сети	
fullName	нет	Полное имя пользователя аккаунта соц. сети	
oldAvatarUrl	нет	Адрес фотографии аккаунта соц. сети для удаления привязки	

oldFullName	нет	Полное имя пользователя аккаунта соц. сети для удаления привязки
socialNetworkId	да	Идентификатор соц. сети (vkontakte, odnoklassniki)

Выходные данные (от виджета к серверу)

имя	обязательный	описание	констрейнты
_eventId	да	Следующий сценарий	"next" для продолжения или "cancel" для отмены

13.3.7. Состояние enter_otp_form

Входные данные (от сервера к виджету)

имя	обязательный	описание	констрейнты
serverUrl	да	URL для следующего запроса	
execution	да	Идентификатор предсессии аутентификации	
step	да	Код состояния	всегда enter_otp_form
expireOtpCodeTime	да	Время действия одноразового пароля в секундах	секунды
otpCodeAvailableAttempts	да	Кол-во попыток ввода OTP кода	
nextOtpCodePeriod	да	Время в секундах до наступления возможности запросить новое sms с OTP кодом	секунды
otpCodeNumber	нет	Номер отправленного смс сообщения	
form.errors[].message	нет	Сообщение об ошибке	

Выходные данные (от виджета к серверу)

имя	обязательный	описание	констрейнты
otpCode	да	Одноразовый пароль	Цифры от 0 до 9, длиной 4 символа

_eventId

да

Следующий сценарий

"send" для повторной отправки смс, "cancel" для отмены, "start" для продолжения

Пример с ошибкой валидации кода

```
{
  "form": {
    "errors": [
      {
        "field": "otpCode",
        "message": "invalid_otp"
      }
    ],
    "name": "otpForm",
    "fields": {
      "otpCode": {
        "constraints": [
          {
            "name": "NotNull"
          },
          {
            "name": "Size",
            "attributes": {
              "min": 4,
              "max": 4
            }
          },
          {
            "name": "Pattern",
            "attributes": {
              "regexp": "[0-9]+"
            }
          }
        ]
      }
    ]
  },
  "view": {
    "otpCodeAvailableAttempts": 2,
    "msisdn": "79876543210",
    "nextOtpCodePeriod": 120,
    "blockedFor": 0,
    "isBlocked": false
  },
  "serverUrl": "https://sso.rooxteam.com/sso/auth/websso",
  "step": "enter_otp_form",
  "execution": "dec7dacf-as2d-4345-aae5-d08410cb8f95_H4sIAAAAAAAAAAN1WTWwcRRYu27GdxE5wEnYF..."
}
```

Пример с превышением кол-ва попыток

```
{
  "form": {
```

```

"errors": [
  {
    "field": "otpCode",
    "message": "too_many_wrong_code"
  }
],
"name": "otpForm",
"fields": {
  "otpCode": {
    "constraints": [
      {
        "name": "NotNull"
      },
      {
        "name": "Size",
        "attributes": {
          "min": 4,
          "max": 4
        }
      },
      {
        "name": "Pattern",
        "attributes": {
          "regexp": "[0-9]+"
        }
      }
    ]
  }
}
},
"view": {
  "otpCodeAvailableAttempts": 2,
  "msisdn": "79876543210",
  "nextOtpCodePeriod": 120,
  "blockedFor": 0,
  "isBlocked": false
},
"serverUrl": "https://sso.rooxteam.com/sso/auth/websso",
"step": "enter_otp_form",
"execution": "dec7dacf-as2d-4345-aae5-d08410cb8f95_H4sIAAAAAAAAAAN1WTWwcRRYu27GdxE5wEnYF..."
}

```

Пример с просроченным кодом

```

{
  "form": {
    "errors": [
      {
        "field": "otpCode",
        "message": "otp_expired"
      }
    ],
    "name": "otpForm",
    "fields": {
      "otpCode": {
        "constraints": [
          {
            "name": "NotNull"
          },

```

```

{
  "name": "Size",
  "attributes": {
    "min": 4,
    "max": 4
  }
},
{
  "name": "Pattern",
  "attributes": {
    "regexp": "[0-9]+"
  }
}
]
}
},
"view": {
  "otpCodeAvailableAttempts": 2,
  "msisdn": "79876543210",
  "nextOtpCodePeriod": 120,
  "blockedFor": 0,
  "isBlocked": false
},
"serverUrl": "https://sso.rooxteam.com/sso/auth/websso",
"step": "enter_otp_form",
"execution": "dec7dacf-as2d-4345-aae5-d08410cb8f95_H4sIAAAAAAAAAAN1WTWwcRRYu27GdxE5wEnYF..."
}

```

13.3.8. Успешное завершение цепочки

Входные данные (от сервера к виджету)

имя	обязательный	описание	констрейнты
step	да	Код состояния	всегда redirect
location	да	URL для редиректа	

Пример

```

{
  "step": "redirect",
  "location": "/sso/auth/complete"
}

```

Выходные данные (от виджета к серверу)

Надо сделать редирект на <location>, если он есть и не пустой. Переход на него проставит куку iPlanetAuthToken с временным JWT токеном. После чего пользователь будет перенаправлен на /UI/Login, где ему проставят WebSSO куку и перенаправят на goto.

13.4. Возврат к защищаемому сервису

В виджете может быть кнопка или ссылка, по которой пользователь сможет вернуться на страницу защищаемого сервиса, который инициировал аутентификацию.

ВАЖНО

Информация о том, отображать ли кнопку, автоматически вызывать cancel при блокировке или просто отображать сообщение, будет пробрасываться в виджет отдельным параметром, детали будут утверждены позже

По клику на эту кнопку или ссылку должен выполняться запрос на сервер следующего вида:

имя	обязательный	описание	констрейнты
_eventId	да	Идентификатор действия	cancel

Сервер должен вернуть ответ аналогичный успешному завершению

```
{
  "step": "redirect",
  "location": "/sso/auth/complete"
}
```

Необходимо выполнить редирект пользователя на указанный URL.

13.5. Возможные ограничения на поля формы

название	описание	атрибуты	описание атрибута
NotNull	Обязательность поля		
Size	Длина строкового параметра	min	Минимальная длина
		max	Максимальная длина
Pattern	Regex для строкового параметра	regex	Регулярное выражение
Min	Минимальное значение целого числового параметра	value	Минимальное значение
Max	Максимальное значение целого числового параметра	value	Максимальное значение
DecimalMin	Минимальное значение десятичного числового параметра	value	Минимальное значение
DecimalMax	Максимальное значение десятичного числового параметра	value	Максимальное значение

	десятичного числового параметра		
FilteredSize	Ограничения на номер телефона	min	Минимальная длина
		max	Максимальная длина
		skip	Регулярное выражение для фильтрации. Перед отправкой нужно удалять все, что ему удовлетворяет. Например, происходит очищение введенных данных от всех символов кроме цифр, дальше удалять все символы слева до первой девятки (не включая).

13.6. Возможные коды ошибок

код	описание	комментарий
ip_blocked	IP адрес, с которого пользователь обращается в WebSSO, заблокирован.	user_blocked
Учетная запись пользователя заблокирована.	system_blocked	Заблокировано приложение (client_id)
	invalid_captcha	Введенная пользователем CAPTCHA неверная.
need_captcha	Для продолжения сценария пользователь должен ввести CAPTCHA.	invalid_credentials
Пользователь ввел неправильные данные учетной записи и не был аутентифицирован.	expired_password	Время действия пароля, введенного пользователем, истекло. Пользователь не был аутентифицирован.
may not be null	Поле не может быть пустым	Ошибка валидации формы, смотреть детали в массиве ошибок на поля формы
size must be between {min} and {max}	Значение должно быть в заданном диапазоне	Ошибка валидации формы, смотреть детали в массиве ошибок на поля формы
required on {field name}	Обязательное поле	Ошибка валидации формы, смотреть детали в массиве ошибок на поля формы
must match "{regexp}"	Соответствие регулярному выражению	Ошибка валидации формы, смотреть детали в массиве ошибок на поля формы
too_many_wrong_code	Превышен лимит попыток ввода OTP кода	
too_many_sms	Превышен лимит попыток заказа нового	

ОТР кода		
too_many_sends	То же, что и too_many_sms, но для EMAIL и нового универсального 2FA	
error_sending_otp	Транспортная ошибка доставки ОТР	
otp-error	Общая неклассифицируемая ошибка обработки ОТР	
login-by-otp-disabled	Системе в client_id запрещено проходить сценарий 2FA	
invalid_otp	Введен неверный ОТР код	
validate-otp-fail	Введенный ОТР неверный.	otp_expired
Время действия введенного ОТР истекло.	otp-expired	Время действия введенного ОТР истекло.
system_error	Общая неклассифицируемая ошибка UIDM	
error	Пользователь не был аутентифицирован в результате неожиданного ответа от внешней системы.	otp-error
Сценарий ввода ОТР завершился неуспешно.	external-api-error	Выполнение сценария невозможно, поскольку получен неожиданный ответ от внешней системы или внешняя система не доступна.
external-system-bad-response	Выполнение сценария невозможно, поскольку получен неожиданный ответ от внешней системы или внешняя система не доступна.	msisdn-is-not-b2b
Введенный msisdn не принадлежит реалму b2b.	too_many_attempts	Превышено число допустимых попыток изменения пароля.
user-is-not-allowed	Данному пользователю запрещено продолжать данный сценарий.	error_password_change
Сценарий изменения пароля не был успешно завершен. Пароль не был изменен.	no_email_found	Пользователь не найден или у пользователя не существует email.
msisdn-not-exists	Пользователя с заданным msisdn не существует.	principal-not-exists
Пользователь не существует.	no-principal-found	Пользователь не существует.
user-not-found	Пользователь не существует.	login-exists
Пароль уже использован.	login_already_exists	Пароль уже использован. login уже существует

невозможно зарегистрировать пользователя с заданным login, поскольку пользователь с заданным login уже существует.	login_already_exists	невозможно изменить login пользователя на заданный, поскольку пользователь с заданным login уже существует.
email-exists	Пользователь с заданным email уже существует.	user-exists
Невозможно зарегистрировать пользователя с заданными msisdn или email или login, поскольку пользователь с такими параметрами уже существует.	email_verification_failed	Невозможно подтвердить email.

13.7. Дополнительные условия

Для работы цепочки аутентификации нужно при каждом запросе передавать cookie JSESSIONID. Если cookie отсутствует или истекло время сессии, вернется ошибка HTTP 400 вида:

```
{
  "error": {
    "code": 400,
    "message": "No flow execution could be found with key 'e2s2'"
  }
}
```

Если в запросе указан execution с валидным идентификатором e но невалидным шагом s, вернется ошибка HTTP 400 вида:

```
{
  "error": {
    "code": 400,
    "message": "No flow execution snapshot could be found with id '4'; perhaps the snapshot has been removed?"
  }
}
```

13.8. Алгоритм формирования socialData

Параметр socialData это данные от JS API соцсети, сериализованные как url-параметры и закодированные в base64.

13.8.1. Пример функции сериализации

```
function encodeSocialParams(params) {
  var result = "";
  for (var param in params) {
    result += param + "=" + params[param] + "&"
  }
}
```

```
result = result.substr(0, result.length - 1);
```

```
return encodeURIComponent(result);
```

```
}
```

13.8.2. Пример получения socialData для Vkontakte

```
function vkontakteAuth(response) {  
  if (response.session) {  
    var socialData = base64.encode(encodeSocialParams(response.session), false, true);  
    var loginType = 'WEBSITE';  
    $('#auth_method').val('vkontakte');  
    $('#_eventId').val('vkontakte');  
    $('#socialData').val(socialData);  
    $('#loginType').val(loginType);  
    $('#authForm').attr('target', '_top');  
    $('#authForm').submit();  
  }  
}
```

```
}
```

```
function loginVK() {  
  VK.Auth.login(vkontakteAuth);
```

```
}
```

13.8.3. Пример получения socialData для Одноклассники

```
function loginOK() {  
  //функция должна открыть окно логина в одноклассники (state можно не передавать)  
  // https://connect.ok.ru/oauth/authorize?client_id={clientId}&scope={  
{scope}&response_type={{response_type}}&redirect_uri={redirectUri}&layout={layout}&state=  
{state}  
  //и обработать  
  //302 редирект в этом окне на зарегистрированный для приложения redirect uri  
{com.rooxteam.sso.endpoint}/ok_callback.jsp/?code={....}&state={...}  
  // после подтверждения юзером разрешения на доступ SSO plugin к профилю  
  //закодировать параметры code и state (если есть) в base64 для получения socialData  
  //закрыть окно логина в Одноклассники  
  // заполнить и отправить форму перехода #authForm  
    $('#auth_method').val('odnoklassniki');  
    $('#_eventId').val('odnoklassniki');  
    $('#socialData').val(socialData);  
    $('#loginType').val('WEBSITE');  
    $('#authForm').attr('target', '_top');  
    $('#authForm').submit();  
}
```

```
}
```

