Авторизация пользователя v1

Ресурс /v1/oauth/authorize

Запрос кода авторизации используется для проверки успешной аутентификации клиента в АС СББОЛ. Если клиент переходит из банка по ссылке, Партнёру требуется предварительно распознать переход клиента по ссылке. После распознавания партнёр должен перенаправить клиента обратно вместе с запросом кода авторизации, сформированным с помощью метода GET. В том случае, если клиент начинает работу на стороне сервиса партнёра без предварительного перехода из банка по ссылке, партнёр должен отправить клиента вместе с запросом кода авторизации, сформированным с помощью метода GET, без предварительных действий на соответствующий сервис SSO СББОЛ.

Если партнёрский сервис переадресовал token-клиента на страницу аутентификации sms-клиента, то после ввода логина и пароля появится подсказка для корректной аутентификации. При первичной аутентификации клиенту будет предложено подписать оферту для возможности Партнера отправлять запросы по счетам клиента.

При переходе клиента из банка на сервис партнёра в ссылке присутствует параметр **loginDCB=true** (признак перехода клиента из банка), указывающий партнёру на необходимость перенаправления клиента обратно для авторизации на банковском сервере. Дополнительно в ссылке передаются два параметра:

- Тип пользователя:
  - userType=WEB — для пользователей, использующих SMS-подтверждение;
  - userType=Token — для пользователей, осуществляющих аутентификацию с помощью устройства защиты.
- Значение порта callbackPort (для пользователей использующих средства защиты/USB-средства аутентификации).

Эти параметры необходимы для корректного формирования хоста для запросов **Authorization Code, Access Token** и **User Info**.
Пример: Для пользователя, использующего SMS-подтверждение при аутентификации, для перехода из банка в сервис партнёра будет сформирована ссылка: ***https://example.ru?loginDCB=true&userType=WEB***

Для пользователя, использующего аутентификацию с помощью устройства защиты, для перехода из банка в сервис партнёра будет сформирована ссылка: ***https://example.ru?loginDCB=true&userType=Token&callbackPort=28016***

Параметры запроса и ответа

Параметр	Описание
scope	Значение "openid" и к нему добавляется **через пробел** наименование дополнительного партнёрского scope,

Параметр	Описание
	полученного от менеджера партнёра при регистрации приложения (обязательный параметр при обращении к ресурсу)
response_type	Тип запроса (значение должно быть всегда "code" согласно требованию по обеспечению безопасности, разрешающее использовать исключительно Auth Code Flow (обязательный параметр))
client_id	Банковский идентификатор сервиса партнёра полученный от менеджера партнёра при регистрации приложения (обязательный параметр)
state	Параметр для предотвращения CSRF-атак (обязательный параметр)
nonce	Значение используется для защиты от атак с повторением токенов (указанное в запросе значение должно соответствовать nonce-значению, возвращаемому в ответе, при этом значение nonce должно быть уникальным в сеансе клиента и сложным для угадывания (обязательный параметр))
redirect_uri	Ссылка на страницу сервиса партнёра, на которую необходимо перенаправить клиента после успешной авторизации (обязательный параметр)
code_challenge	Код вызова, полученный из кода проверки (code_verifier) путем применения метода преобразования code_challenge_method (необязательный параметр)
code_challenge_method	Метод преобразования, который был применен к коду проверки (code_verifier), для получения значения кода вызова (code_challenge) (необязательный параметр)
location	Значение redirect_uri для client_id, указанного в запросе авторизации
code	Значение кода авторизации. Значение кода авторизации формируется произвольно в формате UUID (случайное значение) плюс "1" или "2" через дефис, например, CD6A56FD-A9C7-4152-AA1D-FA57E550F6AC-2.

Пример запроса	Пример ответа
GET ic/sso/api/v1/oauth/authorize HTTP/1.1 Host: *edupirfintech.sberbank.ru:9443* response_type=code &scope=examplescope &client_id=1001 &state=af0ifjsldkj &nonce=n-0S6_WzA2Mj &redirect_uri=*https%3A%2F%2example.ru%2Fcb*	HTTP/1.1 302 Found Location: *https%3A%2F%2example.ru%2Fcb HTTP/1.1* code=CD6A56FD-A9C7-4152-AA1D-FA57E550F6AC-2 &state=af0ifjsldkj &nonce=n-0S6_WzA2Mj

В случае обнаружения ошибок client_id и redirect_uri банк сформирует ответ HTTP 400 с ошибкой invalid_grant и описанием:

- Redirect uri *https://example.ru* is invalid - неверный redirect_uri
- Invalid client secret for authz code 'xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx-x'

В случае если банк вернёт ответ с ошибкой авторизации, сервис партнёра должен корректно его обработать. Так как первый шаг взаимодействия (аутентификация) осуществляется через браузер клиента, банк вернёт HTTP ответ 302 Found. Это необходимо для того, чтобы сервис партнёра имел возможность обработать ошибку и отобразить клиенту необходимую ему страницу, например, страницу авторизации (рекомендовано).

Формирование параметра State

Параметр служит для защиты от CSRF-атак. Для каждого партнёра при регистрации его сервиса в банке указывается ссылка на страницу сервиса партнёра, на которую необходимо перенаправить клиента из банка после успешной авторизации (redirect_uri). При поступлении на redirect_uri запроса с кодом авторизации сервис партнёра должен проверять, что обмен данными происходит в рамках инициированного им самим взаимодействия.
Значение **State** должно быть защищено от подбора и храниться таким образом, чтобы его не мог изменить никто, кроме сервиса партнёра. Рекомендуется использовать идентификатор сессии клиента в сервисе партнёра или один из его производных (например, хэш этого идентификатора), при этом может использоваться любой другой механизм генерации случайного значения достаточной длины для предотвращения подбора. В общем случае оптимально использовать строку длиной не менее 36 символов с проверкой регистра.

Формирование параметра Nonce

Параметр служит для защиты от replay-атак, то есть для предотвращения обработки одного и того же запроса несколько раз. Значение **Nonce** должно быть защищено от подбора и храниться таким образом, чтобы его не мог изменить никто, кроме сервиса партнёра. Рекомендуется использовать случайное значение, сохраняемое в сессии клиента в сервисе партнёра. Для повторного запроса это значение должно заменяться новым. В общем случае оптимально использовать строку длиной не менее 10 символов с проверкой регистра.

Оферта

Для взаимодействия с сервисами партнёра клиент должен подписать оферту, в которой указано к каким данным и на какой срок предоставлен доступ партнёру. При предоставлении партнёру доступа к данным своих счетов клиент в явном виде указывает каждый счёт, к которому предоставляет доступ.
После получения банком запросов со стороны партнёра, касающихся клиентских данных, выполняется проверка наличия между партнёром и клиентом активной оферты, а при её отсутствии или истечения срока действия партнёру возвращается сообщение об ошибке доступа к клиентским данным. После того как партнёр определит по ссылке переход клиента из банка, он может получить код авторизации (Authorization Code), авторизационный токен для получения доступа к данным клиента (Access Token), идентификационные атрибуты клиента, которые клиент разрешил предоставить партнеру в рамках оферты (при наличии активной оферты).

Формирование хоста для отправки запроса Authorization Code

В зависимости от типа клиента (значения параметра userType, переданного при первом переходе клиента со стороны банка на сервис партнёра, либо информации полученной от клиента для перехода клиента на сервис банка со стороны партнера) алгоритм формирования хоста для запросов авторизации должно быть различным:

- Если userType=WEB, то значение хоста будет, например, ***https://edupir.testsbi.sberbank.ru:9443/ic/sso/api/v1/oauth/authorize?{параметры}***
- Если userType=Token, то значение хоста будет, например, ***http:///localhost:28016/ic/sso/api/v1/oauth/authorize?{параметры}***, где 28016 — значение параметра callbackPort, переданного при первом переходе клиента из банка на сервис партнёра.

Указанные выше значения хостов приведены для примера. Реальные значения сообщаются по не автоматизируемым каналам.
Значение userType=Token может быть определено по содержимому ID Token (например, по наличию SID2), также по параметрам acr и amr (см. Декодирование ID Token).

Ресурс /v1/oauth/token

После получения кода авторизации и проверки корректности успешного ответа необходимо запросить авторизационный токен к данным клиента **Access Token**, который возвращается вместе с ID Token, содержащим клиентские идентификационные данные, и ключом **Refresh Token**.

Шаги

1. Определить переход клиента по ссылке (опционально)
2. Получить код авторизации
3. Отправить запрос
4. Декодировать ID Token
5. Актуализация авторизационного токена

Обмен кода авторизации на **Access Token** происходит через меж серверные каналы (REST-сервис), поэтому значение хоста не зависит от типа клиента (userType). В случае возникновения ошибки при попытке получить AccessToken по коду авторизации по каким-либо причинам (например, неправильно передан client_secret) код авторизации удаляется и все дальнейшие попытки получить по нему AccessToken (даже с корректными параметрами) будут приводить к ошибке с текстом "Unknown code = <код_авторизации>".
Запрос должен производиться только методом POST с типом данных application/x-www-form-urlencoded.

Для работы с собственными счетами Партнёру необходимо получить access_token под пользователем своей организации, процесс аналогичен получению клиентского токена.

Модель запроса и ответа

Параметр	Описание
grant_type	Значение должно быть "authorization_code" (обязательный параметр)
code	Значение кода авторизации (обязательный параметр)
client_id	Банковский идентификатор сервиса партнёра полученный от менеджера партнёра при регистрации приложения (обязательный параметр)
client_secret	Авторизационный ключ партнёра (обязательный параметр). Генерируется банком в момент заведения сервиса и передается партнёру оффлайн.
redirect_uri	Ссылка на страницу сервиса партнёра, на которую необходимо перенаправить клиента после успешной авторизации (обязательный параметр)
code_verifier	Код проверки, на основе которого получили код вызова (code_challenge) путем применения метода преобразования code_challenge_method (необязательный параметр)
access_token	Авторизационный токен к данным организации
refresh_token	Токен обмена
token_type	Всегда равен "Bearer"
expires_in	Время жизни access_token в секундах
id_token	Закодированный в Base64URL набор атрибутов клиента, необходимых для идентификации пользователя. Атрибуты разделены символами «.», каждый необходимо декодировать отдельно.

Пример запроса	Пример ответа
POST /ic/sso/api/v1/oauth/token Host: *edupirfintech.sberbank.ru:9443* Content-Type: application/x-www-form-urlencoded grant_type=authorization_code &code= CD6A56FD-A9C7-4152-AA1D-FA57E550F6AC-2 &client_id=1001 &client_secret=Ac03df04fff8 &redirect_uri=*https%3A%2F%2Fclient.example.ru%2Fcb*	HTTP/1.1 200 OK Content-Type: application/json Cache-Control: no-store Pragma: no-cache { "access_token": "c76fb018-27c9-43f7-a751-62646eda7e1a-1", "token_type": "Bearer", "expires_in": 3600, "refresh_token": "03e0be32-e72e-47ec-b740-a00b333a8ac4-1", "id_token": " eyJ0eXAiOiJKV1QiLCJhbGciOiJnb3N0MzQuMTAtMjAxMiJ9. eyJzdWIiOiI2ODM4ZjM1MmI0YzQ0YjZjOGFmYTY0ZTFlZDJ mNjg1NzM0MjE4NDAwNjZiZTU3MTgxZjNiN2IyYjc1NThkY mJlIiwiYXVkIjoiMTAwMTMiLCJhY3IiOiJsb2EtMyIsImF6cCI6I

Пример запроса	Пример ответа
	jEwMDEzIiwiYXV0aF90aW1lIjoxNTgyMzcwNDk5LCJhbXIiOi J7cHdkLCBtY2EsIG1mYSwgb3RwLCBzbXN9IiwiaXNzIjoiaHR 0cDovL3NidC1vYWZzLTYzODo5MDgwL2ljZGsiLCJleHAiOjE 1ODIzNzA4MDEsImlhdCI6MTU4MjM3MDUwMSwibm9uY2 UiOiI3YmU2NmFjOS1kMDdjLTQ5NjctYWRlZC1jYTI3MGEyN 2U5ZTgiLCJ1c2wiOiJQYXJ0bmVyMzMyMiJ9.IWCyZzOk5nT0 GWfhi9n3Nqy8Ii8mJ1eeFS7YRoE-l74lqo6BLksCuaVXt2ErMZ YmDyyZscu7ISm0n-YsSrgZPQ" }

При получении ответа сервис партнёра должен проверить его корректность в соответствии со спецификацией. Рекомендации по проверке ответа на запрос **Access Token** описаны в спецификациях:

- https://tools.ietf.org/html/rfc6749#section-5
- http://openid.net/specs/openid-connect-core-1_0.html#IDTokenValidation
- http://openid.net/specs/openid-connect-core-1_0.html#CodeFlowTokenValidation

Декодирование ID Token

При необходимости декодировать ID Token следует воспользоваться алгоритмом Base64URL Encoding.
Согласно спецификации JSON Web Token (JWT) ID Token должен быть представлен структурой вида:

- Алгоритм подписи (по сертификату технологического криптопрофиля Банка, который можно получить запросом /crypto), Header(Заголовок);
- ID_token, Payload (Полезная нагрузка);
- Электронная подпись

Каждая часть ответа, разделённая точкой, должна декодироваться отдельно. Для проверки подписи в поле id_token на стороне клиента, необходимо вычислить подпись публичным ключом Банка, декодировав блок Payload по Base64URL (содержимое между двумя точками). Далее необходимо сравнить полученное значение с блоком Электронная подпись (содержимое после второй точки), декодированным по Base64URL.

Пример ответа Id Token	Пример декодированного ответа

eyJhbGciOiJnb3N0MzQtMTAuMjAxMiJ9.eyJzdWIiOiJ
hMWNhOGYyMTQ4MDc1M2IyMzI1MTZiYzk4NmNiZ
mM4Yjc5MjNiYWI3ODczODQzZWM0ZjQ1MTA4MmI
0YTg3NjFjIiwiaXNzIjoiaHR0cDovL3NidC1vYWZzLTYz
ODo5MDgwL2ljZGsiLCJhdWQiOiIyMDg1IiwiZXhwIjox
NTE4Njg2MDI1LCJpYXQiOjE1MTg2ODU3MjUsImF1d
GhfdGltZSI6MTUxODY4NTM4NSwiYWNyIjoibG9hLT
MiLCJhbXIiOiJ7cHdkLCBtY2EsIG1mYSwgb3RwLCBz
bXN9IiwiYXpwIjoiMjA4NSIsIm5vbmNlIjoiOTc2MjgwY
2ZmZTg5In0=.AGnvv73vv73vv71iFGM977+9We+/vU
vvv73vv704Re+/ve+/vWnvv73vv71y77+9dQh+77+9
GR1377+9Ml8T77+9ae+/vTbvv73Gle+/vU1t77+977+
977+977+97LKk77+9XyQRPFEl77+977+9
	{
	"typ": "JWT",
	"alg": "gost34.10-2012"
	}
	{
	"sub": "a1ca8f21480753b232516bc986cbfc8b7923bab7873843ec4f451082b4a8761c",
	"iss": "https://edupirfintech.sberbank.ru:9443",
	"aud": "2085",
	"exp": 1518686025,
	"iat": 1518685725,
	"auth_time": 1518685385,
	"acr": "loa-3",
	"amr": "{pwd, mca, mfa, otp, sms}",
	"azp": "2085",
	"nonce": "976280cffe89"
	}
	AGnvv73vv73vv71iFGM977+9We+/vUvvv73vv704Re+/ve+/vWnvv73vv71y77+9dQh+77+9
	GR1377+9Ml8T77+9ae+/vTbvv73Gle+/vU1t77+977+
	977+977+97LKk77+9XyQRPFEl77+977+9

Параметр	Описание
iss	URL сервиса, сформировавшего ID Token
sid2	Идентификатор сессии устройства защиты
sub	Хэш банковского идентификатора пользователя
aud	Идентификатор сервиса партнёра, для которого формируется ID Token
nonce	Значение параметра nonce из запроса кода авторизации
exp	Время, после которого ID Token не принимается для обработки. Формат поля Unix time.
iat	Время формирования ID Token. Формат поля Unix time.
auth_time	Время банковской аутентификации пользователя. Формат поля Unix time.
acr	Уровень аутентификации пользователя: • acr=loa-2 (пользователь аутентифицируется с помощью устройства защиты) • acr=loa-3 (пользователь использует подтверждение по одноразовому SMS-паролю для аутентификации)

Параметр	Описание
amr	Методы аутентификации: • Amr=pwd — пользователь аутентифицируется с помощью устройства защиты; • amr= {pwd, mca, mfa, otp, sms} — пользователь использует подтверждение по одноразовому SMS-паролю для аутентификации
azp	Идентификатор сервиса партнёра (рекомендуется проверить идентификатор сервиса партнёра, для которого был сформирован полученный ID Token)
HashOrgId	Хэш от идентификатора организации, с которой связан пользователь

Если банку не удаётся предоставить какие-то данные, такой атрибут отсутствует в ответе (не должно быть атрибутов со значением NULL).

Актуализация авторизационного токена

Ключ Refresh Token предназначен для получения нового Access Token и Refresh Token в случае компрометации или истечения времени жизни Access Token.

Модель запроса и ответа

Параметр	Описание
grant_type	Значение должно быть "refresh_token" (обязательный параметр)
client_id	Банковский идентификатор сервиса партнёра полученный от менеджера партнёра при регистрации приложения (обязательный параметр)
client_secret	Авторизационный ключ партнёра (обязательный параметр). Генерируется банком в момент заведения сервиса и передается партнёру оффлайн.
refresh_token	Токен обмена (обязательный параметр)

Пример запроса	Пример ответа

POST /ic/sso/api/v1/oauth/token
Host: *edupirfintech.sberbank.ru:9443*
grant_type=refresh_token
&refresh_token= 668c708e-865a-42b1-8fe9-846b92b8b14e-1
&client_id=1001
&client_secret=Ac03df04fff8
	Content-Type: application/json
	Cache-Control: no-store
	Pragma: no-cache
	{
	"access_token": "6e55338d-7a7a-4b66-bc78-248d52eeb1dc-1",
	"token_type": "Bearer",
	"expires_in": 3600,
	"refresh_token":"c77fb018-27c9-43f7-a751-62646eda7e1a-1"
	}

- Рекомендуется обновлять ключи доступа (Access Token/Refresh Token) как минимум один раз за время жизни Refresh Token. В случае если Refresh Token потеряет актуальность (не будет обновлён до истечения времени жизни), партнёр сможет получить актуальные ключи доступа только после повторной авторизации клиента в сервисе партнёра через СББОЛ.

- Если при обновлении ключей доступа не был получен ответ из банка. то рекомендуется повторно отправить запрос на актуализацию ключей в течение 1 часа от момента отправки первой попытки.
- Коды возврата

Ресурс /v1/oauth/user-info

Ресурс позволяет Партнёру получать данные о подключенном клиенте (если клиент "разрешил" Партнеру доступ к своим данным), которые могут включать в себя информацию об организации пользователя, например:

- идентификатор клиента (организации пользователя) (**hashOrgId**)

Каждому партнёру при регистрации подключают одинаковый для всех обязательный системный **scope**, который включает в себя только идентификатор клиента. Помимо этого для каждого Партнёра подключается дополнительный уникальный scope. Перечень данных дополнительного **scope** определяется индивидуально менеджером партнёра.

Шаги

1. Определить переход клиента по ссылке (опционально);
2. Получить код авторизации
3. Получить токен авторизации
4. Отправить запрос

Запрос User Info должен производиться методом GET с использованием заголовка Authorization и передаваемых в нём значения **Bearer** (всегда должно соответствовать ответу в параметре token_type=) и значения из **access_token.** При запросе Партнёром информации о пользователе своей организации не заполняется claim "accounts".

Параметры запроса и ответа

Header Parametrs		**Пример запроса**
Authorization	String Access token полученный через SSO. Пример: Bearer daf9a14c-821d-4bde-9c10-0e56e63d54a0-1	GET/ic/sso/api/v1/oauth/user-info HTTP/1.1 Host: *https://edupirfintech.sberbank.ru:9443* Authorization: Bearer CD6A56FD-A9C7-4152-AA1D-FA57E550F6AC-2

Каждая часть ответа, разделённая точкой, должна декодироваться отдельно.

Пример ответа	**Пример декодированного ответа**
HTTP/1.1 200 OK Content-Type: application/jwt w0Y8g0L_QvtC00L_QuNGB0YwiLCJzdW1tT2ZmZXJTbWFydENyZWRpdCI6MCwib3JnQ WN0dWFsQWRkcmVzcyI6ItCg0J7QodCh0JjQmdCh0JrQkNCvINCk0JXQlNCV0KDQkNC m0JjQrywgMTIxMzUxLCDQsy7QnNC-0YHQutCy0LAsINGD0LsuINCa0L7RhtGO0LHQu NC90YHQutC0LPQviwg0LTQvtC8IDQsINGB0YLRgC4gMywg0L7RhC4gMTLQpiIsIkhhc2 hPcmdJZCI6ImIzNGI2YTQ1MzM4NjJlNTE2N2U5ODc4NWIxZDIzYWE3ODlkYWMyMTY wYjNjN2M5Y2MzOTIyMWJkMzNmZmM4NWEiLCJpc0lkZW50aWZpZWQiOnRydWUsI m9yZ1BwcmJJZCI6MTE5MzkwMzUwMjcyNTI2MTcxMSwidXNlclBvc2l0aW9uIjoi0JHRg 9GF0LPQsNC70YLQtdGAIiwiZW1haWwiOiJtZWRpYSsrQHNidC5ydSIsIm9yZ0d1aWQi OiIxMTFjOWMxMy02NGRhLTQ3OTAtOWM5Mi00YjA1NDk5YjZjNGIiLCJpbm4iOiI3Nz MzODEyOTIwIiwib3JnSnVyaWRpY2FsQWRkcmVzcyI6ItCg0J7QodCh0JjQmdCh0JrQk NCvINCk0JXQlNCV0KDQkNCm0JjQrywgMTI1MzY3LCDQsy7QnNC-0YHQutCy0LAsIN C_0YDQvtC10LfQtC4g0JLRgNCw0YfQtdCx0L3Ri9C5LCDQtNC-0LwgMTAsINC-0YQuID EiLCJhY3RpdmUiOjEsIm9yZ0Z1bGxOYW1lIjoi0J7QsdGJ0LXRgdGC0LLQviDRgSDQvtC z0YDQsNC90LjRh9C10L3QvdC-0Lkg0L7RgtCy0LXRgtGB0YLQstC10L3QvdC-0YHRgtG	{ **"sub":** "7f5e5e42cc66973f31fe8f65bfc4460a808fc197d955582989a01282fac14c9c", **"orgOktmo":** "45000000000", **"orgKpp":** "773301001", **"iss":** "http://edupirfintech.sberbank.ru:9443", **"OrgName":** "ООО \"Мед Экспресс\"", **"orgId":** 1123234, **"individualExecutiveAgency":** 1, **"userSignatureType":** "Единственная подпись", **"summOfferSmartCredit":** 0, **"orgActualAddress":** "РОССИЙСКАЯ ФЕДЕРАЦИЯ, 121351, г.Москва, ул. Коцюбинского, дом 4, стр. 3, оф. 12Ц", **"HashOrgId":** "b34b6a4533862e5167e98785b1d23aa789dac2160b3c7c9cc39221bd33ffc85a", **"isIdentified":** true, **"orgPprbId":** 1193903502725261800, **"userPosition":** "Бухгалтер", **"email":** "media++@sbt.ru", **"orgGuid":** "111c9c13-64da-4790-9c92-4b05499b6c4b", **"inn":** "7733812920", **"orgJuridicalAddress":** "РОССИЙСКАЯ ФЕДЕРАЦИЯ, 125367, г.Москва, проезд. Врачебный, дом 10, оф. 1",

Пример ответа	Пример декодированного ответа
M0Y4gXCLQnNC10LQg0K3QutGB0L_RgNC10YHRgVwiIiwidGJJZGVudENvZGUiOiIzOC IsInVzZXJHdWlkIjoiOTQ3M2YyMzUtMDljOS00ZmNhLTgxYWUtYzAyZTkyYjNiNWRiIiw idXNlcklkIjo3NjgwNTgsInRlckJhbmsiOiLQnNC-0YHQutC0LLRgdC60LjQuSDQkdCw0L3 QuiDQodCx0LXRgNCx0LDQvdC60LAg0KDQpCIsImF1ZCI6Ijc0NjQzIiwidXNlclJvbGVzIjp bImJhbmtDbGllbnQiXSwib3JnQnVzaW5lc3NTZWdtZW50IjoiMDMiLCJ1c2VyQ3J5cHRv VHlwZSI6IlNNUyIsInVzZXJHcm91cHMiOiLQoNGD0LrQvtCy0L7QtNC40YLQtdC70Ywi CJvcmdPa3BvIjoiMTE0MzkyMDciLCJzYmJvbDMiOnRydWUsIm9yZ09ncm4iOiIxMTI3N zQ2NjU5MDQwIiwib2ZmZXJTbWFydENyZWRpdCI6ZmFsc2UsIm5hbWUiOiLQn9Cw YDRgtC90LXRgCDQn9Cw0YDRgtC90LXRgCDQn9Cw0YDRgtC90LXRgCIsImlucXVpcnl PcmRlciI6dHJ1ZX0.cIDbvnQrmmx1eVCku7OmyL6aUfEHZ0pD7FhNLU1FvAA0Yu454vD HtfgJSAkSxJiOgceUr2VGg40u8dw4sYk2A	**"active":** 1, **"orgFullName":** "Общество с ограниченной ответственностью \"Мед Экспресс\"", **"tbIdentCode":** "38", **"userGuid":** "9473f235-09c9-4fca-81ae-c02e92b3b5db", **"userId":** 768058, **"terBank":** "Московский Банк Сбербанка РФ", **"aud":** "74643", **"userRoles":** [**"bankClient"**], **"orgBusinessSegment":** "03", **"userCryptoType":** "SMS", **"userGroups":** "Руководитель", **"orgOkpo":** "11439207", **"sbbol3":** true, **"orgOgrn":** "1127746659040", **"offerSmartCredit":** false, **"name":** "Партнер Партнер Партнер", **"inquiryOrder":** true } cIDbvnQrmmx1eVCku7OmyL6aUfEHZ0pD7FhNLU1FvAA0Yu454vD HtfgJSAkSxJiOgceUr2VGg40u8dw4sYk2A

Список доступны параметров user-Info

Параметр	Описание
sid2	Идентификатор сессии токена
sub	Хэш идентификатора пользователя
iss	URL СББОЛ
aud	Идентификатор внешнего сервиса
name	Фамилия Имя Отчество
inn	ИНН организации
email	Адрес электронной почты

Параметр	Описание
phone_number	Номер телефона
HashOrgId	Хэш от идентификатор организации (orgId)
orgId	Идентификатор организации в СББОЛ
orgKpp	КПП
orgFullName	Название компании
OrgName	Наименование организации
orgOgrn	ОГРН компании
orgOkpo	ОКПО организации
orgOktmo	Общероссийский классификатор территорий муниципальных образований
orgActualAddress	Фактический адрес компании
orgJuridicalAddress	Юридический адрес компании
accounts	Счет, БИК, Кор. счет компании. **Примечание:** не заполняется при запросе Партнёром информации о пользователе своей организации.
terBank	Полное наименование подразделения кредитной организации
offerExpirationDate	Дата окончания оферты
userPosition	Должность
userRoles	Список роли пользователя
userSignatureType	Тип подписи
userGuide	Идентификатор пользователя клиента, который обратился к сервису
userId	Внутренний идентификатор пользователя
userGroups	Группы пользователя
sbbol3	Признак использования клиентом нового дизайна СББОЛ 3.0
userCryptoType	Тип криптопрофиля
tbIdentCode	Код территориального банка
individualExecutiveAgency	Признак Единоличный Исполнительный Орган
inquiryOrder	Признак доступности у пользователя функциональности получения справок
isIdentified	Признак идентификации пользователя
offerSmartCredit	Предодобренные предложения по смарт кредитам
summOfferSmartCredit	Сумма предодобренного предложения по смарт-кредитам
resident	Признак является организация резидент/нерезидент

Параметр	Описание
orgBusinessSegment	Бизнес сегмент. Возможные значения: 02-Микро бизнес (микро), 03-Малый бизнес, 04-Средний бизнес, 05-Крупный бизнес, 06-Крупнейший бизнес, 07-Клиенты машиностроения (ОПК), 08-Рег. госсектор, 09-БМО, 10-Фин. институты, 999-Не найдено сегмента в CRM Корпоративном.

Согласно спецификации JSON Web Token (JWT) User Info должен быть представлен структурой вида:

- Алгоритм подписи (Определяется по сертификату технологического криптопрофиля Банка, который можно получить запросом /crypto), Header (Заголовок);
- User-info, Payload (Полезная нагрузка);
- Электронная подпись;

При получении ответа сервис партнёра должен его декодировать и опционально проверить электронную подпись ответа. Система отправит ответ HTTP 200 OK с типом данных jwt.

User-info необходимо декодировать с помощью алгоритма base64url Encoding (по спецификации JWT). Преобразование BASE64URL, отличается от BASE64. Условно алгоритм можно представить следующим образом: Base64Url(x) := Base64(x).Split('=')[0].Replace('+', '-').Replace('/', '_'). Здесь функция Split(x), разбивает строку на части ([i] означает взятие i–ой части), используя символ разделитель x, функция Replace(x,y) заменяет все вхождения символа x на символ y.

В **user-info** включаются обязательные системные и дополнительные атрибуты пользователя для сервиса партнёра. Обязательные системные атрибуты: sub, iss, aud. Если банку не удаётся предоставить какие-то данные, такой атрибут отсутствует в ответе (не должно быть атрибутов со значением NULL). Проверка электронной подписи(3-я часть ответа) является опциональной и может выполняться партнёром при необходимости проверки неизменности данных user-info.

Дополнительная информация

Код возврата	Расшифровка кода возврата	Описание кода возврата	Причина возникновения
200	ОК	Успешный код возврата	
302	invalid_grant	Unknown code = xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx-x	Указан неверный код авторизации: а) несуществующий код авторизации; б) код авторизации, время жизни которого истекло; в) код авторизации, ранее использованный для получения access_token;
		Invalid client secret for authz code 'xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx-x	Указан неверный client_secret или client_id или client_secret и client_id.
		Redirect uri 'https://.ru' is invalid	Неверный redirect_uri.
		Unknown refresh token = xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx-x	Указан неверный refresh_token: а) несуществующий refresh_token; б) refresh_token, время жизни которого истекло; в) refresh_token, ранее использованный для получения нового access_token и refresh_token;
		Invalid client secret for refresh token xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx-x	Указан неверный client_secret (Запрос на получение новых access_token и refresh_token по действующему refresh_token)
		Ext service for authz code xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx-x is blocked	Внешний сервис заблокирован в системе (запрос на получение access_token и refresh_token по коду авторизации)
			Неверно указан адрес sso для подключения.
	unsupported_grant_type	Grant type 'authorization_code1' is not supported	Указан неверный тип кода авторизации (должен быть authorization_code)
		Grant type 'refresh_token1' is not supported	Указан неверный тип refresh_token (должен быть refresh_token)
	unauthorized_client	Unknown client_id = 'XXXX'	Указан неверный client_id или client_id и client_secret. (Запрос на получение новых

Код возврата	Расшифровка кода возврата	Описание кода возврата	Причина возникновения
			access_token и refresh_token по действующему refresh_token)
		Client 'XXXX' is blocked	Внешний сервис заблокирован в системе (запрос на новых получение access_token и refresh_token по refresh_token)
400	invalid_request	Invalid code verifier	Указан неверный код проверки
	bad_request	Неверный формат client_secret	Длина client_secret меньше 8 или больше 256 символов
			Значение new_client_secret совпадает с client_secret
	unsupported_token_type	Тип токена не поддерживается	Указан неверный тип токена
401	unauthorized	Неверный формат access_token	Указан некорректный или просроченный access_token.
			Access_token выдан клиентом, а не партнёром.
500	unknown_exception	Внутренняя ошибка сервера	
503	incorrect_server_	Сервис временно недоступен	Технологическое окно