

Термины

- **Client (Клиент)** — front-end приложение, отвечающее за работу Ленты комментариев в браузере
- **Server (Сервер)** — сервер, обеспечивающий хранение и работу ленты комментирования
- **Subject** — предмет комментирования
- **Thread** — ветка обсуждения и комментирования
- **TargetNode** — DOM node для встраивания в нее отображения **Thread**
- **markerClass** — имя класса, которым маркируется **TargetNode** для встраивания Клиентом **Thread**
- **secretKey** — ключ передаваемый, сайту, используя который, сайт подписывает подлинность генерируемой информации

Общий принцип работы Клиента

Лента комментирования **Subject'a** маркируется посредством тега с определенным классом, передача дополнительных параметров происходит посредством data-* атрибутов.

```
<div class="markerClass" data-subject-id="news 2018-01-29 Bitcoin hard fork." data-subject-hash="#a2fef8e3a5adf6=" data-subject-title="" data-status="(enabled|disabled|hide)" data-lang="en">
```

На одной загружаемой странице одновременно не может быть дублей по data-subject-id.

При первом запуске Клиента отправляется на сервер **subject-id** из атрибута **data-subject-id**, в ответ Клиент получает от сервера **thread_id** идентифицирующий поток комментирования, если таковой был уже создан, Клиент также получает начальное состояние **Thread** в первом ответе.

Сервер либо находит в своей базе **thread_id** по **subject-id** и возвращает его, либо создает и возвращает стартовое состояние.

Достоверность информации, полученной из **TargetNode** проверяется посредством расчета цифровой подписи в data-subject-hash на стороне сайта и сервера.

Если Клиент получает какое-либо начальное состояние, то оно сразу же отображается, если не получает, то это сценарий ошибки

- Ошибка Хеша. В этом случае будет отображена ошибка.
- Ошибка Сети. В этом случае будет отправлен запрос на сервер повторно через 10 секунд.
- Некорректный ответ сервера. В этом случае будет отправлен запрос на сервер повторно через 30 секунд. Каждый последующий запрос будет отправляться с удвоением интервала.

Клиент подписывается по WebSocket (библиотека Socket.IO) на обновления своего **Thread** и обновляет его только по сообщению с сервера.

Клиент далее определяет статус пользователя. Если есть авторизационный токен, то пробует подтвердить его через Сервер. В случае отсутствия токена или подтверждения предлагает авторизацию, а после получения авторизационного токена подписывает им все сообщения и действия от пользователя.

Возможные запросы Клиента:

- Запросить ленту комментариев. На первом этапе мы считаем что лента комментариев такой длины, что можно ее всю загрузить за один раз, одним запросом на сервер
- Только от имени авторизованных пользователей:
 - Отправить комментарий. Комментарий содержит:
 - [Опционально] идентификатор родительского комментария
 - plain text, со ссылками
 - идентификатор автора
 - прикрепленные изображенияИзображения отправляются предварительно и хранятся во временной папке, связанной с пользователем в момент публикации изображения из временной папки переносятся в комментарий и отображаются под ним. Изображения теряются с сервера если комментарий не был опубликован в течении часа. Максимальное количество изображений не более 5ти
- Доступный функционал подсистемы:
 - отправить изображение на сервер
 - запросить список загруженных изображений
 - удалить изображение из временной папки пользователя
- Отправить голос за комментарий «+» или «-», отмена голоса
- Пожаловаться на комментарий с сообщением

В первой версии **не** поддерживаются, но предполагается к реализации в последующих версиях:

- редактирование и удаление собственных комментариев
- отображение проголосовавших
- сортировка отличная от хронологической
- фильтрация (поиск) по хештегу

Поведение Клиента не зависящее от Сервера:

- Древовидное отображение комментариев
- Древовидное иерархическое с сохранением вложенных состояний сворачивание веток комментариев
- Динамическое привязывание формы ответа к любому комментарию со сменой режима отображения в случае если комментарий идет в самый верхний уровень
- Динамический пересчет времени публикации с шагом в одну минуту

Требования к функционалу Клиента

- Локализация интерфейса через конфигурационный файл
- Микровзаимодействия отображающая внутренние состояния
 - прелоадеры
 - анимация смены состояний