



КРИПТОВАЛЮТА  
БУДУЩЕГО  
V 2022



Описание монеты eGOLD .....	3
Сравнение платёжных систем .....	4
Пропускная способность .....	6
Уникальные преимущества eGOLD .....	7
Дорожная карта .....	12
Почему именно eGOLD? .....	14
Вопросы и ответы .....	23
Работа с API ноды .....	36
1. Запрос к ноде без параметров .....	36
2. Запрос [type=wallet] - данные по кошельку .....	36
3. Запрос [type=nodas] - список доступных нод .....	38
4. Запрос [type=history] - история транзакций .....	38
5. Запрос [type=referrals] - история транзакций по рефералам .....	40
6. Запрос [type=referralwallets] - вывод списка кошельков рефералов у данного кошелька .....	41
7. Запрос [type=referralresults] - вывод общего количества и объёма монет по уровню рефералов .....	42
8. Запрос [type=height] - получение высоты кошелька и возможности отправки транзакции Служит для вспомогательного запроса перед транзакцией .....	43
9. Запрос [type=send] - осуществление транзакции при которой может быть смена публичного и закрытого ключа, смена пароля для управления контактами и E-mail уведомлениями, зарегистрирован новый кошелёк .....	43
10. Запрос [type=synch] - синхронизация ноды .....	44
11. Запрос [type=synchwallets] - обращение к синхронизируемой ноде .....	44
12. Запрос [type=contacts] - отображение контактов кошелька на ноде с которой он совершает транзакции .....	45
13. Запрос [type=email] - E-mail уведомления по транзакциям кошелька на ноде с которой он совершает транзакции .....	46
14. Запрос [type=balanceall] - общий баланс всех кошельков без учёта начисленных процентов .....	46
15. Запрос [type=walletscount] - общее количество кошельков .....	47
16. Запрос [version] - отображение текущей версии ноды файла egold.php .....	47
17. Пример совершения транзакции на PHP .....	47
18. Пример создания кошелька на PHP .....	48
19. Пример смены закрытого ключа на PHP .....	50
20. Пример приёма средств на PHP .....	52

# Описание монеты eGOLD




eGOLD electronic gold - электронное золото - это электронная криптовалюта, получившая своё название по своему прямому назначению также, как email electronic mail - электронное письмо. Email письма прочно вошли в нашу жизнь и вытеснили обычные письма из-за своей простоты, удобства использования и моментальной скорости доставки без каких-либо дополнительных затрат. Email письма не требуют материальных затрат, не требуют использования природных ресурсов. Также и eGOLD позиционируется, как экологичная альтернатива бумажным деньгам со значительными преимуществами над ними, схожими с преимуществами email писем над бумажными.

В монете используется сильнейший мотивационный механизм – реферальная ссылка и майнинг на кошельке в размере 10% в месяц и 13% в месяц на кошельке владельца ноды. Снижение майнинга монет из месяца в месяц и снижение майнинга на кошельках в трёхуровневой структуре мотивирует инвестировать в монету на начальном этапе.







К 2036 году майнинг снизится до 2% и 2,53% в месяц соответственно.



# Сравнение платёжных систем

Платёжная система	Комиссия за перевод	Скорость транзакции	Максимальное количество транзакций в секунду	Объём базы данных	Отсутствие заморозки	Устойчивость к взлому квантовым компьютером	Ежемесячный доход на баланс	Вознаграждение за популяризацию или реферальная программа	Стабильность курса средств платёжной системы	Изменение цены за последние 6 месяцев к доллару
 <b>eGold</b>	2 монеты = 0.035\$	менее 4 секунд	не ограничено благодаря параллельным транзакциям	объём настраиваемый и неизменный благодаря горизонту истории транзакций	+	+	Снижающийся майнинг от 205%-304% за 2022 год до 35,54%-27,54% в 2036 году	от 52%, 26%, 13% в 2022 году с 1,2,3 уровня соответственно до 6,88%, 3,44%, 1,72% в 2036 году	-	-
 <b>Bitcoin</b>	от 0.1\$ до 10\$	10 минут	от 2 до 7	увеличивается по 130 мегабайт в день. Сейчас более 330 гигабайт	+	-	0	0	-	+ 4.52%
 <b>Ethereum</b>	от 0.1\$ до 3\$	13 секунд	от 15 до 25	увеличивается по 240 мегабайт в день и уже более 340 гигабайт	+	-	0	0	-	+ 13.96%
 <b>USDT</b>	от 0\$ до 2\$	4 секунды	от 2 до 25	увеличивается как Ethereum и Bitcoin в зависимости от алгоритма	+	-	0	0	+	0
 <b>PRIZM</b>	от 0.0008\$ до 0.17\$	1 минута	4	увеличивается по 30 мегабайт в день и уже более 27 гигабайт	+	-	от 1.16% до 14.85% сейчас и от 0.2% до 1.31% при целевом PARATAX от 97% до 98% в зависимости от личного баланса и баланса рефералов до 88 уровня и начисляются до полного объёма выпуска монеты	увеличение дохода с личного баланса в зависимости от баланса созданных кошельков лично и рефералами до 88 уровня	-	- 15.70%
 <b>DASH</b>	0.003\$	около 2 минут 40 секунд	40	увеличивается по 12 мегабайт в день и уже более 23 гигабайт	+	-	около 1% - при этом на балансе должно быть 1 000 монет - это 72 000\$ по текущему курсу	0	-	+ 13.16%

# Сравнение платёжных систем

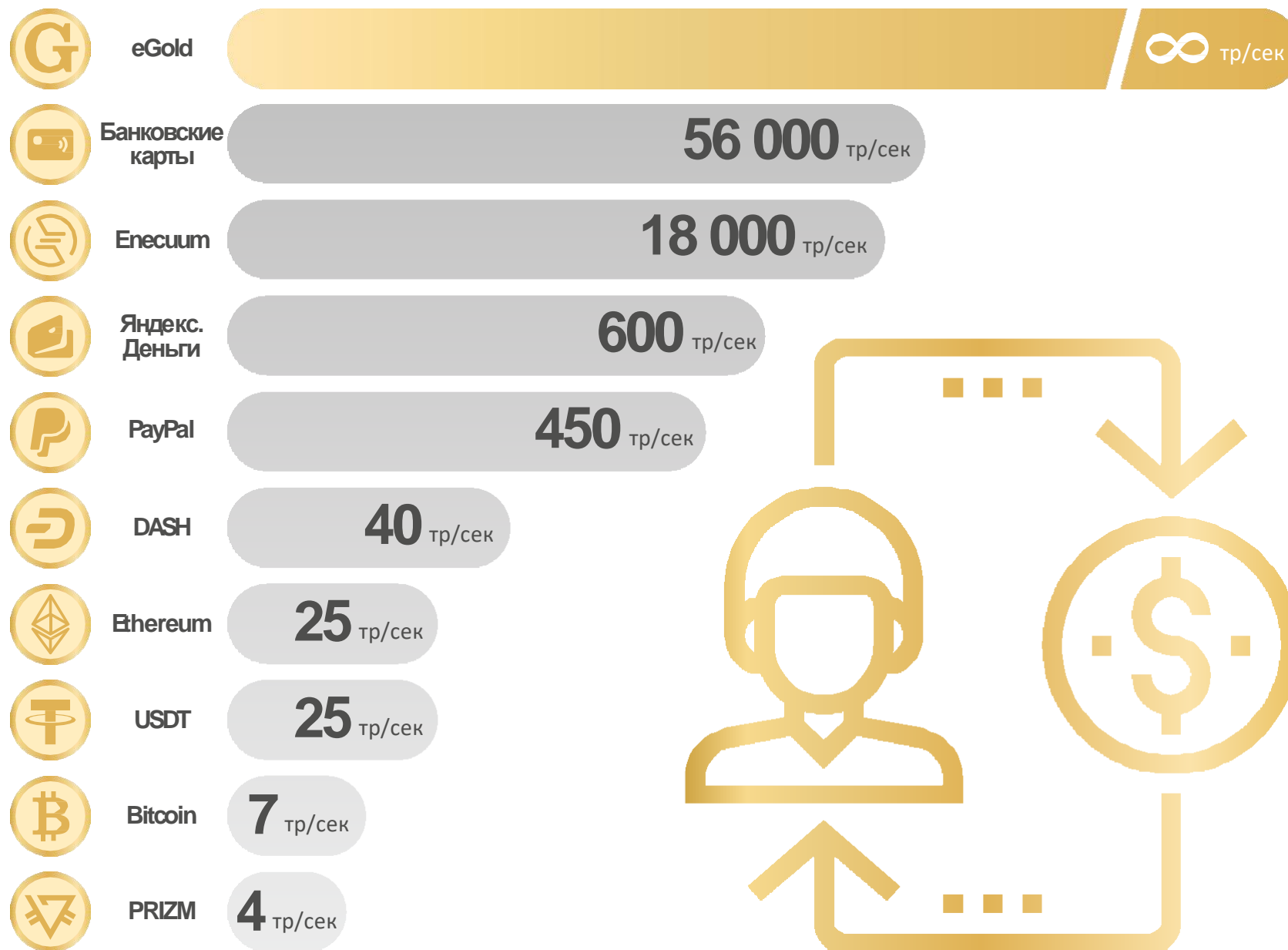
	Платёжная система	Комиссия за перевод	Скорость транзакции	Максимальное количество транзакций в секунду	Объём базы данных	Отсутствие заморозки	Устойчивость к взлому квантовым компьютером	Ежемесячный доход на баланс	Вознаграждение за популяризацию или реферальная программа	Стабильность курса платёжной системы	Изменение цены за последние 6 месяцев
	Bitcoin	?	15 секунд	18 000	увеличивается	+	-	около 2.3%	+10% от майнинга для реферера и реферала	-	- 8.52%
	Банковские карты	от 0% до 5%	от секунды до 7 рабочих дней	56 000	увеличивается	-	+ / ?	от 0% до 0.5%	0	+	0
	Банки	от 1% до 10%	до 5 рабочих дней	?	увеличивается	-	+ / ?	от 0% до 0.55%	0	+	0
	PayPal	от 0% до 4%	несколько секунд	450	увеличивается	-	+ / ?	0	0	+	0
	WebMoney	0.8%	несколько секунд	?	увеличивается	-	+ / ?	0	0	+	0
	Яндекс Деньги	от 0% до 3%	несколько секунд	600	увеличивается	-	+ / ?	0	0	+	0

\* Все данные средние и их сбор осуществлялся 25 июня 2021 года для России. Данные предоставлены только для ознакомления и могут отличаться от реальных цифр.

\* Знак ? в ячейках означает отсутствие данных. Обозначение +/? допускает возможность защиты от квантовых компьютеров с помощью косвенных мер.



## Пропускная способность



## Уникальные преимущества eGOLD



eGOLD работает на энергонезависимом алгоритме Proof of Stake, который осуществляет проверку транзакций путём подтверждения при более чем 50% баланса доступных нод. Что позволяет держать ноду на устройстве практически любой мощности.



Высочайшая защищённость достигается с помощью квантовоустойчивого шифрования FALCON и получения нодами уже подписанных закрытым ключом транзакций на стороне клиента.





В криптовалюте полная анонимность за счёт использования только удобного 18-и значного цифрового номера кошелька.  
Пример: G-7355-87879-8875-80915



Присутствует уникальная возможность для криптовалюты - это смена закрытого ключа. Сделать это позволяет неизменный номер кошелька, при котором новый закрытый ключ подписывается старым закрытым ключом, как и любая транзакция в соответствии с правилами работы большинства криптовалют. При этом, хранится только хэш нового открытого ключа до первой транзакции после смены закрытого ключа, что ещё выше поднимает безопасность использования кошелька.

## Уникальные преимущества eGOLD



-  В eGOLD вместо блокчейна применён многоуровневый алгоритм для параллельных транзакций следующего постблокчейнового поколения.
-  Использование параллельных транзакций позволяет достигать неограниченной пропускной способности в отличие от стандартного блокчейна, где каждая транзакция в сети записана в строгой последовательности. Скорость анонса транзакции достигает 4-х секунд.



Создание монет eGOLD реализовано технологией стекинга с вознаграждением от 10% до 13% в месяц от общего количества монет, позволяющая майнить новые монеты на своём кошельке. Майнинг происходит каждую секунду и монеты зачисляются в автоматическом режиме на баланс. Это сделано для справедливого распределения монет между всеми участниками системы. А первоначальный выпуск сделан только для старта распространения монет. При этом, в соответствии с алгоритмом работы криптовалюты, разработчик не может добавить монеты в систему извне. Каждый держатель ноды является гарантом соблюдения правил заложенных в eGOLD.



Майнинг начисляется каждую секунду по сложному проценту и все монеты можно использовать в любой момент времени без каких-либо дополнительных транзакций.



# Уникальные преимущества eGOLD



Впервые, в криптовалюте реализована уникальная 3-х уровневая реферальная система с постоянным майнингом от баланса кошельков в структуре, как вознаграждение за продвижение и популяризацию монеты, начисляемое при любой исходящей транзакции на кошельке в структуре:

- ✓ 52% за год от баланса рефералов 1 уровня;
- ✓ 26% за год от баланса рефералов 2 уровня;
- ✓ 13% за год от баланса рефералов 3 уровня.

Если у реферала есть нода, вознаграждение возрастает на 25%.

В монете используется халвинг – снижение майнинга монет из месяца в месяц. См. ниже.



Стоимость транзакций всегда составляет 2 монеты для любого количества отправляемых монет, а при совершении транзакции через свою ноду – стоимость составит 1 монету.

## Уникальные преимущества eGOLD



Отсутствие майнинга нодой в традиционном смысле, компенсируется вознаграждением держателям нод равным 1 монете eGOLD от каждой транзакции и повышенным майнингом по сравнению с обычным кошельком.



Существует служебный адрес самой криптовалюты G-1 для уничтожения монет.



Инновационная технология горизонта истории транзакций позволяет держать неизменным и настраивать самостоятельно объём базы данных транзакций на нодах.



Возможна лёгкая интеграция в любые веб проекты по средствам простого API с JSON ответами и POST / GET запросами, а также прямого использования PHP и MySQL.



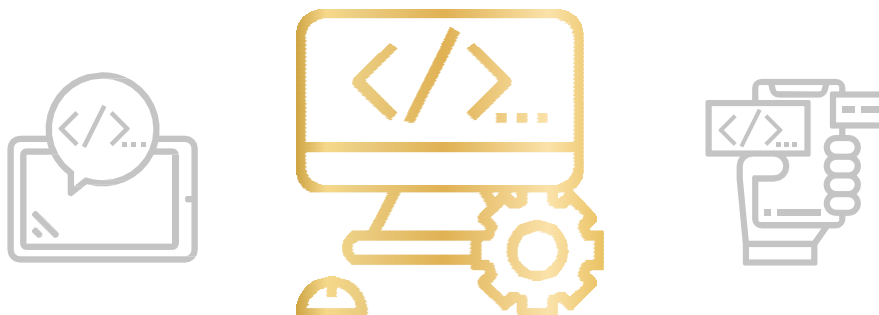
Аналог халвинга - снижение майнинга монеты из месяца в месяц путём снижения на 1% от предыдущего месяца. То есть, через месяц прирост составит 9.9%, через двенадцать 8.86%, через 5 лет 5.47% в месяц и так будет уменьшаться до 1 марта 2035 года 2,02% и 2,53% на ноду в месяц.



Весь исходный код открыт.

## Уникальные преимущества eGOLD

Важной отличительной особенностью eGOLD является её кошелёк. Это один HTML файл, работающий на любых устройствах. Этот файл содержит все необходимые скрипты и картинки.



- ✓ Кошелёк обладает полным функционалом. Отображаются все транзакции, реферальные поступления, возможно управление контактами, а также E-mail уведомления по входящим и исходящим транзакциям.
- ✓ Присутствуют фильтры, чтобы найти транзакцию по дате или цифровому комментарию метке - пинкоду.
- ✓ Все действия защищены множеством алгоритмов, что исключает ошибки и взлом. Даже создание нового закрытого ключа имеет свою систему защиты от случайных действий пользователя.
- ✓ В самом кошельке есть функция смены закрытого ключа и создания нового кошелька.
- ✓ Управление кошельком интуитивно понятно любому человеку, который никогда не имел дела с криптовалютами.



- 📍 **1 -е полугодие 2018 года** - сбор информации, разработка концепции криптовалюты eGOLD и выбор применяемых технологий и технической реализации.
- 📍 **2 -е полугодие 2018 года** - разработка ноды и выбор применяемой технологии для официального кошелька и запуск P2P обменника. Нарботка опыта для создания на основе данного обменника обмена P2P в монете eGOLD.
- 📍 **1 -й квартал 2019 года** - завершение разработки ноды и добавление API по взаимодействию с нодой.
- 📍 **2 -й квартал 2019 года** - разработка дизайна и принципов работы автономного HTML кошелька и внесение корректировок в работу API ноды для подстройки под необходимые запросы и данные кошелька.
- 📍 **2-е полугодие 2019 года** - разработка автономного HTML кошелька и P2P обменника. Тестирование, отладка ноды.

01

02



- 📍 **1-й квартал 2020 года** - завершение разработки автономного HTML кошелька.
- 📍 **2-й квартал 2020 года** - запуск криптовалюты eGOLD. Продажа предмайнинга.
- 📍 **3-й квартал 2020 года** - создание белой книги, промо материалов для продвижения и начало продвижения монеты eGOLD. Выявление ошибок.
- 📍 **4-й квартал 2020 года** - разработка примеров интеграции и помощь для упрощения интеграции монеты в различные системы.
- 📍 **1-е полугодие 2021 года** – разработка нового функционала кошелька и ноды: интегрированного P2P обмена с арбитражом и добавление реферальной ссылки.
- 📍 **2-е полугодие 2021 года** – запуск и отладка полностью децентрализованной монеты за счёт её естественного распространения среди всех участников, уничтожение излишка монет путём отправки на служебный адрес G-1. Повышение майнинга до 200% в год, введение халвинга.
- 📍 **2022-2036 год** децентрализованное распространение монеты, цена на монету определяется рыночным механизмом спросом и предложением. Монета популяризируется реферальной ссылкой и свободным определением цены. Халвинг снижает из месяца в месяц, майнинг монеты и монета становится на первом этапе популярным инвестиционным инструментом, а затем полноценным средством для обмена за товары и услуги.

03

04



## Почему именно eGOLD?

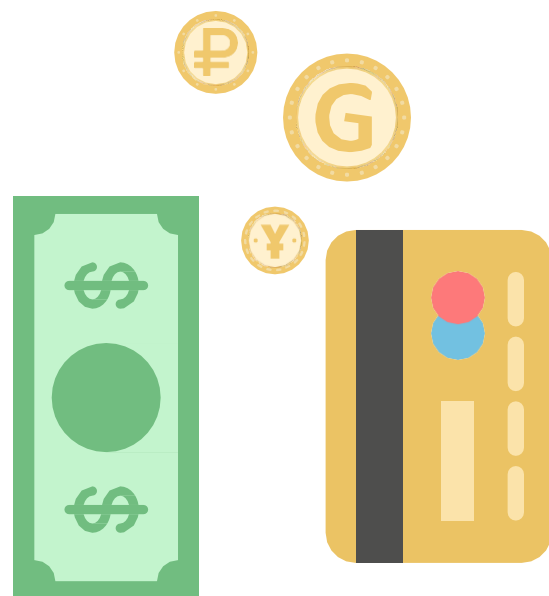
На начальном этапе развития в монете используется сильнейший механизм мотивирующий увеличение количества пользователей монеты, этот механизм используется только на начальном этапе. С течением времени людей, пользующихся монетой, будет становиться всё больше. Каждый владелец монет майнит новые монеты и всё больше товаров и услуг можно будет обменять за монету.

Обменивать за монету товары и оказывать услуги за монету намного выгоднее, чем за обычные средства платежа. Монеты майнятся в кошельке, а биткоины, рубли, доллары не майнятся.

Ликвидность монете обеспечивает простота создания нового кошелька и реферальная ссылка позволяющая превратить любой кошелек в настоящую майнинговую ферму по добыче передовой криптовалюты eGOLD.

Цель создания монеты: плавно, незаметно заменить фиатные деньги, лишить государства возможности налогообложения обращения.

О монете очень интересно и действительно выгодно рассказывать. Можно сказать, что на первоначальном этапе маркетинг eGOLD действительно очень сильно напоминает технологически продвинутую пирамиду. Но халвинг работает и он снижает майнинг из месяца в месяц. Первопроходцы монеты получают больше награду. Процесс развития eGOLD очень напоминает развитие биткоина, когда-то их можно было майнить на домашнем компьютере, но их мало кто майнил. Все сомневались.



## Почему именно eGOLD?

Биткоин созданный, в 2009 году показал пример грамотного маркетинга – полная анонимность и усложнение добычи повышают стоимость биткоина многие годы. Но технология POW – вознаграждение за работу - зашла в тупик. Чтобы майнить нужно иметь постоянно самое новейшее оборудование. Зарабатывают не майнеры, а изготовители оборудования и электроэнергии. Растут и затраты на добычу и централизация.

Технология POS майнинга лишена этих недостатков. Майнинг в зависимости от владения доли монет и выполнения условий более прост, справедлив и понятен. Ни графические процессоры, ни микросхемы не влияют на скорость майнинга в POS.

Майнинг в монете eGOLD зависит от следующих параметров:

- личного баланса;
- баланса структуры на 1,2,3 уровнях;
- наличия ноды привязанной к кошелькам;
- количества месяцев прошедших с 1 ноября 2021 года.

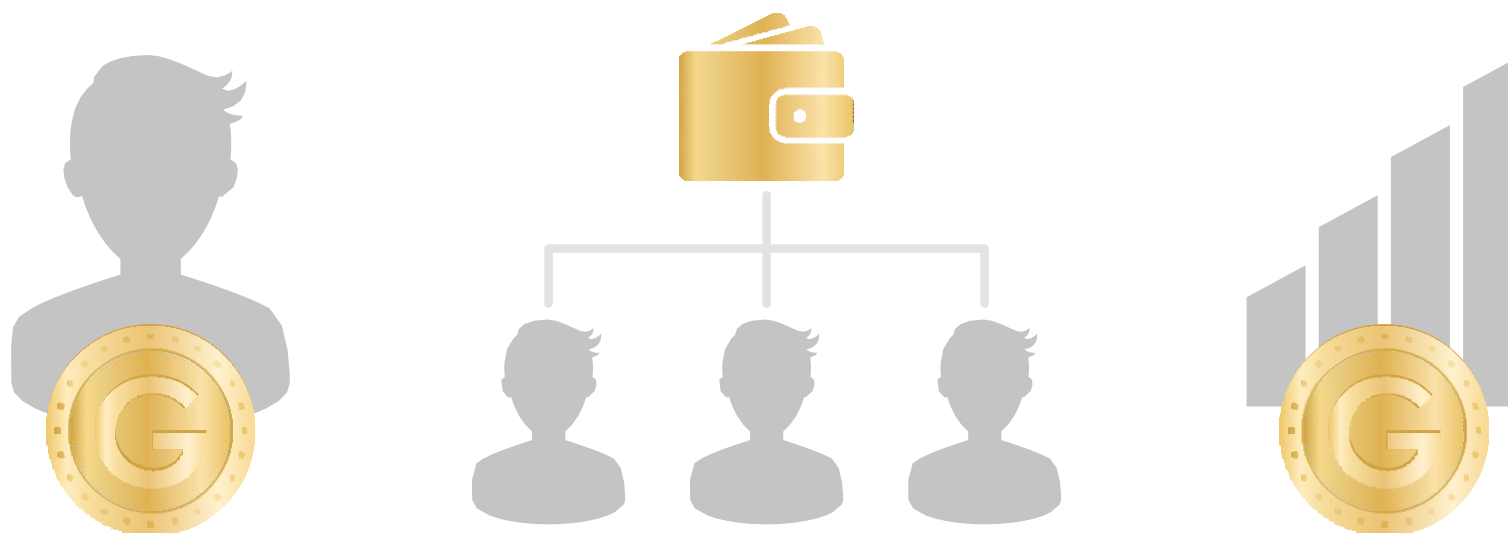
Чем больше личный баланс, тем больше абсолютный майнинг на кошельке.

Майнинг на январь 2022 года - 10% в месяц.

eGOLD – это первая криптовалюта с реферальной ссылкой на сам кошелек в криптовалюте и трехуровневой реферальной системой.



## Почему именно eGOLD?



eGOLD – первая криптовалюта в которой можно заработать без инвестиций, на продвижении криптовалюты, на её рекламе благодаря реферальной ссылке и майнингу от баланса в трёх-уровневой реферальной структуре.

3-х уровневая автоматическая реферальная система встроена в саму криптовалюту. Обычно, любые партнерские вознаграждения реализуются через вспомогательные сервисы: сайты, кабинеты, приложения. Аналогов этой технологии не существует в криптовалютах. Как бы не относились к реферальному вознаграждению – это сильнейшая мотивация к распространению информации о криптовалюте, мотивация к покупке монет вашими приглашенными.

Популярность, распространение и использование монеты, а вместе с этим, и доход от неё, полностью в руках всего сообщества и каждого владельца монеты eGOLD! eGOLD – это монета с мотивационным двигателем огромной силы.



01011010101101010  
1011 FALCON 01  
0100  
1 SHA3 DAG 1  
0 PHP MySQL



**eGOLD** является экологичной **PoS** криптовалютой на постблокчейновой технологии неограниченной пропускной способности параллельных транзакций с использованием быстрого **квантовоустойчивого** алгоритма шифрования **FALCON** и что особенно важно - возможностью смены ключей шифрования (открытого и закрытого ключа) с сохранением номера кошелька, при этом новый публичный ключ закрыт под **SHA3** хэшированием и своим специально разработанным хэшированием до момента следующей транзакции, что делает невозможным вычисление закрытого ключа по открытому в связи с его полным сокрытием. В работе с кошельком используется только сменяемый закрытый ключ и постоянный номер кошелька.

В криптовалюте eGOLD нет блокчейна, вместо этого используется более совершенный многоуровневый алгоритм следующего поколения для синхронизации транзакций похожий на **DAG (directed acyclic graph – направленный ациклический граф)** с существенным отличием, закрывающим проблемы параллельной синхронизации – это рассылка об успешной транзакции всем нодам веером по цепочке, если эта транзакция не приходила ранее и она прошла проверку на валидность, включая проверку на верность и принадлежность подписи кошельку отправителя. Для работы ноды нужен обычный интернет хостинг с PHP и MySQL. Время принятия одной транзакции около **4 секунд** (обработка транзакции около 1-2 минут). Защита от двойной траты сделана с помощью временной задержки и похожа на защиту криптовалюты **NANOCOIN**. Также в нодах существует проверка на пропущенные транзакции и постоянная проверка валидности кошельков по более чем 50% баланса доступных нод. В eGOLD встроена система самоочищения базы данных, путем введения **горизонта истории транзакций**, это позволяет сохранить неизменным её объем при одинаковом количестве транзакций.

## Почему именно eGOLD?

К примеру, у **Биткоина** на текущий месяц (июнь 2020 года) в среднем около 300 тысяч транзакций в день, а объем базы данных блокчейна постоянно увеличивается и превысил 300 гигабайт. В eGOLD же объём базы данных настраиваемый и остаётся неизменным, благодаря горизонту истории транзакций и другим методам самоочистки. Майнинг отсутствует, но есть вознаграждение держателям нод в виде 1 монеты от каждой транзакции ноды и повышенный майнинг держателям нод от баланса кошелька прикрепленного к ноде. Все кошельки имеют майнинг 10% в месяц (на 2022 год\*) монет на баланс кошелька. Все владельцы монет являются майнерами новых монет в зависимости от действий по продвижению монеты. Зачисление в баланс происходит автоматически без создания дополнительных транзакций. Монеты могут уничтожаться путём отправления на адрес G-1. G-1 - это служебный адрес, не имеющий кошелька. Других таких служебных адресов нет и можно отправлять монеты только на адреса кошельков. После реализации предмайнинга, монеты майнятся на кошельках пользователя в соответствии с правилами маркетинга, никакого ограничения генерации монет нет. Пользователи всегда могут посмотреть общий объём монет в ноде с помощью запроса адрес://[IP ноды]/egold.php, где в ответе общий объём монет на всех кошельках без учёта начисления процентов с момента последней транзакции.

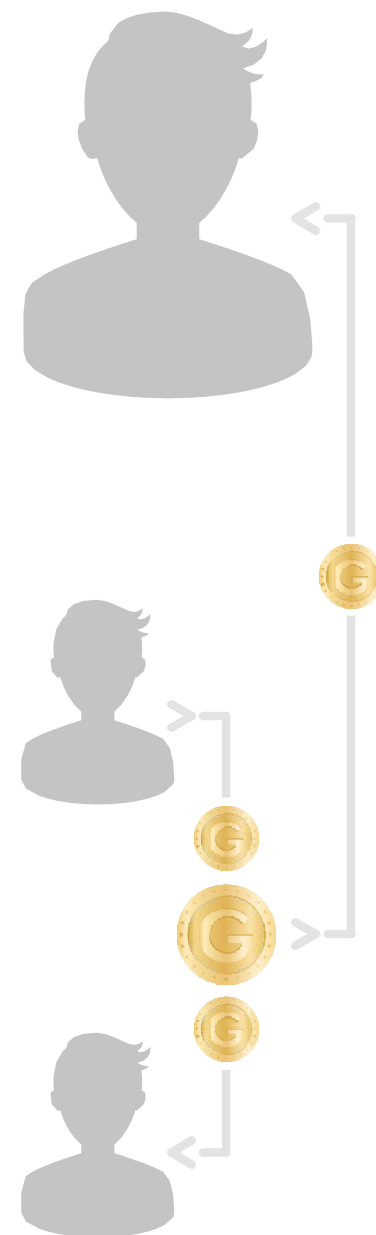




## Почему именно eGOLD?



В eGOLD в качестве единиц используются только целые числа 1,2,3... без сотых. Комиссия на все транзакции 2 монеты и не зависит от суммы транзакции. При этом 1 монета всегда уходит владельцу ноды к которой подключается кошелёк для совершения транзакций и 1 монета списывается системой для защиты её от атак с использованием многочисленных транзакций. Когда пользователь совершит транзакцию и добавит пароль для доступа к ноде, он сможет воспользоваться сервисами ноды, через которую была совершена эта транзакция: книгой контактов (все контакты зашифрованы от владельца ноды) и E-mail уведомлениями о транзакциях на почту. В монету встроена 3-х уровневая автоматическая реферальная система. Каждый раз при любых транзакциях своих рефералов, владелец рефералов получает майнинг в виде монет от своих рефералов, зависящий от баланса своих рефералов, прошедшего времени между транзакциями у рефералов и уровня рефералов: 53% (1-ый уровень), 26% (2-ой уровень) и 13% (3-ий уровень) (2022 год\*).



## Почему именно eGOLD?

Создание кошелька стоит 5 монет, в которые уже входят комиссионные 2 монеты системы, а 3 монеты остаются на вновь созданном кошельке для возможности смены закрытого ключа.

Создать кошелек может настроенная нода, либо владелец уже созданного кошелька имеющий на балансе больше 5 монет.

Создание кошелька происходит вместе с созданием секретной фразы для нового кошелька, которую потом нужно сменить. Это сделано для защиты от атак, использующих многочисленные транзакции.

При отсутствии смены приватного ключа или пополнении баланса на кошельке в течении 72 часов – кошелек уничтожается. База очень компактна по этой причине.



## Почему именно eGOLD?

Код ноды написан на PHP (требуется версия PHP не ниже 7.3, библиотека PHP-BCMATH, GMP и CURL) + MySQL (рекомендуется использовать актуальные версии). Официальный кошелёк написан на HTML с JavaScript, jQuery, Ajax и сделан в виде одного HTML, что делает возможным его запуск на любых устройствах. FALCON шифрование с использованием закрытого ключа происходит непосредственно в самом HTML файле, а в ноду уже отправляется подписанная транзакция, что полностью защищает секретную фразу от кражи, путем перехвата в момент отправления до ноды. Пара закрытого и открытого ключа тоже генерируется на HTML кошельке.



Для удобства пользователей в ноду встроены модули для хранения контактов и отправки E-mail уведомлений о транзакциях, аналогично SMS уведомлениям банков о зачислении или списании средств. Имена контактов в ноде хранятся в зашифрованном виде. Доступ к контактам и управлению E-mail уведомлениями на ноде осуществляется с помощью пароля. Пароль задается и изменяется при любой транзакции. Для удобства, в кошельке для задания пароля сделана отдельная функция. Сервисы контактов и E-mail уведомлений работают на конкретной ноде и доступны участнику только после совершения транзакции на данной ноде, что стимулирует пользователей использовать определенные ноды, а владельцу ноды с каждой транзакции по ней, идёт вознаграждение в виде 1 монеты. Для работы ноды на балансе кошелька, привязанного к ноде должно быть не менее 1000 монет.

## Почему именно eGOLD?



Нода eGOLD обладает возможностью самоочищаться от неиспользуемых данных, что защищает её от чрезмерного использования памяти и ресурсов системы, сохраняя одинаковый объем базы данных при одинаковом количестве транзакций путем удаления:

- ✓ истории транзакций старше 30 дней (параметр настраивается переменной `$history_day` в файле настроек `egold_settings.php`)
- ✓ кошелька от ноды, если по кошельку не было транзакций или нодой не пользовались больше 30 дней
- ✓ учётных записей для использования отправки E-mail уведомлений и контактов кошелька, если кошелёк не пользовался этой нодой более 365 дней
- ✓ кошельков с балансом меньше 10 монет, если ими не пользовались больше 90 дней
- ✓ кошельков, если по ним не было входящих и исходящих транзакций больше 30 лет!

Исходный код монеты открыт и прост, поэтому монета будет пользоваться популярностью у разработчиков. Каждый сможет сделать себе или всему сообществу модули, решающие те или иные задачи. А владельцы сайтов смогут встраивать ноды в любые свои веб проекты и напрямую принимать eGOLD монеты. Генерация открытого и закрытого ключа, подпись транзакций (оплата), также возможна и с ноды, что позволяет автоматизировать сервисы с выводом средств на кошельки участников. Настройки ноды хранятся в файле `egold_settings.php`. Запросы к нодам осуществляются с помощью GET и POST обращения к файлу `egold.php`, а ответы ноды представлены в виде JSON.





01

## **В** Как войти в кошелёк?

**О** Для этого его нужно зайти на сайт и перейти по ссылке на файл **eGOLD.html**. Данный файл является официальным кошельком криптовалюты eGOLD. Его можно также скачать. Проверяйте источник файла.

После запуска файла, откроется страница входа в кошелёк. На ней необходимо ввести 18-и значный цифровой номер кошелька и IP ноды к которой будете подключаться.

## **Как сменить закрытый ключ?**

02

**В** Закрытый ключ изменяется при нажатии на кнопку «сменить закрытый ключ» со стрелками. При нажатии на кнопку открывается страница, где нужно

**О** -двигать курсором мыши, либо пальцем по экрану пока на прогресс-баре не наберётся 100%.

- генерация нового закрытого ключа может занять до 10 минут.

- введите ваш закрытый ключ

- нажмите кнопку «сменить закрытый ключ»

Появится новый закрытый ключ, который нужно сохранить.





После смены закрытого ключа, восстановить старый закрытый ключ будет никак невозможно. Это криптовалюта. Поэтому, проверьте новый закрытый ключ, перед переводом на него больших средств, вдруг вы допустили ошибку при его сохранении. Стоимость смены закрытого ключа кошелька составляет 3 монеты.

03

### **В** Как перевести средства?

**О** Для этого нужно вписать 18ти значный цифровой номер кошелька получателя и сумму и нажать кнопку «ОТПРАВИТЬ».

Если всё сделано верно, то появится большая галочка о совершении транзакции. Комиссия за транзакцию всегда составляет 2 монеты.



### 04 **В** Как создать новый кошелёк?

**О** Для этого нужно нажать на кнопку "кошелька с плюсом" и проделать всё тоже самое, что и при смене закрытого ключа. Появится номер нового кошелька и закрытый ключ. Данный кошелёк с ключом нужно передать новому пользователю. Стоимость операции регистрации нового кошелька составляет 5 монет.

### 05 **В** Почему невозможно зайти в новый кошелёк сразу после его создания, а также нельзя перевести на него средства?

**О** Создание нового кошелька прописывается в транзакции и новый кошелёк начинает работать после принятия транзакции нодами по более чем 50% баланса кошельков нод в соответствии с PoS алгоритмом. В среднем, это занимает около 2 минут и в официальном кошельке eGOLD.html строка транзакция перестаёт быть оранжевого цвета.

### 06 **В** Внастройках официального кошелька eGOLD.html задаётся пароль. Для чего он?


**О** Данный пароль используется для доступа к дополнительным сервисам ноды. На текущий момент, любая нода может хранить контакты пользователя и если есть домен у ноды, и он прописан в настройках



ноды `egold_settings.php`, нода может отправлять E-mail уведомления о входящих и исходящих транзакциях по кошельку в соответствии с установленными порогами по количеству монет в транзакциях. Если пароль изменить, то контакты удаляются. Нода предоставляет доступ к своим сервисам с момента первой транзакции отправленной в неё и до момента отправления транзакции в другую ноду. Чтобы начать пользоваться услугами ноды, нужно вписать пароль и если этот пароль ещё не задан, появится форма для его создания. Для этого, введите любой пароль от 6 символов, напишите свой закрытый ключ и нажмите сохранить. Чтобы удалить пароль ноды, сотрите его в настройках и нажмите сохранить. Чтобы изменить, напишите любые знаки в форме пароля и затем задайте новый пароль. Стоимость операции создания или смены пароля к ноде составит 3 монеты.

### 07 **В** Как совершать операции максимально безопасно?

**О** Правильно - скачать файл кошелька на свой компьютер из архива в разделе кошелька ноды. Проверить MD5 сумму. При работе с кошельком с локального компьютера закрытый ключ остаётся на компьютере и никуда не передаётся. В ноду передаётся лишь уже подписанная транзакция, что делает безопасным работу с любой нодой. При использовании на стороннем ресурсе, доподлинно неизвестно, модифицирован ли кошелёк с целью похищения закрытого ключа или нет. Поэтому, скачав архив с кошельком, проверяйте его с помощью MD5 не модифицирован ли он, сравнив строку **MD5** сразу на



нескольких доверенных ресурсах. Это исключит возможность фишинга или подмены кошелька. Что это такое, как и где это делается можно найти в любой поисковой системе интернета. В целях безопасности, проверять нужно сразу на нескольких ресурсах.

**08** **В** Есть ли у разработчика возможность дополнительной эмиссии монет?

**О** Технически это невозможно, так как после запуска нод, потребуется полный перезапуск всего с остановкой всех нод, чтобы добавить монеты из вне.

**09** **В** Насколько eGOLD зависит от разработчика?

**О** Если с разработчиков что-то случится, то проект будет жить сам. Исходный код открыт, все монеты на руках пользователей. Всё продолжит работать.



10

**В** Как будет регулироваться цена на монету?

**О** Цена будет определяться рыночным механизмом – спросом и предложением. Спрос формируется реферальной ссылкой и трёхуровневой реферальной системой, а предложение зависит от значения халвинга, количества месяцев прошедших с 1 ноября 2021 года, каждый месяц майнинг уменьшается на 1% от предыдущего. К 2036 году майнинг снизится до минимальных значений. Маркетинг в eGOLD схож с маркетингом биткоина, где предложение сокращается в 2 раза после халвинга. После уничтожения предмайнинга остаются только те монеты, которыми владеют пользователи-майнеры. Майнер монет самостоятельно определяют по какой цене он будет продавать монеты исходя из своих соображений. Биржи, которые самостоятельно залистят монету помогут популяризации и определению рыночного курса монеты.

11

**В** Есть ли возможность усложнения генерации новых монет?

**О** Начиная с 1 ноября 2021 года в каждый следующий месяц добыча монет уменьшается на 1% по сравнению с предыдущим месяцем. Снижение будет происходить до марта 2036 года, когда это снижение остановится и составит 35,4% в год на ноде и 27,5% на кошельке без ноды. Это аналог халвинга биткоина, который очень хорошо себя показал положительно влияя на цену биткоина.





- 12** **В** Будет ли увеличение количества площадок на которых происходит обмен монет? Надёжность этих площадок?
- О** В монете реализован P2P обмен с залогом и арбитражем. Этот механизм позволяет быстрее, дешевле, без регистрации, обменивать монеты на другие платёжные средства. Биржи технологически проигрывают P2P обмену. Блокчейн eGOLD позволяет все сделки удалять из блокчейна спустя 7 дней. Анонимность на высочайшем уровне. Биржи могут листить монету самостоятельно или по предложению сообщества пользователей монеты eGOLD. Биржа помогает найти взвешенный курс монеты на момент времени.
- 13** **В** Какие преимущества есть у кошелька перед кошельками других криптовалют?
- О** Кошелек прост, интуитивно понятен, русифицирован, есть английская версия кошелька. Кошелек – это один файл. Его не нужно скачивать с play market и app store. Для совершения транзакции нужно только подключиться к любой ноде. Между вами и вашими монетами только приватный ключ, который вы можете сменить в любой момент.
- 14** **В** При обращении к ноде, для проведения транзакций, передается ли парольная фраза (закрытый ключ)?
- О** Фраза никогда не передаётся в интернет при условии запуска файла кошелька eGOLD.html с локального компьютера. Важно предварительно сверять MD5 архива, из которого брали кошелек, сразу с несколькими доверенными ресурсами. В ноду всегда отправляется уже подписанная

транзакция не содержащая приватного ключа. Закрытый ключ сразу же удаляется после подписания транзакции в кошельке.

15

**В** Может ли мошенник подключиться к ноде и украсть приватный ключ (закрытый ключ)?

**О** Нет никакой возможности узнать приватный ключ. Нода никогда не получает приватный ключ. А если его там нет, то и украсть его невозможно. Нода получает всегда уже подписанную транзакцию.

16

**В** Как происходит майнинг на кошельке?

**О** Майнинг монет происходит каждую секунду на всю сумму и на ту, что уже прибавили в прошлую секунду. То есть происходит начисление майнинга по сложному проценту. Намайненные монеты всегда доступны к использованию. Никаких дополнительных действий для зачисления их на баланс не требуется и всегда можно распоряжаться всеми намайненными монетами.



**17 В** А если кошельком долго не пользоваться, будет ли он майнить монеты?

**О** Если по кошельку или ноде кошелька не было входящих и исходящих транзакций более 3 650 дней (почти 10 лет), то майнинг останавливается. Связано это с целью ускорения работы ноды.

**18 В** Когда происходит фиксация майнинга с рефералов?

**О** Майнинг с рефералов происходит, когда на кошельке реферала происходит транзакция. При этом, количество монет по реферальной программе происходит с округлением до целого числа в меньшую сторону. То есть, если не набирается хотя бы одной монеты по реферальной программе, то майнинга одной монеты не происходит. Начисление осуществляется по сложному проценту и зависит от баланса реферала на момент предыдущей транзакции реферала, уровня реферала и прошедшего времени с момента предыдущей транзакции реферала.

**19 В** Вижу на странице прихода от рефералов зачисление, но оно оранжевого цвета и от него на баланс нет поступления. Когда произойдёт зачисление на баланс?

**О** Сразу после поступления средств от рефералов, они зачисляются на накопительный реферальный баланс и транзакция по рефералам в кошельке становится оранжевой до тех пор, пока монеты с реферального баланса не будут зачислены на основной баланс.



20


**В** Когда начинает работать новый закрытый ключ после смены старого? Нужно ли хранить старый закрытый ключ?

**О** Новый закрытый ключ начинает работать сразу после смены старого закрытого ключа. С этого момента старый закрытый ключ больше не работает и не будет работать в дальнейшем. Единственное, если по какой-то причине транзакция не была принята нодами по более чем 50% баланса кошельков нод в соответствии с PoS алгоритмом, она может быть удалена и новый закрытый ключ не будет работать, потому что он исчезнет из нод. Поэтому не удаляйте старый закрытый ключ, по крайней мере, минут 5. Технология сохранения нового закрытого ключа аналогично технологии сохранения новых транзакций и прописывается в самой транзакции, путём подписания старым закрытым ключом транзакции с новым закрытым ключом.

21

**В** Какой часовой пояс используется в официальном кошельке eGOLD.html и что за время отображается в нём снизу слева?

**О** Снизу слева в транзакциях и начислениях от рефералов отображается текущий часовой пояс устройства на котором открыт кошелек



eGOLD.html. Также, там отображается время получения данных с ноды по транзакциям. У каждой транзакции время и дата в соответствии с часовым поясом устройства.

**22** **В** Имеет ли владелец ноды доступ к сохранённым на ней контактам или почте?

**О** Все номера кошельков и их названия в контактах шифруются паролем к ноде, а владельцу ноды передаётся только хэш пароля для доступа к его ноде. Поэтому, без самого пароля данные по контактам не просмотреть. Сама же почта видна владельцу ноды. При сохранении почты, она один раз передаётся в ноду в зашифрованном виде и больше не передаётся обратно ни в каком виде. Поэтому, вместо почты в настройках кошелька указаны звёздочки.

**23** **В** Что за вторая буква G с плюсом и числом отображается правее баланса?

**О** Это количество намайненных монет от рефералов готовых к зачислению. Они перейдут в баланс после любой транзакции на вашем кошельке.





**24 В** Установлена своя нода. Входящих и исходящих транзакций по кошельку не было и в бонусе G+ было 2 монеты. сейчас G+ стал равен 0. Почему?

**О** Произошло зачисление монет на баланс кошелька. Алгоритмом ноды предусмотрено зачисление монет от транзакций сразу в баланс без совершения транзакций.

**В** Как изменить привязанный к ноде кошелёк?

**25 О** Для смены кошелька у ноды нужно зайти через ноду на кошелек привязанный к ноде и в настройках нажать на «отключить ноду». Прописать новый кошелёк в файле настроек ноды `egold_settings.php` и зайти в этот новый кошелек через эту ноду и привязать новый кошелек к ноде там же в настройках. Стоимость привязки кошелька к ноде 102 монеты. Как только транзакция будет обработана, нода привяжется к кошельку. При потере доступа к кошельку, нода отвяжется от него самостоятельно через 30 дней и перестанет работать, если по ней и! по кошельку, привязанному к ноде, не будет транзакций в течение 30 дней.

**26 В** На каких устройствах работает нода и какие системные требования?

**О** Нода невероятно лёгкая и работает на любых операционных системах и устройствах практически любой мощности. Размер базы данных можно настроить самому в большую или меньшую сторону путём

изменения времени хранения истории транзакций. На текущий момент, рекомендую выделить для хранения данных не менее 1 гигабайта.

Основные требования - это статический IP адрес и установленный PHP версии от 7.3 с MySQL. Всё это можно установить практически на любую операционную систему, включая Windows, MAC OS, Linux и, даже, мобильные устройства. Как это делается можно найти в интернете.

27

**В Нода не работает. Что мне делать?**

**О** Сначала нужно установить последнюю версию ноды (скачать и разместить файлы в папке с заменой старых при необходимости). Затем, если она не заработает, необходимо устранить причины неправильной её работы, настроив среду, в которой она работает. Возможно, что-то в этой среде не установлено, либо не та версия необходимых компонентов среды, либо что-то работает не так в самой среде. Сделать это можно самостоятельно, либо при помощи техподдержки, где установлена нода, либо спросив у других людей, например в чате. Ошибки, возникающие в самой ноде, выводятся сразу при обращении к ней, либо более детально можно посмотреть по логу системы, где она установлена, а также раскомментировав первые строки для логирования с параметрами `ini_set`. При наличии ошибок в ноде, в папке с файлом ноды `egold.php` появится файл лога `egold_error.log`. Он открывается в любом текстовом редакторе.

28

**В Указал домен `mail.ddns.net` в настройках ноды. Через некоторое время письма с уведомлением о транзакциях перестали приходить. Почему?**

**О** Свой домен должен быть платный и не выше 2 уровня. Данный домен бесплатный и 3 уровня (3-и слова и 2-е точки между названиями). Зачастую, почтовые сервера не пропускают письма с таких доменов.



## 01 Запрос к ноде без параметров

Пример обращения к ноде по ссылке: [http://\[ip ноды\]/egold.php](http://[ip ноды]/egold.php)

Пример ответа страницы: {"time":"1586582875", "noda":"91.106.203.179", "owner":"G-1000-00000-0000-00000", "balanceall":"32639969", "walletscount":"2156"}

- **time** - время на ноде в UNIX формате
- **noda** (параметр нужен для работы ноды и он прописывается в файле настроек egold\_settings.php под названием **\$noda\_ip**) - IP текущей ноды
- **owner** (параметр нужен для работы ноды и он прописывается в файле настроек egold\_settings.php под названием **\$noda\_wallet**) - кошелек владельца ноды и нужен для того, чтобы нода заработала
- **balanceall** всего монет в системе
- **walletscount** всего кошельков в системе

## 02 Запрос [type=wallet] - данные по кошельку

Пример GET запроса: [http://\[IP ноды\]/egold.php?type=wallet&wallet=G-1261-21479-1061-51551](http://[IP ноды]/egold.php?type=wallet&wallet=G-1261-21479-1061-51551)

Пример JSON ответа: {"time":"1644761588", "noda":"5.181.110.217", "owner":"G-9960-05006-8165-84807", "percent":"1.00000003706", "percent\_noda":"1.000000046325", "wallet":"G-9960-05006-8165-84807", "ref1":"G-5744-52986-7449-40410", "ref2":"G-9968-28846-4534-44639", "ref3":"G-8141-25257-4368-09093", "nodawallet":"5.181.110.217", "nodawalletuse":"5.181.110.217", "balance":"148997", "percent\_4":"40.850338974393", "percent\_5":"51.064673203516", "percent\_ref":"5", "height":"35", "date":"1644754191", "nodaping":"","signpub":"","sign":"","signpub":" публичный ключ кошелька", "sign":"":"[подпись последней транзакции]"}

В каждом запросе обязательно должен быть указан его тип: **type**. В примере используется тип запроса **wallet** с дополнительным параметром **wallet** равным номеру кошелька.

- **nodawallet** - нода которая принадлежит данному кошельку
- **nodawalletuse** - нода с которой была последний раз исходящая транзакция этим кошельком
- **balance** - текущий баланс
- **percent\_4** - начисление майнинга по текущему значению в общем случае
- **percent\_5** - начисление майнинга по текущему значению при ноды.



- **percent\_ref** - накопленный процент от рефералов, который зачисляется при исходящей транзакции
- **height** - высота последней транзакции кошелька
- **date** - время и дата последней транзакции в UNIX формате
- **signpubnew** - новый открытый ключ под хэшем (если параметр не пустой, значит изменялся закрытый ключ) **signnew**
- - подпись транзакции новым закрытым ключом (если параметр не пустой, значит изменялся закрытый ключ) **signpub**
- - открытый ключ кошелька
- **sign** - подпись транзакции закрытым ключом
- 

\* Дополнительно в запросе можно просмотреть настройки аккаунта на ноде для отправки уведомления, для этого отправляется параметр пароля **password** хэшем SHAKE256 длиной 256 символов (1024 бита) от предыдущего пароля, который был уже записан на ноду под хэшем SHAKE256 длиной 128 символов. То есть, сначала от предыдущего пароля берётся хэш SHAKE256 на 128 символов на стороне клиента и отправляется в ноду для задания пароля, а потом для проверки пароля на ноду уже от имеющегося хэша берётся ещё раз хэш, но длиной 256 символов, используя функцию `gen_sha3([пароль на ноду],256)`, отправляется с клиента и проверяется на ноду. Если соответствует два хэша, тогда считаем всё верно и показываем настройки: есть ли почта и с какими порогами.

\* Если пароль будет изменён (будет отличаться от уже установленного), то все сохранённые контакты будут удалены, так как они зашифрованы под старый пароль.

**Пример GET запроса:** : `http://[IP ноды]/egold.php?type=wallet&wallet=G-1261-21479-1061-51551&password=[пароль под SHAKE256 с длиной хэша 256 от предыдущего который под SHAKE256 с длиной хэша 128]`

**Пример JSON ответа:** `{"time":"1567159668","noda":"[IP ноды]","wallet":"G-1261-21479-1061-51551","ref1":"G-4613-11305-1110-01414","ref2":"G-1514-07813-0081-39731","ref3":"G-1000-00000-0000-00000","nodawallet":"91.106.203.180","nodawalletuse":"91.106.203.180","balance":"11578494","percent_4":"121","percent_5":"151","height":"311","date":"1567243045","usersemal":"[email]","usersemalup":"10","usersemaldown":"20","usersemaldateupdate":"1567159580","signpubnew":"","signnew":"","signpub":"[публичный ключ кошелька]","sign":"[подпись последней транзакции]}"`



## 03

### Запрос [type=**nodas**] - список доступных нод

Выводит список нод со статистикой обращения к ним и с количеством кошельков подключенных к ним.

Использует дополнительные необязательные параметры:

- **balancestart** - минимальный баланс кошелька ноды
- **balancefinish** - максимальный баланс кошелька ноды
- **nodausewalletstart** - минимальное количество кошельков, использующих ноду
- **nodausewallefinish** - максимальное количество кошельков, использующих ноду
- **order** - параметр сортировки вывода истории в прямом порядке от меньшей даты к большей с параметром =asc, по умолчанию в обратном порядке по дате последнего использования. Также можно отобразить ноды по балансу от большего к меньшему при order=balance.
- **start** - с какой транзакции в выдаче начать отображать. Здесь и далее в других типах с множественным отображением: при =0 начинаем без пропусков, при =1 сдвигаем выдачу на 1 и так далее.
- **limit** - количество отображенных нод от параметра start. По умолчанию здесь и далее в других типах с множественным отображением =100.

**Пример GET запроса:** [http://\[IP ноды\]/egold.php?type=nodas](http://[IP ноды]/egold.php?type=nodas)

**Пример JSON ответа:**

```
[{"noda":"87.236.17.35","wallet":"283076041494789597","balance":"2851271","datelastuse":"1644853444"}, {"noda":"94.228.124.143","wallet":"149010181898503611","balance":"989550","datelastuse":"1644836217"}, {"noda":"188.225.56.126","wallet":"166024881534782013","balance":"871760","datelastuse":"1644522379"}, {"noda":"45.12.17.209","wallet":"922613103222822701","balance":"646787","datelastuse":"1644502476"}, {"noda":"95.182.120.202","wallet":"708197125367346809","balance":"500534","datelastuse":"1644825695"}]
```

- **noda** - IP доступной ноды
- **wallet** - кошелёк к которому привязана нода
- **balance** - баланс кошелька к которому привязана нода
- **datelastuse** - дата и время последней транзакции по ноде в UNIX формате

## 04

### Запрос [type=**history**] - история транзакций

Использует дополнительные необязательные параметры:

- **all** - при его задании =3 - отображаются все типы истории, включая те, что еще не приняты или отклонены, all=2 - отображаются ещё не принятые транзакции и принятые, all=1 — отображаются только ещё не принятые транзакции, all=0 -отображаются только принятые транзакции. По умолчанию all=0





- **history** - отображает все входящие и исходящие транзакции кошелька и должен быть равен номеру кошелька
- **pin** - любое число до 18 цифр (для определения транзакции). Смотрится полное соответствие знаком равенства
- **wallet** - номер кошелька по которому смотрятся только исходящие транзакции
- **recipient** - номер кошелька по которому смотрятся входящие транзакции
- **date** - дата с которой смотрятся транзакции в UNIX формате
- **dateto** - дата до которой смотрятся транзакции в UNIX формате
- **height** - высота от которой смотрятся транзакции
- **nodause** - нода которую использовал кошелек в последний раз
- **order** - параметр сортировки вывода истории в прямом порядке от меньшей даты к большей с единственным возможным параметром =asc, по умолчанию в обратном порядке
- **history\_exception** - параметр для исключения вывода истории с массивом кошельков и высот для исключенных транзакций  
Используется для синхронизации транзакций между нодами
- **start** - с какой транзакции в выдаче начать отображать. Здесь и далее в других типах с множественным отображением: при =0 начинаем без пропусков, при =1 сдвигаем выдачу на 1 и так далее
- **limit** - количество отображенных транзакций от параметра start. По умолчанию здесь и далее в других типах с множественным отображением =100, но при history по умолчанию =25

**Пример GETзапроса:** [http://\[IP ноды\]/egold.php?type=history&wallet=101245147116351512&height=2&start=0&limit=2&order=asc](http://[IP ноды]/egold.php?type=history&wallet=101245147116351512&height=2&start=0&limit=2&order=asc)

**Пример JSONответа** (здесь и далее пропущены ответы "signpub" и "sign"):

```
{{"wallet":"10000000000000000000","recipient":"260749343133458182","money":"3","pin":"0","height":"2","nodawallet":"10000000000000000000","nodause":"95.169.185.90","nodaown":"1","date":"1590647968","signpubreg":"076b17561d742...","signreg":"656465383964653361613038...","signpubnew":"","signnew":"","signpub":"07f217eb1ad9100c0902290600e2030419b719d122e505bc223809612d1f2f091d96...","sign":"326239...","checkhistory":"1"},{"wallet":"10000000000000000000","recipient":"932333570717013652","money":"3","pin":"0","height":"3","nodawallet":"10000000000000000000","nodause":"95.169.185.90","nodaown":"1","date":"1590648025","signpubreg":"07dd1ba61b731c561b6000862bee1...","signreg":"33316531386335633864656132306633...","signpubnew":"","signnew":"","signpub":"07f217eb1ad9100c0902290600e2030419b719d122e50e...","sign":"323432633...","checkhistory":"1"}}
```

- **wallet** - кошелек отправителя
- **recipient** - кошелек получателя
- **money** - сумма перевода без комиссии
- **pin** - пинкод из цифр до 18 символов в открытом виде
- **height** - высота транзакции по кошельку
- **nodawallet** - нода данного кошелька
- **nodause** - нода которую использовал кошелек при данной транзакции
- **nodaown** - принадлежит ли нода с которой была совершена транзакция данному кошельку
- **date** - дата и время транзакции в UNIX формате
- **signpubreg** - открытый ключ нового кошелька под хэшем (если параметр не пустой, значит был создан новый кошелек с номером в поле recipient)
- **signreg** - подпись части транзакции для создания нового кошелька (если параметр не пустой, значит был создан новый кошелек с номером в поле recipient)
- **signpubnew** - новый открытый ключ под хэшем (если параметр не пустой, значит изменялся закрытый ключ)
- **signnew** - подпись транзакции новым закрытым ключом (если параметр не пустой, значит изменялся закрытый ключ)
- **signpub** - открытый ключ кошелька
- **sign** - подпись транзакции закрытым ключом
- **checkhistory** - статус транзакции: 0 - транзакция не обработана, 1 - транзакция принята, 2 - транзакция отклонена.



\* Если запрос без параметра **all=3** с параметром **history равным кошельку**, то в конце массива выводится число, означающее общее количество записей в истории по данному запросу без разбивки по страницам. Сделано для того, чтобы знать сколько всего страниц с запросами.

**Пример GETзапроса:** `http://[IP ноды]/egold.php?type=history&history=101245147116351512`

**Пример JSONответа** (здесь и далее пропущены ответы "signpub" и "sign"): `[{"wallet":"101245147116351512", "recipient":"1", "money":"1", "height":"2", "nodawallet":"100000000000000000", "nodaused":"91.106.203.179", "nodaown":"0", "date":"1562182374", "signpubreg":"","signreg":"","signpubnew":"","signnew":"","signpub":"","sign":""}, {"wallet":"101245147116351512", "recipient":"1", "money":"1", "height":"3", "nodawallet":"101245147116351512", "nodaused":"91.106.206.144", "nodaown":"1", "date":"1562182398", "signpubreg":"","signreg":"","signpubnew":"","signnew":"","signpub":"","sign":"","checkhistory":"1"}, 10]`

## 05

### Запрос [type=referrals] - история транзакций по рефералам

Использует дополнительные необязательные параметры:

- **wallet** - номер кошелька который совершил транзакцию и кому он зачислил реферальные бонусы
- **ref** - номер кошелька по которому смотрятся все его рефералы всех уровней
- **ref1** - номер кошелька по которому смотрятся все его рефералы 1-ого уровня, если нет параметра ref
- **ref2** - номер кошелька по которому смотрятся все его рефералы 2-ого уровня, если нет параметра ref
- **ref3** - номер кошелька по которому смотрятся все его рефералы 3-его уровня, если нет параметра ref
- **height** - высота от которой смотрятся транзакции
- **nodaused** - нода которую использовал кошелек в последний раз для транзакции
- **date** - дата с которой смотрятся транзакции в UNIX формате
- **dateto** - дата до которой смотрятся транзакции в UNIX формате
- **order** - параметр сортировки вывода истории в прямом порядке от меньшей даты к большей по дате последней транзакции с единственным возможным параметром =asc, по умолчанию в обратном порядке
- **start** - с какой транзакции в выдаче начать отображать.
- **limit** - количество отображенных транзакций от параметра start.

\* В конце массива выводится число, означающее общее количество записей в истории реферальных начислений по данному запросу без разбивки по страницам. Сделано для того, чтобы знать сколько всего страниц с запросами.

**Пример GETзапроса:** `http://[IP ноды]/egold.php?type=referrals&wallet=126121479106151551&height=2&start=0&limit=1&order=asc`

**Пример JSON ответа:**

`[{"wallet":"126121479106151551", "ref1":"461311305111001414", "ref2":"151407813008139731", "ref3":"100000000000000000", "money1":"201214", "money2":"100607", "money3":"50303", "height":"22", "date":"1562940727"}, 10]`

- **wallet** - кошелек который является последователем рефереров разного уровня
- **ref1** - реферер **1 уровня**, то есть кошелек, который создал кошелек **wallet**
- **ref2** - реферер **2 уровня**, то есть кошелек, который создал кошелек того кошелька, который создал кошелек **wallet**
- **ref3** - реферер **3 уровня**



- **money1** - сумма вознаграждения **1%** в месяц с баланса **wallet**, полагающаяся рефереру **1 уровня** после любой входящей или исходящей операции, а с нодой у реферала 1.25%
- **money2** - сумма вознаграждения **0.5%** в месяц с баланса **wallet**, полагающаяся рефереру **2 уровня** после любой входящей или исходящей операции, а с нодой у реферала 0.625%
- **money3** - сумма вознаграждения **0.25%** в месяц с баланса **wallet**, полагающаяся рефереру **3 уровня** после любой входящей или исходящей операции, а с нодой у реферала 0.3125%
- **height** - высота транзакции с которой был перевод вознаграждения
- **date** - дата и время транзакции в UNIX формате с которой был перевод вознаграждения
- в конце цифры, это количество строк в данном ответе

## 06

### Запрос [type=referralwallets] - вывод списка кошельков рефералов у данного кошелька

Использует дополнительные необязательные параметры:

- **wallet** - кошелёк
- **ref** - номер кошелька по которому смотрятся все его рефералы всех уровней
- **ref1** - номер кошелька по которому смотрятся все его рефералы 1-ого уровня, если нет параметра ref
- **ref2** - номер кошелька по которому смотрятся все его рефералы 2-ого уровня, если нет параметра ref
- **ref3** - номер кошелька по которому смотрятся все его рефералы 3-его уровня, если нет параметра ref
- **height** - высота от которой смотрятся кошельки по последней транзакции
- **noda** - нода которую использовал кошелёк в последний раз для транзакции
- **date** - дата от которой смотрятся кошельки по последней транзакции в UNIX формате
- **dateto** - дата до которой смотрятся кошельки по последней транзакции в UNIX формате
- **order** - параметр сортировки вывода истории в прямом порядке от меньшей даты к большей - возможные параметры =asc - по возрастанию даты, =balanceasc по возрастанию баланса и убывания даты, =balancedesc по убыванию баланса и убывания даты, без параметров сортировка по дате в обратном направлении. Сортировка баланса без учёта начисления процентов
- **start** - с какой транзакции в выдаче начать отображать.
- **limit** - количество отображенных транзакций от параметра start.

По умолчанию сортировка идёт в обратном порядке от большей даты изменения к меньшей.

\* Баланс рефералов выводится без учёта процента от последней транзакции до текущего времени

\* В конце массива выводится число, означающее общее количество записей в истории реферальных начислений по данному запросу без разбивки по страницам. Сделано для того, чтобы знать сколько всего страниц с запросами.

**Пример GET запроса:** [http://\[IP ноды\]/egold.php?type=referralwallets&ref=10000000000000000000&height=2&start=0&limit=2&order=asc](http://[IP ноды]/egold.php?type=referralwallets&ref=10000000000000000000&height=2&start=0&limit=2&order=asc)

**Пример JSON ответа:**

```
[{"wallet": "942940567813543929", "ref1": "735587879887580955", "ref2": "10000000000000000000", "ref3": "0", "noda": "", "nodause": "91.106.2"}
```



```
03.179","balance":"478","date":"1590853829","height":"2"},{"wallet":"381636645604930508","ref1":"10000000000000000000","ref2":"0","ref3":"0","noda":"","nodaused":"5.181.110.217","balance":"100383290","date":"1590901712","height":"7"},62]
```

- **wallet** - кошелёк реферала
- **ref1** - реферер **1 уровня**, то есть кошелёк, который создал кошелёк **wallet**
- **ref2** - реферер **2 уровня**, то есть кошелёк, который создал кошелёк того кошелька, который создал кошелёк **wallet**
- **ref3** - реферер **3 уровня**
- **noda** - нода, которая принадлежит кошельку
- **nodaused** - нода которую использовал кошелёк в последний раз для транзакции
- **balance** - баланс кошелька
- **date** - дата и время последней транзакции в UNIX формате
- **height** - высота последней транзакции
- в конце цифры, это количество строк в данном ответе

## 07

### Запрос [type=referralresults] - вывод общего количества и объёма монет по уровню рефералов

Обязательно должен быть использован один из параметров:

- **ref** - номер кошелька по которому смотрятся все его рефералы всех уровней
- **ref1** - номер кошелька по которому смотрятся все его рефералы 1-ого уровня, если нет параметра **ref**
- **ref2** - номер кошелька по которому смотрятся все его рефералы 2-ого уровня, если нет параметра **ref**
- **ref3** - номер кошелька по которому смотрятся все его рефералы 3-его уровня, если нет параметра **ref**

\* Баланс рефералов выводится без учёта процента от последней транзакции до текущего времени

**Пример GETзапроса:** [http://\[IP ноды\]/egold.php?type=referralresults&ref=461311305111001414](http://[IP ноды]/egold.php?type=referralresults&ref=461311305111001414)

**Пример JSONответа:** {"count1":"1","balance1":"13969582","count2":"6","balance2":"413713","count3":"101","balance3":"300124234"}

- **count1** - количество рефералов 1 уровня данного кошелька, указанного в параметре запроса **ref**
- **balance1** - баланс рефералов 1 уровня данного кошелька, указанного в параметре запроса **ref**
- **count2** - количество рефералов 2 уровня данного кошелька, указанного в параметре запроса **ref**
- **balance2** - баланс рефералов 2 уровня данного кошелька, указанного в параметре запроса **ref**
- **count3** - количество рефералов 3 уровня данного кошелька, указанного в параметре запроса **ref**
- **balance3** - баланс рефералов 3 уровня данного кошелька, указанного в параметре запроса **ref**



08

Запрос [type=**height**] - получение высоты кошелька и возможности отправки транзакции. Служит для вспомогательного запроса перед транзакцией

Используется дополнительный параметр: wallet - кошелёк

Пример GETзапроса: [http://\[IP ноды\]/egold.php?type=height&wallet=158011521111537971](http://[IP ноды]/egold.php?type=height&wallet=158011521111537971)

Пример JSONответа: {"time":"1567159024","noda":"91.106.203.179","balance":99901,"height":"1","date":"1567158999"}

- **time** - время на ноде в UNIX формате
- **noda** - IP текущей ноды
- **balance** - баланс кошелька, указанного в параметре **wallet**
- **height** - высота кошелька, указанного в параметре **wallet**
- **date** - дата последней транзакции кошелька, указанного в параметре **wallet**

09

Запрос [type=**send**] - осуществление транзакции при которой может быть смена публичного и закрытого ключа, смена пароля для управления контактами и E-mail уведомлениями, зарегистрирован новый кошелёк

Использует дополнительные обязательные параметры:

- **wallet** - кошелёк с которого переводим средства
- **recipient** - кошелёк получатель на который переводим средства
- **money** - сумма перевода без учёта комиссии в 2 монеты
- **pin** - любое число до 18 цифр (для определения транзакции). По умолчанию должен быть равен 0
- **height** - высота данной транзакции по кошельку
- **signpub** - открытый ключ переводящего средства кошелька
- **sign** - подпись транзакции закрытым ключом переводящего средства кошелька формируется из строки: wallet + recipient + money + pin + height + noda + signpubreg + signreg + signpubnew + signnew. Где «+» это сложение строк. Signreg = '30' + sha\_dec(signpubreg) (для регистрации отправляется 3 монеты и высота нового кошелька 0), signpubnew= sha\_dec(signpubnew), signnew= wallet + height. sha\_dec - функция в javascript. Signpubreg, signreg, signpubnew и signnew могут быть пустыми значениями.

Использует дополнительные необязательные параметры:

- **signpubnew** - хэш для нового открытого ключа
- **signpubnew\_check** - открытый ключ для проверки корректности нового закрытого ключа в самой ноде. При использовании данного параметра транзакция не пройдёт, если закрытый ключ был неправильный. При этом, открытый ключ используется только для проверки его валидности и только на этой ноде и сразу же удаляется из памяти. Если данный параметр не передаётся, то и корректность закрытого ключа не будет проверяться
- **signnew** - подпись для проверки нового закрытого ключа
- **signpubreg** - открытый ключ для создания нового кошелька
- **signreg** - подпись для создания нового кошелька





- **password** - пароль для управления контактами и E-mail уведомлениями. С клиента получается 128-и символьный пароль. Который сохраняется в ноде как есть и в дальнейшем проверяется его 256-и символьный хэш. Подробнее в пункте 2 по запросу [type=wallet]

**Пример GETзапроса:** `http://[IP ноды]/egold.php?type=send&wallet=10000000000000000&recipient=126121479106151551&money=5&pin=0&height=2&signpub=xxx&sign=xxx`

**Пример JSONответа:** `{"time":"1592593054","noda":"91.106.203.179","date":"1592593000","send":"true","recipient":"G-7355-87879-8875-80955"}`

- **time** - время на ноде в UNIX формате
- **noda** - IP текущей ноды
- **date** - дата транзакции кошелька, указанного в параметре **wallet** в UNIX формате
- **send** - ответ ноды: **true** - значит транзакция прошла
- **signpubnew\_check** - ответ ноды: **true** - значит открытый ключ проверен, поэтому закрытый ключ корректный. Отображается только, если был получен параметр `signpubnew_check` с публичным ключом
- **recipient** - кошелек получателя

## 10 Запрос [type=synch] - синхронизация ноды

Не использует никаких дополнительных параметров и устанавливается в **CRONс ежеминутным** обращением. Рекомендуется в CRON устанавливать обращение через PHP запрос в виде `«/usr/local/bin/php ~/[путь до файла egold.php]/egold.php synch»` Где **synch** - это передаваемый параметр **type** для синхронизации, при этом PHP запрос должен быть от версии не ниже 7.1. Также можно использовать обращение вида `«http://[IP ноды]/egold.php?type=synch»`

## 11 Запрос [type=synchwallets] - обращение к синхронизируемой ноде

Использует дополнительный необязательный параметр только в виде POST запроса: **wallets** - массив кошельков по которым нужно отобразить данные ноды для синхронизации.

Сортировка идёт в прямом порядке от меньшей даты к большей и отображается до 100 кошельков с приоритетом на кошельки перечисленные в массиве.

**Пример POSTзапроса с параметром [wallets=10000000000000000]:** `http://[IP ноды]/egold.php?type=synchwallets`

**Пример JSONответа:** `{"time":"1567160232","noda":"91.106.203.179","synchwallets":{"10000000000000000":{"wallet":"10000000000000000","ref1":"310121260159274912","ref2":"","ref3":"","noda":"","nodause":"","balance":"3","height":"0","date":"1560612600","signpubnew":"","signnew":"","signpub":"","sign":"","413712101472131111":{"wallet":"413712101472131111","ref1":"126121479106151551","ref2":"461311305111001414","ref3":"151407813008139731","noda":"","nodause":"","balance":"3","height":"0","date":"1560917136","signpubnew":"","signnew":"","signpub":"","sign":""}}}}`

- **time** - время на ноде в UNIX формате
- **noda** - IP текущей ноды
- **synchwallets** - массив кошельков, где номер кошелька идёт сначала и потом его данные соответствующие запросу **type=wallet** с теми же типами данных

## 12 Запрос [**type=contacts**] - отображение контактов кошелька на ноде с которой он совершает транзакции

Для использования этого запроса сначала необходимо совершить транзакцию на ноде вместе с заданием пароля для управления настройками на этой ноде. Контакты будут храниться и отображаться только при запросе к этой ноде. Максимально можно хранить 100 контактов

Использует дополнительные обязательные параметры:

- **wallet** - кошелёк по которому смотрятся контакты
- **password** - пароль для доступа к данным кошелька, сохраненным на ноде сравнивается с присланным, используя функцию `gen_sha3([пароль на ноде],256)`. То есть параметр пароля **password** под хэшем SHAKE256 длиной 256 символов (1024 бита). С клиента получается 128-и символьный пароль. Который сохраняется в ноде как есть и в дальнейшем проверяется его 256-и символьный хэш. Подробнее в пункте 2 по запросу [**type=wallet**] и пункте 9 по запросу [**type=send**]

Использует дополнительные необязательные параметры:

- **contacts** - массив контактов передаваемых ноде только по средствам POST запроса. Если этот параметр не передать, тогда отображается список уже имеющихся контактов. Если передан пустым, то все контакты удаляются. На сервер передаются параметры **recipient** и **name**. Фактически, они могут быть любыми строками с символами букв английского языка и цифрами. Длина должна быть не более 255 символов. Параметры для отправки соответствуют параметрам в **JSON** ответе для отображения контактов. Пример **JSON** ниже.

Пример **JSON** ответа по сохранению и изменению контактов: `{"time":"1567160321","noda":"91.106.203.179","contact":"save"}`

Пример **JSON** ответа по удалению контактов (если отправлен запрос без контактов):

`{"time":"1567160493","noda":"91.106.203.179","contact":"del"}`

Пример **JSON** ответа по отображению контактов:

`[{"recipient":"20000000000000000000","name":"name2"}, {"recipient":"30000000000000000000","name":"name3"}]`

- **recipient** - номер кошелька, сохранённый в книге контактов
- **name** - название кошелька



## 13

### Запрос [type=email] - E-mail уведомления по транзакциям кошелька на ноде с которой он совершает транзакции

Для использования этого запроса сначала необходимо совершить транзакцию на ноде вместе с заданием пароля для управления настройками на этой ноде. Только ноды со своим прописанным в файле настроек доменом для почты могут отправлять E-mail уведомления. Об этом можно узнать при обращении к ноде с пустым запросом [http://\[IP ноды\]/egold.php](http://[IP ноды]/egold.php) при наличии заполненного параметра email\_domain

Использует дополнительные обязательные параметры:

- **wallet** - кошелёк с которого переводим средства
- **password** - пароль для доступа к данным кошелька, сохраненным на ноде сравнивается с присланным, используя функцию `gen_sha3([пароль на ноду],256)`. То есть параметр пароля **password** под хэшем SHAKE256 длиной 256 символов (1024 бита). С клиента получается 128-и символьный пароль. Который сохраняется в ноду как есть и в дальнейшем проверяется его 256-и символьный хэш. Подробнее в пункте 2 по запросу [type=wallet] и пункте 9 по запросу [type=send]

Использует дополнительные необязательные параметры:

- **email** - почта на которую будут отправляться уведомления. Если не будет этого параметра при присутствии почты на ноду, она будет удалена и все остальные параметры очищены.
- **up** - целое число указывающее на минимальный порог по сумме исходящих транзакций при котором будут отправлены уведомления
- **down** - целое число указывающее на минимальный порог по сумме входящих транзакций при котором будут отправлены уведомления

Пример JSONответа по сохранению почты и параметров: {"time":"1567160588","noda":"91.106.203.179","emailwallet":"save"}

Пример JSONответа по удалению почты и параметров: {"time":"1567160606","noda":"91.106.203.179","emailwallet":"del"}

- **time** - время на ноду в UNIX формате
- **noda** - IP текущей ноды
- **emailwallet** - ответ по сохранению контактов: **save** - контакты сохранены, **del** - удалены

## 14

### Запрос [type=balanceall] - общий баланс всех кошельков без учёта начисленных процентов

\* Баланс всех кошельков выводится без учёта процента от последней транзакции до текущего времени

Пример GETзапроса: [http://\[IP ноды\]/egold.php?type=balanceall](http://[IP ноды]/egold.php?type=balanceall)

Пример JSONответа: {"time":"1583326557","noda":"91.106.203.179","balanceall":"18446744074137928231"}

- **time** - время на ноду в UNIX формате
- **noda** - IP текущей ноды
- **balanceall** - баланс всех кошельков без начисленных процентов



## 15 Запрос [type=walletscount] - общее количество кошельков

Пример GET запроса: `http://[IP ноды]/egold.php?type=walletscount`

Пример JSON ответа: `{"time":"1583326557","noda":"91.106.203.179","walletscount":"234"}`

- **time** - время на ноде в UNIX формате
- **noda** - IP текущей ноды
- **walletscount** - количество всех кошельков

## 16

## 17 Пример совершения транзакции на PHP

```
<?php
include './egold_crypto/falcon.php'; //подключаем шифрование
function egold_send($params,$noda){
    $url = 'http://'.$noda.'/egold.php';
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, $url);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($ch, CURLOPT_POST, 1);
    curl_setopt($ch, CURLOPT_POSTFIELDS, $params);
    curl_setopt($ch, CURLOPT_TIMEOUT_MS, 15*1000);
    $json = curl_exec($ch);
    curl_close($ch);
    return json_decode($json, true);
}
```

```
$noda = '[IP ноды которую будем использовать]';
$wallet = '[кошелёк с которого отправляем только 18 цифр]';
$recipient = '[кошелёк на который отправляем только 18 цифр]';
$money = '[сумма]';
$pin = '[пин - любые цифры в количестве от 1 до 18 символом]';
$falcon_k = '[Закрытый ключ кошелька с которого отправляем]';
```



```
$params = array(//POST параметры для отправки
    'type' => 'height',
    'wallet' => $wallet
);
$height= egold_send($params,$noda);

if(isset($height['height']) && $height['height']>0){$height=$height['height']+1;}
else {print_r($params);print_r($height);exit();/*ошибка*/}

$falcon_p= Falcon\createPublicKey($falcon_k);//создание публичного ключа
$str_s= $wallet.$recipient.$money.$pin.$height.$noda;//строка которую подписываем
$falcon_s= Falcon\sign($falcon_k, $str_s);//подпись строки закрытым ключом

$params = array(//POST параметры для отправки
    'type' => 'send',
    'wallet' => $wallet,
    'recipient' => $recipient,
    'money' => $money,
    'pin' => $pin,
    'height' => $height,
    'signpub' => $falcon_p,
    'sign' => $falcon_s
);
$json_send= egold_send($params,$noda);

if(isset($json_send['send']) && $json_send['send']=='true'){echo 'true';/*монеты отправлены*/}
else {print_r($params);print_r($json_send);/*ошибка*/}
?>
```

## 18 Пример создания кошелька на PHP

```
<?php
include './egold_crypto/falcon.php';//подключаем шифрование

function egold_send($params,$noda){
    $url = 'http://'.$noda.'/egold.php';
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, $url); curl_setopt
    ($ch, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($ch, CURLOPT_POST, 1);
    curl_setopt($ch, CURLOPT_POSTFIELDS, $params);
    curl_setopt($ch, CURLOPT_TIMEOUT_MS, 15*1000);
```





```
$json = curl_exec ($ch);
curl_close ($ch);
return json_decode($json, true);
}

function bchexdec($hex){//длинные числа
    $dec = 0; $len = strlen($hex);
    for ($i = 1; $i <= $len; $i++)$dec = bcadd($dec, bcmul(strval(hexdec($hex[$i - 1])), bcpow('16', strval($len - $i))));
    return $dec;
}

function sha_dec($str){return substr(bchexdec(gen_sha3($str,19)),0,19);}//генерация хеша из 19 чисел

$node= '[IP ноды которую будем использовать]';
$wallet= '[кошелек с которого отправляем только 18 цифр]';
$recipient= '00';
$money= '3';
$pin= '0';
$gen_pass= '[случайная строка на 50 000 цифр для генерации нового кошелька]';
$falcon_k= '[Закрытый ключ кошелька с которого отправляем]';

$params = array();//POST параметры для отправки
    'type' => 'height',
    'wallet' => $wallet
);
$height= egold_send($params,$node);

if(isset($height['height']) && $height['height']>0){$height=$height['height']+1;}
else {print_r($params);print_r($height);exit();/*ошибка*/}
list($falcon_k_reg,$falcon_p_reg)= Falcon\createKeyPair(128,$gen_pass);//создание новой пары открытого и закрытого ключа
$str_s_reg='30'.sha_dec($falcon_p_reg);//строка для подписи закрытым ключом нового кошелька
$falcon_s_reg= Falcon\sign($falcon_k_reg, $str_s_reg);//подпись ключом нового кошелька
$falcon_p= Falcon\createPublicKey($falcon_k);//создание публичного ключа
$str_s= $wallet.$recipient.$money.$pin.$height.$node.$falcon_p_reg.$falcon_s_reg;//строка которую подписываем ключом уже имеющегося кошелька
$falcon_s= Falcon\sign($falcon_k, $str_s);//подпись строки закрытым ключом уже имеющегося кошелька

$params = array();//POST параметры для отправки
    'type' => 'send',
    'wallet' => $wallet,
    'recipient' => $recipient,
    'money' => $money,
    'pin' => $pin,
    'height' => $height,
    'signpubreg' => $falcon_p_reg,
```



```
'signreg' => $falcon_s_reg,
'signpub' => $falcon_p,
'sign' => $falcon_s
);

$json_send= egold_send($params,$noda);

if(isset($json_send['walletnew']) && strlen(preg_replace("/[^0-9]/i","", $json_send['walletnew']))==18){echo 'wallet_new=
'. $json_send['walletnew']. ' private_key= ' . $falcon_k_reg; /*кошелёк создан*/}
else {print_r($params);print_r($json_send);/*ошибка*/}
?>
```

## 19 Пример смены закрытого ключа на PHP

```
<?php
include './egold_crypto/falcon.php';//подключаем шифрование

function egold_send($params,$noda){
    $url = 'http://'. $noda. '/egold.php';
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, $url); curl_setopt
    ($ch, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($ch, CURLOPT_POST, 1);
    curl_setopt($ch, CURLOPT_POSTFIELDS, $params);
    curl_setopt($ch, CURLOPT_TIMEOUT_MS, 15*1000);
    $json = curl_exec ($ch);
    curl_close ($ch);
    return json_decode($json, true); function
}
bchexdec($hex){//длинные числа
    $dec = 0; $len = strlen($hex);
    for ($i = 1; $i <= $len; $i++)$dec = bcadd($dec, bcmul(strval(hexdec($hex[$i - 1])), bcpow('16', strval($len - $i))));
    return $dec;
}
function sha_dec($str){return substr(bchexdec(gen_sha3($str,19)),0,19);}//генерация хеша из 19 чисел

$noda= '[IP ноды которую будем использовать]';
$wallet= '[кошелёк с которого отправляем только 18 цифр]';
$recipient= '1';
$money= '1';
$pin= '0';
```



```
$gen_pass= '[случайная строка на 50 000 цифр для генерации нового кошелька]';
$falcon_k= '[Закрытый ключ кошелька с которого отправляем]';
$params = array(//POST параметры для отправки
    'type' => 'height',
    'wallet' => $wallet
);
$height= egold_send($params,$noda);

if(isset($height['height']) && $height['height']>=0){$height=$height['height']+1;}
else {print_r($params);print_r($height);exit();}

list($falcon_k_new,$falcon_p_new)= Falcon\createKeyPair(128,$gen_pass);//создание новой пары открытого и закрытого ключа
$signpubnew= sha_dec($falcon_p_new);//генерируем хэш нового закрытого ключа
$signnew= Falcon\sign($falcon_k_new, $wallet.$height);//генерируем подпись для нового закрытого ключа
$falcon_p= Falcon\createPublicKey($falcon_k);//создание публичного ключа
$str_s= $wallet.$recipient.$money.$pin.$height.$noda.$signpubnew.$signnew;//строка которую подписываем
$falcon_s= Falcon\sign($falcon_k, $str_s);//подпись строки закрытым ключом

$params = array(//POST параметры для отправки
    'type' => 'send',
    'wallet' => $wallet,
    'recipient' => $recipient,
    'money' => $money,
    'pin' => $pin,
    'height' => $height,
    'signpubnew' => $signpubnew,
    'signpubnew_check' => $falcon_p_new,
    'signnew' => $signnew,
    'signpub' => $falcon_p,
    'sign' => $falcon_s
);
$json_send= egold_send($params,$noda);

if(isset($json_send['send']) && $json_send['send']=='true'){echo 'private_key_new= '.$falcon_k_new;/*закрытый ключ изменён*/}
else {print_r($params);print_r($json_send);/*ошибка*/}
?>
```



## 20 Пример приёма средств на PHP

*\*Желательно установить на разных серверах сразу несколько нод и проверять поступление средств через 5 минут с момента поступления транзакции и если транзакция подтверждена сразу на всех нодах через 5 минут, то зачислять средства.*

*\*Если средства поступают на общий кошелёк, каждому пользователю нужно присваивать свой числовой номер до 18 цифр. И если входящая транзакция на общий кошелёк будет с пином, соответствующим числовому номеру пользователя для входящих транзакций, то зачисляем средства этому пользователю.*

*\*Настоятельно рекомендую сделать пользователю возможность смены пинкода, чтобы нельзя было точно определить, что средства зачисляются одному и тому же пользователю.*

*\*После приёма транзакции записываем номер кошелька и высоту откуда она пришла и в дальнейшем пропускаем такие транзакции и обрабатываем только новые.*

*\*Входящие транзакции можно проверять сразу в MySQL в таблице [префикс базы]\_history, так как база данных монеты eGOLD хранится в MySQL.*

```
<?php
function egold_send($params,$noda){
    $url = 'http://'. $noda. '/egold.php';
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, $url); curl_setopt
    ($ch, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($ch, CURLOPT_POST, 1);
    curl_setopt($ch, CURLOPT_POSTFIELDS, $params);
    curl_setopt($ch, CURLOPT_TIMEOUT_MS, 15*1000);
    $json = curl_exec ($ch);
    curl_close ($ch);
    return json_decode($json, true);
}

$noda= '[IP ноды которую будем использовать]';
$recipient= '[кошелёк на который поступают средства в виде 18 цифр]';
$pin= '[пин - любые цифры в количестве от 1 до 18 символом для определения кому зачисляем]';
$dateto= 5;//сколько минут должно пройти от поступления транзакции
$date= 24*60;//предел в минутах от поступления транзакции за который не смотрим

$params = array(//POST параметры для отправки
    'type' => 'history',
    'recipient' => $recipient,
    'pin' => ($pin?$pin:0),
```



```
'date' => time()-$date*60,  
'dateto' => time()-$dateto*60,  
'limit' => 100,  
'order' => 'DESC'  
);  
$json_history= egold_send($params,$noda);  
  
if(isset($json_history[0])){print_r($json_history);/*обрабатываем поступление средств*/}  
else {print_r($params);print_r($json_history);/*ошибка*/}  
?>
```