

Типовые сценарии

В зависимости от целей и задач которые преследует внешняя система во взаимодействии с системой лояльности ProBonus можно выделить несколько типовых сценариев интеграции.

В данном разделе описаны базовые кейсы и указаны конкретные методы API необходимые для их реализации.

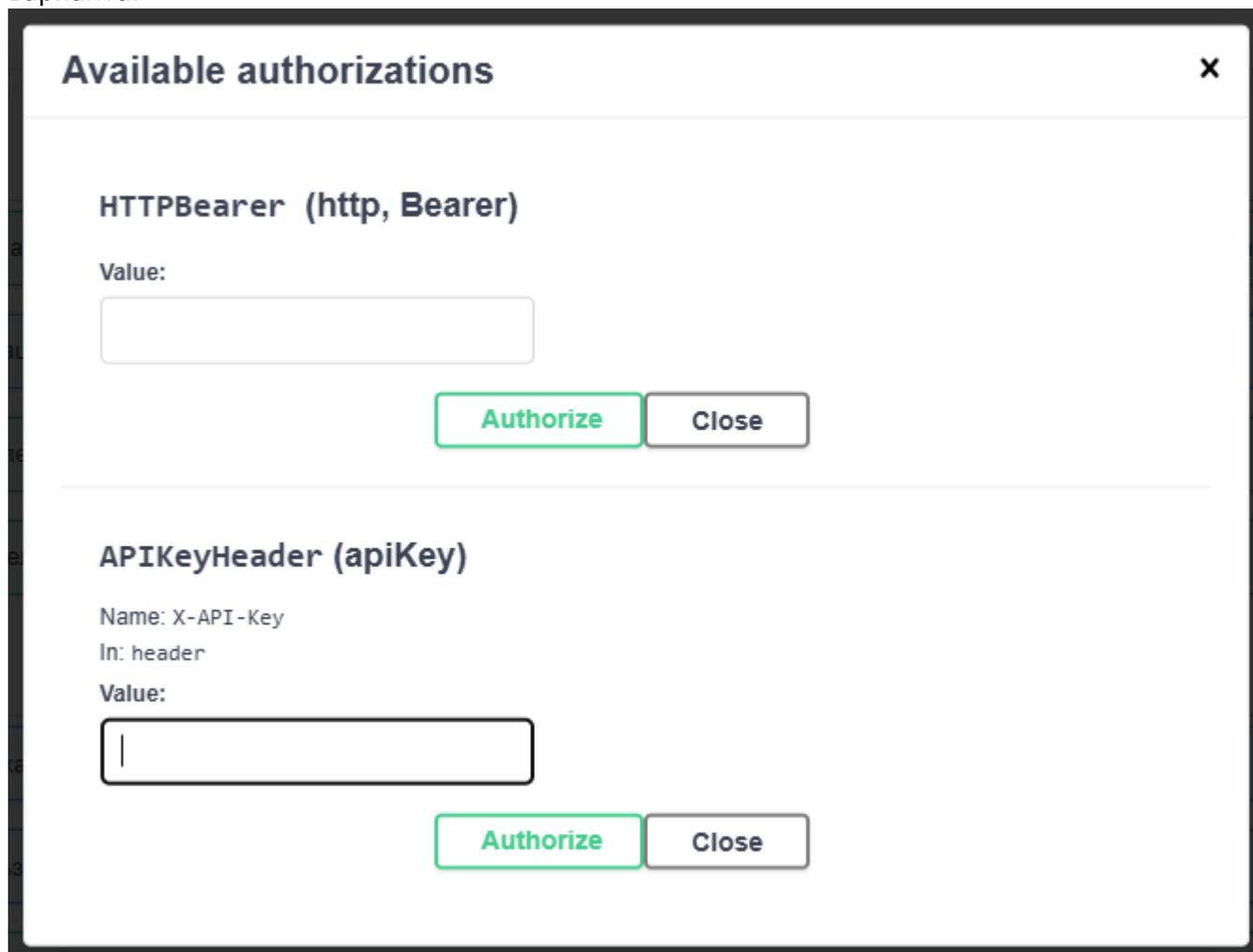
Если требуемая вам логика взаимодействия не описана в данном разделе, вы можете написать нам на soft@carbis.ru и мы обязательно поможем найти решение.

Авторизация для работы с API

API ProBonus предусматривает два различных способа авторизации:

- 1. Получение bearer токена через авторизацию (/api/v0/auth/login) и его дальнейшее обновление (/api/v0/auth/refresh_token).
- 2. Работа через токен для API. Все запросы в заголовках содержат постоянный токен.

Поддерживаются оба варианта. Но вариант 2 проще и удобнее. В сваггере доступны оба варианта.



Работа с анкетными данными на уровне личного кабинета внешней системы

1. В качестве сквозного идентификатора гостя используем номер телефона.
2. Если бизнес-логика подразумевает создание новых счетов из внешней системы, тогда каждого нового гостя вы создаете в ProBonus методом **/api/v0/accounts/insert_by_template**.

По минимуму нужно передать только анкетные данные и телефон (email и доп. идентификаторы - по желанию). Счет для гостя будет создан в группу accountGroupId и данному гостю будет автоматически назначена лояльность по шаблону настроенному для этой группы в рамках системы ProBonus (уровень, приветственные бонусы, установлено сгорание и т. п.)

Id нужной группы мы можете получить по API методом **/api/v0/account_groups/list**, или просто посмотреть в админке (<http://probonus.pro:6896/account-groups/>) и вынести в настройки на своей стороне.

ВАЖНО: Для корректного создания новых счетов по шаблону нужна группа с признаком «Шаблон для API» (haveTemplate=true). В тестовой БД это «Самостоятельно зарегистрировавшиеся» (id=3).

3. Проверить гостя на существование в системе лояльности можно при помощи метода **/api/v0/accounts/find_by_phone**. Ищет гостя по номеру телефона. Если гость с указанным номером телефона существует, то в ответ вернется практически вся информация о нем: анкета, текущий остаток на счете, текущий уровень, купоны и так далее.

Дополнительные методы API ProBonus

1. **Начисление бонусов гостю** по инициативе внешней системы (например за какую то активность в вашем приложении).
`/api/v0/accounts/refill`.
2. **Транзакции по счету** (показать в ЛК гостю его транзакции).
`/api/v0/accounts/transactions` – вернет все транзакции по счету за указанный период.
`/api/v0/accounts/transactions` – вернет N последних транзакций.
3. **Обновить данные счета** (изменить анкету, изменить скидку/бонус, срок жизни бонусов и т.п.)
`/api/v0/accounts/update`
4. **Получить данные конкретного чека** гостя (вплоть до блюд)
`/api/v0/checks/get_by_id`
5. **Если используются сгорающие бонусы.**
`/api/v0/accounts/get_balance_extended` – вернет детализацию баланса счета, с указанием когда и сколько сгорит.

Начисление лояльности за покупки во внешней системе

Учет лояльности (скидки/бонусы/потраты) за все покупки совершенные гостем через систему r_keeper происходит автоматически и все данные об этих транзакциях попадают в базу ProBonus. В некоторых случаях транзакция (которую нужно учесть в рамках ProBonus)

происходит за рамками контура `r_keeper` (например, заказ на доставку который по итогу не будет создан в `r_keeper`, т.к. внешний модуль доставки не имеет интеграции с кассовой системой и существует отдельно). При этом сумма покупки должна быть учтена в оборотах по счету гостя (учет затрат для расчета лестницы, начисление/списание бонусов за внешнюю покупку и т.п.). Для этой цели используется метод `/api/v0/external/register_check`.

Описание переменных `restCode` - Код ресторана (из справочника Настройки-Рестораны), к которому нужно отнести данную транзакцию для отчетов. **`checkGuid`** - Уникальный номер внешней транзакции (в формате GUID). Важно! Система ProBonus не контролирует уникальность переданного GUID в рамках своей БД. Контроль уникальности лежит на внешней системе. **`accountId`** - id счета по которому совершена транзакция. **`rkUnit`** - Код станции (рабочего места) в ресторане, к которому нужно отнести данную транзакцию для отчетов. Рекомендуем указывать код, заведомо определяющий что это не настоящая станция в ресторане. Например, 999 или 888 или любой другой трёхзначный. **`externalCheck`** - Номер транзакции во внешней системе (может использоваться для дальнейшей сверки транзакций между двумя отчетами). **`bonusSpentSum`** - Сумма (в копейках), которую нужно списать со счета в оплату транзакции. **`discountSum`** - Сумма скидки (в копейках), которую гость получил во внешней системе **`bonusAccruedSum`** - Сумма (в копейках), которую нужно начислить на счет за данную транзакцию. **`moneySpentSum`** - Сумма (в копейках), которую гость потратил (оплатил) в рамках данной транзакции. **`checkData`** - Любая текстовая информация о содержимом внешнего чека.

Метод `/api/v0/external/return_check` используется для отката всех начислений сделанных в ProBonus в случае отмены покупки во внешней системе.

From:

<https://wiki.carbis.ru/> - База знаний ГК Карбис

Permanent link:

https://wiki.carbis.ru/external/pro_bonus/10_probonus_api/01_scheme

Last update: **2026/02/16 10:29**