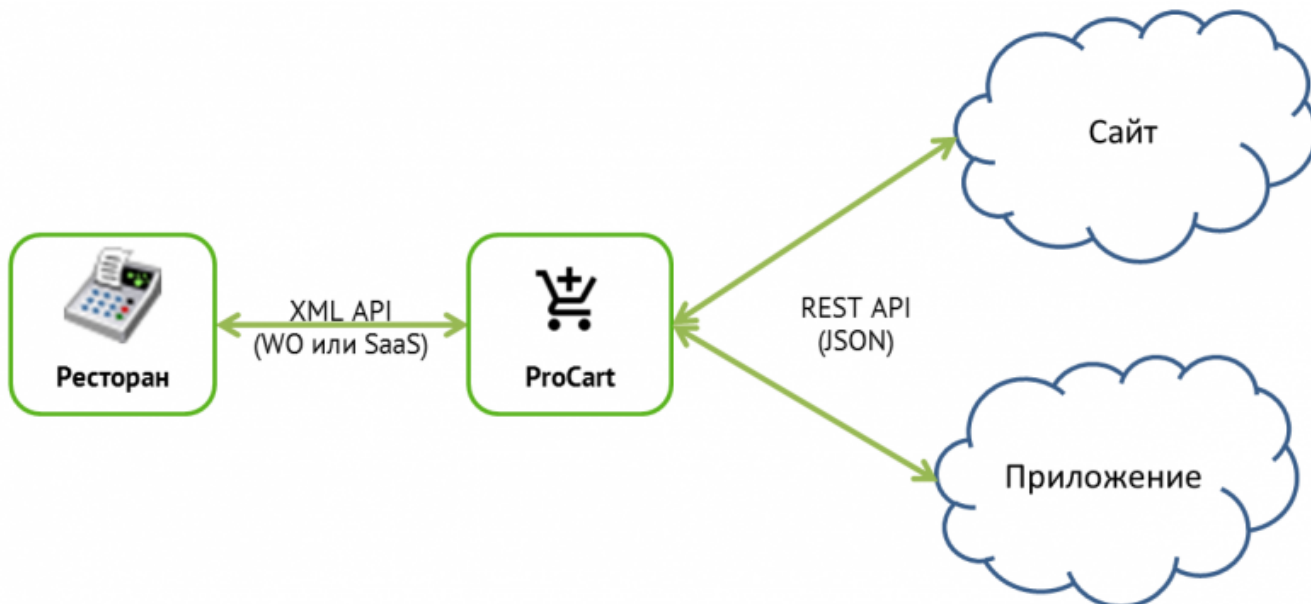


ProCart API

Общий принцип работы

Перед работой с API рекомендуется ознакомиться с данной статьей, для понимания основных принципов работы модуля ProCart и возможностей его API.



ProCart обеспечивает работу с системой **r_keeper** через Rest API (json). Для получения запросов из внешней системы в ресторане должен быть «белый» IP, а так же необходимо будет пробросить порт.

Один модуль **ProCart** может обрабатывать запросы полученные из разных внешних систем. Каждый источник инициализирует себя своим уникальным токеном. Перечень разрешенных токенов указывается в настройках модуля ProCart. Удалив токен из настроек можно ограничить работу модуля с определенным источником.

Внешняя система выполняет **http** запросы в порт модуля **ProCart** и получает ответы. Дополнительно возможна подписка внешней системы на push уведомления от модуля **ProCart** по факту изменения статуса заказа (удален, пречек, оплачен)

Работа со столами

По умолчанию все заказы из внешней системы создаются на выделенный стол (**код** нужного стола указывается в настройках ProCart). Если в работе находится несколько заказов одновременно, то они будут создаваться на дополнительных столах. **Пример:**

- В настройках указан код стола №100
- Первый заказ из внешней системы будет создан на столе «100»
- Следующий заказ (если первый еще не закрыт) будет создан в столе «100.1»
- Следующий заказ (если первые два еще не закрыты) будет создан в столе «100.2»

- И так далее.

Если из внешней системы приходят заказы разных типов (например «Доставка» и «Самовывоз») то дополнительно (хотя и не обязательно) можно сделать разные столы, для разных типов заказа. **Пример:**

- Для заказов на Доставку выделен стол с именем «Дост» (код 101)
- Для заказов на Самовывоз выделен стол с именем «Вывоз» (код 102)
- В настройках ProCart в качестве основного стола указан код 101 (Доставка)
- Внешняя система передает заказ (запрос /postorder)
 - Если это заказ на Доставку, то в запросе можно не передавать код стола, и тогда заказ будет создан на столе по умолчанию (код 101, «Дост»). Если таких заказов будет много, то далее они автоматически будут создаваться на столах «Дост.1», «Дост.2», «Дост.3» и так далее.
 - Если это заказ на Самовывоз, то в запросе можно передать код 102, и тогда заказ будет создан на столе «Вывоз». Если таких заказов будет много, то далее они будут автоматически создаваться на столах «Вывоз.1», «Вывоз.2», «Вывоз.3» и так далее.

При большом количестве внешних заказов имеет смысл создать отдельный «план зала» и в рамках него завести стол (столы) для приема внешних заказов. При такой настройке на кассе можно будет легко переключаться между заказами в зале и заказами из внешних источников, просто переключая залы.

Классификации блюд

В системе `r_keeper` дерево меню может иметь сложную древовидную структуру с большим уровнем вложенности веток. Такая форма удобна для работы на кассе, но не нужна при работе с внешними источниками. Более того, далеко не все блюда, которые заведены в меню, должны быть доступны для продажи во внешнем источнике.

Вопрос формирования структуры меню и доступности блюд для внешнего источника решается при помощи «Классификаций». В системе `r_keeper` создается отдельная классификация «Использовать для внешних продаж». Эта классификация наполняется перечнем категорий (разделов меню), например так: Салаты, Закуски, Супы, Напитки. Далее, для всех блюд которые должны продаваться во внешних источниках нужно проставить значение этого классификатора (Для Оливье - Салаты, для Борща - Супы и т.д.). А для тех блюд которые не должны быть доступны для внешнего заказа - оставить этот классификатор пустым. Запрос `/getmenu` вернет перечень категорий (Салаты, Закуски, Супы, Напитки) и перечень всех доступных блюд с привязкой к соответствующей категории.

Категория заказа

В системе `r_keeper` возможность заказа конкретных блюд, а так же цена на них может зависеть от различных факторов, например, «категории заказа» (КЗ). В настройках **ProCart** указывается код основной категории заказа. Запрос `/getmenu` возвращает блюда и цены для данной категории заказа. В запросах `/validate` и `/postorder` можно дополнительно передать код другой КЗ (обязательно должна существовать в справочниках кассовой системы) и тогда будет сделан расчет заказа для данной категории.

Пример №1: В ресторане есть 3 типа цены:

- Основная - используется для заказов в зале
- Доставка (дороже) - используется для продажи блюд через службу доставки
- Самовывоз (дешевле) - используется для заказа блюд на самовывоз из ресторана

В системе `r_keeper` заведены соответствующие типы цен и три категории заказа: Основная (предустановка), Доставка, Самовывоз. Через «использование типов цен» типы цен связаны с соответствующими категориями заказа.

В настройках ProCart указывается тип заказа по умолчанию (например, «Доставка»). В запросе `/getmenu` будут возвращаться цены на блюда по типу цены «Доставки». Гость наполнил корзину блюда, перешел в оформление заказа и указал «Самовывоз». В запросе `/validate` (а далее в `/postorder`) необходимо указать верный код категории заказа (код Самовывоза) и тогда в ответе вернутся цены, по которым можно продать эти блюда на самовывоз.

Пример №2: В ресторане одна цена, но на Самовывоз дается дополнительная скидка 20%. В `r_keeper` заведено 2 категории заказа (Основная и Самовывоз). Код основной категории указывается в настройках ProCart. Дополнительно заводим и настраиваем скидку «для самовывоза» и через «использование скидок и наценок» включаем ее автоматическое применение для всех заказов с категорией «Самовывоз». При валидации корзины и создании заказа на самовывоз, в запросах `/validate` и `/postorder` дополнительно передаем код соответствующей категории заказа (самовывоз). В итоге будет создан стол с блюдами и в него автоматически будет добавлена соответствующая скидка.

Комментарии к заказу

В системе `r_keeper` у заказов есть свойство, которое можно использовать для хранения дополнительной текстовой информации (комментария к заказу), но к сожалению, длина этого поля крайне мала (30-100 символов) и зачастую вся необходимая доп. информация по заказу туда не помещается. В рамках ProCart проблема длинных комментариев решена следующим образом: В запросе `/postorder` внешняя система может передать текстовый комментарий (поле «comment»). Он будет сохранен в локальной базе ProCart и связан с реальным заказом в системе `r_keeper`. В печатных формах системы `r_keeper` (сервис-чек, пречек, чек) можно добавить небольшой скрипт, который при помощи служебного метода `/internal/print_info` будет получать комментарий по конкретному заказу из БД ProCart и выводить его на печать.

Так же в запросе `/postorder` есть поле «short_external_info», куда можно передать краткую информацию о заказе (идентификатор заказа во внешней системе, тел или ФИО гостя и т.п.) Этот комментарий (30 символов) будет виден персоналу на экране кассы `r_keeper` (в списке заказов) и может помочь быстрее сориентироваться и найти нужный заказ.

Комментарии к блюду в заказе (открытый модификатор)

С версии 1.5.0.30 В рамках ProCart к блюду может быть прикреплен комментарий. Для этого в запросе `/postorder`, для элемента меню, можно передать комментарий в поле «item_comment». Данный комментарий будет виден персоналу на экране кассы, как модификатор для блюда. Данный комментарий, имеет ограничение на длину от 1 до 255

символов. **Важно!** Требуется дополнительная настройка в `r_keeper`

Заказы с оплатой

Через систему **ProCart** можно создать в `r_keeper` заказ в котором есть только блюда (нет оплат) и далее этот заказ будет рассчитан сотрудником на кассе как обычно. Также, при создании заказа (метод `/postorder`), можно передать информацию об оплатах этого заказа. Оплат может быть передано несколько (разными валютами, например, частично деньгами, частично бонусами). Фискальность каждого типа оплаты определяется настройками соответствующих валют в системе `r_keeper`. Все переданные оплаты добавляются в заказ как «предоплата». Если заказ полностью оплачен во внешней системе, то на кассе `r_keeper` будет создан стол с блюдами, внесена предоплата переданными валютами. Заказ будет открыт на кассе пока его не завершит сотрудник ресторана. Метод учета предоплат (фискально или нет) настраивается в рамках причины внесения предоплат в `r_keeper`.

Скидки (суммовая и словарная)

Модуль **ProCart** поддерживает работы со скидками. Это может быть словарная скидка. «Словарная» - значит заранее настроенная в редакторе меню `r_keeper`, с заранее известным процентом и другими тонкими настройками. В запросах `/validate` и `/postorder` нужно передать код скидки (можно передать несколько словарных скидок в один заказ), существующей в кассовой системе. Применимость данной скидки для текущего заказа определяется на стороне кассовой системы (с учетом стола, категории заказа, времени суток, правил комбинации с другими скидками и т. п.). В ответе на запросы будет возвращен заказ пересчитанный со всеми возможными скидками.

Также поддержана работа с произвольными (суммовыми) скидками. В данном сценарии сумма скидки и правила ее применения находятся на стороне внешней системы, в запросах `/validate` и `/postorder` передается итоговая сумма скидки, рассчитанная внешней системой. Такой сценарий можно применять для случая когда гость тратит в оплату заказа какие-то купоны или бонусы. Тогда сумма всех этих скидок передается одной строкой и применяется на кассе `r_keeper` в виде суммовой скидки.

Также поддержана работа с автоматическими скидками кассы `r_keeper`. Если в ресторане используются какие-то автоматические скидки по определенным правилам, и заказ сделанный внешней системой удовлетворяет этим условиям, то запросы `/validate` и `/postorder` вернут содержимое заказа с учетом автоматических скидок.

Наценка суммовая

Модуль **ProCart** поддерживает работу с наценками, от версии ProCart 1.9.1.64. Для суммовой наценки поддержана работа с произвольной (суммовыми) наценкой. В данном сценарии сумма наценки и правила ее применения находятся на стороне внешней системы, в запросах `/validate` и `/postorder` передается итоговая сумма наценки, рассчитанная внешней системой.

Пример того, как в запросе в поле «discount» можно передать одновременно Скидку словарную

(Ref), Скидку суммовую(Sum) и Наценку суммовую(MARKUPSUM):

```
"discounts": [  
  {  
    'type': 'REF',  
    'code': 10,  
  },  
  {  
    "type": "SUM",  
    "sum": 120,  
    "code": 1137  
  },  
  {  
    "type": "MARKUPSUM",  
    "sum": 40,  
    "code": 1143  
  }  
]
```

Поддержка стоп-листов и ограниченных блюд

Запрос **/getmenu** возвращает список блюд с учетом стоп-листов и остатка (если таковые введены на кассе **r_keeper**) Таким образом, если на кассе какое то блюдо поставлено в стоп-лист, то внешняя система получит эту информацию. Если на кассе установили остаток по какой-то позиции, то в ответе на запрос **/getmenu** кроме основных свойств данного блюда вернется максимальное количество порций, которые можно продать. Запросы **/validate** и **/postorder** также учитывают стоп-листы и ограниченные остатки. Т. е. нельзя провалидировать и создать заказ содержащий в себе блюда из стоп-листа и нельзя сделать заказ, в котором указано количество порций больше, чем остаток на кассе.

Оповещения о заказах

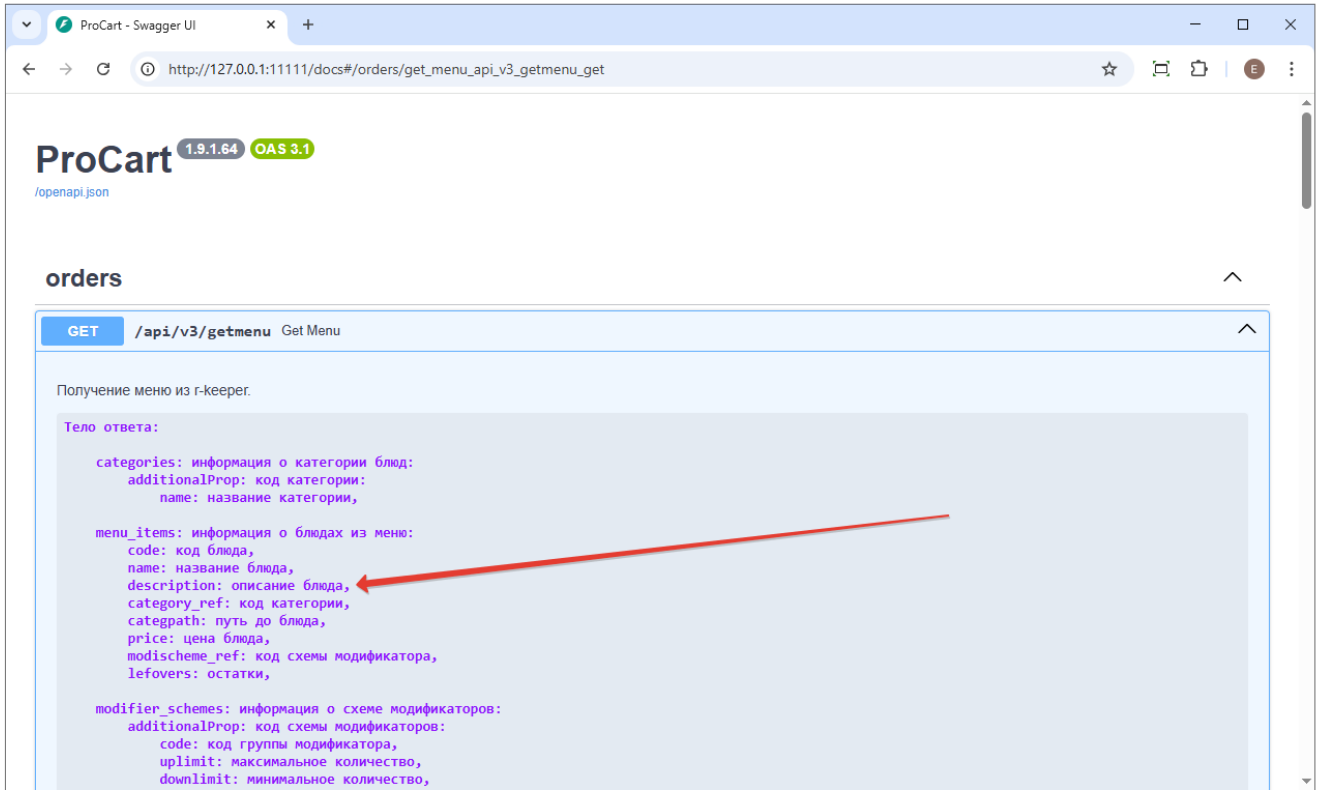
При работе с модулем ProCart персонал заведения может получать уведомления о новых заказах через механики внешних систем (push уведомления в приложении, письма на почту, СМС сообщения и т. п.) Когда заказ создан в кассовой системе, автоматически происходит сервис печать (по настройкам **r_keeper**). Этот механизм можно дополнительно задействовать для уведомления персонала, если в используемую схему печати добавить еще одно правило, согласно которому, все заказы полученные из внешней системы будут печататься по отдельному макету сервис-печати на выделенный принтер.

Описание блюда

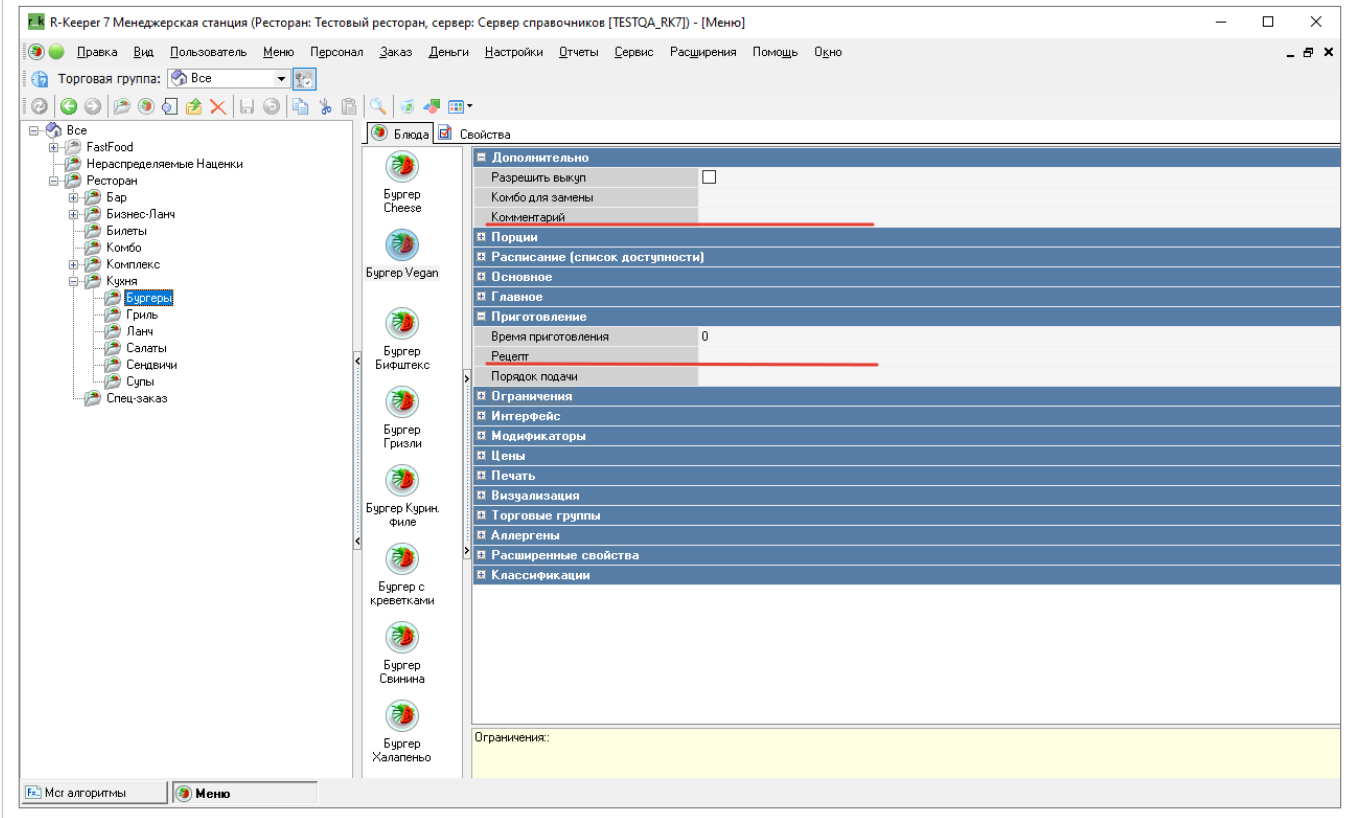
Описание блюда передаётся при запросе меню. Логика этого поля имеет постоянное свойство и дополнительно не настраивается. В приоритете в ProCart поле «description»(описание блюда) заполняется значение из поля «Комментарий» из карточки блюда в **r_keeper**, если это поле

пустое, то заполняется из поля «Рецепт». Если на стороне r_keeper оба поля не заполнены, то и в ProCart в поле «description» будет пустым.

Пример поля на стороне Procart:



Пример передаваемых полей на стороне r_keeper (приоритет у поля «Комментарий»)



From:

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

Permanent link:

https://wiki.carbis.ru/external/procart/02_procart_api

Last update: **2025/03/28 08:55**