

# etherpad protocol notes

nik gaffney

2021-11-28

## Contents

<b>etherpad &amp; the easysync protocol</b>	<b>2</b>
<b>websockets &amp; socket.io</b>	<b>2</b>
<b>websockets in emacs</b>	<b>2</b>
<b>protocol, probes &amp; partials</b>	<b>2</b>
overview .....	3
comment plugin .....	4
example messages .....	4
changesets .....	5
attributes, colours, authors, etc .....	6
CLIENT_VARS .....	6
<b>various tools &amp; accessories</b>	<b>10</b>
<b>testing &amp; tracing</b>	<b>10</b>

## etherpad & the easysync protocol

- [easysync protocol description & notes](#)
- [How etherpad-lite, a real time collaborative editor, works?](#)

possibly relevant parts of the etherpad code

- <https://github.com/ether/etherpad-lite/blob/develop/src/node/handler/PadMessageHandler.js>
- <https://github.com/payload/ethersync/blob/master/src/ethersync.coffee>
- code for a [cli-client](#) and the [javascript client](#)

## websockets & socket.io

etherpad uses socket.io for realtime communication between server and clients. socket.io uses ws and wss as transport protocols, but not directly compatible. appears to work as expected by adding headers and/or prefixes to json data sent via wss ( “Socket.IO is NOT a WebSocket implementation” according to the [socket.io docs](#))

## websockets in emacs

see <https://blog.abrochard.com/websockets.html> and [emacs-websocket](#) for details

## protocol, probes & partials

browser client sends url with pad name to server (e.g. <https://etherpad.wikimedia.org/p/test> ) establishes session (sid) and receives token (in cookie data). updates, changes & pad metadata are sent via wss connection.

e.g. <wss://etherpad.wikimedia.org/socket.io/?EIO=3&transport=websocket&sid=Ap47gBZD98dHcW38AoqY>

# overview

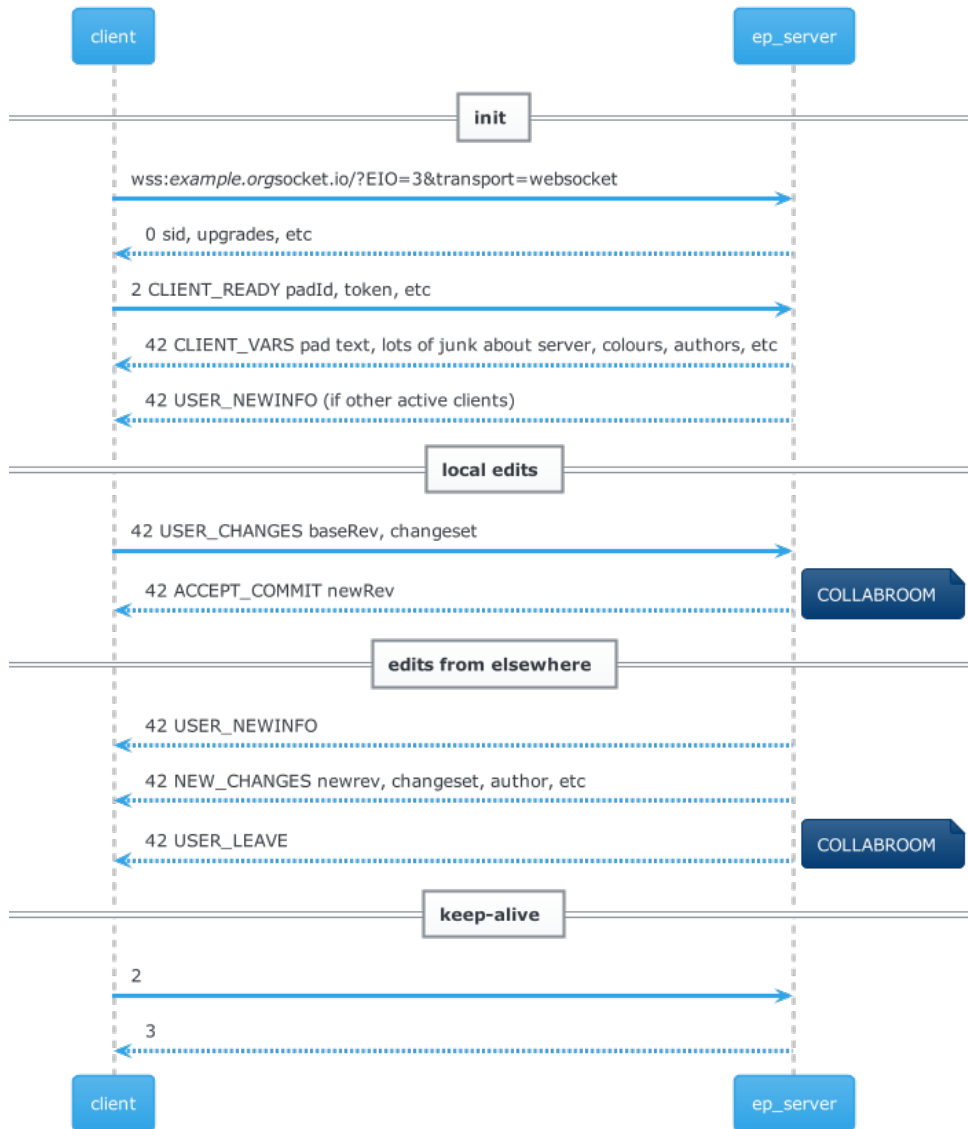


Figure 1: overview of etherpad/easysync protocol

## comment plugin

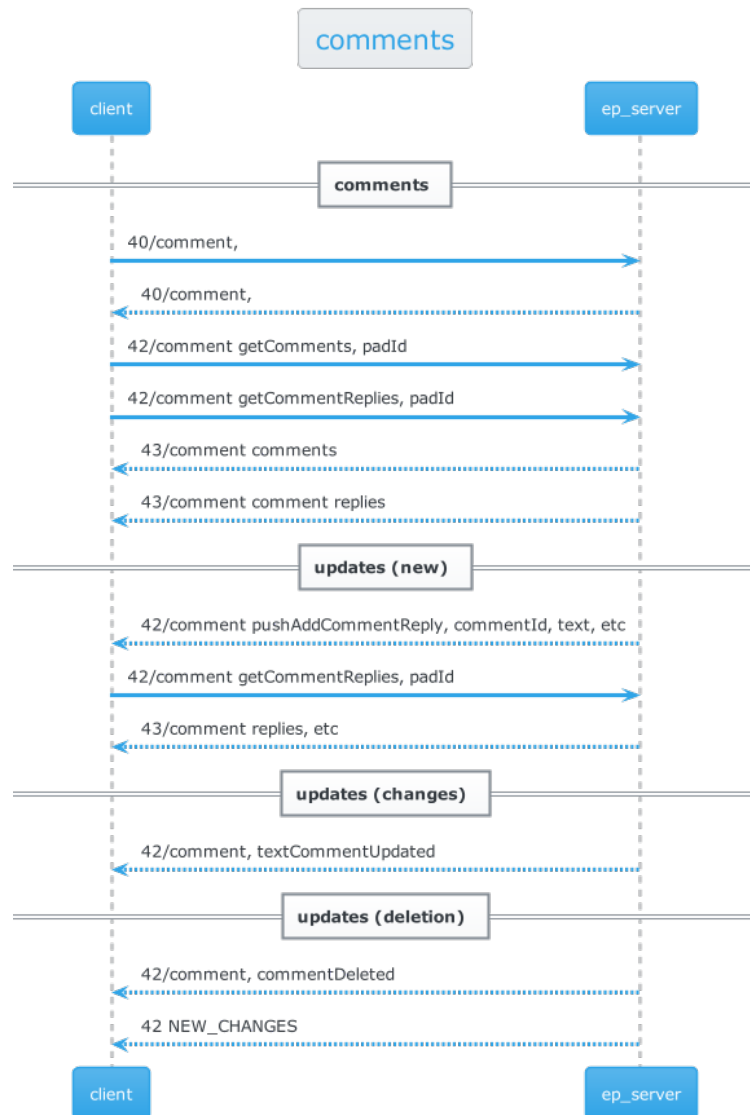


Figure 2: comments

## example messages

### init/request

40/comment,

42/comment,0["getComments",{"padId":"test"}]

42/comment,1["getCommentReplies",{"padId":"test"}]

43/comment,0[{"comments":{"c-4U2BW8J2Lp0r68ZL":{"author":"a.0iRJZx7ji0AxVNMP","name":"zzkt","text":

43/comment,1[{"replies":{}}]

### updates (new)

42/comment,["pushAddCommentReply","c-reply-vMSgWSY4bFhaCCLR",{"commentId":"c-4U2BW8J2Lp0r68ZL","text":

42/comment,2["getCommentReplies",{"padId":"test"}]

43/comment,2[{"replies":{"c-reply-vMSgWSY4bFhaCCLR":{"commentId":"c-4U2BW8J2Lp0r68ZL","text":"no

## updates (changes)

```
42/comment,["textCommentUpdated","c-reply-vMSgWSY4bFhaCCLR","not yet"]
```

## updates (deletion)

```
42/comment,["commentDeleted","c-4U2BW8J2Lp0r68ZL"]
```

```
42["message",{ "type":"COLLABROOM", "data":{"type":"NEW_CHANGES", "newRev":234, "changeset":"Z:e>0=7
```

## changesets

via <https://github.com/ether/etherpad-lite/> ...and [easysync notes](#)

An "attribute" is a (key,value) pair such as (author,abc123456) or (bold,true). Sometimes an attribute is treated as an instruction to add that attribute, in which case an empty value means to remove it. So (bold,) removes the "bold" attribute. Attributes are interned and given numeric IDs, so the number "6" could represent "(bold,true)", for example. This mapping is stored in an attribute "pool" which may be shared by multiple changesets.

Entries in the pool must be unique, so that attributes can be compared by their IDs. Attribute names cannot contain commas.

A changeset looks something like the following:

```
Z:5g>1|5=2p=v*4*5+1$x
```

With the corresponding pool containing these entries: ... 4 -> (author,1059348573) 5 -> (bold,true) ... This changeset, together with the pool, represents inserting a bold letter "x" into the middle of a line. The string consists of:

- a letter Z (the "magic character" and format version identifier)
- a series of opcodes (punctuation) and numeric values in base 36 (the alphanumerics)
- a dollar sign (\$)
- a string of characters used by insertion operations (the "char bank")

If we separate out the operations and convert the numbers to base 10, we get:

```
Z :196 >1 |5=97 =31 *4 *5 +1 $"x"
```

Here are descriptions of the operations, where capital letters are variables:

" :N" : Source text has length N (must be first op)  
">N" : Final text is N (positive) characters longer than source text (must be second op)  
"<N" : Final text is N (positive) characters shorter than source text (must be second op)  
">0" : Final text is same length as source text  
"+N" : Insert N characters from the bank, none of them newlines  
"-N" : Skip over (delete) N characters from the source text, none of them newlines  
"=N" : Keep N characters from the source text, none of them newlines  
"|L+N" : Insert N characters from the source text, containing L newlines. The last character inserted MUST be a newline, but not the (new) document's final newline.  
"|L-N" : Delete N characters from the source text, containing L newlines. The last character inserted MUST be a newline, but not the (old) document's final newline.  
"|L=N" : Keep N characters from the source text, containing L newlines. The last character kept MUST be a newline, and the final newline of the document is allowed.  
"\*I" : Apply attribute I from the pool to the following +, =, |+, or |= command. In other words, any number of \* ops can come before a +, =, or | but not between a | and the corresponding + or =.  
If +, text is inserted having this attribute. If =, text is kept but with the attribute applied as an attribute addition or removal.  
Consecutive attributes must be sorted lexically by (key,value) with key and value taken as strings. It's illegal to have duplicate keys for (key,value) pairs that apply to the same text. It's illegal to have an empty value for a key in the case of an insertion (+), the pair should just be omitted.

Characters from the source text that aren't accounted for are assumed to be kept with the same attributes.

### Additional Constraints:

- Consecutive +, -, and = ops of the same type that could be combined are not allowed. Whether combination is possible depends on the attributes of the ops and whether each is multiline or not. For example, two multiline deletions can never be consecutive, nor can any insertion come after a non-multiline insertion with the same attributes.
- "No-op" ops are not allowed, such as deleting 0 characters. However, attribute applications that don't have any effect are allowed.
- Characters at the end of the source text cannot be explicitly kept with no changes; if the change doesn't affect the last N characters, those "keep" ops must be left off.
- In any consecutive sequence of insertions (+) and deletions (-) with no keeps (=), the deletions must come before the insertions.
- The document text before and after will always end with a newline. This policy avoids a lot of special-casing of the end of the document. If a final newline is always added when importing text and removed when exporting text, then the changeset representation can be used to process text files that may or may not have a final newline.

### Attribution string:

An "attribution string" is a series of inserts with no deletions or keeps. For example, "**\*3+8|1+5**" describes the attributes of a string of length 13, where the first 8 chars have attribute 3 and the next 5 chars have no attributes, with the last of these 5 chars being a newline. Constraints apply similar to those affecting changesets, but the restriction about the final newline of the new document being added doesn't apply.

Attributes in an attribution string cannot be empty, like "(bold,)", they should instead be absent.

### attributes, colours, authors, etc

the "apool"

```
"apool":{"numToAttrib":{"0":["author","a.touCZaixjPgKDSiN"]},"nextNum":1}
```

author ids, names & colour mapping

### CLIENT\_VARS

```
42["message",{"type":"CLIENT_VARS","data":{"... [etc]
```

most directly useful

- pad name - [1]["data"]["padId"] (and also [1]["data"]["collab\_client\_vars"]["padId"])
- revision - [1]["data"]["collab\_client\_vars"]["rev"]
- pad text - [1]["data"]["collab\_client\_vars"]["initialAttributedText"]["text"]
- text attributes (as changset)- [1]["data"]["collab\_client\_vars"]["initialAttributedText"]["attribs"]

authors

- author list - [1]["data"]["collab\_client\_vars"]["historicalAuthorData"]
- e.g. "a.ltSpokLpHyziPkDn": {"name": "someone", "colorId": 46}}

colo[u]rs

- array of hex values - [1]["data"]["colorPalette"]
- map authors -> colour - e.g. [1]["data"]["collab\_client\_vars"]["historicalAuthorData"]["a.TcyaduN3

plugins available

- listed in [1]["data"]["plugins"]
- e.g. [1]["data"]["plugins"]["plugins"]["ep\_comments\_page"]
  - ["data"]["plugins"]["plugins"]["ep\_etherpad-lite"]["package"]["description"]

```
- ["data"]["plugins"]["plugins"]["ep_etherpad-lite"]["package"]["version"]
```

example/reduced

```
[
  "message",
  {
    "type": "CLIENT_VARS",
    "data": {
      "skinName": "colibris",
      "skinVariants": "super-dark-toolbar super-dark-background dark-editor",
      "randomVersionString": "0ec6de15",
      "accountPrivs": {
        "maxRevisions": 100
      },
      "automaticReconnectionTimeout": 5,
      "initialRevisionList": [],
      "initialOptions": {
        "guestPolicy": "deny"
      },
      "savedRevisions": [],
      "collab_client_vars": {
        "initialAttributedText": {
          "text": "ethereal\n",
          "attrs": "*0+8|1+1"
        },
        "clientId": "127.0.0.1",
        "padId": "test2",
        "historicalAuthorData": {
          "a.ltSpoKLpHyziPkDn": {
            "name": null,
            "colorId": 46
          },
          "a.touCZaixjPgKDSiN": {
            "name": null,
            "colorId": 7
          },
          "a.TcyaduN34UmzJIxa": {
            "name": null,
            "colorId": 31
          }
        }
      },
      "apool": {
        "numToAttrib": {
          "0": [
            "author",
            "a.touCZaixjPgKDSiN"
          ]
        }
      },
      "nextNum": 1
    },
    "rev": 174,
    "time": 1607568522484
  },
  "colorPalette": [
    "#ffc7c7",
    "#fff1c7",
    "#e3ffc7",
    "#c7ffd5",
    "#c7ffff",
    "#c7d5ff",
  ]
]
```

```
"#e3c7ff",
"#ffc7f1",
"#ffa8a8",
"#ffe699",
"#cfff9e",
"#99ffb3",
"#a3ffff",
"#99b3ff",
"#cc99ff",
"#ff99e5",
"#e7b1b1",
"#e9dcAf",
"#cde9af",
"#bfedcc",
"#b1e7e7",
"#c3cdee",
"#d2b8ea",
"#eec3e6",
"#e9cece",
"#e7e0ca",
"#d3e5c7",
"#bce1c5",
"#c1e2e2",
"#c1c9e2",
"#cfc1e2",
"#e0bdd9",
"#baded3",
"#a0f8eb",
"#b1e7e0",
"#c3c8e4",
"#cec5e2",
"#b1d5e7",
"#cda8f0",
"#f0f0a8",
"#f2f2a6",
"#f5a8eb",
"#c5f9a9",
"#ececbb",
"#e7c4bc",
"#daf0b2",
"#b0a0fd",
"#bce2e7",
"#cce2bb",
"#ec9afe",
"#edabbd",
"#aeaea",
"#c4e7b1",
"#d722bb",
"#f3a5e7",
"#ffa8a8",
"#d8c0c5",
"#eaaedd",
"#adc6eb",
"#bedad1",
"#dee9af",
"#e9afc2",
"#f8d2a0",
"#b3b3e6"
],
"clientIp": "127.0.0.1",
```



```
"userIsGuest": true,
"userColor": 7,
"padId": "test2",
"padOptions": {
  "noColors": false,
  "showControls": true,
  "showChat": false,
  "showLineNumbers": false,
  "useMonospaceFont": false,
  "userName": true,
  "userColor": true,
  "alwaysShowChat": false,
  "chatAndUsers": false,
  "ShowComments": true,
  "lang": "en-gb",
  "rtl": false
},
"padShortcutEnabled": {
  "altF9": true,
  "altC": true,
  "cmdShift2": true,
  "delete": true,
  "return": true,
  "esc": true,
  "cmdS": true,
  "tab": true,
  "cmdZ": true,
  "cmdY": true,
  "cmdI": true,
  "cmdB": true,
  "cmdU": true,
  "cmd5": true,
  "cmdShiftL": true,
  "cmdShiftN": true,
  "cmdShift1": true,
  "cmdShiftC": true,
  "cmdH": true,
  "ctrlHome": true,
  "pageUp": true,
  "pageDown": true
},
"initialTitle": "Pad: test2"
}
}
]
```

## various tools & accessories

- Firefox/Chrome/Safari → network/ws/messages/console/log etc
- `git clone https://github.com/guyzmo/PyEtherpadLite`
- `wscat`
- `netcat`

## testing & tracing

```
(defun ethertest-loop ()
  (interactive)
  (with-current-buffer *etherpad-buffer*
    (let ((server-url "wss://example.org/socket.io/?EIO=3&transport=websocket")
          (pad "test"))
      (text-mode)
      (etherpad-mode)
      (ethersync-current-socket
       (websocket-open subtest
                       :on-message #'ethersync-parse-wsframe
                       :on-error (lambda (_websocket type err)
                                  (message "ws error: %s %s" type err))
                       :on-close (lambda (_websocket)
                                   (message "websocket closed")))))
      (let* ((*subtest-socket* (ethersync-current-socket)))
        (message "protocols: %s" (websocket-negotiated-protocols *subtest-socket*))
        (message "extensions: %s" (websocket-negotiated-extensions *subtest-socket*))
        (message "cookies? %s" url-cookie-storage)
        ;; init & keep alive
        (ethersync-heartbeat-start)
        (sleep-for 1)
        ;; request data
        (wss-send (ethersync--request-client-ready pad))
        ;; etcn
        ))))
```

```
❏ wscat -c "wss://example.org/socket.io/?EIO=3&transport=websocket"
```

```
Connected (press CTRL+C to quit)
```

```
< 0{"sid":"6_TVij3sJug26KFLAAGc","upgrades":[],"pingInterval":25000,"pingTimeout":5000}
```

```
< 40
```

```
>
```

```
[
  "message",
  {
    "type": "COLLABROOM",
    "data": {
      "type": "NEW_CHANGES",
      "newRev": 969,
      "changeset": "Z:2r>2*0=1=5*1|1+1*1*2*3*4*5+1|1=1*6=1|1=4*7=1|1=4*8=1|1=a*9=1|1=5*0=1|1=f*0",
      "apool": {
        "numToAttrib": {
          "0": [
            "start",
            "1"
          ],
          "1": [
```

```
    "author",
    "a.TA0tv04870h304Up"
  ],
  "2": [
    "insertorder",
    "first"
  ],
  "3": [
    "list",
    "number1"
  ],
  "4": [
    "lmlr",
    "1"
  ],
  "5": [
    "start",
    "2"
  ],
  "6": [
    "start",
    "3"
  ],
  "7": [
    "start",
    "4"
  ],
  "8": [
    "start",
    "5"
  ],
  "9": [
    "start",
    "6"
  ]
],
"attribToNum": {
  "start,1": 0,
  "author,a.TA0tv04870h304Up": 1,
  "insertorder,first": 2,
  "list,number1": 3,
  "lmlr,1": 4,
  "start,2": 5,
  "start,3": 6,
  "start,4": 7,
  "start,5": 8,
  "start,6": 9
},
"nextNum": 10
},
"author": "a.TA0tv04870h304Up",
"currentTime": 1638017098574,
"timeDelta": 8923
}
]
```