



Opportunities in Software Engineering Research for Web API Consumption

WAPI'17

Erik Wittern, Annie Ying, Yunhui Zheng, Jim A. Laredo, Julian Dolby, Christopher C. Young, Aleksander A. Slominski

IBM T.J. Watson Research Center

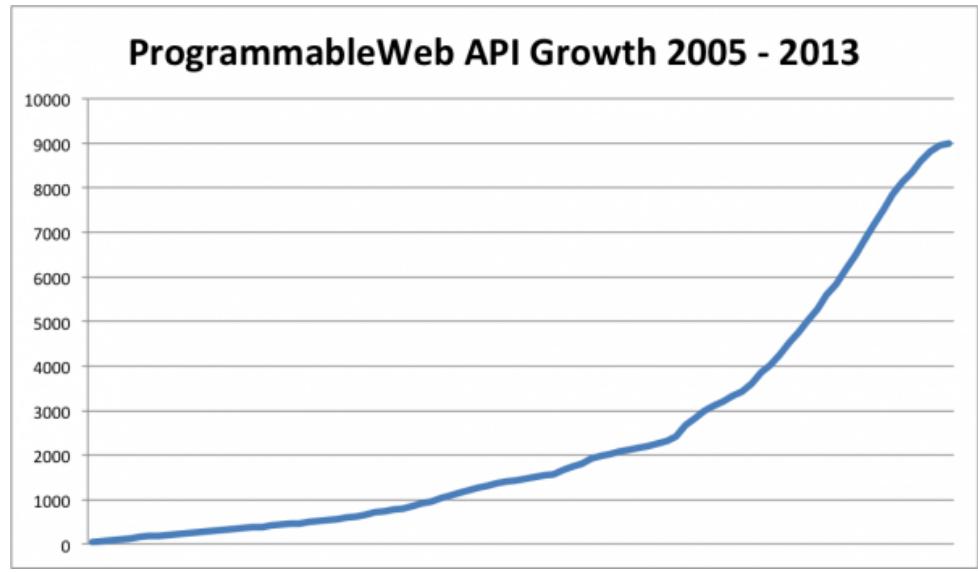


Part 1: Web APIs and challenges for consumption



Web APIs are ubiquitous

- The number of web APIs is continuously growing
- Enable programmatic interaction with remote resources
- ...using of existing, ubiquitous Web technologies
(mostly HTTP + JSON / XML)
- APIs are of great importance for industry to create application ecosystems
- Focus on simplicity & flexibility – few conventions or rules
(contrast with SOAP/WS-*)



Source: ProgrammableWeb.com, now > 17k



IBM API Connect

StrongLoop



Consumption is challenging, though

String-based interface;
no type-checking

```
request({  
  url: 'http://api.example.com/v1/events',  
  method: 'post',  
  body: JSON.stringify({name: 'WAPI \'17'})  
}, (err, response, strData) => {  
  let responseData = JSON.parse(strData)  
  // ...  
})
```

Frequent changes

The screenshot shows three entries from the Twilio API Changelog:

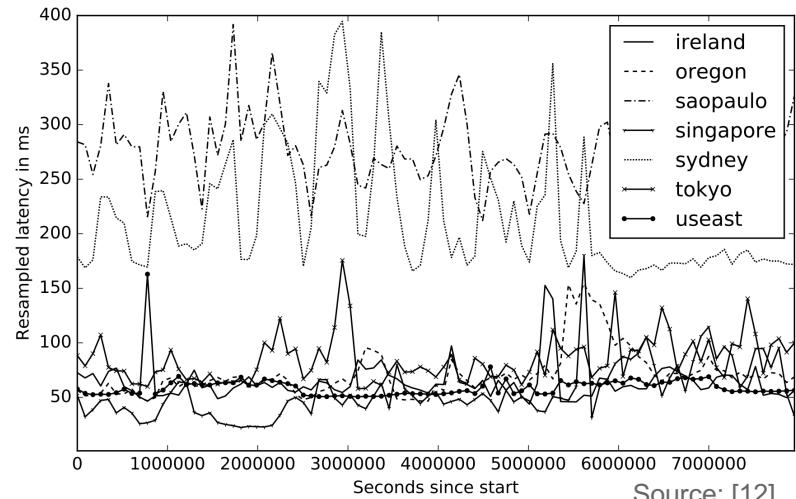
- May 15 2017 Twilio Large edit Changed REST API: Available Phone Numbers - Twilio
- May 15 2017 Twilio Small edit Changed Creating Tasks and Accepting Reservations: Accept a Reservation using Assignment Callback Instructions - Twilio
- May 15 2017 Twilio Small edit Changed How to Modify Calls in Progress in C# - Twilio

[a]

APIs are controlled by third-party



Remote calls with varying QoS



Opinion: Gap between practice & challenges and software engineering research

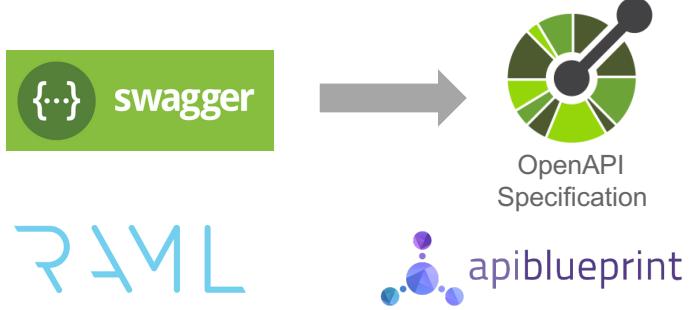


Part 2: IDE support for checking web API requests



Specifications are central to support web API consumption

- Specifications describe possible interactions with an API – they depict the *contract* between client and API
- Machine-readable
- Various formats:



- They describe...
 - Base URL
 - Endpoints (path template + HTTP method)
 - Data required by / returned from endpoints
 - Additional parameters (e.g. in query)
 - Other constraints like headers, authentication, rate limits...

```
swagger: '2.0'  
host: api.twilio.com  
basePath: /2010-04-01  
schemes:  
  - http  
paths:  
  /Accounts/{userId}/Messages.json:  
    post:  
      description: 'Send a message'  
      parameters:  
        - in: path  
          name: userId  
          required: true  
          type: string  
        - in: body  
          name: body  
          schema:  
            $ref: '#/definitions/Message'  
        - in: query  
          name: 'retry'  
          required: false  
          type: boolean  
      ...  
  definitions:  
    Message:  
      type: object  
      properties:  
        from: ...  
        to: ...  
        body: ...  
      required: from, to, body
```

Overview of the approach

```
1 var getPicturesForTag = function (tag) {  
2     var query = {  
3         count: 10  
4     }  
5     var url = 'https://api.instagram.com/v1/tags/' + tag + '/media/recent'  
6     var settings = {  
7         method: "get"  
8         url: url,  
9         data: query  
10    }  
11    sendRequest(settings)  
12 }  
13  
14 var sendRequest = function (settings) {  
15     jQuery.ajax(settings).done(function (response) {  
16         console.log(response)  
17     })  
18 }
```

1. Extract request

Details: ICSE Web applications Thursday 11:00-12:30

→ = data flow
⇒ = control flow

2. Match request against specifications

```
swagger: "2.0"  
info:  
  title: "API Test Site"  
  version: "2016-04-01"  
  schemes:  
    - http  
    - https  
paths:  
  /Accounts/{userId}/Messages.{json}  
  post:  
    description: "Send a message"  
    parameters:  
      - in: query  
        name: order  
        required: true  
        type: integer  
      - in: body  
        name: body  
        schema:  
          $ref: "#/definitions/Message"  
definitions:  
  Message:  
    type: object  
    properties:  
      from: ...  
      to: ...  
      body: ...  
    required: from, to, body
```

3. Report results

(a)

```
File 0 Project 0 ✓ No Issue LF ⚠ 1 deprecation UTF-8 JavaScript master +21 3 u
```

Instagram detected 1 errors

- Method "post" not found for path "/tags/{tag-name}/media/recent".

```
1 var getPicturesForTag = function (tag) {  
2     var query = {  
3         count: 10  
4     }  
5     var url = 'https://api.instagram.com/v1/tags/' + tag + '/media/recent'  
6     var settings = {  
7         method: "POST"  
8         url: url,  
9         data: query  
10    }  
11    sendRequest(settings)  
12 }  
13  
14 var sendRequest = function (settings) {  
15     $.ajax(settings).done(function (response) {  
16         console.log(response)  
17     })  
18 }
```

(b)

```
File 0 Project 0 ✓ No Issue LF ⚠ 1 deprecation UTF-8 JavaScript master +21 3 u
```

API Harmony approves of this request

Matching OpenAPI Specification "Instagram" found.

- Request matches path "/tags/{tag-name}/media/recent".
- Method "get" available for path "/tags/{tag-name}/media/recent".
- All required query parameters are present.
- Payload is in the right format.

```
1 var getPicturesForTag = function (tag) {  
2     var query = {  
3         count: 10  
4     }  
5     var url = 'https://api.instagram.com/v1/tags/' + tag + '/media/recent'  
6     var settings = {  
7         method: "get"  
8         url: url,  
9         data: query  
10    }  
11    sendRequest(settings)  
12 }  
13  
14 var sendRequest = function (settings) {  
15     $.ajax(settings).done(function (response) {  
16         console.log(response)  
17     })  
18 }
```

Example: Request to Medical Lab Inferencing Service

Video: <https://youtu.be/8IJKs7rMjJI>



Many research opportunities

- **Generation / maintenance of specifications**
 - From dynamic traces [7] or via proxies [21]
 - From documentation
 - Through code annotations [b]
- **Static checking of code [8] [*this work*]**
 - During development
 - On existing code bases
- **Automatic testing of APIs**
- **Mining API usage**
 - How are APIs used in the wild?
 - Lessons learned
- **QoS measurement & mitigation [12]**
- **Emerging API paradigms**
 - GraphQL / Apollo (Facebook, GitHub...)
 - Falcor (Netflix...)
- ...





@apiHarmony



<http://ibm.biz/apiharmony>
<http://www.apiful.io>



witternj@us.ibm.com
annie.ying@gmail.com
zhengyu@us.ibm.com
laredoj@us.ibm.com
dolby@us.ibm.com
aslom@us.ibm.com

Discussion points

- Why has web API consumption barely been subject of SE research?
- And how can we change this situation?
- Which existing SE tools & methods from library APIs can be used in the context of web APIs?



Detect inconsistencies between analysis results and specifications

```
{  
  "request": {  
    "data": {},  
    "success": "JSFunction",  
    "error": "JSFunction",  
    "type": "GET",  
    "url": "https://api.instagram.com/v1  
           /tags/<$global#tag$>  
           /media/recent?count=10"  
  }  
}
```

Example data produced by analysis

```
swagger: '2.0'  
info: ...  
host: api.instagram.com  
basePath: /v1  
schemes:  
  - https  
paths:  
  /tags/{tag_id}/media/recent:  
    get:  
      description: 'Get recent media for tag'  
      parameters:  
        - in: query  
          name: count  
          required: true  
          type: number  
      responses:  
        '200':  
          schema:  
            $ref: '#/definitions/Message'  
definitions:  
  Message:  
    type: object  
    properties:  
      from: ...  
      to: ...  
      body: ...  
    required: from, to, body
```

Example specification

System context: API Harmony - *find, learn about, and use web APIs*

The screenshot shows the API Harmony interface. At the top, there's a search bar with placeholder text "Search for APIs...". Below it, a large banner says "Welcome to API Harmony!".

Find APIs: API Harmony currently crawls [APIs.guru](#), from crawling the searching above!

Learn about APIs: API Harmony others use APIs in their applications

Use APIs: API Harmony contains GitHub projects using it, and

API Harmony sidebar includes links to Instagram API Information, Characteristics & Relations, StackOverflow questions, Usages on GitHub, Client Libraries on NPM, Endpoints (Geographies, Locations, Media, Comments, Likes, Tags, Users), and a "DOWNLOAD OPENAPI SPECIFICATION..." button.

Instagram API Details:

- Request:** Set to "JavaScript + jQuery".
Code example:

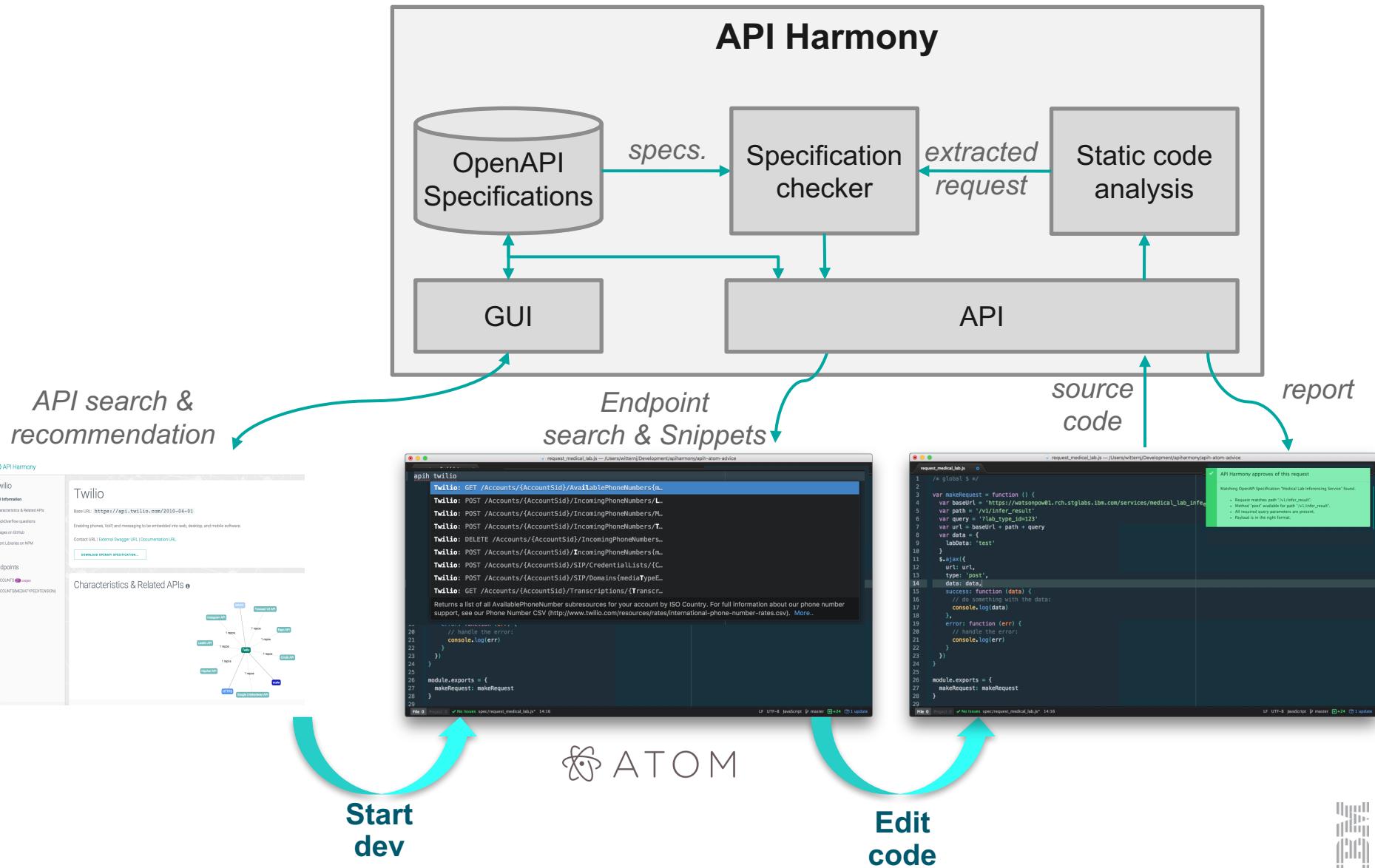
```
var settings = {
  "async": true,
  "crossDomain": true,
  "url": "https://api.instagram.com/v1/media/search?lat=40&long=-74",
  "method": "GET",
  "headers": {
    "access_token": "REPLACE_KEY_VALUE"
  }
}

$.ajax(settings).done(function (response) {
  console.log(response);
});
```
- Responses:** 200
Found media resources (without likes information) in a given area.
- Example Schema:**

```
{
  "data": [
    {}
  ],
  "meta": {
    "code": 0
  }
}
```
- Top response fields:**
 - data (used 16 times)
 - responseText (used 4 times)
 - data.length (used 3 times)
- Parameters:**
 - lat - Latitude of the center search coordinate. If used, 'long' is required. (used 44 times - set to {latitude} 4 times [1] [2] [3])
 - long - Longitude of the center search coordinate. If used, 'lat' is required. (used 44 times - set to {longitude} 4 times [1] [2] [3])
- Encryption scheme:** HTTPS
PI: Instagram



System integration & developer work-flow



References (1/3)

- [a] Espinha, T., Zaidman, A., & Gross, H.-G. (2014). Web API growing pains: Stories from client developers and their code. (pp. 84–93). Presented at the CSMR-WCRE, IEEE.
- [b] <https://github.com/swagger-api/swagger-core>



References (2/3)

- [1] R. T. Fielding, "Architectural styles and the design of network-based software architectures," Ph.D. dissertation, U. of California, Irvine, 2000.
- [2] IBM API Harmony. <https://apiharmony-open.mybluemix.net/>.
- [3] PublicAPIs. <https://www.publicapis.com/>.
- [4] ProgrammableWeb. <http://www.programmableweb.com/>.
- [5] M. Slee, A. Agarwal, and M. Kwiatkowski, "Thrift: Scalable cross-language services implementation," *Facebook White Paper*, 2007.
- [6] M.P.Robillard,E.Bodden,D.Kawrykow,M.Mezini, andT.Ratchford, "Automated API property inference techniques," *IEEE Transactions on Software Engineering*, vol. 39, no. 5, pp. 613–637, 2013.
- [7] P. Suter and E. Wittern, "Inferring Web API Descriptions From Usage Data," in *Proc. of the Workshop on Hot Topics in Web Systems and Technologies*, 2015.
- [8] E. Wittern, A. T. T. Ying, Y. Zheng, J. Dolby, and J. A. Laredo, "Statically Checking Web API Requests in JavaScript," in *Proc. of ICSE, to appear*, 2017.
- [9] B. Dagenais and M. P. Robillard, "Recommending adaptive changes for framework evolution," *ACM TOSEM*, vol. 20, no. 4, p. 19, 2011.
- [10] W. Wu, Y.-G. Gue 'neuc, G. Antoniol, and M. Kim, "Aura: a hybrid approach to identify framework evolution," in *Proc. of ICSE*, 2010, pp. 325–334.
- [11] Open API Initiative. <https://openapis.org/specification>.
- [12] D. Bermbach and E. Wittern, "Benchmarking web API quality," in *Proc. of the International Conference in Web Engineering*, 2016, pp. 188–206.



References (3/3)

- [13] D. Li, S. Hao, J. Gui, and W. G. J. Halfond, "An empirical study of the energy consumption of android applications," in *Proc. of ICSME*, 2014, pp. 121–130.
- [14] E. Chin, A. P. Felt, V. Sekar, and D. Wagner, "Measuring user confidence in smartphone security and privacy," in *Proc. of the Symposium on Usable Privacy and Security*, 2012.
- [15] WADL - Web Application Description Language. <http://www.w3.org/Submission/wadl/>.
- [16] RAML - RESTful API Modeling Language. <http://raml.org/>.
- [17] APIs.guru - Wikipedia for Web APIs. <https://apis.guru/>.
- [18] IBM API Connect. <https://developer.ibm.com/apiconnect/>.
- [19] Swagger Core Library. [https://github.com/swagger-api/swagger- core](https://github.com/swagger-api/swagger-core). [20] swagger-jsdoc. <https://github.com/Surnet/swagger-jsdoc>.
- [21] S.M.Sohan,C.Anslow, and F.Maurer, "SpyREST: AutomatedRESTful API Documentation Using an HTTP Proxy Server," in *Proc. of ASE*, 2015, pp. 271–276.
- [22] A. Feldthaus, M. Schäfer, M. Sridharan, J. Dolby, and F. Tip, "Efficient construction of approximate call graphs for JavaScript IDE services," in *Proc. of ICSE*, 2013, pp. 752–761.
- [23] M. Schäfer, M. Sridharan, J. Dolby, and F. Tip, "Dynamic determinacy analysis," in *Proc. of PLDI*, 2013, pp. 165–174.
- [24] E. Andreasen and A. Møller, "Determinacy in static analysis for jQuery," in *Proc. of OOPSLA*, 2014, pp. 17–31.
- [25] Y. Ko, H. Lee, J. Dolby, and S. Ryu, "Practically tunable static analysis framework for large-scale JavaScript applications," in *Proc. of ASE*, 2015, pp. 541–551.
- [26] Usage Statistics of JavaScript Libraries for Websites, August 2016. <https://w3techs.com/technologies/overview/javascript/library/all/>. [27] Atom Editor. <https://atom.io/>.

