



THE ENGINE MOBILE REST API

VERSION 1.2.2

Julie Montoya 2026

This document is probably out-of-date. A more up-to-date version may be downloaded from our website <https://www.parentshield.co.uk/>

Contents

Important Concepts.....	3
Status Codes.....	3
JSON.....	3
Common properties.....	4
Writing Boolean properties.....	4
Times and Dates.....	5
Number Formats.....	5
Multiple SIMs.....	5
Non-JSON responses.....	5
Endpoints.....	6
GET /API/.....	6
GET /API/homenumbers/.....	7
GET /API/homenumbers/\$id.....	8
PUT /API/homenumbers/\$id/\$property.....	9
GET /API/sims/.....	10
GET /API/sims/\$iccid.....	11
PUT /API/sims/\$iccid/\$property.....	12
GET /API/calls/.....	13
GET /API/calls/\$iccid/.....	14
GET /API/calls/in/\$iccid.....	14
GET /API/calls/out/\$iccid.....	14
GET /API/sms/.....	15
GET /API/sms/in/.....	16
GET /API/sms/out/.....	16
GET /API/mp3/\$call_id.....	17
GET /API/wav/\$call_id.....	17
GET /API/allowlist/.....	17
GET /API/blocklist/.....	17
GET PUT DELETE /API/allowlist/\$iccid/.....	18
GET PUT DELETE /API/blocklist/\$iccid/.....	18
GET DELETE /API/allowlist/\$iccid/\$number.....	20
GET DELETE /API/blocklist/\$iccid/\$number.....	20

The Engine Mobile REST API can be accessed at

[https://\\$EMAIL:\\$PASSWORD@api.engine-mobile.co.uk/API/](https://$EMAIL:$PASSWORD@api.engine-mobile.co.uk/API/)

where \$EMAIL and \$PASSWORD are your usual login credentials for the parent portal. Any URLs given within the response will always be relative to the base address

<https://api.engine-mobile.co.uk/> .

The API would ordinarily be accessed under program control, and the exact details of how this is done depends on the programming language in which you are working. You can access it “manually” for testing using cURL at the bash prompt.

Important Concepts

The API provides a number of “endpoints”, each of which performs one specific function. An endpoint may also provide explicit links to other endpoints, in the form of a list of URLs.

Status Codes

Successful API requests will give the status code **200** “OK”, and will usually return a JSON response body (the exceptions being the download URLs for MP3 or WAV files of recorded conversations). But note that just because the HTTP status is 200, does not mean by itself that everything is perfect! You will need to parse the response to make sure it is really what you expected.

If the username or password are incorrect, status code **401** “Unauthorized” will be returned.

If you try to access a non-existent endpoint that would return anything other than JSON, status code **404** “Not found” will be returned.

If you try to use an inappropriate method, status code **405** “Method not allowed” will be returned.

If you try to write to an endpoint which is subject to a timeout before the end of the timeout period, status code **429** “Too many requests” will be returned.

JSON

Most endpoints output JSON data. This is a well-defined standard for data serialisation, based on JavaScript object literal construction and supported by most modern programming languages.

Example JSON output:

```
{"summer":true,"urls":["/API/homenumbers/","/API/sims/","/API/calls/","/API/sms/","/API/allowlist/","/API/blocklist/"],"unix_time":1619621296,"ok":true,"date":"2021-04-28 15:48:16","method":"GET","browser":"curl/7.64.0","referrer":"","ip_addr":"81.128.160.130"}
```

For the sake of readability, examples in this guide have been formatted with line breaks and indentation as the following example:

```
{
  "ok" : true,
  "date" : "2021-04-28 15:48:16",
  "unix_time" : 1619621296,
  "referrer" : "",
  "urls" : [
    "/API/homenumbers/",
```

```

    "/API/sims/",
    "/API/calls/",
    "/API/sms/",
    "/API/allowlist/",
    "/API/blocklist/"
  ],
  "browser" : "curl/7.64.0",
  "method" : "GET",
  "summer" : true,
  "ip_addr" : "81.128.160.130"
}

```

Common properties

Although the actual content will necessarily depend on the endpoint, every JSON response from the API will include at least one property: **ok**. This is a Boolean (i.e., `true` or `false`) value indicating success or failure. If `false`, there will be an additional property **error** containing a message describing what went wrong.

Any “writeable” properties will be listed in an array property named **writeable**. Writeable parameters would normally be changed using a PUT request to a specific endpoint. If you cannot reliably make PUT requests, you can send a POST request with the data to write in a field called `data`, or a GET request with a `data` field in the query string. Therefore, the following are functionally equivalent, and will have the effect of setting the phone’s “off” time to one minute before midnight:

```
$ curl -XPUT -d'23:59'
https://api.engine-mobile.co.uk/API/sims/8944538531000580007/time_
off
```

```
$ curl -d'data=23:59'
https://api.engine-mobile.co.uk/API/sims/8944538531000580007/time_
off
```

```
$ curl
https://api.engine-mobile.co.uk/API/sims/8944538531000580007/time_
off?data=23:59
```

Writing Boolean properties

If a Boolean property is set to the text “false”, the number 0 or an empty string (i.e., a pair of speech marks with nothing enclosed between them), this will be treated as `false`. Anything else will be treated as `true`.

Some endpoints require a DELETE request; for example, to remove numbers from the “allowed” (in ultra-secure mode) or “blocked” lists. If you cannot reliably make DELETE requests, you can

send a POST request with a field called `forget` containing anything but a figure 0 or an empty string; or a GET request with a non-empty `forget` value in the query string. Therefore, the following are functionally equivalent, and will have the effect of removing the number 01283707058 from the blocklist:

```
$ curl -XDELETE
https://api.engine-mobile.co.uk/API/blocklist/8944538525006706764/
01283707058
```

```
$ curl -d'forget=1'
https://api.engine-mobile.co.uk/API/blocklist/8944538525006706764/
01283707058
```

```
$ curl
https://api.engine-mobile.co.uk/API/blocklist/8944538525006706764/
01283707058?forget=1
```

Times and Dates

The canonical time and date format used by the API looks like this:

2021-04-28 13:37:05

This has the year, month and day separated by minus signs; a single space; and then the time in VCR notation, with seconds, separated by colons.

The backend will attempt to make sense of imperfect date/time entries with e.g. missing leading zeros or no seconds.

Number Formats

The Engine Mobile REST API can accept numbers specified in national (e.g. 01283707050) or international (e.g. 441283707050) formats. There should be no spaces or punctuation in a number.

Multiple SIMs

Even if you have only one SIM card on your account, the API will require you to specify its ICCID in order to keep behaviour consistent on accounts with more than one SIM.

Non-JSON responses

Not all endpoints return data in JSON format; for example, the links to download recorded conversations return MP3 or WAV data.

Endpoints

There follows a full description of each endpoint, with example JSON output and an explanation of what each field represents.

GET /API/

=> an endpoint intended mainly as a simple test, which returns a JSON object with the following properties:

```
{
  "ok" : true,
  "date" : "2021-04-28 13:32:06",
  "unix_time" : 1619613126,
  "referrer" : "",
  "urls" : [
    "/API/homenumbers/",
    "/API/sims/",
    "/API/calls/",
    "/API/sms/",
    "/API/allowlist/",
    "/API/blocklist/"
  ],
  "browser" : "curl/7.64.0",
  "method" : "GET",
  "summer" : true,
  "ip_addr" : "81.128.160.130"
}
```

browser – your user agent string. (This is something you can configure; our server will ignore it.)

date – the current date and time in canonical format. The time zone is UTC in Winter or UTC+1 in Summer.

error – if **ok** is **false**, this will contain a short message explaining what went wrong. This property will be absent if **ok** is **true**.

ip_addr – originating IP address of request.

method – usually GET or POST, but this endpoint also supports PUT and DELETE for testing purposes (nothing will be changed).

ok – should always be true.

putdata – if the PUT method is used, this will contain the request data. Otherwise this will be absent.

referrer – this probably will be empty.

summer (= **true** if daylight savings time is in operation).

unix_time (= the number of seconds since midnight, 1 Jan 1970).

urls – an array of other possible endpoints that you can call.

If you manage to get this output, it means all is working well! (You can actually enter this URL in a web browser, supply your e-mail and password when prompted, and you should see the raw JSON.) The **method** and **putdata** properties in particular may be useful for making sure your requests are properly formed; you can also check any custom user agent string you may have configured.

GET /API/homenumbers/

=> returns information about your home numbers, including the dates they were last changed and whether or not calls will be recorded.

Properties:

```
{
  "homenumbers": [
    {
      "changed": "2021-03-02",
      "id": "n1",
      "info_url": "/API/homenumbers/n1/",
      "number": "441283707058",
      "record": true
    },
    {
      "changed": "0000-00-00",
      "id": "n2",
      "info_url": "/API/homenumbers/n2/",
      "number": "443301121180",
      "record": true
    }
  ],
  "ok": true
}
```

homenumbers – an array with two members (or three, on accounts with three home numbers). Each member has the following properties:

changed – the date of the most recent change to the number.

id – an identifier for the number.

info_url – an endpoint specific to just that number.

number – the home number itself, represented in International format.

record – a boolean value indicating whether or not calls to that home number will be recorded.

ok – this should be `true`.

error – an error message. This will not be present if **ok** is `true`.

GET /API/homenumbers/\$id

=> returns information about an individual home number. (\$id needs to be n1 or n2; some tariffs also support a third home number n3.)

```
{
  "changed": "2021-03-02",
  "number": "441283707058",
  "ok": true,
  "record": true,
  "writeable": [
    "number",
    "record"
  ]
}
```

changed – the date the number was last changed. (Home numbers can ordinarily be changed only once per month, to prevent abuse.)

number – the home number itself. (Writeable)

record – a Boolean value indicating whether or not calls to or from this number are to be recorded. (Writeable)

ok – this should always be `true`.

writeable – an array containing the names of the writeable attributes.

error – an error message. This will not be present if **ok** is `true`.

PUT /API/homenumbers/\$id/\$property

=> updates an individual home number or its recording status.

This should be used with a HTTP PUT request, with the new value in the request body.

```
{
  "changed": "2020-06-17",
  "number": "441543504166",
  "ok": true,
  "record": true,
  "updated": 1
}
```

changed – the current date (if the update was successful), or else the date the home number was last changed.

number – the new home number.

ok – this should be `true`.

record – the new recording status.

updated – this will be `1` if the number or recording status was changed, or `0` if no change was made (either because the number was changed too recently, or because the old and new numbers were the same).

error – an error message. This will not be present if **ok** is `true`.

GET /API/sims/

=> returns brief information about SIMs associated with your account.

```
{
  "ok": true,
  "sims": [
    {
      "iccid": "8944501301210822323",
      "info_url": "/API/sims/8944501301210822323/",
      "label": "Alice",
      "number": "447397900641"
    },
    {
      "iccid": "8944500909204631590",
      "info_url": "/API/sims/8944500909204631590/",
      "label": "Bob",
      "number": "447412233610"
    }
  ]
}
```

This has the following properties:

ok – this should be `true`.

sims – an array, each element of which contains the following properties:

`iccid` – the ICCID of the SIM card.

`info_url` – a link to an endpoint with further information relating to the SIM.

`label` – a name associated with the SIM; usually the child's name.

`number` – the telephone number associated with the SIM.

error – an error message (this will not be present if **ok** is `true`).

More detailed information about a SIM can be obtained from the endpoint given in its **info_url** property.

GET /API/sims/\$iccid

=> returns full information about the SIM with a given ICCID.

```
{
  "always_on_clid": false,
  "block_anon": true,
  "data": true,
  "data_budgeting": true,
  "data_remain": 133675652,
  "iccid": "8944500909204631590",
  "label": "Bob",
  "last_used": "2021-04-27 12:37:46",
  "msgs_remain": 500,
  "not_school_hours": false,
  "number": "447412233610",
  "ok": true,
  "paranoia": false,
  "record_calls": true,
  "record_m2p": false,
  "record_sms": true,
  "secs_remain": 30000,
  "tariff": "Safe Stage 2",
  "time_off": "17:00",
  "time_on": "12:30",
  "timed": false,
  "writeable": [
    "paranoia",
    "data_budgeting",
    "time_off",
    "time_on",
    "record_sms",
    "record_m2p",
    "record_calls",
    "timed",
    "not_school_hours",
    "block_anon",
    "always_on_clid",
    "data"
  ]
}
```

always_on_clid – whether or not CLID is always sent despite phone settings. (Writeable)

block_anon – whether or not anonymous call blocking is active. (Writeable)

data – whether or not data is active. (Writeable)

data_budgeting – whether or not daily data budgeting (only available on all-network roaming SIMs) is active. (Writeable)

data_remain – bytes of data remaining.

iccid – the ICCID of the SIM.

label – SIM label / child's name.

last_used – date and time phone was last used for a call, text or data connection.

msgs_remain – number of messages remaining.

next_data_change – date and time when **data** setting can next be changed. (Only on single-network SIMs, and only present at times when data setting cannot be changed).

not_school_hours – whether or not phone is blocked 09:00-15:00. (Writeable)

ok – usually true.

number – in international format.

paranoia – whether or not ultra-secure mode is active. (Writeable)

record_calls – whether or not voice calls will be recorded. (Writeable)

record_m2p – whether or not to forward machine-to-person messages. (Writeable)

record_sms – whether or not SMS messages will be recorded. (Writeable)

secs_remain – amount of time remaining in seconds.

tariff – tariff name.

time_off – in VCR format, between 00:00 and 23:59. (Writeable)

time_on – as above. (Writeable)

timed – whether or not timed mode is active. (Writeable)

error – an error message (only present if **ok** is false).

PUT /API/sims/\$iccid/\$property

=> sets a writeable property for a particular SIM. The request should use the PUT method, with the new value supplied in the request body. The response will be a much-abridged version of the response from the parent endpoint, including only the changed property.

```
$ curl -XPUT -d'1'
```

```
https://api.engine-mobile.co.uk/API/sims/8944500909204631590/timed
```

```
{
  "ok": true,
  "timed": true,
  "rows": 1,
  "updated": 1
}
```

ok – usually **true**.

\$property – the new value of the property.

rows – number of rows matched by database query.

updated – this will be **1** if the value was changed, or **0** otherwise.

error – an error message, if **ok** is **false**.

The **time_on** and **time_off** properties should be a time of day, with hours (00-23) and minutes (00-59) separated by a colon. Missing leading zeros are tolerated (e.g. “6:5” will be treated as “06:05”). If there is no sensible way to parse the value as a time of day, then an error will be reported:

```
$ curl -XPUT -d'99:99'
https://api.engine-mobile.co.uk/API/sims/8944500909204631590/time_
off
{
  "ok":false,
  "error":"Bad value"
}
```

If the **time_on** setting is after the **time_off** setting, **error** will contain **Warning: time_off is before time_on**. In this state, the phone will become **unusable at any time** (except to call home numbers) while timed mode is active. This setting is allowed in case it is useful.

The **record_m2p** property, if **true**, allows machine-to-person text messages to be sent directly to the phone rather than held in the Portal for review.

Important note for data property:

On both the single-network and roaming SIMs, data may continue to be available after the **data** property has been set to **false**. This is because data is allocated in fixed-sized packets, and a packet already allocated remains available until it has been completely used up. (Obligatory plumbing analogy: You can still flush a toilet, even after the stop valve has been turned off, using the water that was already in the cistern.)

On the single-network SIMs only, when the **data** property is changed, there is a delay while it propagates around the network. A successful PUT to a `.../data` endpoint will return an additional **next_data_change** property. Attempting to change **data** before the specified date and time will return error 429.

GET /API/calls/

=> returns information about calls.

The top of the “calls” hierarchy simply returns a list of URLs for specific information about individual SIMs:

```
{
  "ok": true,
  "urls": [
    "/API/calls/8944500909204631590/",
    "/API/calls/8944501301210822323/"
  ]
}
```

ok – should always be **true**.

urls – an array of other possible endpoints that you can call.

error – an error message, if **ok** is **false**.

GET /API/calls/\$iccid/

=> Returns information about calls made using a given SIM. Ordinarily, this will be limited to the five most recent calls. However, if you specify (either by appending in the query string, or as data in a POST request) **from**=YYYY-mm-dd+HH:ii:ss and/or **to**=YYYY-mm-dd+HH:ii:ss, you can restrict this to a date and time range. (Note the space between date and hours needs to be URL-encoded as a + sign. Take care not to URL-encode this twice as %2b, which will decode to an actual plus sign!) If only a date is specified and the time is absent, this will be treated as 00:00:00 for **from** or 23:59:59 for **to**. You can also specify a date range in the form of Unix timestamps, with parameters **from_unix**= and/or **to_unix**=.

GET /API/calls/in/\$iccid

GET /API/calls/out/\$iccid

=> These endpoints return information about only calls in one direction (incoming or outgoing) with a given SIM. Otherwise they behave exactly as above. Without a time/date range, the default limit will be five calls in the chosen direction (i.e., not just whichever of the five most recent were in that direction).

Sample output (here showing just two calls):

```
{
  "calls": [
    {
      "billsec": 30,
      "call_id": "9fbfe268-0e31-4719-a3d7-591582484e86",
      "destination": "447778429726",
      "direction": "O",
      "finish_date": "2021-04-27 12:37:46",
      "finish_time": 1619523466,
      "mp3_url": "/API/mp3/9fbfe268-0e31-4719-a3d7-591582484e86",
      "recorded": true,
      "source": "447412233610",
      "start_date": "2021-04-27 12:37:37",
      "start_time": 1619523457,
      "wav_size": 44,
      "wav_url": "/API/wav/9fbfe268-0e31-4719-a3d7-591582484e86"
    },
    {
      "billsec": 30,
      "call_id": "b2590363-4fa4-4f78-8ff1-844aa1c2e95d1",
      "destination": "447412233610",
      "direction": "I",
      "finish_date": "2021-03-17 09:30:59",
      "finish_time": 1615973459,
      "mp3_url": "/API/mp3/b2590363-4fa4-4f78-8ff1-844aa1c2e95d1",
      "recorded": true,
      "source": "447778429726",
      "start_date": "2021-03-17 09:30:44",
      "start_time": 1615973444,
      "wav_size": 199404,
      "wav_url": "/API/wav/b2590363-4fa4-4f78-8ff1-844aa1c2e95d1"
    }
  ],
  "ok": true
}
```

ok – this should be `true`.

calls – an array with one element for each call.

error – an error message, if **ok** is false

Each entry within the **calls** array has the following properties:

`call_id` – an internal identifier, which is used to form download URLs.

`direction` – "I" for incoming, "O" for outgoing.

`source` – originating number.

`destination` – the destination number.

`start_date` – in usual, YYYY-mm-dd HH:ii:ss format.

`start_time` – as a Unix timestamp.

`finish_date` – in usual, YYYY-mm-dd HH:ii:ss format.

`finish_time` – as a Unix timestamp.

`billsec` – the billable duration of the call, in seconds.

`recorded` – `true` if recording is available for download.

`wav_url` – URL to download recording in WAV format (if available; may be absent).

`wav_size` – size of WAV file in bytes (if available; may be absent).

`mp3_url` – URL to download recording in MP3 format (if available; may be absent).

GET /API/sms/

=> returns information about SMS messages.

GET /API/sms/in/

GET /API/sms/out/

=> returns information about only SMS messages in one direction (incoming or outgoing).

The SMS endpoints work in almost exactly the same way as the calls endpoints, particularly with regard to date ranges. The main difference is that the message content is short enough to be returned directly in the HTTP response, so there are no download links.

Sample output (here showing just two messages):

```
{
  "messages": [
    {
      "blocked": true,
      "content": "Testing School Block",
      "destination": "447412233610",
      "direction": "I",
      "has_bad_words": false,
      "id": 1996588,
      "reason": "In School hours",
      "sent_date": "2021-04-27 12:23:51",
      "sent_time": 1619522631,
      "source": "447778429726"
    },
    {
      "blocked": false,
      "content": "Are texts going through? ",
      "destination": "447412233610",
      "direction": "I",
      "has_bad_words": false,
      "id": 2563694,
      "sent_date": "2021-04-27 12:25:39",
      "sent_time": 1619522739,
      "source": "447778429726"
    }
  ],
  "ok": true
}
```

The fields are as follows:

ok – this should be `true`.

messages – an array

error – an error message, if **ok** is false

Each entry within the **messages** array has the following properties:

id – an internal identifier.

direction – "I" for incoming, "O" for outgoing.

source – originating number.

destination – the destination number.

sent_date – in usual, YYYY-mm-dd HH:ii:ss format.

sent_time – as a Unix timestamp.

content – the actual message content.

has_bad_words – will be **true** if the message content triggered a word filter.

blocked – will be **true** if the message was blocked.

reason – if **blocked** is **true**, this will contain the reason why the message was blocked.
Otherwise, this field will be absent.

GET /API/mp3/call_id

GET /API/wav/call_id

=> download links for recorded calls. If you specify a nonexistent **call_id**, or one which does not belong to you, HTTP error 404 will be returned.

GET /API/allowlist/

GET /API/blocklist/

=> returns information about allow / block lists

These endpoints work in the same manner and will be described together. If the SIM is in ultra-secure mode (referred to as “paranoia” internally), then numbers other than home numbers must be on an allowlist to be contactable. If ultra-secure mode is turned off, the phone can call numbers which are not on a blocklist. The two lists are stored independently from one another, except that only one will be used depending whether or not ultra-secure mode is in use. Changes to the inactive list will still be honoured. If you have multiple SIMs, each one has its own ultra-secure mode setting and its own allow/block lists.

Sample JSON from **/API/allowlist/** :

```
{
  "ok": true,
  "urls": [
    "/API/allowlist/8944500909204631590/",
    "/API/allowlist/8944501301210822323/"
  ]
}
```

Sample JSON from **/API/blocklist/** :

```
{
  "ok": true,
  "urls": [
    "/API/blocklist/8944500909204631590/",
    "/API/blocklist/8944501301210822323/"
  ]
}
```

The fields are:

ok – usually true.

urls – an array of URLs for information relating to the allow/blocklist for each SIM. (Even if you have only one SIM, this will still be an array.)

error – if ok is false, an error message.

GET|PUT|DELETE /API/allowlist/\$iccid/

GET|PUT|DELETE /API/blocklist/\$iccid/

=> returns numbers on allow/block list. If the PUT request method is used, **adds** a number to the allow or block list. If the DELETE method is used, **removes all** numbers from the list.

```
{
  "block_anon": false,
  "numbers": [
    "01283707058",
    "03301221180"
  ],
  "ok": true,
  "paranoia": false
}
```

The fields are:

ok – usually true.

block_anon – a Boolean indicating whether anonymous numbers are blocked. (This can be written using a PUT request to **/API/sims/\$iccid/block_anon** ; see above.)

paranoia – a Boolean indicating whether ultra-secure mode is active. (This can be written using a PUT request to **/API/sims/\$iccid/paranoia** ; see above.)

numbers – an array containing the list of numbers. Numbers on the allowlist and blocklist are currently presented in National form. Numbers being added will automatically be converted from International form if necessary.

error – if **ok** is false, an error message indicating what went wrong.

A number can be added to the allowlist or blocklist using a PUT request (or a specially-crafted GET or POST request):

```
$ curl -XPUT -d'01283707050'
```

```
https://api.engine-mobile.co.uk/API/blocklist/8944500909204631590/
```

```
{
  "block_anon": false,
  "numbers": [
    "01283707058",
    "01283707050",
    "03301221180"
  ],
  "ok": true,
  "paranoia": false
}
```

If the request is successful, the response will include the number that has just been added to the list in the numbers array. If the number is invalid, or is a duplicate of a number which is already on the list, the response will contain **ok=false** and an error message:

```
$ curl -XPUT -d'666'  
https://api.engine-mobile.co.uk/API/blocklist/8944500909204631590/  
{  
  "block_anon": false,  
  "error": "Bad number",  
  "numbers": [  
    "01283707058",  
    "03301221180"  
  ],  
  "ok": false,  
  "paranoia": false  
}
```

```
$ curl -XPUT -d'03301221180'  
https://api.engine-mobile.co.uk/API/blocklist/8944500909204631590/  
{  
  "block_anon": false,  
  "error": "Number already on blocklist",  
  "numbers": [  
    "01283707058",  
    "03301221180"  
  ],  
  "ok": false,  
  "paranoia": false  
}
```

GET|DELETE /API/allowlist/\$iccid/\$number

GET|DELETE /API/blocklist/\$iccid/\$number

=> checks whether or not a number is on the allow/blocklist for a SIM. If the DELETE method is used, the number will be **removed** from the list. If the number is not found on the list, HTTP error 404 will be returned. If the number is found on the list, the list of numbers will be returned as usual.

```
{
  "ok":true,
  "block_anon":false,
  "paranoia":false,
  "numbers":[
    "01283707058",
    "03301221180"
  ]
}
```

The fields in the response are:

ok – normally true.

block_anon – a Boolean indicating whether anonymous numbers are blocked.

paranoia – a Boolean indicating whether ultra-secure mode is active.

numbers – an array with the list of numbers.