



Extending libphonenumber

(for fun and profit)

Daniel Bruhn

PayPal

Outline

- Concepts
- Requirements
- Libphonenumber & 3rd-party packages
- fullstack-phone
- Challenges
- Concerns

Concepts - GB example

1212345678

Concepts - GB example

national significant number

1212345678

Concepts - GB example

national significant number

0 1212345678

national prefix

Concepts - GB example

	national significant number
<u>44</u>	<u>1212345678</u>
country code (GB)	

Concepts - GB example

exit code

+

44

country code
(GB)

national significant number

1212345678

Concepts - GB example

exit code (US)

011 44

country code
(GB)

national significant number

1212345678

Concepts - GB example

exit code (CH)

00 44

country code
(GB)

national significant number

1212345678

Clarification

44 = “country code”

GB = “region code”

Concepts - formats

	National	International
US	(510) 555-1234	+1 510-555-1234
GB	0121 234 5678	+44 121 234 5678
RU	8 (301) 123-45-67	+7 301 123-45-67
IT	02 1234 5678	+39 02 1234 5678

Requirements - i18n phone module

- Client-side
- Format. Validate. Normalize.
- Canonical, idiomatic-JS phone object

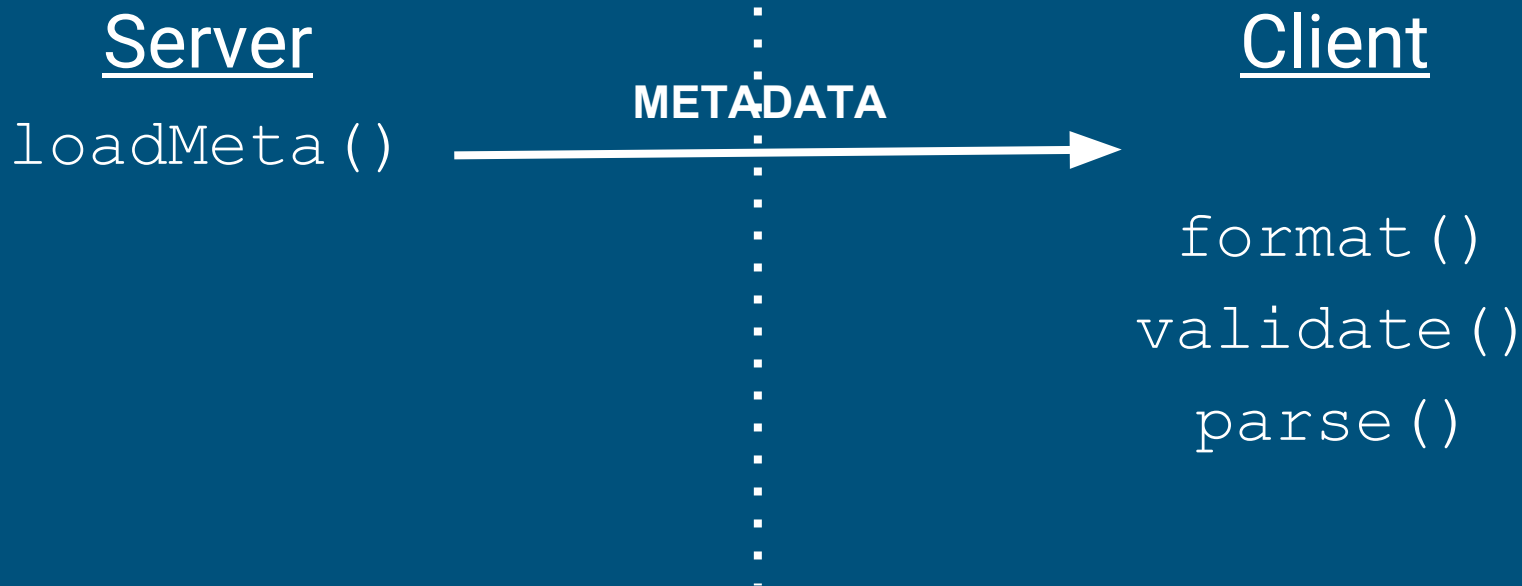
Canonical phone object

```
{  
  countryCode: '1',  
  nationalNumber: '5105551234',  
  extension: '999'  
}
```

Requirements - i18n phone module

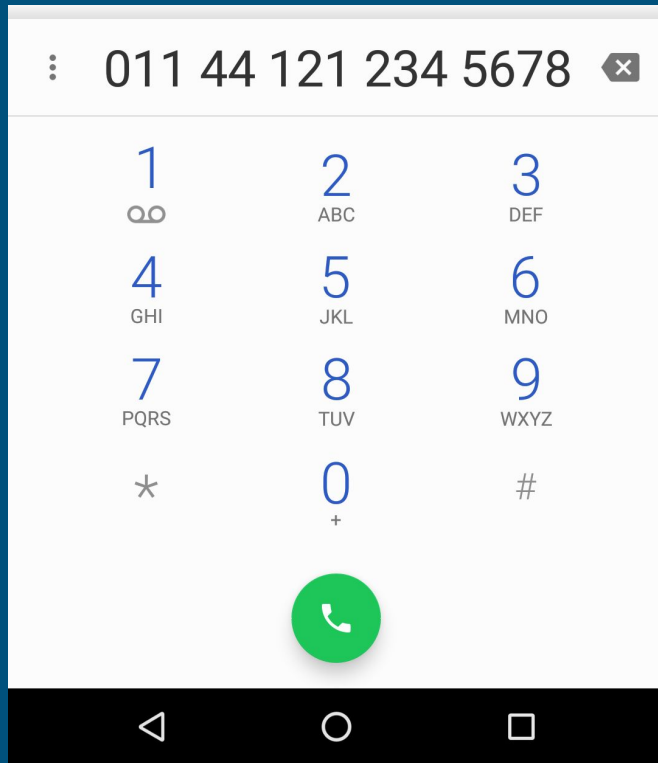
- Client-side
- Format. Validate. Normalize.
- Canonical, idiomatic-JS phone object
- Reliable and up to date
- Small code, decoupled from metadata
- Pluggable & combinable regional metadata

Architecture



Google libphonenumber

- <https://github.com/googlei18n/libphonenumber>
- “Google's common Java, C++ and JavaScript library for parsing, formatting, and validating international phone numbers.”



Google libphonenumber

- JavaScript port
 - Relies on Google Closure library
 - `goog.require`, `goog.array`, `goog.object`, **etc.**
 - Compile with Closure Compiler

Multiple third-party packages


36 packages found for "**libphonenumber**"













- <https://github.com/ruimarinho/google-libphonenumber>
- <https://github.com/grantila/awesome-phonenumber>
- <https://github.com/leodido/i18n.phonenumbers.js>
- <https://github.com/catamphetamine/libphonenumber-js>

Multiple third-party packages

- Two factors mitigated by 3rd-party packages:
 - Google Closure dependency
 - Metadata size

Branch: master ▾ libphonenumber / javascript / i18n /

 **SumathiSundara** Metadata updates for release 8.7.1 (#1887)

17 contributors            

8002 lines (7997 sloc) | 202 KB

```
1  /**
2   * @license
3   * Copyright (C) 2010 The Libphonenumber Authors
4   *
5   * Licensed under the Apache License, Version 2.0 (th
6   * you may not use this file except in compliance wit
7   * You may obtain a copy of the License at
8   *
```

ruimarinho/google-libphonenumber

- Popular
- Thin wrapper
- Closure-compiled
- Metadata + code
- 416 KB
 - .min.gz: 72 KB

4956 lines (4942 sloc) | 416 KB

```
1  (function(f){if(typeof exports==="object"&&typeof modu
2  var COMPILED = !0, goog = goog || {});
3  goog.global = this;
4  goog.isDef = function(a) {
5    return void 0 !== a;
6  };
7  goog.isString = function(a) {
8    return "string" == typeof a;
9  };
10 goog.isBoolean = function(a) {
11    return "boolean" == typeof a;
```

grantila/awesome-phonenumbers

- Simplified API
- Closure-compiled
- Metadata + code
- 223 KB
 - .min.gz: 50 KB

472 lines (459 sloc) | 223 KB

```
1  (function(){var h,aa=this;function l(a){return"st
2  function p(a,b){function c(){}}c.prototype=b.proto
3  function oa(a,b){for(var c=fa(a.h()),d=0;d<c.leng
4  function z(a,b){var c=a.b[b];if(null==c)return nu
5  function y(a,b){return z(a,b)||[]}}function C(a,b)
6
7  Protocol Buffer 2 Copyright 2008 Google Inc.
8  All other code copyright its respective owners.
9  Copyright (C) 2010 The Libphonenumber Authors
10
```

leodido/i18n.phonenumbers.js

- Closure-compiled
- Regional bundles
- Metadata + code
- 212 KB (minified)
 - .gz: 49 KB

410 lines (409 sloc) | 212 KB

```
1  (function(){'use strict';var
2  var p={1:"US AG AI AS BB BM B
3  86:["CN"],90:["TR"],91:["IN"]
4  253:["DJ"],254:["KE"],255:["T
5  386:["SI"],387:["BA"],389:["M
6  691:["FM"],692:["MH"],800:["0
7  [,,"(?:[267]\\d|3[0-5]|4[4-69
8  ,,,[["(\\d{3})(\\d{3})", "$1
```

catamphetamine/libphonenumber-js

- Pure JS rewrite
- Code separate from metadata
- Regional metadata
- Design differences
- Full **metadata** JSON: 127 KB (.gz: 32 KB)
- Code + metadata (.min.giz): 28 KB

```
1 lines (1 sloc) 127 KB
```

```
1 {"country_phone_code_to_countries":{"1":["US","AG
```

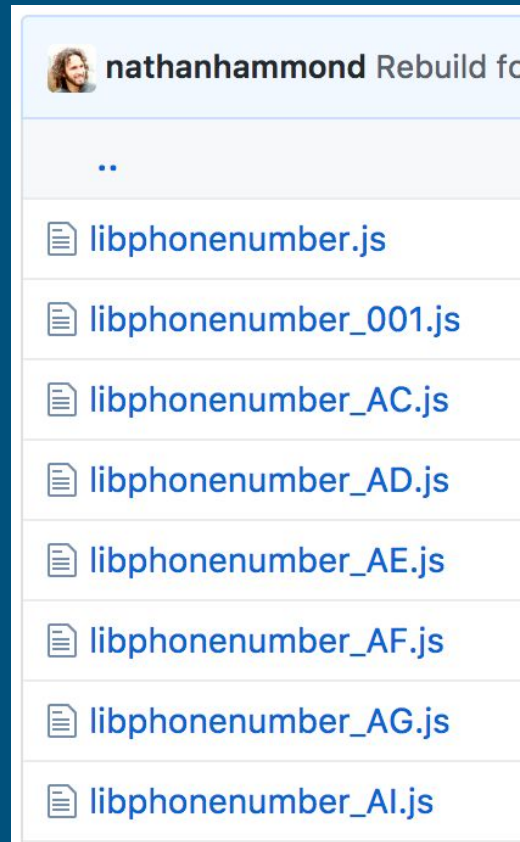
What now?

None meet all requirements.

Time to extend.

Strategy

- Initial fork:
[nathanhammond/libphonenumber](https://github.com/nathanhammond/libphonenumber)
- Created dual-module system
 - Server: metadata loading
 - Client: phone functionality



Server - build time

- Slice metadata.js into regions

```
"AC":{"regionCodes":["AC"],"countryCodeToRegionCodeMap":{"247":["AC"]},"countryToMetadata":{"AC":[null,[null,null,"[46]\\"  
"AD":{"regionCodes":["AD"],"countryCodeToRegionCodeMap":{"376":["AD"]},"countryToMetadata":{"AD":[null,[null,null,"[16]\\"  
"AE":{"regionCodes":["AE"],"countryCodeToRegionCodeMap":{"971":["AE"]},"countryToMetadata":{"AE":[null,[null,null,"[2-79]"  
"AF":{"regionCodes":["AF"],"countryCodeToRegionCodeMap":{"93":["AF"]},"countryToMetadata":{"AF":[null,[null,null,"[2-7]\\"  
"AG":{"regionCodes":["AG"],"countryCodeToRegionCodeMap":{"1":["AG"]},"countryToMetadata":{"AG":[null,[null,null,"[2589]\\"  
"AI":{"regionCodes":["AI"],"countryCodeToRegionCodeMap":{"1":["AI"]},"countryToMetadata":{"AI":[null,[null,null,"[2589]\\"  
"AL":{"regionCodes":["AL"],"countryCodeToRegionCodeMap":{"355":["AL"]},"countryToMetadata":{"AL":[null,[null,null,"[2-57]"  
"AM":{"regionCodes":["AM"],"countryCodeToRegionCodeMap":{"374":["AM"]},"countryToMetadata":{"AM":[null,[null,null,"[1-9]\\"
```

Server - build time

- Extract dependency map

```
"regionCodeToCountryCodeMap":{"US":["1"],"AG":["1"]  
  
"countryCodeToRegionCodeMap":{  
  "1":["US","AG","AI","AS","BB","BM","BS","CA","D  
  "7":["RU","KZ"],  
  "20":["EG"],  
  "27":["ZA"],  
  "30":["GR"],  
  "31":["NL"],  
  "32":["BE"],  
  "33":["FR"],  
  "34":["ES"],  
  "36":["HU"],  
  "39":["IT","VA"],  
  "40":["RO"],  
  "41":["CH"].
```

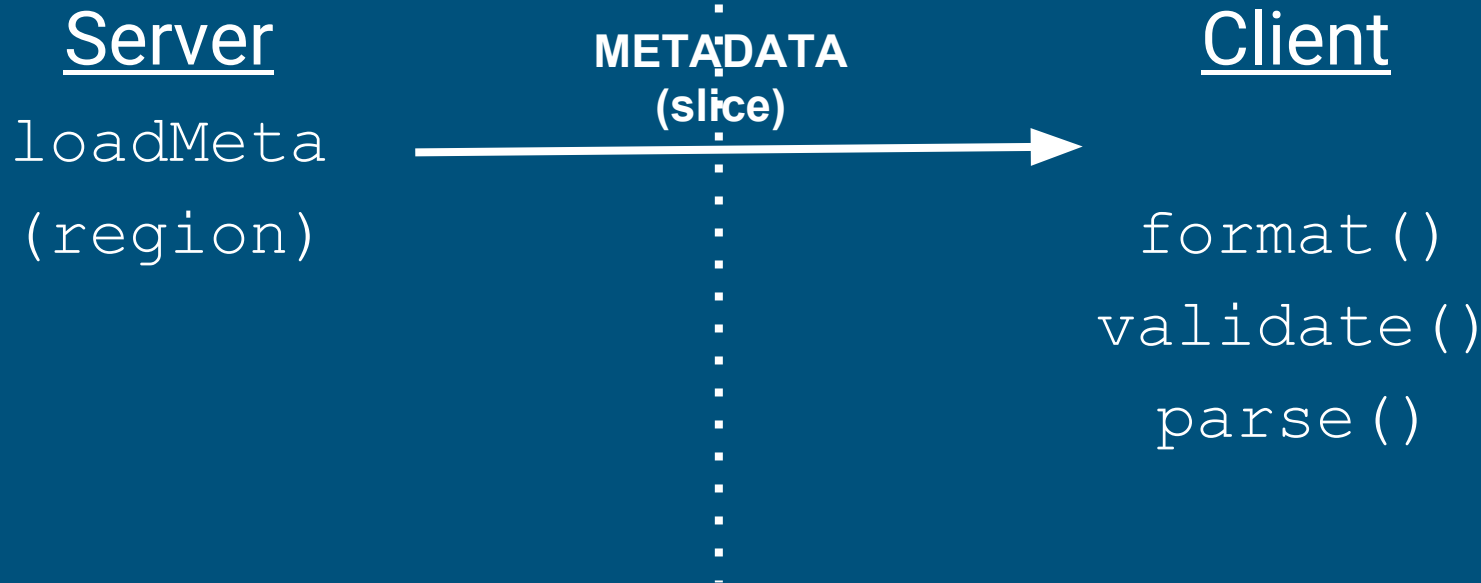
Server - runtime

- Metadata loader utilizes dependency map:
 - CA, TT, BS depend on US (country code 1)
 - KZ --> RU (7)
 - CX --> AU (67)

Client

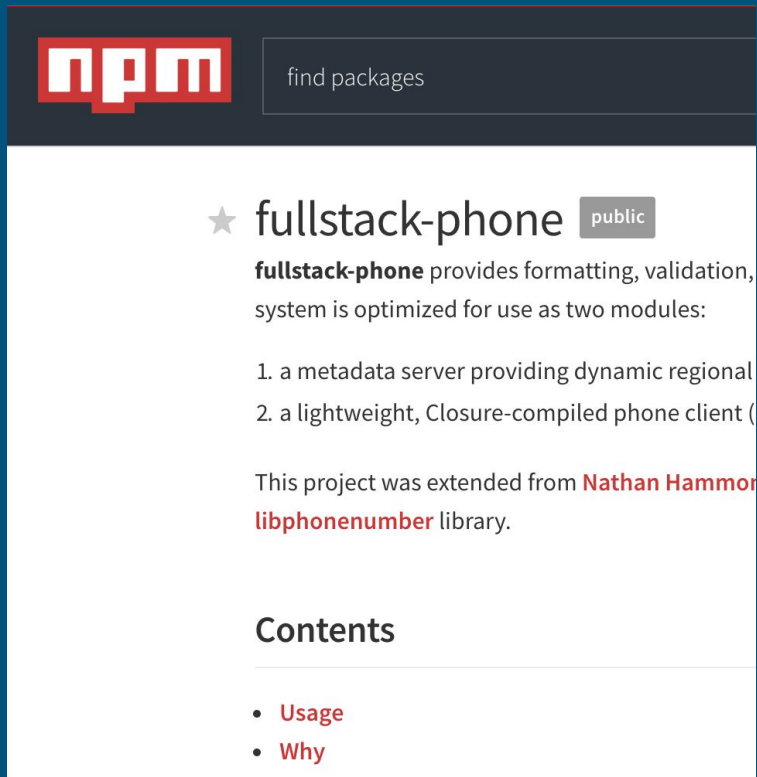
- Client:
 - Write adapter for libphonenumber
 - Expose only necessary functionality
 - Compress with Closure Compiler
 - ADVANCED_OPTIMIZATIONS

Architecture (again)



fullstack-phone

- [dwbruhn/fullstack-phone](https://github.com/dwbruhn/fullstack-phone)
- Pluggable regional metadata
 - US JSON: 2.2 KB (.gz 542 b)
 - JP JSON: 9.9 KB (.gz 1.3 KB)
- Client code: 24 KB
 - gzipped: 8.6 KB



The screenshot shows the npm package page for 'fullstack-phone'. At the top is the npm logo and a search bar. The package name 'fullstack-phone' is displayed with a 'public' badge. The description states that the package provides formatting, validation, and is optimized for use as two modules: a metadata server and a phone client. It also mentions that the project was extended from the 'libphonenumber' library. A 'Contents' section lists 'Usage' and 'Why' as sub-topics.

npm find packages

★ **fullstack-phone** public

fullstack-phone provides formatting, validation, system is optimized for use as two modules:

1. a metadata server providing dynamic regional
2. a lightweight, Closure-compiled phone client (

This project was extended from **Nathan Hammon** **libphonenumber** library.

Contents

- **Usage**
- **Why**

Demo

- [dwbruhn/fullstack-phone-demo](#)

Challenges

Challenges

- Support independent, regional handlers:

```
let usHandler = createPhoneHandler(usMeta);  
let gbHandler = createPhoneHandler(gbMeta);  
Let deHandler = createPhoneHandler(deMeta);
```

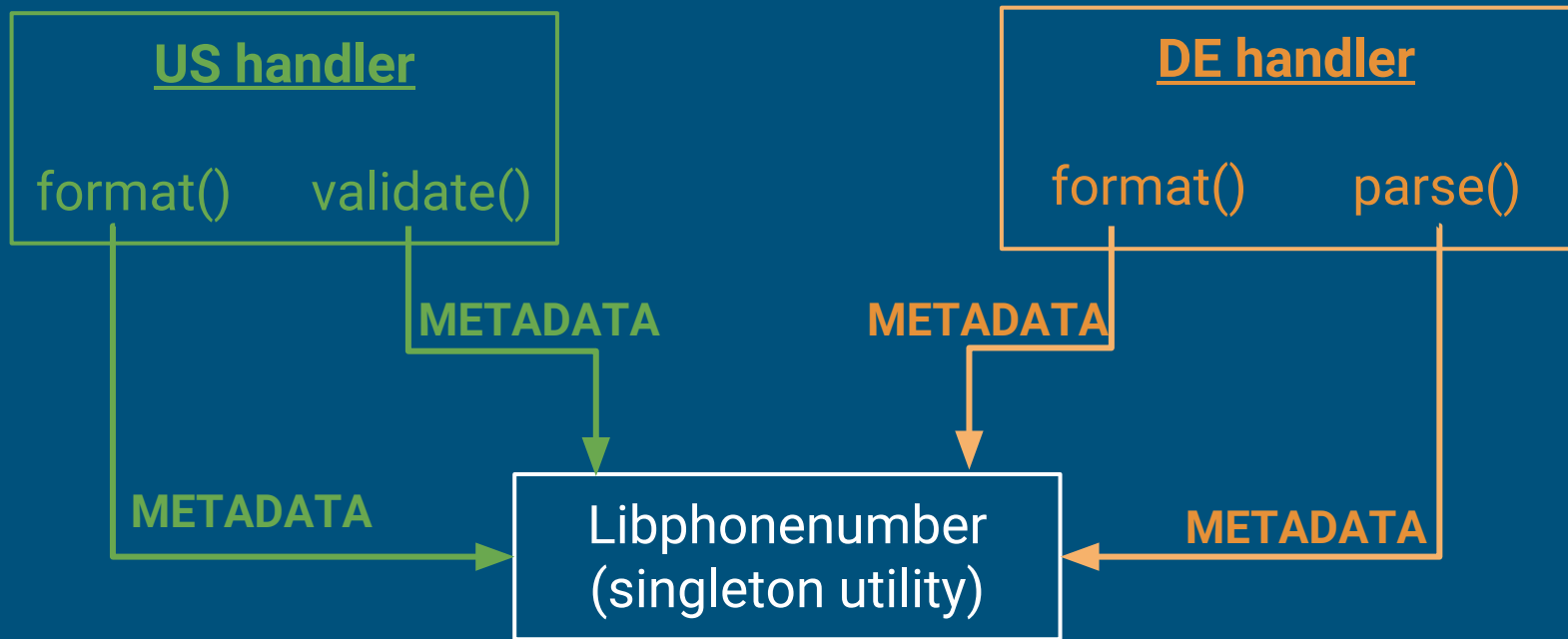
Challenges

```
/**
 * @constructor
 * @private
 */
i18n.phonenumbers.PhoneNumberUtil = function() {
  // ...
};
goog.addSingletonGetter(i18n.phonenumbers.PhoneNumberUtil);
```

Challenges - singleton

- Libphonenumber is a singleton utility
- Constructor is marked as `@private`
 - Can be bypassed, but is there another way?

Challenges - singleton



Metadata injection

```
function injectMeta(metadata) {  
    i18n.phonenumbers.metadata.countryCodeToRegionCodeMap =  
        metadata['countryCodeToRegionCodeMap'];  
  
    i18n.phonenumbers.metadata.countryToMetadata =  
        metadata['countryToMetadata'];  
}
```

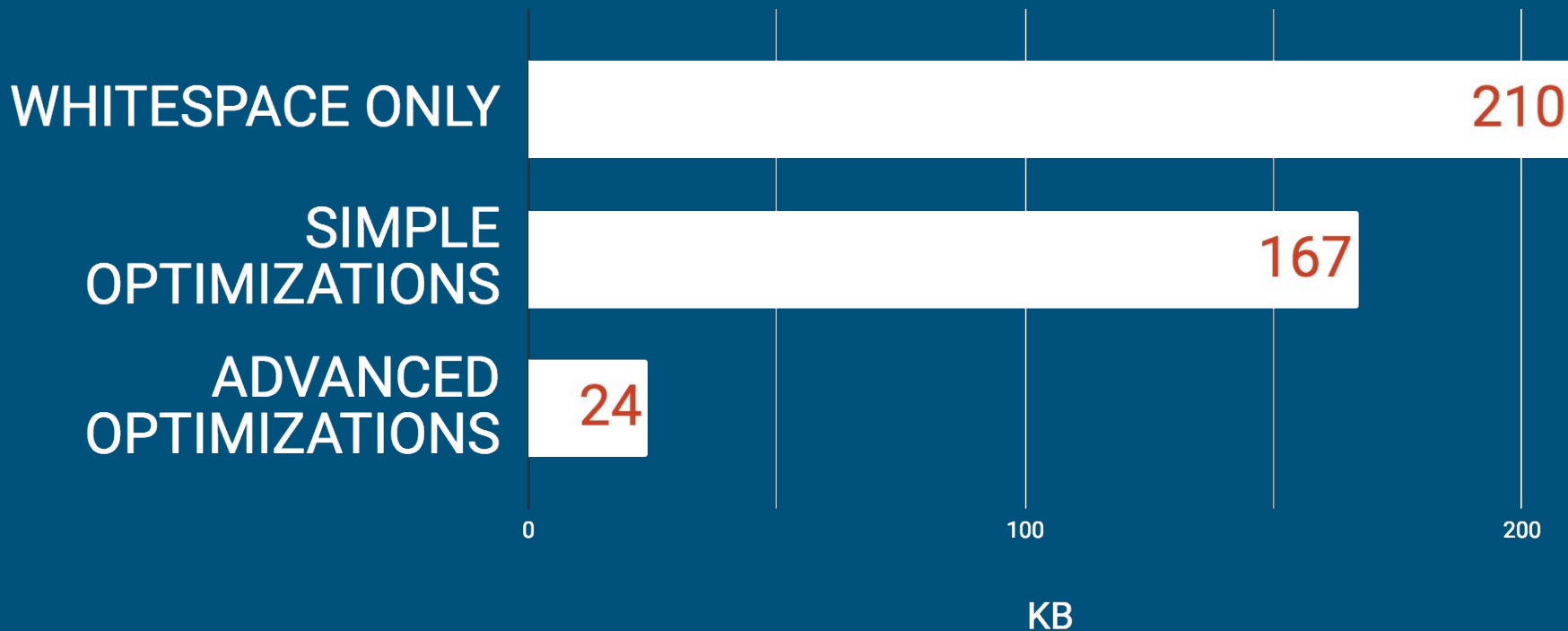
Why not this?

```
function injectMeta(metadata) {  
    i18n.phonenumbers.metadata.countryCodeToRegionCodeMap =  
        metadata.countryCodeToRegionCodeMap;  
  
    i18n.phonenumbers.metadata.countryToMetadata =  
        metadata.countryToMetadata;  
}
```

Challenges - ADVANCED_OPTIMIZATIONS

- Closure Compiler ADVANCED_OPTIMIZATIONS
 - <https://developers.google.com/closure/compiler/docs/api-tutorial3# dangers>
 - “Use quoted string property names only when you don't want Closure Compiler to rename a property at all.”

Closure Compiler Levels



Challenges - ADVANCED_OPTIMIZATIONS

- Dead code removal
- To prevent elimination, functions must be:
 - Called
 - or
 - **Exported** (`goog.exportSymbol`)

Challenges - ADVANCED_OPTIMIZATIONS

Attempting to expose the `AsYouTypeFormatter` constructor from `libphonenumber`

Round 1

```
'AsYouTypeFormatter1': function AsYouTypeFormatter1() {  
    injectMeta(metadata);  
    return i18n.phonenumbers.AsYouTypeFormatter;  
},
```

Round 1: Fail

```
> var AsYouTypeFormatter = handler.AsYouTypeFormatter1()  
undefined  
> var formatter = new AsYouTypeFormatter('US')  
undefined  
> formatter  
V {  
  ja: / /,  
  ca: ' ',  
  o: q {},  
  '$': ' ',  
  i: q {},  
  ..
```

Round 2

```
'AsYouTypeFormatter2': function AsYouTypeFormatter2(regionCode) {  
    injectMeta(metadata);  
    assertSupportedRegion(regionCode, metadata);  
    return new i18n.phonenumbers.AsYouTypeFormatter(regionCode);  
},
```

Round 2: Fail

```
> var formatter = handler.AsYouTypeFormatter2('US')
undefined
> formatter
V {
  ja: / /,
  ca: ' ',
  o: q {},
  '$': ' ',
  i: q {},
}
```

Round 3

```
'AsYouTypeFormatter3': function AsYouTypeFormatter3(regionCode) {  
    injectMeta(metadata);  
    assertSupportedRegion(regionCode, metadata);  
    var formatter = new i18n.phonenumbers.AsYouTypeFormatter(regionCode);  
    return {  
        'inputDigit': formatter.inputDigit,  
        'clear': formatter.clear  
    };  
},
```


Round 3: Fail

```
> var formatter = handler.AsYouTypeFormatter3('US')
undefined
> formatter
{ inputDigit: [Function], clear: [Function] }
> formatter.inputDigit(1)
TypeError: Cannot read property 'a' of undefined
```

Round 4

```
'AsYouTypeFormatter4': function AsYouTypeFormatter4(regionCode) {  
    injectMeta(metadata);  
    assertSupportedRegion(regionCode, metadata);  
    var formatter = new i18n.phonenumbers.AsYouTypeFormatter(regionCode);  
    return {  
        'inputDigit': function inputDigit(x) {  
            return formatter.inputDigit(x);  
        },  
        'clear': function clear() {  
            return formatter.clear();  
        }  
    };  
},
```

Round 4: Success!

```
> var formatter = handler.AsYouTypeFormatter4('US')
undefined
> formatter
{ inputDigit: [Function], clear: [Function] }
> formatter.inputDigit(1)
'1'
> formatter.inputDigit(5)
'15'
```

Concerns

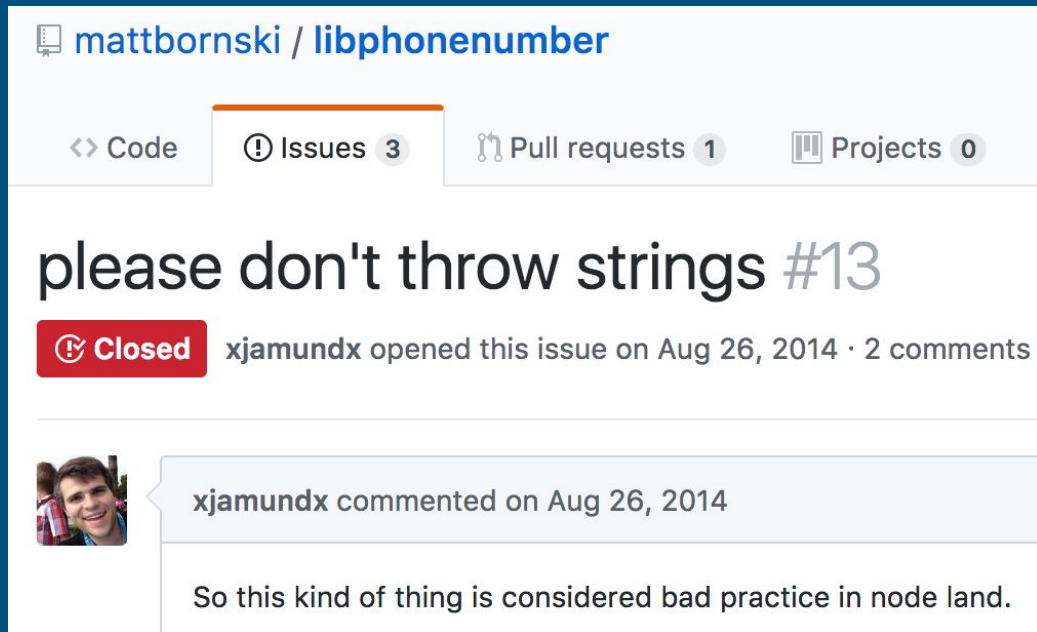
- A friendly appeal

Concerns

- Release notes miss crucial changes

Release notes

- Libphonenumber throws **strings**
- So do some 3rd-party packages
- Handle by mapping string → Error



The screenshot shows a GitHub interface for the repository 'mattbornski / libphonenumber'. The 'Issues' tab is selected, showing 3 issues. A specific issue titled 'please don't throw strings #13' is highlighted, marked as 'Closed'. It was opened by user 'xjamundx' on August 26, 2014, and has 2 comments. A comment from 'xjamundx' is visible, stating: 'So this kind of thing is considered bad practice in node land.'

mattbornski / libphonenumber

<> Code ⓘ Issues 3 🔗 Pull requests 1 📁 Projects 0

please don't throw strings #13

Closed xjamundx opened this issue on Aug 26, 2014 · 2 comments

xjamundx commented on Aug 26, 2014

So this kind of thing is considered bad practice in node land.

Release notes

- Updated fullstack-phone to libphonenumber v8.0.0
- Compiled
- Test failures!
- Validation errors not showing up

10) Phone Data-Driven Tests (Negative Parsing) Region: AU Item: 2 Should return parse error:

```
AssertionError: Unexpected parsing result for AU 0011  
+ expected - actual  
  
+PHONE_TOO_SHORT_AFTER_IDD  
  
at Context.<anonymous> (test/test-phone-data-parsing.js:44:32)
```

11) Phone Data-Driven Tests (Negative Parsing) Region: AU Item: 3 Should return parse error:

```
AssertionError: Unexpected parsing result for AU +15105261234  
+ expected - actual  
  
+PHONE_INVALID_COUNTRY_CODE  
  
at Context.<anonymous> (test/test-phone-data-parsing.js:44:32)
```

12) Phone Data-Driven Tests (Negative Parsing) Region: AU Item: 4 Should return parse error:

```
AssertionError: Unexpected parsing result for AU +997510  
+ expected - actual  
  
+PHONE_INVALID_COUNTRY_CODE  
  
at Context.<anonymous> (test/test-phone-data-parsing.js:44:32)
```

Release notes

- [v8.0.0 release notes:](#)

Dec 07, 2016: v8.0.0

Code changes:

- Deleting deprecated methods in ShortNumberInfo that work on strings, instead of phone number objects. These have been marked deprecated for months. Any users of these methods should call PhoneNumberUtil.parse first to create a PhoneNumber object, and pass this in.
- Support semicolon as extension character while parsing phone numbers. This is not applicable when you are trying to find the phone numbers.

Release notes

- [This commit:](#)

```
-      throw i18n.phonenumbers.Error.INVALID_COUNTRY_CODE;
```

```
+      throw new Error(i18n.phonenumbers.Error.INVALID_COUNTRY_CODE);
```

Release notes

- [This commit:](#)

googlei18n / libphonenumber

Watch 239 Star 5

<> Code Issues 82 Pull requests 4 Projects 0 Wiki Insights

Updating the JS code to throw an error rather than a "string" (#1459)

This could break client code. For anyone who has custom behaviour based on the value of the error, please compare against error.message instead of the error itself.

master (#1459) v8.6.0 ... v8.0.0

Concerns

- Release notes miss crucial changes
- Breaking API changes
 - Especially in non-major releases
 - Difficulties for third-party package maintainers

Concerns

- “If any of the changes ... may cause libphonenumber (Java, C++, or JS) clients to have to change their code to keep building, we publish a major release.”

<https://github.com/googlei18n/libphonenumber#versioning-and-announcements>

Breaking changes

- v7.7.4: Removed fallback to GENERAL in metadata functions
- v7.7.5: Removed `getPossiblePhoneNumberPattern`
- v8.4.1: Changed output of `isPossibleNumberWithReason`
- v8.6.0: Removed `isLeadingZeroPossible`

Appeals

- libphonenumber maintainers:
 - Please think of your clients!
- Potential third party package developers:
 - Be prepared to refactor!

Fun

- Hacking libphonenumbers
- Coding challenges
- Breaking changes!
- Open Source

Profit

- <https://github.com/dwbruhn/fullstack-phone>
- Browser phone module
- Canonical phone object
- Small code & pluggable metadata for quicker loading
- Validation: reduce risk
- Formatting: enhance user experience

Thank you!

