Правила информационного взаимодействия между Банком и Предприятием

Вход по Сбербанк ID

Содержание

Общая информация

Данный протокол взаимодействия разработан на основе стандарта Open ID Connect.

Реализовано 7 сценариев взаимодействия:

1. Web to Web (клиент находится в браузере);
2. mWeb to App (клиент в браузере на мобильном устройстве входит на сайт Партнера через мобильное приложение Сбербанк Онлайн, если оно установлено);
3. App to webview (клиент бесшовно переходит из мобильного приложения Сбербанк Онлайн на сайт Партнера);
4. Web to Web SSO (клиент бесшовно переходит из веб Сбербанк Онлайн на сайт Партнера);
5. App to App (сценарий, когда у клиента установлено мобильное приложение партнера и приложение Сбербанк Онлайн);
6. App to App SSO (клиент бесшовно переходит из мобильного приложения Сбербанк Онлайн в приложение Партнера);
7. App to Web (сценарий, когда у клиента установлено мобильное приложение партнера, но не установлено мобильного приложения Сбербанк Онлайн).

Предусловия:

1. Партнер уже получил от Банка Client ID и Client Secret, зарегистрировав свое приложение в системах Банка. Инструкция по получению Client ID и Client Secret доступна в разделе "Подтверждение регистрации приложения в Личном кабинете".
2. Партнер получил от Банка сертификаты безопасности, настроил свое серверное ПО. Если вы еще не получили сертификаты безопасности от Банка, Вы можете узнать как это сделать в разделе "Шаги подключения".

Терминология:

Аббревиатура/ Сокращение	**Определение**
Пользователь	Пользователь Сбер ID
МП Банка	Мобильное приложение Сбербанк Онлайн
Back Банка	Бэковое программное обеспечение Банка
МП Партнера	Программное обеспечение Предприятия, предназначенное для работы на смартфонах, планшетах и других мобильных устройствах
Back Партнера	Бэковое программное обеспечение Предприятия

Общая схема взаимодействия

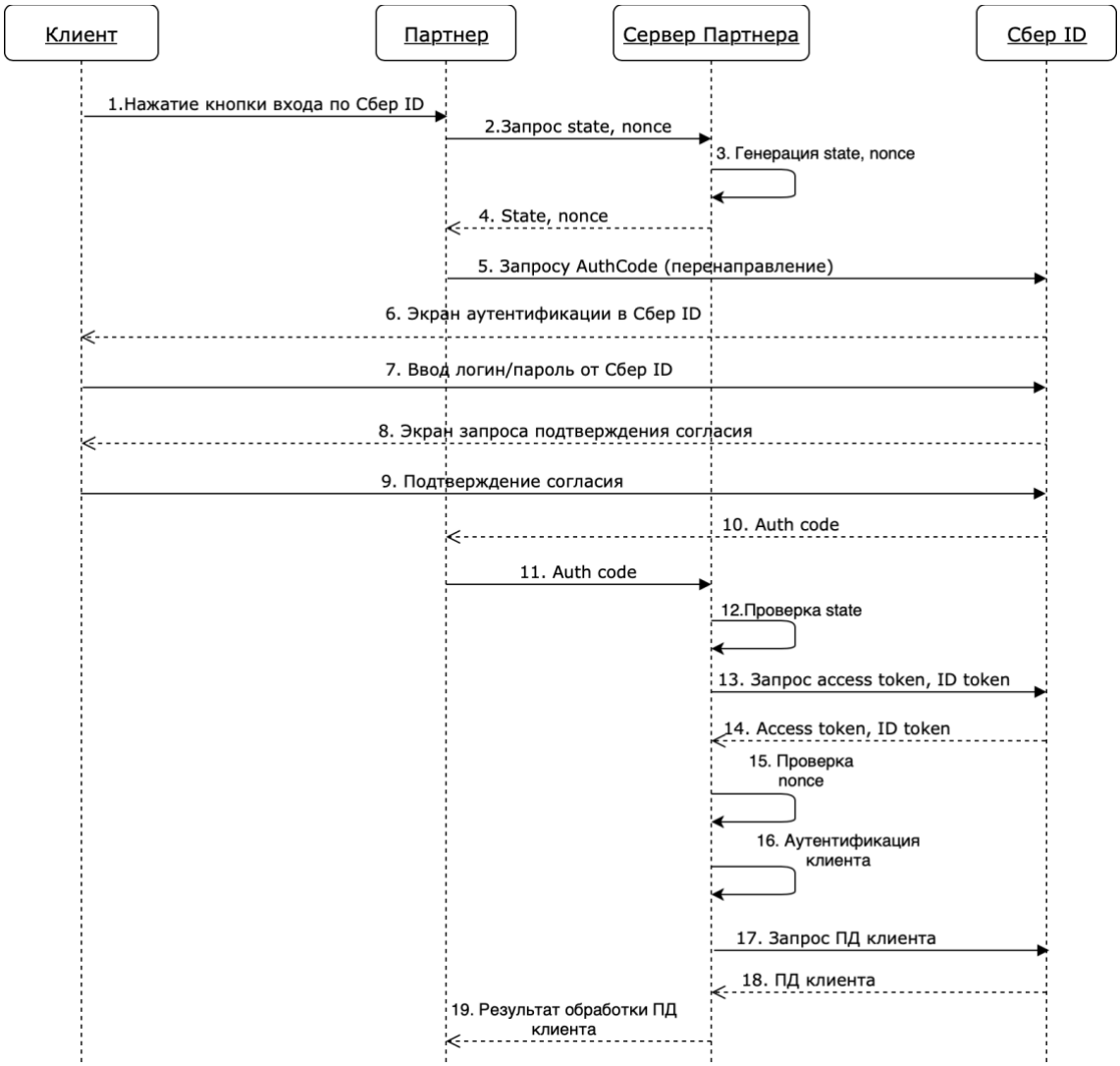


Таблица 1. Описание шагов

№ п/п	Описание
1	Клиент нажимает кнопку входа по Сбер ID в мобильном приложении или на сайте партнера.
2	Мобильное приложение/сайт партнера делает запрос на свой сервер для получения nonce и state.
3	Сервер партнера генерирует значения nonce и state и сохраняет его в текущем сеансе пользователя (при желании можно сохранить в этой сессии адрес текущей страницы, на которой клиент нажал кнопку для входа и при успешной авторизации вернуть клиента на эту страницу).
4	Сервер партнера возвращает в мобильное приложение/сайт партнера значение nonce, state и идентификатор сеанса пользователя. Если запрос на код авторизации инициирован из мобильного приложения партнера, то на этом шаге передаются параметры PKCE -code_challenge и code_challenge_method. Партнер привязывает эти значения к открытой сессии своего мобильного приложения. Подробнее см. 1.1.2.1. Параметры запроса
5	Мобильное приложение (или web-сайт) партнера сохраняет у себя идентификатор сеанса и делает запрос кода авторизации (мобильное приложение или web Сбербанка) с параметрами state, nonce, client_id, redirect_uri (и параметрами PKCE в случае, если запрос инициирует мобильное приложение партнера).
6	Отображение экрана аутентификации в Сбер ID.
7	Клиент аутентифицируется на сервере авторизации.
8	Отображение экрана подтверждения согласия (при отсутствии активного согласия).
9	Клиент подтверждает согласие (при отсутствии активного согласия).
10	Сервер авторизации выдает значение AuthCode и редиректит его на redirect_uri, включая в ответ значение state, которое пришло в запросе.
11	Мобильное приложение/web-сайт партнера, получив ответ, передает его на свой сервер в связке со значением текущего сеанса пользователя.
12	Сервер партнера *должен сверить значение state* из ответа с сохраненным в сеансе значением.
13	*Если значения совпадают*, то сервер партнера в рамках текущей сессии клиента делает запрос на Token Endpoint (СберAPI) для получения кода доступа (access token) - отправляет в запросе полученное значение AuthCode, client_id, client_secret и redirect_uri (то же самое uri, которое было отправлено в запросе на код авторизации). Если запрос на код авторизации был инициирован из мобильного приложения партнера, то необходимо в запросе на код доступа указать параметр PKCE (code_verifier).
14	Token Endpoint авторизует сервер партнера, проверяет AuthCode и выдает access token и ID_token.
15	Сервер партнера получив ID_token *должен проверить значение параметра nonce*, что полученное значение nonce равно значению параметра nonce, отправленного в запросе на аутентификацию.

16	Если значения совпадают сервер партнера может идентифицировать клиента по значению sub, полученного из ID_token.
17	Запрос профиля клиента (userInfo).
18	Передача профиля клиента.
19	Сервер партнера авторизует клиента у себя в сервисе и возвращает успешный ответ в мобильное приложение (зависит от бизнес-логики партнера).

1. Описание запросов

1.1. Запрос кода авторизации

На "Схеме взаимодействия" обозначен как "5. Запрос AuthCode (перенаправление)".

1.1.1. Веб-браузер

1.1.1.1. Параметры запроса

Ниже описано 4 сценария:

1. Аутентификация на веб-странице Партнера (Web to Web)

2. Аутентификация на веб-странице Партнера через мобильное приложение Сбербанк Онлайн (mWeb to App)

3. Аутентификация на веб-странице Партнера внутри мобильного приложения Сбербанк Онлайн (App to webview)

4. Бесшовный переход из веб Сбербанк Онлайн на сайт Партнера (Web to Web SSO)

1. Аутентификация на веб-странице Партнера (Web to Web)

Front-end партнера инициирует запрос на front-end банка на получение кода авторизации, направляя GET или POST запрос из браузера клиента.

> Страницу входа по Сбер ID можно открывать и в popup-окне. Рекомендуемый размер окна - 600x600 пикселей.

Пример запроса на получение кода авторизации:

```
GET  /CSAFront/oidc/authorize.do
Host: online.sberbank.ru
response_type=code
&client_type=PRIVATE
&scope=openid+name
&client_id=DA5278AC-A07F-C01A-B2D3-C231DBB2E20F
&state=af0ifjsldkj
&nonce=n-0S6_WzA2Mj
&redirect_uri=https%3A%2F%2Fclientresource.ru%2Fcb HTTP/1.1
```

Важно!

Для сценариев, в которых присутствует риск перехвата AuthCode необходимо использовать защиту PKCE (https://tools.ietf.org/html/rfc7636). По рекомендациям RFC 7636 для смягчения атак перехвата кода авторизации используется динамически создаваемое случайное значение - "code verifier". Данное значение должно быть уникальным для каждого запроса кода авторизации.

Требования к генерации значения code_verifier (см. также https://tools.ietf.org/html/rfc7636#section-4.1):

- Значение code_verifier - это высокоэнтропийная криптографическая случайная строка.
- Строка генерируется с использованием допустимых символов [AZ] / [az] / [0- 9] / "-" / "." / "_" / "~".
- Минимальная длина - 43 символа.
- Максимальная длина - 128 символов.

Один из возможных алгоритмов:

Партнер, используя подходящий генератор случайных чисел, создает последовательность длиной от 32 до 96 байт, которую затем кодирует способом base64url. Например, для 32-байтной последовательности [116, 24, 223, 180, 151, 153, 224, 37, 79, 250, 96, 125, 216, 173, 187, 186, 22, 212, 37, 77, 105, 214, 191, 240, 91, 88, 5, 88, 83, 132, 141, 121] кодирование base64url в результате даст code_verifier в виде “dBjftJeZ4CVP-mB92K27uhbUJU1p1r_wW1gFWFOEjXk” (см. https://tools.ietf.org/html/rfc7636#appendix-B).

Важно! Партнер для вызова аутентификации не должен использовать браузер, встроенный в приложение, который предоставляет приложению доступ к cookie, или содержимому веб-страниц.

При открытии веб-страницы Сбер ID в браузере, встроенном в приложение, будет отображена страница-заглушка, сообщающая клиенту о небезопасности входа.

Таблица 2. Описание полей запроса кода авторизации

№ п/п	Наименование заголовка/поля	Описание	Обязательность поля	Пример
1	response_type	Указывается равным code	Да	code
2	client_type	Указывается равным PRIVATE	Нет	PRIVATE
3	scope	Наименование групп данных, на которые подписана система партнера (выдается при регистрации системы в банке). Значение openid является обязательным и располагается на первой позиции. Разделитель – "+".	Да	openid+name+maindoc+email+mobile
4	client_id	Идентификатор системы партнера, полученный партнером в личном кабинете после регистрации приложения.	Да	DA5278AC-A07F-C01A-B2D3-C231DBB2E20F
5	state	Значение, включенное в запрос, которое также возвращается в ответе. Может быть строка любого контента. Для предотвращения подделки межсайтовых запросов используется генерируемое случайным образом уникальное значение.	Да	af0ifjsldkj
6	nonce	Значение, сгенерированное внешней АС для предотвращения атак повторения. Это значение обычно представляет собой случайную уникальную строку или глобальный уникальный идентификатор, которые можно использовать для определения источника запроса. Ограничение по длине значения составляет 64 символа.	Да	n-0S6_WzA2Mj
7	redirect_uri	Адрес страницы партнера, на которую будет перенаправлен клиент после успешной аутентификации в системе банка. Временное ограничение: недопустимы символы “;” и “=“.	Да	https://clientresource.ru/cb
8	code_challenge	Хэшированное значение секретного кода code_verifier партнера (генерацию code_verifier см. выше в блоке «Важно»). Хэширование выполняется методом, указанным в code_challenge_method, в нашем случае – всегда S256, поэтому code_challenge = BASE64URL-ENCODE(SHA256 (ASCII (code_verifier))) (см. https://tools.ietf.org/html/rfc7636#section-4.2). Например, для code_verifier = “dBjftJeZ4CVP-mB92K27uhbUJU1p1r_wW1gFWFOEjXk” выполнение хэширования SHA256 этого	Нет	E9Melhoa2OwvFrEMTJguCHaoeK1t8URWbuGJSstw-cM

№ п/п	Наименование заголовка/поля	Описание	Обязательность поля	Пример
		значения в результате выдаст последовательность байтов [19, 211, 30, 150, 26, 26, 216, 236, 47, 22, 177, 12, 76, 152, 46, 8, 118, 168, 120, 173, 109, 241, 68, 86, 110, 225, 137, 74, 203, 112, 249, 195], преобразование которой через Base64Url дает нам значение code_challenge = “E9Melhoa2OwvFrEMTJguCHaoeK1t8URWbuGJSstw-cM” (см. https://tools.ietf.org/html/rfc7636#appendix-B).		
9	code_challenge_method	Метод преобразования секретного кода code_verifier партнера. Допустимым значением является S256.	Нет	S256

2. Аутентификация на веб-странице Партнера через мобильное приложение Сбербанк Онлайн (mWeb to App)

Протокол взаимодействия поддерживает функционал универсальных ссылок. Если на устройстве пользователя установлено мобильное приложение Сбербанк Онлайн, то операционная система направит запрос авторизации в мобильное приложение, иначе страница аутентификации откроется во внешнем браузере.

Для поддержки сценария Партнеру необходимо:

1. Встроить скрипт, формирующий универсальную ссылку
2. Реализовать экран ошибки входа при несовпадении параметра state

Для минимизации риска перехвата AuthCode в данном сценарии рекомендуется использовать метод защиты PKCE.

Важно!

При вызове универсальной ссылки из браузера сценарий входа по Сбер ID через мобильное приложение в некоторых случаях может завершиться ошибкой - параметр state запросе и ответе не будут совпадать, т.к. сценарий начался и закончился в разных браузерах или нет возможности сохранить state локально (пример: приватный режим браузера; in-app браузер; все браузеры на iOS, кроме Safari).

Скрипт Сбер ID формирует универсальную ссылку в зависимости от браузера и режима просмотра с целью минимизации ошибочных сценариев описанных выше.

При обратном редиректе из МП СБОЛ (по redirect_uri) в браузере открывается новая вкладка. Для корректного продолжения сценария пользователя на сайте Партнера следует сохранять контекст сессии (например, в cookie) при нажатии кнопки "Войти по Сбер ID" и восстанавливать его при обратном редиректе.

Скрипт можно скачать здесь, в разделе "Скрипт для формирования универсальной ссылки".

Пример встраивания скрипта:

```
<body>
    <div class="navigator"></div>
    <div class="result">
        <a
href="https://online.sberbank.ru/CSAFront/oidc/authorize.do?response_type
=code&scope=openid+gender+snils&client_id=AAAABBBB-XXXX-YYYY-ZZZZ-
A12A618A4C3C&state=7bQfJTePBhlW&nonce=w9H8bh2yN4mQ&redirect_uri=https://e
xample.com/external_login">Войти по Сбер ID</a>
    </div>
 <!-- Начало скрипта -->
 <script
 src = "dist/sberid-deeplink.min.js"> </script>
 <script>
 (function () {
     function mySberidUniversallink(result) {
          console.log(result);
         /*{
              "isPrivate": true,
              "isUniversalLink": false,
              "os": "windows",
              "browser": "chrome",
              "link":
"https://online.sberbank.ru/CSAFront/oidc/authorize.do?response_type=code
&scope=openid+gender+snils&client_id=AAAABBBB-XXXX-YYYY-ZZZZ-
4C3C&state=7bQfJTePBhlW&nonce=w9H8bh2yN4mQ&redirect_uri=https://example.c
om/external_login",
              "deeplink":
"sberbankidlogin://sberbankid/sso?response_type=code&scope=openid+gender+
snils&client_id=AAAABBBB-XXXX-YYYY-ZZZZ-
4C3C&state=7bQfJTePBhlW&nonce=w9H8bh2yN4mQ&redirect_uri=https://example.c
om/external_login",
              "universalLinkUrl:
"https://online.sberbank.ru/CSAFront/oidc/sberbank_id/authorize.do",
              "defaultLinkUrl":
"https://online.sberbank.ru/CSAFront/oidc/authorize.do",
              "deeplinkUrl": "sberbankidlogin://sberbankid/sso",
             oidc: {
                  response_type: "code",
                client_type: "PRIVATE",
                client_id: "AAAABBBB-XXXX-YYYY-ZZZZ-A12A618A4C3C",
                state: "7bQfJTePBhlW",
                redirect_uri: "https://example.com/external_login",
                scope: "openid+gender+snils",
                nonce: "w9H8bh2yN4mQ",
                display: "popup",
                ext_redirect_uri:
"googlechrome://example.com/external_login"
              }
         }*/
      }
      try {
         var sberidUniversallink = new SberidUniversallink({
              params:
'response_type=code&scope=openid+gender+snils&client_id=AAAABBBB-XXXX-
YYYY-ZZZZ-
4C3C&state=7bQfJTePBhlW&nonce=w9H8bh2yN4mQ&redirect_uri=https://example.c
om/external_login',
              callback: mySberidUniversallink,
         });
      } catch (error) {
         console.log(error);
      }
 })();
 </script>
```

```
  <!-- Конец скрипта -->
</body>
```

В результате выполнения скрипта:

1. Обычная ссылка заменяется на универсальную в элементах, указанных в параметре **selector**, подставляется универсальная ссылка, сформированная из параметров **params**
2. В функцию обратного вызова передаются данные, позволяющие сформировать универсальную ссылку самостоятельно

Таблица 3. Описание параметров скрипта

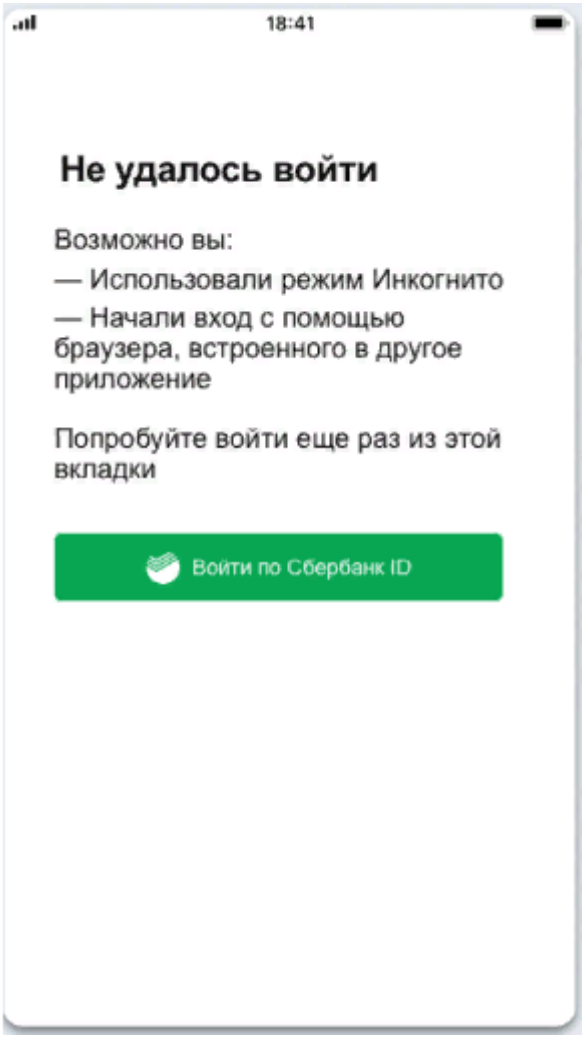
№ п/п	Наименование параметра	Описание	Обязательность
1	params	Параметры запроса кода авторизации	Нет
2	selector	HTML-элемент (id, стиль или вид), в котором нужно заменить ссылку запроса кода авторизации	Нет
3	callback	Функция обратного вызова, в которой Партнер может реализовать свою логику формирования универсальной ссылки. Актуально, если Партнер формирует ссылку на сервере.	Нет

Таблица 4. Описание объекта response

№ п/п	Наименование параметра	Описание	Пример
1	isPrivate	Режим браузера. true - приватный false - обычный	false
2	isUniversalLink	true - из данного браузера можно запускать универсальную ссылку false - универсальную ссылку запускать нельзя, т.к. это может привести к ошибке на стороне Партнера	true
3	os	Операционная система	ios
4	browser	Браузер	safari
5	link	Полная ссылка запроса кода авторизации	https://online.sberbank.ru/CSAFront/oidc/sberbank_id/authorize.do? response_type=code&scope=openid+gender+snils& client_id=AAAABBBB-XXXX-YYYY-ZZZZ-

№ п/п	Наименование параметра	Описание	Пример
			A12A618A4C3C&state=7bQfJTePBhlW& nonce=w9H8bh2yN4mQ&redirect_uri=https://example.com/external_login
6	universalLinkUrl	Универсальная ссылка без параметров	https://online.sberbank.ru/CSAFront/oidc/sberbank_id/authorize.do
7	defaultLinkUrl	Обычная ссылка без параметров	https://online.sberbank.ru/CSAFront/oidc/authorize.do
8	oidc	Объект. Содержит oidc параметры, переданные в скрипт при вызове.	
9	ext_redirect_uri	Параметр, вложенный в объект oidc. Deeplink для вызова браузера в котором находился клиент при нажатии на кнопку "Войти по Сбер ID" и открытия в нем ссылки, указанной в redirect_uri. Указывается только если os=ios	googlechrome://example.com/external_login
10	package	Параметр, вложенный в объект oidc. Deeplink для вызова браузера в котором находился клиент при нажатии на кнопку "Войти по Сбер ID" и открытия в нем ссылки, указанной в redirect_uri. Указывается только если os=android	googlechrome://example.com/external_login

Пример экрана ошибки (на стороне Партнера), в случае несовпадения параметра state:



Кнопка "Войти по Сбер ID" в этом случае должна вести строго на стандартный сценарий входа в веб (Web to Web).

3. Аутентификация на веб-странице Партнера внутри мобильного приложения Сбербанк Онлайн (App to webview)

Внутри мобильного приложения Сбербанк Онлайн есть возможность открывать сайт Партнера во встроенном браузере (Safari View Controller/Google Chrome Custom Tabs) по клику на различные баннеры.

Для того, чтобы клиент мог бесшовно (без ввода логина и пароля) войти по Сбер ID на сайт Партнера необходимо вызвать сценарий входа по Сбер ID по deeplink.

Важно! Вызывать сценарий входа по Сбер ID по deeplink можно только из встроенного браузера внутри мобильного приложения Сбербанк Онлайн.

Если вызвать его из стороннего браузера, то вход по Сбер ID завершится ошибкой.

Для открытия веб-страницы Сбер ID в браузере внутри мобильного приложения Сбербанк Онлайн следует вызывать обычную ссылку, а не универсальную (см. сценарий mWeb to App).

Есть два сценария запуска входа по Сбер ID:

1. Клиент самостоятельно нажимает на кнопку "Войти по Сбер ID"
2. Сайт Партнера автоматически запускает сценарий входа по Сбер ID

Сайту Партнера рекомендуется реализовать сервис, который будет формировать страницу не с обычной веб-ссылкой на кнопке Сбер ID, а с deeplink.

Сценарий:

1. Клиент внутри мобильного приложения Сбербанк Онлайн нажимает на какой-либо баннер (сториз, каталог, ссылка в чате и т.д.)
2. В результате в Safari View Controller/Google Chrome Custom Tabs вызывается сервис Партнера вида https://partner.ru/auth?type=deeplink&source=StoryGD20
3. Сервис Партнера по входным параметрам понимает, что на кнопку Сбер ID нужно поставить не обычную ссылку, а deeplink и добавить к ней параметр source:

Code Block 1 Пример запроса

```
sberbankidlogin://sberbankidsso?

response_type=code

&scope=openid+phones+place_of_birth+gender

&client_id=AAAABBBB-XXXX-YYYY-ZZZZ-A12A618A4C3C

&state=UhdlDQS6vG4m&nonce=TGfhfBxr5ak1

&redirect_uri=https://exapmle.com/external_login

&source=StoryGD20
```

,где source – любая строка (пример: Story, SmartBanner, Catalog). Опциональный параметр, в который передается точка входа в МП СБОЛ из которой был открыт Safari View Controller/Google Chrome Custom Tabs.

Остальные параметры являются стандартными для любого входа по Сбер ID и описаны выше в таблице 2.

Рекомендуем воспользоваться SberID JS SDK или скриптом для формирования универсальной ссылки, который возвращает уже сформированный deeplink в объекте oidc.

Параметр source можно указать при инициализации скрипта в params.

Пример работы со скриптом представлен выше в описании сценария mWeb to App.

4. Бесшовный переход из веб Сбербанк Онлайн на сайт Партнера (Web to Web SSO)

Внутри веб Сбербанк Онлайн есть возможность разместить баннер (или ссылку), ведущую на сайт Партнера.

Для того, чтобы клиент при переходе мог бесшовно (без ввода логина и пароля) войти по Сбер ID на сайт Партнера необходимо вызвать сценарий входа по Сбер ID в веб (Web to Web). Пример формирования запроса описан выше в сценарии (Web to Web).

Партнеру рекомендуется реализовать сервис, который будет автоматически формировать и вызывать запрос кода авторизации.

Сценарий:

1. Клиент внутри веб Сбербанк Онлайн нажимает на какой-либо баннер (или ссылку)
2. В результате клиент перенаправляется на сервис Партнера вида https://partner.ru/auth?type=auto&source=StoryGD20&to=cabinet, где source и to - дополнительные параметры, по которым сервис Партнера определяет откуда пришел клиент и куда его нужно перенаправить после успешного входа (то есть какой адрес указать в параметре redirect_uri в запросе кода авторизации). Сбер ID эти параметры никак не использует и они не участвуют в запросе кода авторизации.
3. Сервис Партнера по входным параметрам (type=auto) определяет, что нужно сформировать и вызвать запрос кода авторизации
4. Сервис Партнера запрашивает код авторизации (перенаправляет клиента на страницу Сбер ID)
5. Клиент автоматически аутентифицируется в Сбер ID (по уже активной сессии), подтверждает вход и согласие (если оно не активно)
6. В результате успешного входа Сбер ID перенаправляет клиента на сервис Партнера по адресу, указанному в параметре redirect_uri

1.1.1.2. Параметры ответа

Ответ на запрос кода авторизации

В случае успешной обработки запроса front-end партнера получает код авторизации.

Тип ответа HTTP 302 Found.

Пример ответа в случае успешного выполнения запроса:

```
HTTP/1.1 302 Found
Location: https%3A%2F%2Fclientresource%2Fcb HTTP/1.1
code=FA2154AC-3451-C01A-B2D3-C231DBB2E20F
&state=af0ifjsldkj
```

Таблица 3. Описание полей ответа на запрос кода авторизации

№ п/п	Наименование	Заголовок/поле	Описание
1	Location	Заголовок	redirect_uri, указанное в запросе кода авторизации.
2	code	Поле	Сгенерированный код авторизации.
3	state	Поле	Значение атрибута **state**, указанное в запросе кода авторизации.

1.1.1.3. Описание ошибок

В случае неуспешной обработки запроса, front-end партнера получает сообщение, в котором содержится тип ошибки. Тип ответа HTTP 302 Found.

Пример ответа в случае ошибки:

```
HTTP/1.1 302 Found
Location: https%3A%2F%2Fclientresource%2Fcb HTTP/1.1
error=invalid_request
&state=af0ifjsldkj
```

Таблица 4. Описание полей ответа на неуспешный запрос кода авторизации

№ п/п	Наименование	Заголовок/поле	Описание
1	Location	Заголовок	redirect_uri, указанное в запросе кода авторизации.
2	error	Поле	Наименование ошибки.
3	state	Поле	Значение атрибута state, указанное в запросе кода авторизации.

Таблица 5. Типы возвращаемых ошибок

№ п/п	Описание	Тип возвращаемой ошибки
1	В запросе отсутствуют обязательные атрибуты	invalid_request
2	АС-источник запроса не зарегистрирована в банке	unauthorized_client
3	АС-источник запроса заблокирована в банке	unauthorized_client
4	Значение атрибута client_id не соответствует формату	unauthorized_client
5	Значение атрибута response_type не равно «code»	unsupported_response_type
6	Значение атрибута scope не содержит параметр openid в начальной позиции	invalid_scope
7	Запрошенный scope содержит значения, недоступные для АС-источника запроса	invalid_scope

Примечание: в случае отсутствия в запросе атрибута redirect_uri или в случае, если значение redirect_uri не зарегистрировано для данного партнера, банк перенаправляет клиента на страницу, информирующую клиента о недоступности сервиса.

1.1.2. Мобильное приложение

1.1.2.1. Параметры запроса

Ниже описаны 2 сценария взаимодействия:

- 1. Аутентификация в мобильном приложении Партнера через приложение Сбербанк Онлайн и через веб Сбер ID (App to App & App to Web)
  - Android. Запрос кода авторизации через deeplink
  - iOS. Запрос кода авторизации через deeplink
- 2. Бесшовный переход из мобильного приложения Сбербанк Онлайн в приложение Партнера (App to App SSO)

Получение кода авторизации состоит из следующих шагов:

1. Клиент в приложении Партнера нажимает кнопку входа по Сбер ID
2. Приложение Партнера делает запрос в back-end Партнера для получения параметров аутентификации (client_id, redirect_url, state, nonce)
3. Приложение Партнера генерирует параметр **code_verifier** и хэширует его методом **code_challenge_method** (S256). В результате хэширования получается значение **code_challenge**
4. Мобильное приложение Партнера проверяет наличие на устройстве клиента приложения Сбербанк Онлайн.
   Если оно есть - запрашивает код авторизации через приложение Сбербанк Онлайн (по deeplink), если нет - запрашивает код авторизации путем перенаправления на веб-страницу Сбер ID https://online.sberbank.ru/CSAFront/oidc/authorize.do (get запрос) в отдельном браузере.
   В запросе указываются параметры client_id, scope, redirect_url, state, nonce, code_challenge, code_challenge_method.
5. Клиент в приложении Сбербанк Онлайн/веб-странице Сбер ID подтверждает аутентификацию
6. В результате аутентификации Сбербанк Онлайн/веб-страница Сбер ID перенаправляет клиента в приложение Партнера по адресу, указанному в redirect_uri. В случае успеха возвращается код авторизации и state, в ином случае - ошибка.

Важно! Партнер для вызова аутентификации не должен использовать браузер, встроенный в приложение, который предоставляет приложению доступ к cookie, или содержимому веб-страниц.

При открытии веб-страницы Сбер ID в браузере, встроенном в приложение, будет отображена страница-заглушка, сообщающая клиенту о небезопасности входа.

Важно!

Для сценариев, в которых присутствует риск перехвата AuthCode необходимо использовать защиту PKCE (https://tools.ietf.org/html/rfc7636). По рекомендациям RFC 7636 для смягчения атак перехвата кода авторизации используется динамически создаваемое случайное значение - "code verifier". Данное значение должно быть уникальным для каждого запроса кода авторизации.

Требования к генерации значения code_verifier (см. также https://tools.ietf.org/html/rfc7636#section-4.1):

- Значение code_verifier - это высокоэнтропийная криптографическая случайная строка.
- Строка генерируется с использованием допустимых символов [AZ] / [az] / [0- 9] / "-" / "." / "_" / "~".
- Минимальная длина - 43 символа.
- Максимальная длина - 128 символов.

Один из возможных алгоритмов:

Партнер, используя подходящий генератор случайных чисел, создает последовательность длиной от 32 до 96 байт, которую затем кодирует способом base64url. Например, для 32-байтной последовательности [116, 24, 223, 180, 151, 153, 224, 37, 79, 250, 96, 125, 216, 173, 187, 186, 22, 212, 37, 77, 105, 214, 191, 240, 91, 88, 5, 88, 83, 132, 141, 121] кодирование base64url в результате даст code_verifier в виде “dBjftJeZ4CVP-mB92K27uhbUJU1p1r_wW1gFWFOEjXk” (см. https://tools.ietf.org/html/rfc7636#appendix-B).

1. Аутентификация в мобильном приложении Партнера через приложение Сбербанк Онлайн и через веб Сбер ID (App to App & App to Web)

Android. Запрос кода авторизации через deeplink

Code Block 2 Схема запроса

```
sberbankidlogin://sberbankid?client_id={client_id}&state={state}&scope={s
cope}&redirect_uri={redirect_uri}&code_challenge_method={S256}
```

Описание полей запроса смотри в **Таблице 6. Описание полей запроса кода авторизации.**

На экран авторизации в приложении Партнера необходимо добавить кнопку Войти через Сбер ID, при нажатии на которую инициируется запрос кода авторизации.

Получение кода авторизации состоит из следующих шагов:

1. Добавить ключи параметров в ваше приложение.

Code Block 3 Формирование параметров запроса

```
//Название ключей, для параметров запроса
private static final String CLIENT_ID = "client_id";
private static final String STATE = "state";
private static final String NONCE = "nonce";
private static final String SCOPE = "scope";
private static final String BROWSER = "package";
private static final String REDIRECT_URI = "redirect_uri";
private static final String CODE_CHALLENGE = "code_challenge";
private static final String CODE_CHALLENGE_METHOD =
"code_challenge_method";
```

2. Создать Uri с параметрами из п.1.

Code Block 4 Пример

```
//Создание Uri с параметрам
 Uri uri = new Uri.Builder()
        .scheme(SCHEME_MP)
        .authority(HOST_MP)
        .appendQueryParameter(CLIENT_ID, clientID)
        .appendQueryParameter(STATE, state)
        .appendQueryParameter(NONCE, nonce)
        .appendQueryParameter(SCOPE, scope)
     .appendQueryParameter(REDIRECT_URI, redirectUri)
        .appendQueryParameter(CODE_CHALLENGE, codeChallenge)
        .appendQueryParameter(CODE_CHALLENGE_METHOD, codeChallengeMethod)
        .build();
```

3. Проверить установлено ли приложение Сбербанк Онлайн.

Code Block 5 Пример

```
//Проверка, что приложение Сбербанк онлайн установлено на телефоне
private boolean isSberbankDeeplinkCanBeHandle(Context context) {
    Uri testSberbankDeeplink = new Uri.Builder()
            .scheme(SCHEME_MP)
            .authority(HOST_MP)
            .build();
    Intent testSberbankIntent = new Intent(Intent.ACTION_VIEW,
testSberbankDeeplink);
    return
testSberbankIntent.resolveActivity(context.getPackageManager()) != null;
}
```

4. Запросить код авторизации:

- если приложение Сбербанк Онлайн установлено, то его необходимо запустить для запроса кода авторизации

Code Block 6 Пример

```
//Запуск приложения Сбербанк Онлайн
Intent intent = new Intent(Intent.ACTION_VIEW, uri);
startActivity(intent);
```

- если не установлено, то необходимо запросить код авторизации путем перенаправления на веб-страницу Сбер ID https://online.sberbank.ru/CSAFront/oidc/authorize.do (get запрос) в отдельном браузере. (подробнее можно узнать в разделе 1.1.1.1. Параметры запроса).

Code Block 7 Пример

```
//Замените хост и схему, чтобы использовать web версию для аутентификации
if (checkSbolIsNotInstalled(context)) {
    uri.buildUpon()
    .scheme("https")
    .authority("online.sberbank.ru")
    .appendEncodedPath("CSAFront/oidc/authorize.do")
    .appendQueryParameter("response_type", "code")
    .build();
}
```

5. В результате аутентификации вернется deeplink, указанный в redirect_uri. Чтобы обработать диплинк, укажите в *AndroidManifest.xml*

Code Block 8 Пример

```
<activity
    android:name="com.example.MainActivity"
    android:theme="@style/AppTheme.NoActionBar">
    <intent-filter>
        <action android:name="android.intent.action.VIEW" />

        <category android:name="android.intent.category.DEFAULT" />
        <category android:name="android.intent.category.BROWSABLE" />

        <data
            android:host="merchant_host"
            android:scheme="app" />

    </intent-filter>
</activity>
```

Host и *scheme* должны соответствовать *redirect_uri.*

Code Block 9 Пример обратного deeplink

```
Положительный результат:
app://apphost?state=Jt2dvD9a9tmZ&code=0BC4A121-F75F-8A3B-BE7E-
8C2412209B17

Негативный результат:
app://apphost?result=FAILURE&error_code=5&error={error}
```

Код ошибки 5 означает, что в запросе от приложения Партнера пришли некорректные данные. Также можно посмотреть описание возможных ошибок в **Таблице 8. Типы возвращаемых ошибок**

Если в ответе пришел error и не пришел код ошибки, то необходимо обрабатывать это как ошибку.

Результат стоит проверять по наличию параметров state и code.

iOS. Запрос кода авторизации через deeplink

Code Block 10 Схема запроса

```
sberbankidexternallogin://sberbankid?client_id={client_id}&scope={scope}&
state={state}&redirect_uri={redirect_uri}
```

Описание полей запроса смотри в **Таблице 6. Описание полей запроса кода авторизации.**

На экран авторизации в приложении Партнера необходимо добавить кнопку Войти по Сбер ID, при нажатии на которую инициируется запрос кода авторизации.

Получение кода авторизации состоит из следующих шагов:

1. Первоначально в настройках проекта необходимо создать redirecturi по примеру: sberbankidexternallogin:/ , где merchantScheme должен соответствовать в info.plist/URL types/URL Scheme
пример:

Code Block 11 Пример

```
<key>CFBundleURLTypes</key>
 <array>
  <dict>
   <key>CFBundleTypeRole</key>
   <string>Editor</string>
   <key>CFBundleURLName</key>
   <string>$(PRODUCT_BUNDLE_IDENTIFIER)</string>
   <key>CFBundleURLSchemes</key>
   <array>
    <string>sberbankinvestor</string>
    <string>merchantApp</string>
    <string>messenger</string>
    <string>merchant</string>
   </array>
  </dict>
```

2. Проверить установлено ли приложение Сбербанк Онлайн.

Code Block 12 Пример

```
NSURL *URL = [NSURL URLWithString:@"sberbankidexternallogin://"];
BOOL appToAppEnabled = [[UIApplication sharedApplication]
canOpenURL:URL];
if (appToAppEnabled){
    // Сценарий AppToApp
} else {
    // Сценарий AppToWeb
}
```

3. Запросить код авторизации:

- если приложение Сбербанк Онлайн установлено, то его необходимо запустить для запроса кода авторизации

Code Block 13 Пример

```
NSString *strURL = [NSString
stringWithFormat:@"sberbankidexternallogin://sberbankid?%@=%@&%@=%@&%@=%@
&%@=%@",
      @"client_id", sbID.clienID,
      @"scope", sbID.scope,
      @"state", sbID.state,
      @"nonce", sbID.nonce/*,
            @"image_url", [imagePath
stringByAddingPercentEncodingWithAllowedCharacters:[NSCharacterSet
alphanumericCharacterSet]]*/
      ];
 if (sbID.PKCEEEnabled)
 {
  NSString *codeChallenge = [PKCEHelper generateCodeChallenge];
  NSString *codeChallengeMethod = [PKCEHelper codeChallengeMethod];
  strURL = [strURL
stringByAppendingFormat:@"&code_challenge=%@&code_challenge_method=%@",
codeChallenge, codeChallengeMethod];
 }

 strURL = [strURL stringByAppendingFormat:@"&%@=%@", @"redirect_uri",
sbID.redirectURI];

 NSURL *URL = [NSURL URLWithString:[strURL
stringByAddingPercentEncodingWithAllowedCharacters:[NSCharacterSet
URLQueryAllowedCharacterSet]]];
 [[UIApplication sharedApplication] openURL:URL];
```

- если не установлено, то необходимо запросить код авторизации путем перенаправления на веб-страницу Сбер ID https://online.sberbank.ru/CSAFront/oidc/authorize.do (get запрос) в отдельном браузере. (подробнее можно узнать в разделе 1.1.1.1. Параметры запроса).

Code Block 14 Пример

```
NSURLComponents *URLComponents = [NSURLComponents
componentsWithString:@"https://online.sberbank.ru/CSAFront/oidc/sberbank_
id/authorize.do"];
NSURLQueryItem *responseTypeQueryItem = [NSURLQueryItem
queryItemWithName:@"response_type" value:responseType];
NSURLQueryItem *clientIDQueryItem = [NSURLQueryItem
queryItemWithName:@"client_id" value:clientID];
NSURLQueryItem *stateQueryItem = [NSURLQueryItem
queryItemWithName:@"state" value:state];
NSURLQueryItem *nonceQueryItem = [NSURLQueryItem
queryItemWithName:@"nonce" value:nonce];
NSURLQueryItem *scopeQueryItem = [NSURLQueryItem
queryItemWithName:@"scope" value:scope];
NSURLQueryItem *redirectURIQueryItem = [NSURLQueryItem
queryItemWithName:@"redirect_uri" value:redirectURI];
NSURLQueryItem *codeChallengeQueryItem = [NSURLQueryItem
queryItemWithName:@"code_challenge" value:codeChallenge];
NSURLQueryItem *codeChallengeMethodItem = [NSURLQueryItem
queryItemWithName:@"code_challenge_method" value:codeChallengeMethod];

URLComponents.queryItems = @[responseTypeQueryItem, clientIDQueryItem,
stateQueryItem, nonceQueryItem, scopeQueryItem, redirectURIQueryItem,
         codeChallengeQueryItem, codeChallengeMethodItem
         ];

NSURL *URL = URLComponents.URL;
[[UIApplication sharedApplication] openURL:URL];
```

4. Обработка редиректа (по redirect_uri) происходит в методах AppDelegate.

Code Block 15 Пример

```
// < iOS 9
- (BOOL)application:(UIApplication *)application handleOpenURL:(NSURL
*)url
// >= iOS 9
- (BOOL)application:(UIApplication *)app openURL:(NSURL *)url
options:(NSDictionary<NSString*, id> *)options
```

В выбранном методе происходит обработка redirect_uri, отправленного в запросе на получение кода авторизации

Code Block 16 Пример

```
NSDictionary *dictParams = [self URLQueryParametersFoURL:url];
NSString *state = dictParams[@"state"];
NSString *code = dictParams[@"code"];
NSString *status = dictParams[@"status"];
 if (status.length && [status isEqualToString:@"success"] && state.length
&& code.length) {
  // Обработка успешного результата
 }
 else if ([status isEqualToString:@"fail"]) {
  // Обработка не успешного результата с возможной ошибкой
 }
```

Код ошибки 5 означает, что в запросе от приложения Партнера пришли некорректные данные. Также можно посмотреть описание возможных ошибок в **Таблице 8. Типы возвращаемых ошибок**

Если в ответе пришел error и не пришел код ошибки, то необходимо обрабатывать это как ошибку.

Результат стоит проверять по наличию параметров state и code.

Таблица 6. Описание полей запроса кода авторизации

№ п/п	Наименование заголовка/поля	Описание	Обязательность поля	Пример
1	response_type	Указывается равным code	Да	code
2	client_type	Указывается равным PRIVATE	Нет	PRIVATE
3	scope	Наименование групп данных, на которые подписана система Партнера (выдается при регистрации системы в банке). Значение openid является обязательным и располагается на первой позиции. Разделитель – пробел.	Да	openid name maindoc email mobile
4	client_id	Идентификатор системы Партнера, полученный партнером в личном кабинете после регистрации приложения.	Да	DA5278AC-A07F-C01A-B2D3-C231DBB2E20F
5	state	Значение, включенное в запрос, которое также возвращается в ответе. Может быть строка любого контента. Для предотвращения подделки межсайтовых запросов используется генерируемое случайным	Да	af0ifjsldkj

		образом уникальное значение.		
6	nonce	Значение, сгенерированное внешней АС для предотвращения атак повторения. Это значение обычно представляет собой случайную уникальную строку или глобальный уникальный идентификатор, которые можно использовать для определения источника запроса. Ограничение по длине значения составляет 64 символа.	Да	n-0S6_WzA2Mj
7	redirect_uri	Адрес страницы Партнера, на которую будет перенаправлен клиент после успешной аутентификации в системе банка. Временное ограничение: недопустимы символы “;” и “=“.	Да	iOS/Android deeplink: https://app://sberbankid
8	code_challenge	Хэшированное значение секретного кода code_verifier Партнера (генерацию code_verifier см. выше в блоке «Важно»). Хэширование выполняется методом, указанным в code_challenge_method, в нашем случае – всегда S256, поэтому code_challenge = BASE64URL-	Нет	E9Melhoa2OwvFrEMTJguCHaoeK1t8URWbuGJSstw-cM

		ENCODE(SHA256 (ASCII (code_verifier))) (см. https://tools.ietf.org/html/rfc7636#section-4.2). Например, для code_verifier = “dBjftJeZ4CVP-mB92K27uhbUJU1p1r_wW1gFWFOEjXk” выполнение хэширования SHA256 этого значения в результате выдаст последовательность байтов [19, 211, 30, 150, 26, 26, 216, 236, 47, 22, 177, 12, 76, 152, 46, 8, 118, 168, 120, 173, 109, 241, 68, 86, 110, 225, 137, 74, 203, 112, 249, 195], преобразование которой через Base64Url дает нам значение code_challenge = “E9Melhoa2OwvFrEMTJguCHaoeK1t8URWbuGJSstw-cM” (см. https://tools.ietf.org/html/rfc7636#appendix-B).		
9	code_challenge_method	Метод преобразования секретного кода code_verifier Партнера. Допустимым значением является S256.	Нет	S256

2. Бесшовный переход из мобильного приложения Сбербанк Онлайн в приложение Партнера (App to App SSO)

Внутри мобильного приложения Сбербанк Онлайн есть возможность разместить баннер (или ссылку), ведущую на мобильное приложение Партнера.

Для того, чтобы клиент при переходе мог бесшовно (без ввода логина и пароля) войти по Сбер ID в мобильное приложение Партнера необходимо вызвать сценарий входа по Сбер

ID в веб (App to App). Пример формирования запроса описан выше в сценарии (App to App).

Для реализации сценария необходимо**:**

1. Внутри мобильного приложения Сбербанк Онлайн предварительно разместить баннер (или ссылку) с помощью специалистов Сбера
2. Партнеру реализовать внедрить App2App вход по Сбер ID (описан в разделе 1)
3. Партнеру внутри своего мобильного приложения реализовать сервис, который будет автоматически формировать и вызывать запрос кода авторизации (шаг 2 в сценарии ниже)

Сценарий:

1. Клиент внутри мобильного приложения Сбербанк Онлайн нажимает на какой-либо баннер (или ссылку)
2. В результате клиент перенаправляется в мобильное приложение Партнера по deeplink (или universal link), например partner://auth?type=auto&source=StoryGD20&to=cabinet, где source и to - дополнительные параметры, по которым сервис Партнера определяет откуда пришел клиент и куда его нужно перенаправить после успешного входа (то есть какой адрес указать в параметре redirect_uri в запросе кода авторизации). Сбер ID эти параметры никак не использует и они не участвуют в запросе кода авторизации.
3. Мобильное приложение Партнера по входным параметрам (type=auto) определяет, что нужно сформировать и вызвать запрос кода авторизации по deeplink. Пример запроса описан выше в сценарии App to App.
4. Мобильное приложение Партнера запрашивает код авторизации (перенаправляет клиента в мобильное приложение Сбербанк Онлайн)
5. Клиент автоматически аутентифицируется в мобильном приложении Сбербанк Онлайн (по уже активной сессии), подтверждает согласие (если оно не активно)
6. В результате успешного входа мобильное приложение Сбербанк Онлайн перенаправляет клиента в приложение Партнера по адресу, указанному в параметре redirect_uri, например partner://sberidauthtocabinet

При реализации данного сценария Партнеру необходимо учесть следующие случаи:

1. У клиента нет мобильного приложения Партнера. В этом случае нужно вести клиента сначала в Store, и после установки автоматически продолжать открытие deeplink.
2. Клиент уже аутентифицирован в мобильном приложении Партнера каким-либо другим способом. В этом случае клиенту нужно предложить связать учетные записи.

1.1.2.2. Параметры ответа

В случае успешной обработки запроса мобильное приложение партнера получает код авторизации.

Пример обработки ответа:

iOS

```
NSDictionary *dictParams = [self URLQueryParametersFoURL:url];
NSString *authCode = dictParams[@"code"];
NSString *state = dictParams[@"state"];
NSString *code = dictParams[@"code"];
NSString *status = dictParams[@"status"];
if (status.length && [status isEqualToString:@"success"] && state.length
&& code.length) {
 // Успешный сценарий
}
else if ([status isEqualToString:@"fail"]) {
 //Сценарий ошибки
}
- (NSDictionary *)URLQueryParametersFoURL:(NSURL *)URL
{
 NSString *queryString = [URL query];
 NSMutableDictionary *result = [NSMutableDictionary dictionary];
 NSArray *parameters = [queryString componentsSeparatedByString:@"&"];
 for (NSString *parameter in parameters)
 {
  NSArray *parts = [parameter componentsSeparatedByString:@"="];
  if ([parts count] > 1)
  {
   NSString *key = [parts[0] stringByRemovingPercentEncoding];
   NSString *value = [parts[1] stringByRemovingPercentEncoding];
   result[key] = value;
  }
 }
 return result;
}
```

Android

```
if(intent.getData() != null) {
    mAuthCode = intent.getData().getQueryParameter(CODE);
    mState = intent.getData().getQueryParameter(STATE);
}
```

Таблица 7. Описание полей ответа на запрос кода авторизации

№ п/п	Название параметра	Описание	Обязательность поля	Пример
1	code	Код авторизации клиента.	[0-1] 0 - только в случае ошибки	FA2154AC-3451-C01A-B2D3-C231DBB2E20F
2	error	Текст ошибки (возможные значения приведены в таблице 8).	[0-1] 1 - только в случае ошибки	invalid_request
3	state	Значение, включенное в запрос возвращается в ответе.	Да	af0ifjsldkj

1.1.2.3. Описание ошибок

В случае неуспешной обработки запроса, мобильное приложение партнера получает сообщение, в котором содержится тип ошибки.

Пример ответа с ошибкой:

iOS

```
Components merchantScheme://redirect?status=fail&error=invalid_request
```

Android

```
appSchem://appName/appPath?result=FAILURE&error_code=5
```

Таблица 8. Типы возвращаемых ошибок

№ п/п	Описание	Тип возвращаемой ошибки
1	В запросе отсутствуют обязательные атрибуты.	invalid_request
2	АС-источник запроса не зарегистрирована в банке.	unauthorized_client
3	АС-источник запроса заблокирована в банке.	unauthorized_client
4	Значение атрибута client_id не соответствует формату.	unauthorized_client
5	Значение атрибута response_type не равно «code».	unsupported_response_type
6	Запрошенный scope содержит значения, недоступные для АС-источника запроса.	invalid_scope
7	Значение code_challenge_method не соответствуют допустимым значениям.	invalid_request

Важно!

- Для любого из перечисленных типов ошибок МП СБОЛ на платформе Android возвращает в приложение партнера код 5 в параметре error_code без указания типа ошибки.
- В случае отсутствия в запросе атрибута redirect_uri или в случае если значение redirect_uri не зарегистрировано для данного партнера, банк перенаправляет клиента на экран, информирующий клиента о недоступности сервиса.

1.2. Запрос access token и id token

На "Схеме взаимодействия" обозначен как "13. Запрос access token, id token ".

1.2.1. Параметры запроса

Back-end Партнера инициирует запрос в back-end Банка на получение access token и Id token, направляя в запросе полученный ранее код авторизации.

Пример запроса на получение access token и Id token:

```
curl --request POST \
  --url https://api.sberbank.ru/ru/prod/tokens/v2/oidc \
  --header 'accept: application/json' \
  --header 'content-type: application/x-www-form-urlencoded' \
  --header 'rquid: REPLACE_THIS_VALUE' \
  --header 'x-ibm-client-id: REPLACE_THIS_KEY' \
  --data
'grant_type=authorization_code&scope=openid+name+maindoc&client_id=5e7668
0a-6344-4978-8ee4-
5ff6370695ddd&client_secret=****yTd7r1&code=*****B6199B56184&redirect_uri
=https%3A%2F%2Fwww.sberbank.ru%2F&code_verifier=dBjftJeZ4CVP-
mB92K27uhbUJU1p1r_wW1gFWFOEjXk'
```

Таблица 9. Описание полей запроса на получение access token и id token

№ п/п	Наименование заголовка/поля	Заголовок / Поле	Описание	Обязательность поля
1	X-IBM-Client-ID	Заголовок	Идентификатор системы Партнера, полученный Партнером в Личном Кабинете после регистрации приложения.	Да
2	RqUID	Заголовок	Уникальный идентификатор сообщения, «maxLength=32 и pattern=([0-9]|[a-f]|[A-F]){32})», переданный во входящем сообщении. Необходим для журналирования входящих вызовов и удобства разбора инцидентов. Чтобы обеспечить уникальность, можно использовать стандартные библиотеки и классы для генерации UUID/GUID (https://ru.wikipedia.org/wiki/UUID), убрав из результата разделители «-».	Да

3	content-type	Заголовок	Всегда принимает значение "application/x-www-form-urlencoded"	Да
4	Accept	Заголовок	Опционально. Может принимать значения application/json или application/jwt.	Нет
5	grant_type	Поле	Указывается равным authorization_code.	Да
6	code	Поле	Код авторизации, полученный от Банка. (См. Ответ на запрос кода авторизации).	Да
7	redirect_uri	Поле	Значение параметра redirect_uri, которое было указано в запросе кода авторизации.	Да
8	client_id	Поле	Идентификатор системы Партнера, полученный партнером в личном кабинете после регистрации приложения. Значение client_id должно совпадать с параметром x-ibm-client-id.	Да
9	client_secret	Поле	Пароль системы Партнера, полученный партнером в личном кабинете после регистрации приложения. Значение client_secret должно совпадать с параметром x-ibm-client-secret.	Да
10	code_verifier	Поле	Секретное значение приложения Партнера, сгенерированное для защиты от атак перехвата кода авторизации. *Данный параметр добавляется в запрос, только если код авторизации (AuthCode) был получен с использованием параметров PKCE (code_challenge и code_challenge_method).* *Параметр поддерживается начиная api v2 для запроса access token, id token.* Требования к значению code_verifier см. в разделе "1.1.2.1. Параметры запроса. Блок Важно!".	Нет

1.2.2. Шлюзы вызова API

Шлюзы вызова API:

DEVELOPMENT
https://dev.api.sberbank.ru/ru/prod/tokens/v2/oidc

PRODUCTION
https://sec.api.sberbank.ru/ru/prod/tokens/v2/oidc - для подключений через ФПСУ
https://api.sberbank.ru/ru/prod/tokens/v2/oidc - для подключений через двусторонний TLS

Также, актуальные шлюзы для вызова API размещены на Портале разработчиков в документации по API - https://developer.sberbank.ru/api/5c9f5313e4b0388ba0f08b42 (ссылка доступна только зарегистрированным пользователям).

1.2.3. Параметры ответа

В случае успешной обработки запроса Back партнера получает access token и Id token. Ответ типа HTTP 200 OK. Описание полей ответа приведено в таблице 10.

Пример JSON сообщения - ответа в случае успешной обработки запроса:

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
rquid: 01234567890123456789012345678 9FF
Cache-Control: no-store
Pragma: no-cache
{
  "access_token":"f213a511-58d7-4e7c-88b3-a6de380c81da",
  "token_type":"Bearer",
  "expires_in":864000,
  "scope": "openid name https://api.sberbank.ru/sberbankid/userinfo",
  "id_token": "eyJhbGciOi-
IgZ29zdDM0LTEwLjIwMTIifQ.ewogImlzcyI6ICJodHRwOi8vc2VydmVyLmV4YW1wbGU
uY29tIiwKICJzdWIiOiAiMjQ4Mjg5NzYxMDAxIiwKICJhdWQiOiAiMTAwMSIsCiAibm9uY2Ui
OiAib
i0wUzZfV3pBMk1qIiwKICJleHAiOiAxMzExMjg0NTcwLAogImlhdCI6IDEzMTEyODA5NzAK4o
Ccc2l
kMuKAnTrigJ0xMjM0NTY4Nzk5MjU14oCdIAog4oCcbmFtZSI6IOKAnNCY0LLQsNC90L7QsiDQ
mNCy0
LDQvSDQmNCy0LDQvdC+0LLQuNGH4oCdCn0.EJqZnRKZVo0Q1ZQLW1COTJLMjd1aGJVSlUxcDF
yX3dX
MWdGV0ZPRWpYaw."
}
```

Таблица 10. Описание полей ответа

№ п/п	Наименование поля	Описание	Обязательность	Пример
1	access_token	Сгенерированный Access token.	Да	f213a511-58d7-4e7c-88b3-a6de380c81da
2	token_type	Тип запрашиваемого токена. Всегда передается значение «Bearer».	Да	Bearer
3	expires_in	Время в секундах, в течение которого действует Access Token.	Да	864000
4	scope	Список групп персональных данных, на получение которых выдан данный токен. В список так же по умолчанию включается название сервиса API	Да	openid name https://api.sberbank.ru/sberbankid/userinfo
5	id_token	Закодированный набор атрибутов пользователя в формате base64_url, необходимый для идентификации пользователя.	Да	eyJhbGciOi-IgZ29zdDM0LTEwLjIwMTIifQ.ewogImlzcyI6ICJodHRwOi8vc2VydmVyLmV4YW1wbGU uY29tIiwKICJzdWIiOiAiMjQ4Mjg5NzYxMDAxIiwKICJhdWQiOiAiMTAwMSIsCi-Aibm9uY2UiOiAib i0wUzZfV3pBMk1qIiwKICJleHAiOiAxMzExMjg0NTcwLAogImlhdCI6IDEzMTEyODA5NzAK4oCcc2l kMuKAnTrigJ0xMjM0NTY4Nzk5MjU14oCdIAog4oCcbmFtZSI6IOKAnNCY0LLQsNC90L7QsiDQmNCy0LDQvSDQmNCy0LDQvdC+0LLQu NGH4oCdCn0.EJqZnRKZVo0Q1ZQLW1COTJLMjd1aGJVSlUxcDFyX3dX MWdGV0ZPRWpYaw.

Пример ID Token:

```
{
 "iss":       "https: //online.sberbank.ru/ESAFront/index.do",
 "sub":
"74c64d08bdd5e6f2b94770e9fed9342b9054f22bea1571e68448c8cae83e0d80ec206549
e11d13fc",
 "aud":       "DA5278AC - A07F - C01A - B2D3 - C231DBB2E20F",
 "exp":       141254872,
 "iat":       1411250872,
 "auth_time": 1411250871,
 "nonce":     "n - 0 S6_WzA2Mj"
 }
```

Таблица 11. Описание полей ID Token

№ п/п	Наименование поля	Описание	Примечание
1	sub	Неизменный уникальный идентификатор клиента в АС Банка, передаваемый внешним потребителям. Максимальная длина 96 символов.	Идентификатор клиента банка, используемый внешними АС. Идентификатор может быть сохранен у Партнера.
2	iss	URL сервиса (URL АС банка), сформировавшего ID Token.	
3	aud	Идентификатор внешней АС/ресурсной системы (client_id).	При получении ID_Token партнеру необходимо сравнить значение aud со значением своего client_id. Если значения не совпадают, то партнер должен завершить сценарий без авторизации клиента на своем ресурсе.
4	exp	Время, до наступления которого гарантируется неизменность информации о клиенте в составе ID Token.	
5	iat	Время формирования ID Token.	
6	auth_time	Время аутентификации клиента в АС Банка при запросе кода авторизации.	
7	nonce	Значение переменной, сгенерированная ресурсной системой/внешней АС для предотвращения атак повторения при запросе кода авторизации.	Партнер получив ID_Token должен проверить значение параметра nonce, что полученное значение nonce равно значению параметра nonce, отправленного в запросе на аутентификацию. Если значения не совпадают, то Партнер должен завершить сценарий без авторизации клиента на своем ресурсе.

1.2.4. Описание ошибок

Пример ответа в случае ошибки:

```
HTTP/1.1 400 Bad Request
Content-Type: application/json
Cache-Control: no-store
Pragma: no-cache
{
    "httpCode": "400",
    "httpMessage": "Bad Request",
    "moreInformation": "invalid_grant"
}
```

Таблица 12. Типы возвращаемых ошибок

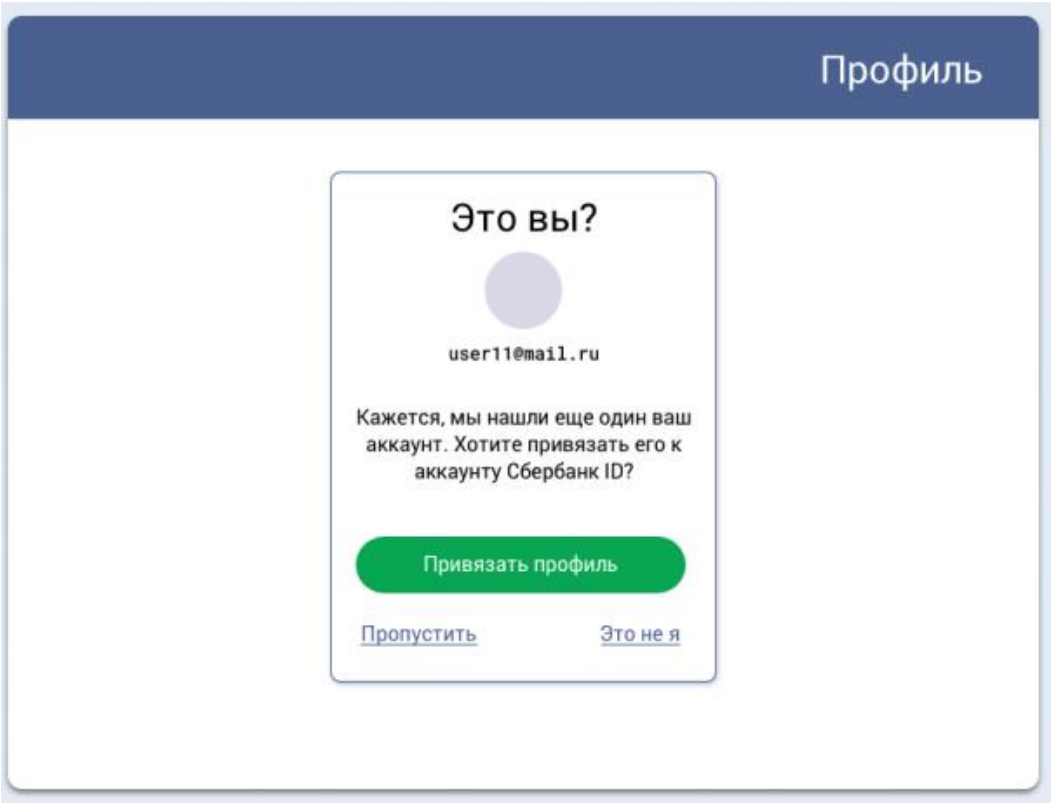
№ п/п	Тип ошибки	Тип возвращаемой ошибки
1	В запросе отсутствуют обязательные атрибуты.	invalid_request
2	Значение атрибута grant_type не равно «authorization_code».	unsupported_grant_type
3	Предъявленный Authorization_code не является действующим.	invalid_grant
4	Предъявленный Authorization_code выдан АС с другим client_id.	invalid_grant
5	Значение redirect_uri не соответствует значению, указанному в параметре redirect_uri в запросе кода авторизации URI АС-источника с указанным в запросе значением client_id.	invalid_grant
6	Authorization_code был выдан с использованием параметров PKCE, но в запросе токена не был указан code verifier.	invalid_request
7	Значение code_verifier не соответствует параметрам PKCE, переданным в первом запросе на Authorization_code.	invalid_grant
8	Аутентифицированному клиенту не разрешено использовать данный тип разрешений (grant_type).	unauthorized_client
9	Предоставленные разрешения (учетные данные владельца ресурса) не валидны, просрочены, отозваны.	invalid_grant

1.2.5. Мэтчинг учетных записей

После получения идентификатора в Сбер ID (sub в ID Token) рекомендуется следующий сценарий:

1. Партнер ищет у себя учетную запись (по sub из ID Token), к которой привязан Сбер ID
2. Если такая учетная запись найдена по sub, Партнер аутентифицирует Клиента
3. Если такой учетной записи не найдено, то Партнер может попробовать поискать учетную запись Клиента по данным, полученным через Сбер ID (например, адрес электронной почты, номер телефона и т.д.)
    a. Если такая учетная запись найдена (по дополнительным данным), Партнер предлагает Клиенту привязать ее к уже существующей. Клиент может пропустить этот шаг. В случае, если клиент пропускает этот шаг, то переход к шагу b.
    b. Если такой учетной записи не найдено, Партнер аутентифицирует Клиента

Пример окна с предложением привязать учетную запись:



1.3. Запрос на получение данных

На "Схеме взаимодействия" обозначен как "17. Запрос ПД клиента".

- 1.3.1. Параметры запроса
- 1.3.2. Шлюзы вызова API
- 1.3.3. Параметры ответа
- 1.3.4. Описание ошибок.

1.3.1. Параметры запроса

Back-end партнера инициирует запрос в back-end банка на получение профиля клиента. Тип запроса - GET.

Ниже приведен пример запроса:

```
curl --request GET \
  --url https://api.sberbank.ru/ru/prod/sberbankid/v2.1/userinfo \
  --header 'accept: application/json' \
  --header 'authorization: Bearer DC3641EC-A0C1-F61A-B2DE-A331C0B2E20F' \
  --header 'x-ibm-client-id: 5e76680a-6344-4978-8ee4-5ff6370695ddd' \
  --header 'x-introspect-rquid: L4hE5nH3wB5lC6sP0bO7bA666yM5bH5h'
```

Таблица 13. Описание полей запроса на получение профиля клиента

№ п/п	Наименование поля	Заголовок/поле	Описание	Обязательность поля
1	Authorization	Заголовок	Полученный ранее токен, см. Ответ на запрос access token и Id token. В начало необходимо добавить «Bearer», например: Bearer DC3641EC-A0C1-F61A-B2DE-A331C0B2E20F	Да
2	x-introspect-rquid	Заголовок	Уникальный идентификатор сообщения, «maxLength=32 и pattern=([0-9]|[a-f]|[A-F]){32})», переданный во входящем сообщении. Необходим для журналирования входящих вызовов и удобства разбора инцидентов. Чтобы обеспечить уникальность, можно использовать стандартные библиотеки и классы для генерации UUID/GUID (https://ru.wikipedia.org/wiki/UUID), убрав из результата разделители «-».	Да
3	X-IBM-Client-ID	Заголовок	Идентификатор системы партнера, полученный партнером в Личном кабинете после регистрации приложения.	Да
4	Accept	Заголовок	Опционально. Может принимать значения application/json или application/jwt.	Нет

1.3.2. Шлюзы вызова API

Шлюзы вызова API:

DEVELOPMENT
https://dev.api.sberbank.ru/ru/prod/sberbankid/v2.1/userinfo

PRODUCTION
https://sec.api.sberbank.ru/ru/prod/sberbankid/v2.1/userinfo - для подключений через ФПСУ
https://api.sberbank.ru/ru/prod/sberbankid/v2.1/userinfo - для подключений через двусторонний TLS

Также, актуальные шлюзы для вызова API размещены на Портале разработчиков в документации по API - https://developer.sberbank.ru/api/5c9f5314e4b0388ba0f08b43 (ссылка доступна только зарегистрированным пользователям).

1.3.3. Параметры ответа

При успешной проверке запроса, Банк формирует ответ, содержащий данные клиента, и возвращает его в ответе типа HTTP 200 OK. Ниже приведен пример сообщения.

Ответ в случае успешной обработки запроса:

```
{
   "sub":"e327493e-979a-461f-9ca5-edfab9d6fbab",
   "family_name":"Фамилия",
   "given_name":"Имя",
   "middle_name":"Отчество",
   "birthdate":"0000-00-00",
   "identification":{
      "series":"00 00",
      "number":"000000",
      "issued_by":"Орган выдавший документ",
      "issued_date":"0000-00-00",
      "code":"000-000"
   },
   "inn":{
      "number":"0000000"
   },
   "snils":{
      "number":"0000000"
   },
   "email":"электронная почта",
   "phone_number": "+7 (000) 000000”,
 "email": “personal@mail.ru”,
   "gender":1
}
```

Ответ содержит JSON - объект, в котором находятся запрошенные данные пользователя.

Таблица 14. Описание полей ответа

№ п/п	Наименование поля	Описание	Обязательность	Пример
1	iss	URL сервиса, сформировавшего ID Token.	Да	"iss":"https://online.sberbank.ru/CSAFront/index.do"
2	sub	Неизменный уникальный идентификатор клиента, передаваемый внешним потребителям.	Да	"sub":"e327493e-979a-461f-9ca5-edfab9d6fbab"
3	aud	Идентификатор внешней АС (client_id) (получено в письме от Банка). При получении ответа UserInfo Партнеру необходимо сравнить полученное значение aud со значением своего client_id. Если значения не совпадают, то Партнер должен завершить сценарий без авторизации клиента на своем ресурсе.	Да	"aud":"7dd4327b-765f-4e1a-88f2-42cb6007ef52"
4	family_name	Фамилия	Нет	"family_name":"Фамилия"
5	given_name	Имя	Нет	"given_name":"Имя"
6	middle_name	Отчество	Нет	"middle_name":"Отчество"
7	birthdate	Дата рождения (формат ГГГГ-ММ-ДД)	Нет	"birthdate":"0000-00-00"

8	identification
	series
	number
	issued_by
	issued_date
	code
		Полные данные паспорта
		серия
		номер
		орган, выдавший паспорт
		дата выдачи паспорта (формат ГГГГ-ММ-ДД)
		код подразделения
			Нет	"identification":{
				"series":"00 00",
				"number":"000000",
				"issued_by":"Орган выдавший документ",
				"issued_date":"0000-00-00",
				"code":"000-000"
				}

9	inn number	ИНН номер	Нет	"inn":{
				"number":"0000000"
				}

10	snils number	СНИЛС номер	Нет	"snils":{
				"number":"0000000"
				}

11	phone_number	Номер телефона	Нет	"phone_number": "+7 (000) 000000”
12	email	Адрес электронной почты	Нет	"email": “personal@mail.ru”
13	gender	Пол 1 – мужчина; 2 – женщина;	Нет	"gender":1,

14	driving_license	Номер водительского удостоверения	Нет	"driving_license":{ "number":"111111" },
15	international_passport	Заграничный паспорт гражданина РФ серия документа (формат 00 00) номер документа (формат 000000) кем выдан дата выдачи (формат ГГГГ-ММ-ДД) дата окончания (формат ГГГГ-ММ-ДД) имя фамилия	Нет	"international_passport":{ "series":"777", "number":"333", "issued_by":"рога и копыта", "issued_date":"1981-01-01", "planned_end_date":"1999-02-01", "name":"name", "surname":"surname" },
16	priority_doc	Данные документа, удостоверяющего личность тип серия документа (формат 00 00) номер документа (формат 000000) кем выдан дата выдачи (формат ГГГГ-ММ-ДД) код Выводится один документ в соответствии со списком приоритетов: • Паспорт РФ • Загранпаспорт гражданина РФ • Военный билет • Паспорт моряка • Временное удостоверение • Паспорт иностранного граданина • Вид на жительство иностранного гражданина	Нет	"priority_doc":{ "type":17, "series":"777", "number":"333", "issued_by":"рога и копыта", "issued_date":"1981-01-01", "code":"adasd" },
17	citizenship	Гражданство: последняя по актуальности страна гражданства (наименование на русском языке) и ее код (ОКСМ, буквенное обозначение – Альфа-3)	Нет	"citizenship":{ "country_code":"countryCode", "country_name":"countryName" },
18	place_of_birth	Место рождения – город рождения клиента (текстовое поле, не кодируется)	Нет	" place_of_birth":”Nsk”
19	address_reg	Адрес регистрации полный адрес код ФИАС почтовый индекс	Нет	"address_reg":{ "full_address":"fullAddress", "fias_code":"fiasCode", "post_index":"postIndex",

		страна регион район город поселение улица дом строение корпус квартира		"country":"country", "region":"region", "district":"district", "city":"city", "settlement":"settlement", "street":"street", "house":"house", "building":"building", "bulk":"bulk", "apartment":"apartment" },
20	work_address	Рабочий адрес полный адрес код ФИАС почтовый индекс страна регион район город поселение улица дом строение корпус квартира	Нет	"work_address":{ "full_address":"fullAddress", "fias_code":"fiasCode", "post_index":"postIndex", "country":"country", "region":"region", "district":"district", "city":"city", "settlement":"settlement", "street":"street", "house":"house", "building":"building", "bulk":"bulk", "apartment":"apartment" },
21	address_of_actual_residence	Адрес места жительства полный адрес код ФИАС почтовый индекс страна регион район город поселение улица дом строение корпус квартира	Нет	"address_of_actual_residence":{ "full_address":"fullAddress", "fias_code":"fiasCode", "post_index":"postIndex", "country":"country", "region":"region", "district":"district", "city":"city", "settlement":"settlement", "street":"street", "house":"house", "building":"building", "bulk":"bulk", "apartment":"apartment" },
22	addresses	Адрес регистрации полный адрес код ФИАС почтовый индекс страна регион район город поселение улица дом строение корпус квартира +	Нет	"address_of_actual_residence":{ "full_address":"fullAddress", "fias_code":"fiasCode", "post_index":"postIndex", "country":"country", "region":"region", "district":"district", "city":"city", "settlement":"settlement", "street":"street", "house":"house", "building":"building", "bulk":"bulk", "apartment":"apartment" },

		Адрес места жительства полный адрес код ФИАС почтовый индекс страна регион район город поселение улица дом строение корпус квартира		"address_reg":{ "full_address":"fullAddress", "fias_code":"fiasCode", "post_index":"postIndex", "country":"country", "region":"region", "district":"district", "city":"city", "settlement":"settlement", "street":"street", "house":"house", "building":"building", "bulk":"bulk", "apartment":"apartment" },
23	is_company_employee	Признак сотрудника ПАО "Сбербанк"	Нет	"is_company_employee":true,
24	sts	Номер СТС	Нет	"sts":{ "number":"00 00 00000" },
25	is_self_employed	Признак самозанятого	Нет	"is_self_employed":true, "is_self_employed":false,
26	previous_maindoc	Передается полный список данных предыдущих документов удостоверяющих личность: Серия документа Номер документа Кем выдан Дата выдачи	Нет	"previous_identification":{ "series":"00 00", "number":"000 000", "issued_by":"Орган выдавший документ", "issued_date":"0000-00-00", },
27	previous_name	Передается полный список предыдущих ФИО. При этом предыдущие ФИО могут содержать лишь корректировку опечаток. Предыдущая Фамилия Предыдущее Имя Предыдущее Отчество	Нет	"previous_family_name": "Предыдущая фамилия", "previous_given_name": "Предыдущее имя", "previous_middle_name": "Предыдущее отчество"
28	education	Сведения об образовании Код + описание • начальное, • среднее, • среднее специальное, • высшее, • магистратура и пр.	Нет	"education":{ "code":"1", "description":"начальное", },
29	place_of_work	Наименование организации	Нет	"place_of_work": "место работы",

		(место работы)		
30	job_title	Наименование должности	Нет	"job_title": "должность"
31	marital_status	Семейное положение Код + описание • холост, • женат, • разведен, • вдовец/ вдова, • раздельное проживание, • гражданский брак и пр.	Нет	"marital_status":{ "code":"1", "description":"холост", },
32	work_number	Номер рабочего телефона	Нет	"work_phone_number": "00000000000"
33	home_number	Номер домашнего телефона	Нет	"home_phone_number": "000000"

1.3.4. Описание ошибок.

Пример ответа в случае ошибки:

```
HTTP/1.1 400 Bad Request
Content-Type: application/json
Cache-Control: no-store
Pragma: no-cache
{
«error»: «invalid_request»
}
```

Если предъявленный Access Token не найден или уже использован, возвращается ответ с ошибкой типа **HTTP 401 unauthorized**.

Таблица 15. Типы возвращаемых ошибок

№ п/п	Описание типа ошибки	Тип возвращаемой ошибки
1	В заголовке Authorization отсутствует какое-либо значение или в запросе присутствуют дополнительные атрибуты.	invalid_request
2	Указан неверный тип токена.	invalid_request

2. Список доступных данных профиля

Партнер сможет получать данные по клиенту согласно обозначенным в запросе скоупам (группам данных), которые выдаются при регистрации системы в банке. Таблица с максимальным набором скоупов представлена ниже. Однако система партнера может быть подписана на меньшее количество скоупов, чем указано в таблице. При запросе неразрешенных данному партнеру скоупов будет возвращена ошибка (invalid_scope).

Важно! Если по клиенту в банке нет запрошенных данных или клиент запретил передачу этих данных, то данный скоуп в ответе будет отсутствовать.

В рамках данной версии API по клиентам могут быть возвращены следующие данные:
фамилия, имя, отчество (при наличии), номер телефона (при наличии), реквизиты паспорта гражданина РФ (при наличии), адрес электронной почты (при наличии), ИНН (при наличии), СНИЛС (при наличии), дата рождения, пол, номер водительского удостоверения (при наличии), реквизиты заграничного паспорта гражданина РФ (при наличии), реквизиты документа, удостоверяющего личность (при наличии), гражданство (при наличии), место рождения (при наличии), адрес регистрации (при наличии), рабочий адрес (при наличии), адрес места жительства (при наличии), признак сотрудника Сбера.

Таблица 15. Список данных профиля

№ п/п	Наименование поля	Передаваемые данные	Ключ	Обязательность
1	Идентификатор клиента	Идентификатор клиента (sub в ID_Token и UserInfo)	openid	Да
2	Фамилия, имя, отчество	Фамилия Имя Отчество	name	Да
3	Паспорт гражданина РФ	Серия документа Номер документа Кем выдан Дата выдачи Код подразделения, выдавшего документ	maindoc	Нет
4	Адрес электронной почты	E-mail	email	Нет
5	ИНН	Идентификационный номер налогоплательщика	inn	Нет
6	СНИЛС	Страховой номер индивидуального лицевого счета	snils	Нет
7	Номер мобильного телефона	Номер мобильного телефона	mobile	Нет
8	Дата рождения	Дата рождения	birthdate	Нет

9	Пол	Пол	gender	Нет
10	Водительское удостоверение	Номер водительского удостоверения	driving_license	Нет
11	Заграничный паспорт РФ	Серия документа Номер документа Кем выдан Дата выдачи Дата окончания Имя Фамилия	international_passport	Нет
12	Документ по приоритету	Тип Серия документа Номер документа Кем выдан Дата выдачи Код	priority_doc	Нет
13	Гражданство	Код страны (ОКСМ, буквенное обозначение – Альфа-3) Название страны (на русском языке)	citizenship	Нет
14	Место рождения	Город рождения (текстовое поле, не кодируется)	place_of_birth	Нет
15	Адрес регистрации	Полный адрес Код ФИАС Почтовый индекс Страна Регион Район Город Поселение Улица Дом Строение Корпус Квартира	address_reg	Нет
16	Рабочий адрес	Полный адрес Код ФИАС Почтовый индекс Страна Регион Район Город Поселение Улица Дом Строение	work_address	Нет

		Корпус Квартира		
17	Адрес места жительства	Полный адрес Код ФИАС Почтовый индекс Страна Регион Район Город Поселение Улица Дом Строение Корпус Квартира	address_of_actual_residence	Нет
18	Адрес регистрации + Адрес места жительства	Возвращает полный список данных адреса регистрации и адреса места жительства	addresses	Нет
19	Признак сотрудника	Признак сотрудника ПАО "Сбербанк"	is_company_employee	Нет
20	Свидетельство о регистрации транспортного средства (СТС)	Номер СТС	sts	Нет
21	Признак самозанятого	Признак самозанятого	is_self_employed	Нет
22	Реквизиты ранее выданного документа, удостоверяющего личность	Передается полный список данных предыдущих документов удостоверяющих личность: Серия документа Номер документа Кем выдан Дата выдачи	previous_maindoc	Нет
23	Предыдущие фамилия, имя, отчество	Передается полный список предыдущих ФИО: Предыдущая Фамилия Предыдущее Имя Предыдущее Отчество	previous_name	Нет
24	Образование	Сведения об образовании	education	Нет
25	Место работы	Наименование организации	place_of_work	Нет

26	Должность	Наименование должности	job_title	Нет
27	Семейное положение	Семейное положение	marital_status	Нет
28	Рабочий телефон	Номер рабочего телефона	work_number	Нет
29	Домашний телефон	Номер домашнего телефона	home_number	Нет

Пример UserInfo для групп данных **name+birthdate+mobile**:

```
{
"iss":  "https://online.sberbank.ru/CSAFront/index.do",
"sub":
"74c64d08bdd5e6f2b94770e9fed9342b9054f22bea1571e68448c8cae83e0d80ec206549
e11d13fc",
"aud":  "DA5278AC-A07F-C01A-B2D3-C231DBB2E20F",
"family_name":  "Иванов",
"given_name":  "Иван",
"middle_name":  "Викторович",
"birthdate":  "1981-01-01",
"phone_number":  "+7 (964) 6735442"
}
```

Пример UserInfo для групп данных **name+birthdate**:

```
{
 "iss":  "https://online.sberbank.ru/CSAFront/index.do",
 "sub":
"74c64d08bdd5e6f2b94770e9fed9342b9054f22bea1571e68448c8cae83e0d80ec206549
e11d13fc",
 "aud":  "DA5278AC-A07F-C01A-B2D3-C231DBB2E20F",
 "family_name":  "Иванов",
 "given_name":  "Иван",
 "middle_name":  "Викторович",
 "birthdate":  "1981-01-01"
}
```

Пример UserInfo для групп данных **name+mobile,** в случае отсутствия номера телефона клиента в профиле клиента:

```
{
 "iss":  "https://online.sberbank.ru/CSAFront/index.do",
 "sub":
"74c64d08bdd5e6f2b94770e9fed9342b9054f22bea1571e68448c8cae83e0d80ec206549
e11d13fc",
 "aud":  "DA5278AC-A07F-C01A-B2D3-C231DBB2E20F",
 "family_name":  "Иванов",
 "given_name":  "Иван",
 "middle_name":  "Викторович"
}
```