



## Описание сервиса extracard.net.

Сервис [extracard.net](https://extracard.net) - сервис по продаже карт MasterCard Platinum, которые работают для всех стран в Facebook для работы арбитражников.

Преимущества extracard.net:

- покупку карт без документов,
- пополнение через сторонние платежные системы (5),
- уникальный БИН,
- получение 3ds кодов и смс в личном кабинете,
- платиновые кредитные пластиковые карты;

Сайт полностью выполнен на английском языке, но также имеет русскоязычную версию. На **главной странице** сосредоточена основная информация о ресурсе: как начать пользоваться картой, отзывы, контактная информация, документация сайта, там же можно найти список преимуществ.

В "шапке" находится главное меню, при помощи которого пользователь перемещается по сайту, линк на форму регистрации, смена языка. В "подвале" - копирайт (только на английском языке) и общая информация.

Вход на сайт может осуществляться через привязанный e-mail и пароль или же через аккаунты google и reddit.

После регистрации открывается **личный кабинет** со скрывающимся боковым меню, с помощью которого можно перейти к "Моим картам", "Подпискам", "Списаниям", "Платежам", "3ds кодам", "Настройкам", "Новостям", "FAQ" или обратиться в службу поддержки через телеграмм или скайп.

Пополнить баланс можно из любой части личного кабинета, так как в линк на оплату находится в хедере ЛК.

Карты можно пополнять через Bitcoin 1%, Webmoney 1%, Payoneer 3%, Wire 5%, Безналичный расчет 7%.

% - за пополнение с любой системы - до 9% (зависит от объема), расчёт комиссии также есть в форме оплаты.

## Мои карты.

Одна карта стоит 6900 рублей. Здесь её и можно приобрести, перед этим пополнив баланс. Также в этом разделе представлен список всех ваших карт и есть возможность производить поиск по номеру, комментарию или дате покупки карты, а также фильтровать их по дате/времени, статусу, номеру карты, балансу и комментариям.

## Подписка.

Аналогично разделу “мои карты” представлен и этот раздел с таким же списком подписок и способом поиска, и фильтра подписок. И, соответственно, здесь можно приобрести подписку на карты от одной до неограниченного количества. Указав количество необходимых карт, вам будет представлена сумма для оплаты, дата окончания подписки, а если средств будет недостаточно на вашем счете, это также будет видно.

### **Списания, Платежи, 3ds коды.**

В этих разделе представлены все списания/платежи/3ds коды, которые производились с ваших карт/начисления/отправленные коды, поиск по ним и фильтры, как в предыдущих разделах.

### **Настройки.**

В разделе настроек есть возможность редактировать уведомления, которые приходят вам на почту, выбирать язык сайта, сменить пароль, e-mail, а также установить двухфакторную авторизацию через Телеграмм.

### **Новости.**

На данный момент в разделе отсутствует какая-либо информация.

### **FAQ.**

Здесь представлены ответы на самые популярные вопросы, которые могут возникнуть при работе с картами и сайтом, в целом. Также здесь контакты для связи с поддержкой.

### **Бизнес логика extracard.net.**

**ping/timeline** - получение от точки информации о платежах, обновление в базе. Вызывается в кроне.

1. Запрос в банк:
  - банк не найден;
  - авторизация прошла.
2. Создание банка/синхронизация с существующим (Синхронизировать ли данные с банком по этому кастомеру)
  - Синхронизация прошла/ банк найден - проверяется информация о списаниях, выдается массив время, список списаний. Берет время редактирует формат, если он некорректен.
  - Создается запись со значениями/ берет имеющиеся данные: Type ownerId, sum, sumCurrency, originalSum, originalCurrency, purpose, eventId, cardId, cardOperationType, verboseTerminalName, cardTrnInfo, trnId, cardTrnInfo, "stateInfo.
  - Получить владельца карты на определенную дату (смотрим кто когда назначен на карту, если ничего нет в логе назначений по этой карте, то возвращаем текущего пользователя карты)
3. Проверим, были ли изменения в объекте.

Если новый банк:

Операция не прошла:

Возвращает информацию о схеме таблицы БД, связанной с этим классом AR.  
Возвращаемая схема таблицы информации о схеме таблицы БД, связанной с этим классом AR. Вызывает исключение `InvalidConfigurationException`, если таблица для класса AR не существует.

Логируем изменения по стейтменту. Добавить лог изменения стейтментов(данные `"$statement_id, $key, $old, $new, $objectId"/ "Cancel транзакция id= %s на сумму %s %s %s (%s->%s)"`).

Выдается сообщение, если транзакция отменена

Важен владелец карты на момент транзакции

Сообщение Клиент %s, Cancel транзакция id= %s с карты %s на сумму %s %s %s (%s->%s)"

Операция прошла:

Обновляется дата и время, сохраняется информация, проверяется БД - если все окей - проверим, достаточно ли сейчас денег на карте для этой транзакции.

Если это новая запись- Регистрирует информационное сообщение для пользователя.

Сообщение на какую сумму транзакция/ Если карта заблокирована сообщение "сейчас карта заблокирована"

4. Подозрительные транзакции - в специальный тг чат. Если это новая не режестед транзакция или если не новая отмененная транзакция.

Сообщение типа "Подозрительная транзакция: user id=%s %s card number %s type %s sum %s руб. title %s".

**cards/get-limits** - Обновление лимитов карт по всем банкам. Вызывается в кроне. Можно дополнительно передать `$getLimitsAccountId` - выбрать по id аккаунта, где `cards.user_id` is null.

1. Авторизация с банком
2. Создание нового таска
3. Считаем время выполнения по аккаунту
4. Считаем время выполнения по батчу
5. Лимит, который пришел от Банка, сверяем с нашим, при несоответствии логируем и шлем сообщение
6. Каунтер, который пришел от Банка, сверяем с нашим, при ошибке логируем и шлем сообщение
7. Если хотя бы одна карта банк-аккаунта сохранилась, то обновляем флаг
8. После сохранения карты.  
Если карта с отрицательным балансом - пробуем пополнить ее с баланса пользователя  
Не делаем этого в промежуток смены месяца - когда мы обновляем лимиты.

Если Точка не успеет выставить counter в 0, то имеем заведомо отрицательную карту

9. Сохраняем в базу последнее время сохранения лимита по картам банк аккаунта
10. Закрываем соединение
11. Отправляем сообщения по картам об отрицательном балансе, если такие карты есть  
Если карта заблокирована - данные не актуальны(limit = 0, counter != 0 => balance < 0)

**cards/priority-limits** - Запросить лимиты по картам в приоритетном порядке.

1. Проверяем авторизацию банка
2. При успешной авторизации и имеющихся лимитах получаем лимиты по картам формата "Priority Check Limits. Count cards in batch: %s. Count received limits: %s"
3. При отсутствии лимита сообщение 'No card limit received from Bank'
4. Лимит, который пришел от Банка, сверяем с нашим, при несоответствии логируем и шлем сообщение
5. Каунтер, который пришел от Банка, сверяем с нашим, при ошибке логируем и шлем сообщение
6. закрываем соединение

**tasks/exec** - Таски на создание, блокировку, разблокировку, штрафы, сброс лимитов, изменение лимитов по картам. Вызывается в кроне.

Самый высокий приоритет - сброс каунтеров в точке, затем штрафы, чтобы не смог вывести деньги с карты - во всех тасках пересчет реального лимита по карте!!!

При блокировке карты (BLOCK) создается также таск на вывод средств (LIMIT)

Необходимо чтобы таски на вывод средств выполнялись первыми

Таск move - тоже после limit

Таск Replace - после Limit и после Block

1. Проверка синхронизации с банком. Если банк разлогинен - пропускаем
2. Если по этой карте есть выполненные таски (status=1) НО с невыполненной работой (made=0) - ждем. Полезная работа по таску - в cards/priority-limits

Если не смогли обновить каунтеры по карте - ждем, идем к следующему таску. Для типов TYPE\_RESET\_COUNTER и TYPE\_UNBLOCK не нужно обновлять.

**TYPE\_NEW** - Таск на выпуск новой карты.

Из-за того, что лимит вычисляется в коде, и могут быть поставлены несколько тасков, изменяющих лимит, то этот лимит может быть неверный.

Перед запросом в точку пересчитывается лимит, и записывать в real\_limit.

- Установка лимита.

Если все okay - **STATUS\_DONE** и сохранение.

Если нет - Сообщения в "денежный чат" (Может отправлять информацию по событию не при первом его возникновении, а 2й, 3й, ... раз, т.е. пропустить какое-то количество отправок) - "NEW TASK: Ошибка выставления лимита ".\$task->real\_limit." по карте ".\$card->cmsGuid." ( ".\$card->account->customer->name." )"

- Возвращаем деньги за покупку карты. Сообщение "Возвращаем деньги %s на баланс пользователя %s за покупку карты № %s (%s)"

Если карта куплена по подписке - сообщение (\$card->is\_subscribe\_free? 'Бесплатная подписочная карта. ': 'Подписочная карта. ').\$msg

- Забираем карту. Если карта была подписочная - \$card->is\_subscribe = false;
- Лог назначения карт(card\_id = \$card->id; user\_id = 0)
- Возвращаем деньги на баланс, за пополнение карты, сообщение "Возвращаем деньги %s на баланс пользователя %s за неудавшуюся попытку положить на карту № %s (%s)"
- Преобразует псевдоним пути в реальный путь ('@runtime').'/t.txt', 'a+')
- При условии, что таск, лог и карта успешно сохраняются - транзакция завершена, в ином случае - отменена.

#### TYPE\_BLOCK - Таск на блокировку карты

- Выполняем родительские таски

Все родительские таски должны быть выполнены успешно к этому времени. Если есть неуспешные родительские таски - фейлим и текущий. Если есть еще не выполненные таски - ждем, идем к следующему таску. Все родительские таски должны быть применены к этому времени, made = 1. Если не все применились - ждем, идем к следующему таску.

Если карта заблокирована - сообщение "task id = %s: bank request was failed"

- После блокировки статус сменяется на STATUS\_INACTIVE
- Сохранения в БД
- При откате транзакции - STATUS\_FAIL и сообщение "Карта " . \$card->number . " не заблокирована. ".\$e->getMessage()

#### TYPE\_UNBLOCK - Таск на разблокировку карты

- При успешной разблокировке (с применением информации по карте и task id) статус карты меняется на STATUS\_ACTIVE.
- При неудачной разблокировке - сообщение "TaskController UnblockCard task\_id=%s. Проблема с сохранением карты: %s"
- Сохранение таска. При неудаче - сообщение "TaskController UnblockCard task\_id=%s. Проблема с сохранением таска: %s"
- После разблокировки - если есть траты по карте - новый таск на лимит на сумму каунтера. При неудаче - сообщение "TaskController UnblockCard task\_id=%s. Проблема с сохранением корректировочного таска: %s"
- При условии, что таск, карта и баланс успешно сохраняются - транзакция завершена, в ином случае - отменена.

### TYPE\_PENALTY - Таск на списание штрафа - в высоком приоритете

- Устанавливаем флаг приоритетной проверки
- Если баланс карты недостаточен - пересчитываем сумму таска
- Денег хватает - ок, нулевой баланс - сообщение "PENALTY TASK: нет денег на балансе карты, task\_id=%s", \$task->id
- ☐ Новый лимит для запроса в Банк Тут лимит карты - тот что у нас в базе
- ☐ Пробуем установить лимиты в точке
- При успешной попытке выходим из цикла  
При неудачной попытке слип, сообщение "PENALTY TASK: bank response is fail, task\_id=%s", \$task->id
- Устанавливаем флаг приоритетной проверки
- ☐ При неуспешном таске устанавливаем real\_sum в 0, чтобы потом знать сколько надо дописать (sum - real\_sum)  
Сообщение в чатик, что денег на штраф не хватило("Штрафной таск выполнен со статусом FAIL, task id: %s", \$task->id)

**TYPE\_RESET\_COUNTER** - Таск изменение лимита при сбросе каунтера в Банке, когда Банк в начале месяца иницирует сброс каунтера в 0.

Этот тип тасков ставится в cards/balance - после того, как Банк в смену месяца сбрасывает каунтер в 0

Чтобы баланс карты был корректный после этой процедуры

Новый лимит для запроса в Банк - Берем то что посчитано в коде

- Пробуем установить лимиты в Точке
- При успешной попытке выходим из цикла  
При неудачной попытке сообщение ""RESET\_COUNTER TASK: Ошибка выставления лимита ".\$task->real\_limit." по карте " и слип
- ☐ Устанавливаем флаг приоритетной проверки

**ping/get-info** - Обновление авторизационных данных аккаунтов. Вызывается в кроне.

- Подключение к банку
- Получаем данные аутентификации (Формирует массив для запроса)
- Получение кастомеров с его аккаунтами
- Синхронизация с банком
- Получение трат и лимитов по счету (работает только для рублевых счетов)

При сбое подключения выводит сообщение 'Error getting bank account limits '.

`$bankAccount->id . ' - ' . $e->getMessage()`

- Если сохранение прошло успешно - сохраняем время обновления баланса аккаунта ('Y-m-d H:i:s')

Обновление настройки по ключу

Название составного ключа в базе

`(BANK_ACCOUNT_LAST_TIME_UPDATE_BALANCE, $bankAccount->id)`

