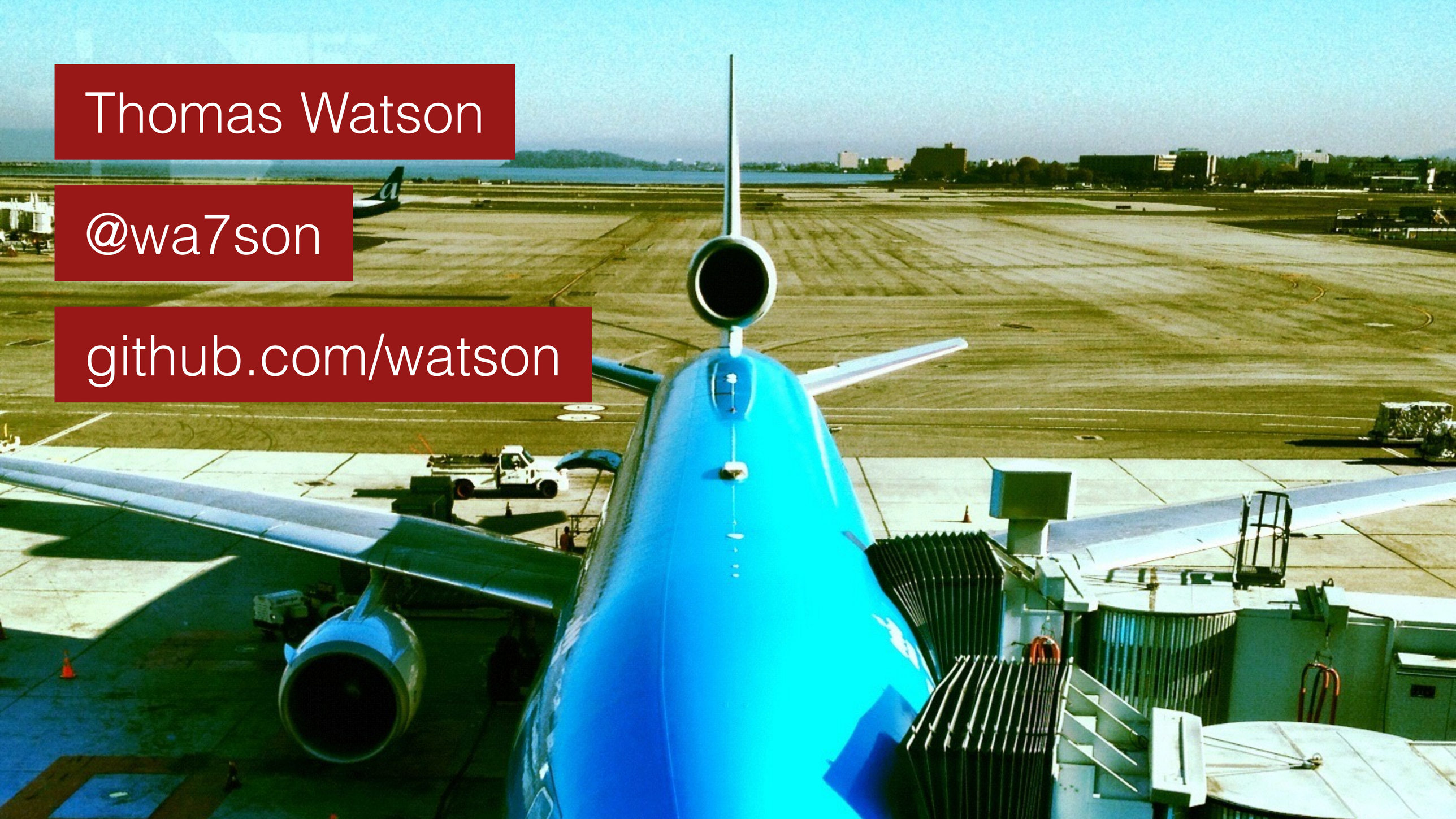
Thomas Watson
@wa7son
github.com/watson

Build your own printer in JavaScript

… no hardware required!

AirPlay

AirPlay

Bonjour / Zeroconf

AirPlay

Bonjour / Zeroconf

Multicast DNS

AirPlay

Bonjour / Zeroconf

Multicast DNS

DNS Service Discovery (DNS-SD)

inova-ontrack
Inova Solutions OnTrack Display Protocol
Protocol description: Proprietary
Defined TXT keys: None

idcws
Intermec Device Configuration Web Services
Thaddeus Ternes <thaddeus.ternes at intermec.com>
Protocol description: Proprietary web service for configuring embedded devices
Defined TXT keys: version=<x.xx>

ipbroadcaster
IP Broadcaster
10base-t Interactive <support at 10base-t.com>
Defined TXT keys: None

ipp
IPP (Internet Printing Protocol)
Carl-Uno Manros <manros at cp10.es.xerox.com>
Defined TXT keys: See BonjourPrinting.pdf.
Flagship Protocol: printer

ipspeaker
IP Speaker Control Protocol
Dan Mahn <dan.mahn at digidescorp.com>
Protocol description: Proprietary
Defined TXT keys: None

irelay
iRelay application discovery service
Marc Diamante <pgmp at pgmpsolutions.com>
Protocol description: Proprietary
Defined TXT keys: Proprietary

```
POST / HTTP/1.1
Content-Length: 635
Content-Type: application/ipp
Host: watson-2.local:3000
User-Agent: CUPS/2.0.0 (Darwin 14.5.0; x86_64) IPP/2.0
Expect: 100-continue
```

Printer operations

- Print Job
- Print URI
- Validate Job
- Create Job
- Get Printer Attributes
- Get Jobs
- Pause Printer
- Resume Printer
- Purge Jobs

Job operations

- Send Document
- Send URI
- Cancel Job
- Get Job Attributes
- Hold Job
- Release Job
- Restart Job

Printer operations

- Print Job 0x02
- Print URI 0x03
- Validate Job 0x04
- Create Job 0x05
- Get Printer Attributes 0x0b
- Get Jobs 0x0a
- Pause Printer 0x10
- Resume Printer 0x11
- Purge Jobs 0x12

Job operations

- Send Document 0x06
- Send URI 0x07
- Cancel Job 0x08
- Get Job Attributes 0x09
- Hold Job 0x0c
- Release Job 0x0d
- Restart Job 0x0e

```
'0101'                  // version (1.1)
'000b'                  // operation id (get printer attributes)
'00000001'              // request id (1)
'01'                    // delimiter tag (operation attributes)
  '44'                  // value tag (keyword)
    '0006'              // name length
    '737472696e67'      // name
    '0003'              // value length
    '666f6f'            // value
  '22'                  // value tag (boolean)
    '0004'              // name length
    '626f6f6c'          // name
    '0001'              // value length
    '01'                // value
  ...
'03'                    // end of attributes tag
...                     // PostScript document...
```

```
'0101'                  // version (1.1)
'000b'                  // operation id (get printer attributes)
'00000001'              // request id (1)
'01'                    // delimiter tag (operation attributes)
  '44'                  // value tag (keyword)
    '0006'              // name length
    '737472696e67'      // name
    '0003'              // value length
    '666f6f'            // value
  '22'                  // value tag (boolean)
    '0004'              // name length
    '626f6f6c'          // name
    '0001'              // value length
    '01'                // value
  ...
'03'                    // end of attributes tag
...                     // PostScript document...
```

```
'0101'                  // version (1.1)
'000b'                  // operation id (get printer attributes)
'00000001'              // request id (1)
'01'                    // delimiter tag (operation attributes)
  '44'                  // value tag (keyword)
    '0006'              // name length
    '737472696e67'      // name
    '0003'              // value length
    '666f6f'            // value
  '22'                  // value tag (boolean)
    '0004'              // name length
    '626f6f6c'          // name
    '0001'              // value length
    '01'                // value
  ...
'03'                    // end of attributes tag
...                     // PostScript document...
```

```
'0101'                  // version (1.1)
'000b'                  // operation id (get printer attributes)
'00000001'              // request id (1)
'01'                    // delimiter tag (operation attributes)
  '44'                  // value tag (keyword)
    '0006'              // name length
    '737472696e67'      // name
    '0003'              // value length
    '666f6f'            // value
  '22'                  // value tag (boolean)
    '0004'              // name length
    '626f6f6c'          // name
    '0001'              // value length
    '01'                // value
  ...
'03'                    // end of attributes tag
...                     // PostScript document...
```

```
'0101'                  // version (1.1)
'000b'                  // operation id (get printer attributes)
'00000001'              // request id (1)
'01'                    // delimiter tag (operation attributes)
  '44'                  // value tag (keyword)
    '0006'              // name length
    '737472696e67'      // name
    '0003'              // value length
    '666f6f'            // value
  '22'                  // value tag (boolean)
    '0004'              // name length
    '626f6f6c'          // name
    '0001'              // value length
    '01'                // value
  ...
'03'                    // end of attributes tag
...                     // PostScript document...
```

```
'0101'                  // version (1.1)
'000b'                  // operation id (get printer attributes)
'00000001'              // request id (1)
'01'                    // delimiter tag (operation attributes)
  '44'                  // value tag (keyword)
    '0006'              // name length
    '737472696e67'      // name
    '0003'              // value length
    '666f6f'            // value
  '22'                  // value tag (boolean)
    '0004'              // name length
    '626f6f6c'          // name
    '0001'              // value length
    '01'                // value
  ...
'03'                    // end of attributes tag
...                     // PostScript document...
```

```
'0101'                  // version (1.1)
'000b'                  // operation id (get printer attributes)
'00000001'              // request id (1)
'01'                    // delimiter tag (operation attributes)
  '44'                  // value tag (keyword)
    '0006'              // name length
    '737472696e67'      // name
    '0003'              // value length
    '666f6f'            // value
  '22'                  // value tag (boolean)
    '0004'              // name length
    '626f6f6c'          // name
    '0001'              // value length
    '01'                // value
  ...
'03'                    // end of attributes tag
...                     // PostScript document...
```

```
'0101'                // version (1.1)
'000b'                // operation id (get printer attributes)
'00000001'            // request id (1)
'01'                  // delimiter tag (operation attributes)
  '44'                // value tag (keyword)
    '0006'            // name length
    '737472696e67'    // name
    '0003'            // value length
    '666f6f'          // value
  '22'                // value tag (boolean)
    '0004'            // name length
    '626f6f6c'        // name
    '0001'            // value length
    '01'              // value
  ...
'03'                  // end of attributes tag
...                   // PostScript document...
```

```
'0101'                // version (1.1)
'000b'                // operation id (get printer attributes)
'00000001'            // request id (1)
'01'                  // delimiter tag (operation attributes)
  '44'                // value tag (keyword)
    '0006'            // name length
    '737472696e67'    // name
    '0003'            // value length
    '666f6f'          // value
  '22'                // value tag (boolean)
    '0004'            // name length
    '626f6f6c'        // name
    '0001'            // value length
    '01'              // value
  ...
'03'                  // end of attributes tag
...                   // PostScript document...
```

```
'0101'                  // version (1.1)
'000b'                  // operation id (get printer attributes)
'00000001'              // request id (1)
'01'                    // delimiter tag (operation attributes)
  '44'                  // value tag (keyword)
    '0006'              // name length
    '737472696e67'      // name
    '0003'              // value length
    '666f6f'            // value
  '22'                  // value tag (boolean)
    '0004'              // name length
    '626f6f6c'          // name
    '0001'              // value length
    '01'                // value
  ...
'03'                    // end of attributes tag
...                     // PostScript document...
```

```
'0101'                  // version (1.1)
'000b'                  // operation id (get printer attributes)
'00000001'              // request id (1)
'01'                    // delimiter tag (operation attributes)
  '44'                  // value tag (keyword)
    '0006'              // name length
    '737472696e67'      // name
    '0003'              // value length
    '666f6f'            // value
  '22'                  // value tag (boolean)
    '0004'              // name length
    '626f6f6c'          // name
    '0001'              // value length
    '01'                // value
  ...
'03'                    // end of attributes tag
...                     // PostScript document...
```

github.com /
watson /
ipp-encoder

```
var ipp = require('ipp-encoder')
var C = ipp.CONSTANTS

// decode binary buffer from IPP client
var decoded = ipp.request.decode(buf)

// ...handle request...

// prepare response
var response = {
  statusCode: C.SUCCESSFUL_OK, // set `operationId` instead if encoding a request
  requestId: decoded.requestId,
  groups: [
    { tag: C.OPERATION_ATTRIBUTES_TAG, attributes: [
      { tag: C.CHARSET, name: 'attributes-charset', value: 'utf-8' },
      { tag: C.NATURAL_LANG, name: 'attributes-natural-language', value: 'en-us' },
      { tag: C.TEXT_WITH_LANG, name: 'status-message', value: { lang: 'en-us', value: 'successful-ok' }
    ] },
    { tag: C.JOB_ATTRIBUTES_TAG, attributes: [
      { tag: C.INTEGER, name: 'job-id', value: 147 },
      { tag: C.NAME_WITH_LANG, name: 'job-name', value: { lang: 'en-us', value: 'Foobar' } }
    ] }
  ]
}

// encode response to binary buffer
ipp.response.encode(response) // <Buffer 01 01 00 00 ... >
```

```
var ipp = require('ipp-encoder')
var C = ipp.CONSTANTS

// decode binary buffer from IPP client
var decoded = ipp.request.decode(buf)

// ...handle request...

// prepare response
var response = {
  statusCode: C.SUCCESSFUL_OK, // set `operationId` instead if encoding a request
  requestId: decoded.requestId,
  groups: [
    { tag: C.OPERATION_ATTRIBUTES_TAG, attributes: [
      { tag: C.CHARSET, name: 'attributes-charset', value: 'utf-8' },
      { tag: C.NATURAL_LANG, name: 'attributes-natural-language', value: 'en-us' },
      { tag: C.TEXT_WITH_LANG, name: 'status-message', value: { lang: 'en-us', value: 'successful-ok' }
    ] },
    { tag: C.JOB_ATTRIBUTES_TAG, attributes: [
      { tag: C.INTEGER, name: 'job-id', value: 147 },
      { tag: C.NAME_WITH_LANG, name: 'job-name', value: { lang: 'en-us', value: 'Foobar' } }
    ] }
  ]
}

// encode response to binary buffer
ipp.response.encode(response) // <Buffer 01 01 00 00 ... >
```

```
var ipp = require('ipp-encoder')
var C = ipp.CONSTANTS

// decode binary buffer from IPP client
var decoded = ipp.request.decode(buf)

// ...handle request...

// prepare response
var response = {
  statusCode: C.SUCCESSFUL_OK, // set `operationId` instead if encoding a request
  requestId: decoded.requestId,
  groups: [
    { tag: C.OPERATION_ATTRIBUTES_TAG, attributes: [
      { tag: C.CHARSET, name: 'attributes-charset', value: 'utf-8' },
      { tag: C.NATURAL_LANG, name: 'attributes-natural-language', value: 'en-us' },
      { tag: C.TEXT_WITH_LANG, name: 'status-message', value: { lang: 'en-us', value: 'successful-ok' }
    ] },
    { tag: C.JOB_ATTRIBUTES_TAG, attributes: [
      { tag: C.INTEGER, name: 'job-id', value: 147 },
      { tag: C.NAME_WITH_LANG, name: 'job-name', value: { lang: 'en-us', value: 'Foobar' } }
    ] }
  ]
}

// encode response to binary buffer
ipp.response.encode(response) // <Buffer 01 01 00 00 ... >
```

```
var ipp = require('ipp-encoder')
var C = ipp.CONSTANTS

// decode binary buffer from IPP client
var decoded = ipp.request.decode(buf)

// ...handle request...

// prepare response
var response = {
  statusCode: C.SUCCESSFUL_OK, // set `operationId` instead if encoding a request
  requestId: decoded.requestId,
  groups: [
    { tag: C.OPERATION_ATTRIBUTES_TAG, attributes: [
      { tag: C.CHARSET, name: 'attributes-charset', value: 'utf-8' },
      { tag: C.NATURAL_LANG, name: 'attributes-natural-language', value: 'en-us' },
      { tag: C.TEXT_WITH_LANG, name: 'status-message', value: { lang: 'en-us', value: 'successful-ok' }
    ] },
    { tag: C.JOB_ATTRIBUTES_TAG, attributes: [
      { tag: C.INTEGER, name: 'job-id', value: 147 },
      { tag: C.NAME_WITH_LANG, name: 'job-name', value: { lang: 'en-us', value: 'Foobar' } }
    ] }
  ]
}

// encode response to binary buffer
ipp.response.encode(response) // <Buffer 01 01 00 00 ... >
```

```
var ipp = require('ipp-encoder')
var C = ipp.CONSTANTS

// decode binary buffer from IPP client
var decoded = ipp.request.decode(buf)

// ...handle request...

// prepare response
var response = {
  statusCode: C.SUCCESSFUL_OK, // set `operationId` instead if encoding a request
  requestId: decoded.requestId,
  groups: [
    { tag: C.OPERATION_ATTRIBUTES_TAG, attributes: [
      { tag: C.CHARSET, name: 'attributes-charset', value: 'utf-8' },
      { tag: C.NATURAL_LANG, name: 'attributes-natural-language', value: 'en-us' },
      { tag: C.TEXT_WITH_LANG, name: 'status-message', value: { lang: 'en-us', value: 'successful-ok' }
    ] },
    { tag: C.JOB_ATTRIBUTES_TAG, attributes: [
      { tag: C.INTEGER, name: 'job-id', value: 147 },
      { tag: C.NAME_WITH_LANG, name: 'job-name', value: { lang: 'en-us', value: 'Foobar' } }
    ] }
  ]
}

// encode response to binary buffer
ipp.response.encode(response) // <Buffer 01 01 00 00 ... >
```

```
var ipp = require('ipp-encoder')
var C = ipp.CONSTANTS

// decode binary buffer from IPP client
var decoded = ipp.request.decode(buf)

// ...handle request...

// prepare response
var response = {
  statusCode: C.SUCCESSFUL_OK, // set `operationId` instead if encoding a request
  requestId: decoded.requestId,
  groups: [
    { tag: C.OPERATION_ATTRIBUTES_TAG, attributes: [
      { tag: C.CHARSET, name: 'attributes-charset', value: 'utf-8' },
      { tag: C.NATURAL_LANG, name: 'attributes-natural-language', value: 'en-us' },
      { tag: C.TEXT_WITH_LANG, name: 'status-message', value: { lang: 'en-us', value: 'successful-ok' }
    ] },
    { tag: C.JOB_ATTRIBUTES_TAG, attributes: [
      { tag: C.INTEGER, name: 'job-id', value: 147 },
      { tag: C.NAME_WITH_LANG, name: 'job-name', value: { lang: 'en-us', value: 'Foobar' } }
    ] }
  ]
}

// encode response to binary buffer
ipp.response.encode(response) // <Buffer 01 01 00 00 ... >
```

```
var ipp = require('ipp-encoder')
var C = ipp.CONSTANTS

// decode binary buffer from IPP client
var decoded = ipp.request.decode(buf)

// ...handle request...

// prepare response
var response = {
  statusCode: C.SUCCESSFUL_OK, // set `operationId` instead if encoding a request
  requestId: decoded.requestId,
  groups: [
    { tag: C.OPERATION_ATTRIBUTES_TAG, attributes: [
      { tag: C.CHARSET, name: 'attributes-charset', value: 'utf-8' },
      { tag: C.NATURAL_LANG, name: 'attributes-natural-language', value: 'en-us' },
      { tag: C.TEXT_WITH_LANG, name: 'status-message', value: { lang: 'en-us', value: 'successful-ok' }
    ] },
    { tag: C.JOB_ATTRIBUTES_TAG, attributes: [
      { tag: C.INTEGER, name: 'job-id', value: 147 },
      { tag: C.NAME_WITH_LANG, name: 'job-name', value: { lang: 'en-us', value: 'Foobar' } }
    ] }
  ]
}

// encode response to binary buffer
ipp.response.encode(response) // <Buffer 01 01 00 00 ... >
```

```
var ipp = require('ipp-encoder')
var C = ipp.CONSTANTS

// decode binary buffer from IPP client
var decoded = ipp.request.decode(buf)

// ...handle request...

// prepare response
var response = {
  statusCode: C.SUCCESSFUL_OK, // set `operationId` instead if encoding a request
  requestId: decoded.requestId,
  groups: [
    { tag: C.OPERATION_ATTRIBUTES_TAG, attributes: [
      { tag: C.CHARSET, name: 'attributes-charset', value: 'utf-8' },
      { tag: C.NATURAL_LANG, name: 'attributes-natural-language', value: 'en-us' },
      { tag: C.TEXT_WITH_LANG, name: 'status-message', value: { lang: 'en-us', value: 'successful-ok' }
    ] },
    { tag: C.JOB_ATTRIBUTES_TAG, attributes: [
      { tag: C.INTEGER, name: 'job-id', value: 147 },
      { tag: C.NAME_WITH_LANG, name: 'job-name', value: { lang: 'en-us', value: 'Foobar' } }
    ] }
  ]
}

// encode response to binary buffer
ipp.response.encode(response) // <Buffer 01 01 00 00 ... >
```

```
var ipp = require('ipp-encoder')
var C = ipp.CONSTANTS

// decode binary buffer from IPP client
var decoded = ipp.request.decode(buf)

// ...handle request...

// prepare response
var response = {
  statusCode: C.SUCCESSFUL_OK, // set `operationId` instead if encoding a request
  requestId: decoded.requestId,
  groups: [
    { tag: C.OPERATION_ATTRIBUTES_TAG, attributes: [
      { tag: C.CHARSET, name: 'attributes-charset', value: 'utf-8' },
      { tag: C.NATURAL_LANG, name: 'attributes-natural-language', value: 'en-us' },
      { tag: C.TEXT_WITH_LANG, name: 'status-message', value: { lang: 'en-us', value: 'successful-ok' }
    ] },
    { tag: C.JOB_ATTRIBUTES_TAG, attributes: [
      { tag: C.INTEGER, name: 'job-id', value: 147 },
      { tag: C.NAME_WITH_LANG, name: 'job-name', value: { lang: 'en-us', value: 'Foobar' } }
    ] }
  ]
}

// encode response to binary buffer
ipp.response.encode(response) // <Buffer 01 01 00 00 ... >
```

```
var ipp = require('ipp-encoder')
var C = ipp.CONSTANTS

// decode binary buffer from IPP client
var decoded = ipp.request.decode(buf)

// ...handle request...

// prepare response
var response = {
  statusCode: C.SUCCESSFUL_OK, // set `operationId` instead if encoding a request
  requestId: decoded.requestId,
  groups: [
    { tag: C.OPERATION_ATTRIBUTES_TAG, attributes: [
      { tag: C.CHARSET, name: 'attributes-charset', value: 'utf-8' },
      { tag: C.NATURAL_LANG, name: 'attributes-natural-language', value: 'en-us' },
      { tag: C.TEXT_WITH_LANG, name: 'status-message', value: { lang: 'en-us', value: 'successful-ok' }
    ] },
    { tag: C.JOB_ATTRIBUTES_TAG, attributes: [
      { tag: C.INTEGER, name: 'job-id', value: 147 },
      { tag: C.NAME_WITH_LANG, name: 'job-name', value: { lang: 'en-us', value: 'Foobar' } }
    ] }
  ]
}

// encode response to binary buffer
ipp.response.encode(response) // <Buffer 01 01 00 00 ... >
```

```
var ipp = require('ipp-encoder')
var C = ipp.CONSTANTS

// decode binary buffer from IPP client
var decoded = ipp.request.decode(buf)

// ...handle request...

// prepare response
var response = {
  statusCode: C.SUCCESSFUL_OK, // set `operationId` instead if encoding a request
  requestId: decoded.requestId,
  groups: [
    { tag: C.OPERATION_ATTRIBUTES_TAG, attributes: [
      { tag: C.CHARSET, name: 'attributes-charset', value: 'utf-8' },
      { tag: C.NATURAL_LANG, name: 'attributes-natural-language', value: 'en-us' },
      { tag: C.TEXT_WITH_LANG, name: 'status-message', value: { lang: 'en-us', value: 'successful-ok' }
    ] },
    { tag: C.JOB_ATTRIBUTES_TAG, attributes: [
      { tag: C.INTEGER, name: 'job-id', value: 147 },
      { tag: C.NAME_WITH_LANG, name: 'job-name', value: { lang: 'en-us', value: 'Foobar' } }
    ] }
  ]
}

// encode response to binary buffer
ipp.response.encode(response) // <Buffer 01 01 00 00 ... >
```

Printer operations

- Print Job 0x02
- Print URI 0x03
- Validate Job 0x04
- Create Job 0x05
- Get Printer Attributes 0x0b
- Get Jobs 0x0a
- Pause Printer 0x10
- Resume Printer 0x11
- Purge Jobs 0x12

Job operations

- Send Document 0x06
- Send URI 0x07
- Cancel Job 0x08
- Get Job Attributes 0x09
- Hold Job 0x0c
- Release Job 0x0d
- Restart Job 0x0e

Printer operations

- Print Job 0x02 👍
- Print URI 0x03
- Validate Job 0x04 👍
- Create Job 0x05
- Get Printer Attributes 0x0b 👍
- Get Jobs 0x0a 👍
- Pause Printer 0x10
- Resume Printer 0x11
- Purge Jobs 0x12

Job operations

- Send Document 0x06
- Send URI 0x07
- Cancel Job 0x08 👍
- Get Job Attributes 0x09 👍
- Hold Job 0x0c
- Release Job 0x0d
- Restart Job 0x0e

github.com /
watson /
ipp-printer

NCFACP

(NSA/CIA/FBI auto censoring printer)

[redacted] earlier tests. [redacted] phenomenon [redacted]

Typical symptoms include:

* Panic
* Increased Heart Rate
* [redacted]
* REM
* [redacted]

[redacted] visibly disoriented [redacted] exposure, [redacted] hallucinogen [redacted] Castle [redacted] changes in their [redacted] all for naught.

Subject hostility [redacted] sedate, [redacted] high resilience towards pain [redacted] chilling result. [redacted] solutions are shortcoming. [redacted] knawing their hands [redacted] winter hit. [redacted] on-base generators [redacted]

printb.in

Attack vector:
Zeroconf / Bonjour

Man in the middle

- Attacker: Malicious mDNS announcement
- Target: Forced name change
- Clients: Update reference to new host/port
- Attacker: Intercept and forward all jobs

github.com /
watson /
bcc

Thank you! -> Q&A

@wa7son
github.com / watson

github.com / watson / bonjour
github.com / watson / ipp-encoder
github.com / watson / ipp-printer
github.com / watson / printcat
github.com / watson / bcc
printb.in