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
Consumption is challenging, though

String-based interface; no type-checking

```javascript
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

Remote calls with varying QoS

APIs are controlled by third-party
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:

  - 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…

They describe…

Excerpt of an OpenAPI Specification

**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

...
Overview of the approach

1. Extract request

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

2. Match request against specifications

3. Report results
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…)

- …
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

```json
{
    "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

---

**Procedure**

1. Find all specifications that match **base URL**
2. Check usage of **schema**
3. Select specifications that also match **path**
4. Select specifications that also match **method**
5. Determine whether URL contains **required query parameters***
6. Determine whether payload matches the required **schema***

```json
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

* If multiple spec-path-method triples match, consider least demanding requirement
System context: API Harmony - *find, learn about, and use web APIs*

Welcome to API Harmony!

**Find APIs:** API Harmony currently finds APIs.guru, from crawling the web and searching above!

**Learn about APIs:** API Harmony learns about other APIs in their applications!

**Use APIs:** API Harmony connects GitHub projects using it, and...

---

**Instagram**

*API Information*
- Characteristics & Relations
- StackOverflow questions
- Usages on GitHub
- Client Libraries on NPM

*Endpoints*
- GEOGRAPHIES
- LOCATIONS
- MEDIA
- COMMENTS
- LIKES
- TAGS
- USERS

*Characteristics & Relations*
- lat - Latitude of the center search coordinate. If used, 'lng' is required. (used 45 times - set to (latitude) 3 times [1][2][3])
- lng - Longitude of the center search coordinate. If used, 'lat' is required. (used 45 times - set to (longitude) 3 times [1][2][3])

*Request*

JavaScript + jQuery

```javascript
var settings = {
  "async": true,
  "crossdomain": true,
  "url": "https://api.instagram.com/v1/media/search?query=instagram&method="GET",
  "headers": {
    "access_token": "REPLACE_KEY_VALUE"
  }
};

$.ajax(settings).done(function (response) {
  console.log(response);
});
```

*Parameters*

**QUERY**
- lat - Latitude of the center search coordinate. If used, 'lng' is required. (used 45 times - set to (latitude) 3 times [1][2][3])
- lng - Longitude of the center search coordinate. If used, 'lat' is required. (used 45 times - set to (longitude) 3 times [1][2][3])

*Responses*

Found media resources (without likes information) in a given area.

**Top response fields**
- data (used 30 times)
- resposetext (used 5 times)
- data.length (used 3 times)

---

http://ibm.biz/apiharmony
System integration & developer work-flow

API Harmony

OpenAPI Specifications → Specification checker → Static code analysis

- GUI
- Specification checker
- Static code analysis

API search & recommendation

Edit code

Start dev

Source code

Report

Endpoint search & Snippets

API search & recommendation
References (1/3)


- [b] https://github.com/swagger-api/swagger-core
References (2/3)


References (3/3)