

API Salad

API <x>

- **Specification** — A technical standard, like OpenAPI 3.0, that defines how to describe an API's interface in a general
- **Description** — A file articulating a single API's interface that fulfills the requirements and expectations of an API specification.
- **Documentation** — The collection of things necessary to convey sufficient meaning so that clients can successfully use the deployed API through the interface. Includes, but not limited to, the API Description.

Approaches

- API-first
- Implementation-first

Spec: OpenAPI

- Describes both the service model and the request/response body based on JSON Schema
- Was called Swagger before it was donated to Linux Foundation
- Now Swagger is only used for tools from SwaggerHub

Spec: JSON Schema

- Describes an instance of JSON data
- OpenAPI is an “extended subset” of JSON Schema (perhaps will be fixed in OAS 3.1)
- Has additional tooling for client-side validations or HATEOAS support

Spec: JSON:API

- Anti-bikeshedding
- HATEOASish

```
GET /articles?page[number]=3&page[size]=1 HTTP/1.1
```

```
HTTP/1.1 200 OK
Content-Type: application/vnd.api+json

{
  "meta": {
    "totalPages": 13
  },
  "data": [
    {
      "type": "articles",
      "id": "3",
      "attributes": {
        "title": "JSON:API paints my bikeshed!",
        "body": "The shortest article. Ever.",
        "created": "2015-05-22T14:56:29.000Z",
        "updated": "2015-05-22T14:56:28.000Z"
      }
    }
  ],
  "links": {
    "self": "http://example.com/articles?page[number]=3&page[size]=1",
    "first": "http://example.com/articles?page[number]=1&page[size]=1",
    "prev": "http://example.com/articles?page[number]=2&page[size]=1",
    "next": "http://example.com/articles?page[number]=4&page[size]=1",
    "last": "http://example.com/articles?page[number]=13&page[size]=1"
  }
}
```

Spec: AsyncAPI

- OpenAPI for event-driven architectures

```
1  asyncapi: '2.0.0'
2  id: 'urn:com.my-application.example'
3  info:
4    title: Example
5    version: 0.1.0
6  channels:
7    user/signedup:
8      subscribe:
9        message:
10          description: An event describing that a user just signed up.
11          payload:
12            type: object
13            additionalProperties: false
14            properties:
15              fullName:
16                type: string
17              email:
18                type: string
19                format: email
20              age:
21                type: integer
22                minimum: 18
```

Spec: moar

- gRPC
- GraphQL
- OData
- ...

\$ref

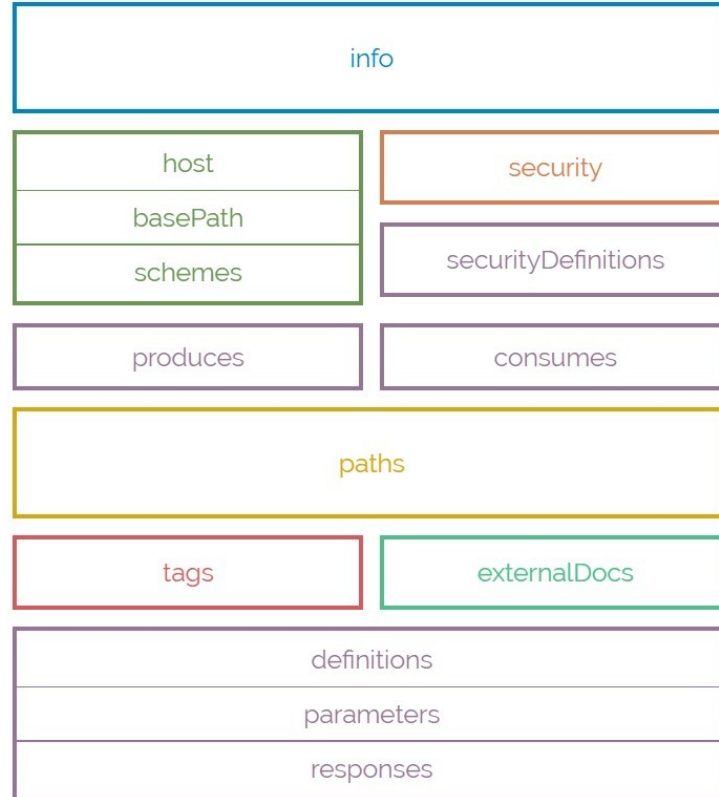
```
paths:  
  /users:  
    $ref: '../resources/users.yaml'  
  /users/{userId}:  
    $ref: '../resources/users-by-id.yaml'
```

Resolving \$ref

- **Lookup** — looking for value of \$ref, no changes to file
- **Bundling (External Inlining)** — replacing \$refs to external files with internal references
- **Dereferencing (Internal Inlining)** — replacing \$refs to external or internal data with actual data (no more \$refs)

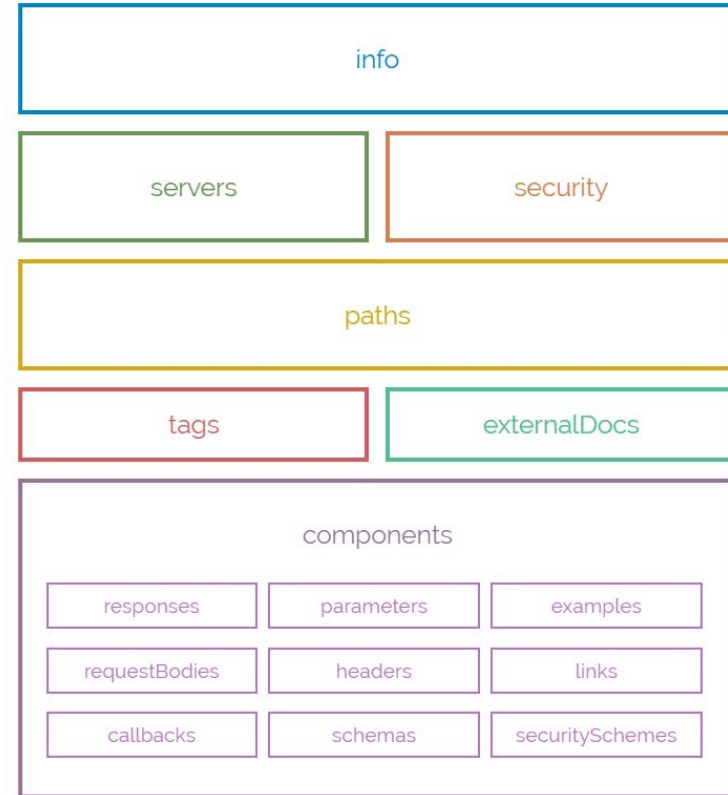
OAS 2 vs 3

Swagger 2



VS

OpenAPI 3



Server (Host)

- Can define several servers
- Path templating support:
 - url: `https://{subdomain}.site.com/{version}`
- Paths:
 - can have different base URLs (thanks to servers)
 - body for GET and DELETE is not allowed
 - TRACE support

Components

- Improves reusage and modularity
 - Existing: responses, parameters
 - Updated: schemas, schemas
 - New: examples, requestBodies, headers, links, callbacks
- Evolution of “definitions”:
 - `#/components/schemas/Pet`

Body

- Split into response and request bodies
- formData is merged with requestBody
- Content negotiation
- Supports cookies
- Supports wildcard codes: 4xx
- Callbacks!

Links

HATEOAS FTW

Various

- Security — improved, including support for multiple flows in OAuth 2
- Better documentation (examples for bodies, descriptions for paths)
- Parameters styles (arrays and objects)

Closer to JSON Schema

- oneOf
- anyOf
- nullable
- deprecated
- writeOnly
- not

P.S.

Hypermedia controls == HATEOAS
(but for humans)