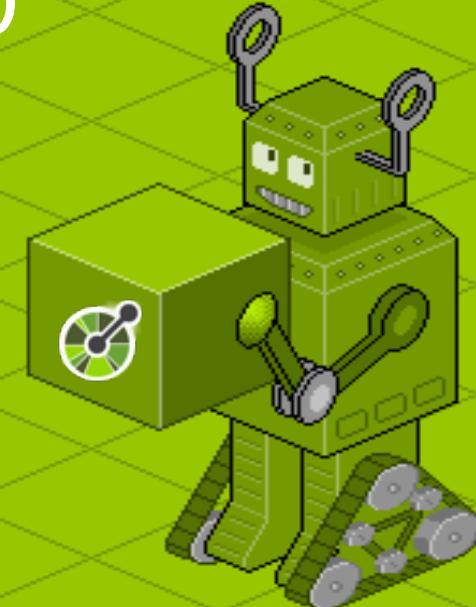
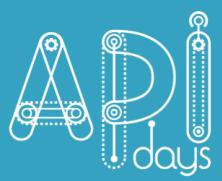


Giovanni Vigorelli
Software Specialist



Introduction to SWAGGER





Code of Conduct

Be respectful to one another

1. Respect yourself, and respect others.
2. Unwelcome behaviour will not be tolerated whether to other attendees, organisers, venue staff or other patrons of the APIDays NZ venues. Unwelcome behaviours include intimidation, harassment, abusive, discriminatory or derogatory conduct by any attendees of APIDays NZ and related events. All APIDays NZ venues may be shared with members of the public; please be respectful to all patrons of these locations.
3. Be careful. Please keep an eye out for dangerous activity or situations, and alert the conference team if you become aware of any safety hazards.

If we were at Amazon...

Everybody must use API (“Anyone who doesn't do this will be fired”)

We mustn't use PowerPoint

OpenAPI Specification (fka Swagger RESTful API Documentation Specification) Version 2.0 The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119. The Swagger specification is licensed under The Apache License, Version 2.0. Introductions Swagger™ is a project used to describe and document RESTful APIs. The Swagger specification defines a set of files required to describe such an API. These files can then be used by the Swagger-UI project to display the API and Swagger-Codegen to generate clients in various languages. Additional utilities can also take advantage of the resulting files, such as testing tools. Revision History Version Date Notes 2.0 2014-09-08 Release of Swagger 2.0 1.2 2014-03-14 Initial release of the formal document. 1.1 2012-08-22 Release of Swagger 1.1 1.0 2011-08-10 First release of the Swagger Specification Definitions Path Templating Path templating refers to the usage of curly braces ({}) to mark a section of a URL path as replaceable using path parameters. Mime Types Mime type definitions are spread across several resources. The mime type definitions should be in compliance with RFC 6838. Some examples of possible mime type definitions: text/plain; charset=utf-8 application/json application/vnd.github+json application/vnd.github.v3+json application/vnd.github.v3.raw+json application/vnd.github.v3.text+json application/vnd.github.v3.html+json application/vnd.github.v3.full+json application/vnd.github.v3.diff application/vnd.github.v3.patch HTTP Status Codes The HTTP Status Codes are used to indicate the status of the executed operation. The available status codes are described by RFC 7231 and in the IANA Status Code Registry. Specification Format The files describing the RESTful API in accordance with the Swagger specification are represented as JSON objects and conform to the JSON standards. YAML, being a superset of JSON, can be used as well to represent a Swagger specification file. For example, if a field is said to have an array value, the JSON array representation will be used: { "field" : [...] } While the API is described using JSON it does not impose a JSON input/output to the API itself. All field names in the specification are case sensitive. The schema exposes two types of fields. Fixed fields, which have a declared name, and Patterned fields, which declare a regex pattern for the field name. Patterned fields can have multiple occurrences as long as each has a unique name. File Structure The Swagger representation of the API is made of a single file. However, parts of the definitions can be split into separate files, at the discretion of the user. This is applicable for \$ref fields in the specification as follows from the JSON Schema definitions. By convention, the Swagger specification file is named swagger.json. Data Types Primitive data types in the Swagger Specification are based on the types supported by the JSON-Schema Draft 4. Models are described using the Schema Object which is a subset of JSON Schema Draft 4. An additional primitive data type "file" is used by the Parameter Object and the Response Object to set the parameter type or the response as being a file. Primitives have an optional modifier property format. Swagger uses several known formats to more finely define the data type being used. However, the format property is an open string-valued property, and can have any value to support documentation needs. Formats such as "email", "uuid", etc., can be used even though they are not defined by this specification. Types that are not accompanied by a format property follow their definition from the JSON Schema (except for file type which is defined above). The formats defined by the Swagger Specification are: Common Name type format Comments integer integer int32 signed 32 bits long integer int64 signed 64 bits float number float double number double string string byte string byte base64 encoded characters binary string binary any sequence of octets boolean boolean date string date-time As defined by full-date - RFC3339 date-time string date-time As defined by date-time - RFC3339 password string password Used to hint UIs the input needs to be obscured. Schema Swagger Object This is the root document object for the API specification. It combines what previously was the Resource Listing and API Declaration (version 1.2 and earlier) together into one document. Fixed Fields Field Name Type Description swagger string Required. Specifies the Swagger Specification version being used. It can be used by the Swagger UI and other clients to interpret the API listing. The value MUST be "2.0". info Info Object Required. Provides metadata about the API. The metadata can be used by the clients if needed. host string The host (name or ip) serving the API. This MUST be the host only and does not include the scheme nor sub-paths. It MAY include a port. If the host is not included, the host serving the documentation is to be used (including the port). The host does not support path templating. basePath string The base path on which the API is served, which is relative to the host. If it is not included, the API is served directly under the host. The value MUST start with a leading slash (/). The basePath does not support path templating. schemes [string] The transfer protocol of the API. Values MUST be from the list: "http", "https", "ws", "wss". If the schemes is not included, the default scheme to be used is the one used to access the Swagger definition itself. consumes [string] A list of MIME types the APIs can consume. This is global to all APIs but can be overridden on specific API calls. Value MUST be as described under Mime Types. produces [string] A list of MIME types the APIs can produce. This is global to all APIs but can be overridden on specific API calls. Value MUST be as described under Mime Types. paths Paths Object Required. The available paths and operations for the API. definitions Definitions Object An object to hold data types produced and consumed by operations. parameters Parameters Definitions Object An object to hold parameters that can be used across operations. This property does not define global parameters for all operations. responses Responses Definitions Object An object to hold responses that can be used across operations. This property does not define global responses for all operations. securityDefinitions Security Definitions Object Security scheme definitions that can be used across the specification. security [Security Requirement Object] A declaration of which security schemes are applied for the API as a whole. The list of values describes alternative security schemes that can be used (that is, there is a logical OR between the security requirements). Individual operations can override this definition. tags [Tag Object] A list of tags used by the specification with additional metadata. The order of the tags can be used to reflect on their order by the parsing tools. Not all tags that are used by the Operation Object must be declared. The tags that are not declared may be organized randomly or based on the tools' logic. Each tag name in the list MUST be unique. externalDocs External Documentation Object Additional external documentation. Patterned Objects Field Pattern Type Description ^x- Any Allows extensions to the Swagger Schema. The field name MUST begin with x-, for example, x-internal-id. The value can be null, a primitive, an array or an object. See Vendor Extensions for further details. Info Object The object provides metadata about the API. The metadata can be used by the clients if needed, and can be presented in the Swagger-UI for convenience. Fixed Fields Field Name Type Description title string Required. The title of the application. description string A short description of the application. GFM syntax can be used for rich text representation. termsOfService string The Terms of Service for the API. contact Contact Object The contact information for the exposed API. license License Object The license information for the exposed API. version string Required Provides the version of the application API (not to be confused with the specification version). Patterned Objects Field Pattern Type Description ^x- Any Allows extensions to the Swagger Schema. The field name MUST begin with x-, for example, x-internal-id. The value can be null, a primitive, an array or an object. See Vendor Extensions for further details. Info Object Example: { "title": "Swagger Sample App", "description": "This is a sample server Petstore server.", "termsOfService": "http://swagger.io/terms/", "contact": { "name": "API Support", "url": "http://www.swagger.io/support", "email": "support@swagger.io" }, "license": { "name": "Apache 2.0", "url": "http://www.apache.org/licenses/LICENSE-2.0.html" }, "version": "1.0.1" } title: Swagger Sample App description: This is a sample server Petstore server. termsOfService: http://swagger.io/terms/ contact: name: API Support url: http://www.swagger.io/support email: support@swagger.io license: name: Apache 2.0 url: http://www.apache.org/licenses/LICENSE-2.0.html version: 1.0.1 Contact Object Contact information for the exposed API. Fixed Fields Field Name Type Description name string The identifying name of the contact person/organization. url string The URL pointing to the contact information. MUST be in the format of a URL. email string The email address of the contact person/organization. MUST be in the format of an email address. Patterned Objects Field Pattern Type Description ^x- Any Allows extensions to the Swagger Schema. The field name MUST begin with x-, for example, x-internal-id. The value can be null, a primitive, an array or an object. See Vendor Extensions for further details. Contact Object Example: { "name": "API Support", "url": "http://www.swagger.io/support", "email": "support@swagger.io" } name: API Support url: http://www.swagger.io/support email: support@swagger.io License Object License information for the exposed API. Fixed Fields Field Name Type Description name string Required. The license name used for the API. url string A URL to the license used for the API. MUST be in the format of a URL. Patterned Objects Field Pattern Type Description ^x- Any Allows extensions to the Swagger Schema. The field name MUST begin with x-, for example, x-internal-id. The value can be null, a primitive, an array or an object. See Vendor Extensions for further details. License Object Example: { "name": "Apache 2.0", "url": "http://www.apache.org/licenses/LICENSE-2.0.html" } name: Apache 2.0 url: http://www.apache.org/licenses/LICENSE-2.0.html Paths Object Holds the relative paths to the individual endpoints. The path is appended to the basePath in order to construct the full URL. The Paths may be empty, due to ACL constraints. Patterned Fields Field Pattern Type Description /{path} Path Item Object A relative path to an individual endpoint. The field name MUST begin with a slash. The path is appended to the basePath in order to construct the full URL. Path templating is allowed. ^x- Any Allows extensions to the Swagger Schema. The field name MUST begin with x-, for example, x-internal-id. The value can be null, a primitive, an array or an object. See Vendor Extensions for further details. Paths Object Example { "/pets": { "get": { "description": "Returns all pets from the system that the user has access to", "produces": ["application/json"], "responses": { "200": { "description": "A list of pets.", "schema": { "type": "array", "items": { "\$ref": "#/definitions/pet" } } } } } } /pets: get: . It combines what previously was the Resource Listing and API Declaration (version 1.2 and earlier) together into one document. Fixed Fields Field Name Type Description swagger string Required. Specifies the Swagger Specification version being used. It can be used by the Swagger UI and other clients to interpret the API listing. The value MUST be "2.0". info Info Object Required. Provides metadata about the API. The metadata can be used by the clients if needed. host string The host (name or ip) serving the API. This MUST be the host only and does not include the scheme nor sub-paths. It MAY include a port. If the host is not included, the host serving the documentation is to be used (including the port). The host does not support path templating. basePath string The base path on which the API is served, which is relative to the host. If it is not included, the API is served directly under the host. The value MUST start with a leading slash (/). The basePath does not support path templating. schemes [string] The transfer protocol of the API. Values MUST be from the list: "http", "https", "ws", "wss". If the schemes is not included, the default scheme to be used is the one used to access the Swagger definition itself. consumes [string] A list of MIME types the APIs can consume. This is global to all APIs but can be overridden on specific API calls. Value MUST be as described under Mime Types. produces [string] A list of MIME types the APIs can produce. This is global to all APIs but can be overridden on specific API calls. Value MUST be as described under Mime Types. paths Paths Object Required. The available paths and operations for the API. definitions Definitions Object An object to hold data types produced and consumed by operations. parameters Parameters Definitions Object An object to hold parameters that can be used across operations. This property does not define global parameters for all operations. responses Responses Definitions Object An object to hold responses that can be used across operations. This property does not define global responses for all operations. securityDefinitions Security Definitions Object Security scheme definitions that can be used across the specification. security [Security Requirement Object] A declaration of which security schemes are applied for the API as a whole. The list of values describes alternative security schemes that can be used (that is, there is a logical OR between the security requirements). Individual operations can override this definition. tags [Tag Object] A list of tags used by the specification with additional metadata. The order of the tags can be used to reflect on their order by the parsing tools. Not all tags that are used by the Operation Object must be declared. The tags that are not declared may be organized randomly or based on the tools' logic. Each tag name in the list MUST be unique . It combines what previously was the Resource Listing and API Declaration (version 1.2 and earlier) together into one document. Fixed Fields Field Name Type Description swagger string Required. Specifies the Swagger Specification version being used. It can be used by the Swagger UI and other clients to interpret the API listing. The value MUST be "2.0". info Info Object Required. Provides metadata about the API. The metadata can be used by the clients if needed. host string The host (name or ip) serving the API. This MUST be the host only and does not include the scheme nor sub-paths. It MAY include a port. If the host is not included, the host serving the documentation is to be used (including the port). The host does not support path templating. basePath string The base path on which the API is served, which is relative to the host. If it is not included, the API is served directly under the host. The value MUST start with a leading slash (/). The basePath does not support path templating. schemes [string] The transfer protocol of the API. Values MUST be from the list: "http", "https", "ws", "wss". If the schemes is not included, the default

A bit of history

We should always learn from past

1462 - First specs released by Leonardo (“The Da Vinci code”)

2010 - Wordnik started (again) as internal project

2015 - SmartBear (maintainer with other founding members such as Google, IBM and Microsoft) created a new organization called Open API initiative.

2016 - Swagger renamed to Open API Specification and moved to github.



2016 – OpenApi v3 working in progress

Can we start without a definition? (Swagger)

- ④ Rumors say that Swagger was the answer to WADL (walking like a duck)
- ④ How someone presents him or her self to the world. Swagger is shown from how the person handles a situation. It can also be shown in the person's walk.
 - ④ A person's style- they way they walk, talk, dress.
"I'm loving his swagger!"



<http://www.urbandictionary.com/define.php?term=swagger>

And now?
(Open API)



It's all about making themselves visible

REST Web API is like browsing a web site with your eyes closed!

In the ‘old days’ or in ‘the other room’ there’s WSDL (SOAP web services), complex and ugly as you like, but up for the job

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="AktienKurs"
  targetNamespace="http://loc
  xmlns:xsd="http://schemas.xmlsoap.org/wsd
  xmlns="http://schemas.xmlsoap.org/wsd
  <service name="AktienKurs">
    <port name="AktienSoapPort" binding
      <soap:address location="http://loc
    </port>
    <message name="Aktie.HoleWert">
      <part name="body" element="xsd:Tra
    </message>
  </service>
</definitions>
```

Which problems it tries (successfully) to solve



API discovery

API readability

Development speed up
- client / server decoupling
- creating API from code
- generate code from API

Remove barriers
between Business
(product owners) and IT
(developers)

API definition (discovery)

```
swagger: "2.0"

info:
  version: 0.0.1
  title: session rating
  description: set of api to rate and get ratings of API Days session
  termsOfService: totally free
  contact:
    name: jg
    url: jg.co.nz
    email: g@jg.co.nz
  license:
    name: MIT
    url: http://opensource.org/licenses/MIT

schemes:
  - https
host: jg.co.nz
basePath: /api

paths: {}
```

swagger

info

schemes

host

basePath

paths

(JSON or YAML, you chose)

API definition (discovery)

```
paths:  
  /ratings:  
    get:  
      summary: get some ratings  
      description: returns a list of all ratings unless differently requested (paging)  
      parameters:  
        - name: pageSize  
          in: query  
          description: number of ratings returned  
          type: integer  
        - name: pageNumber  
          in: query  
          description: page number  
          type: integer  
      responses:  
        200:  
          description: list of ratings  
          schema:  
            type: array  
            items:  
              required:  
                - rating  
                - session  
              properties:  
                rating:  
                  type: integer  
                session:  
                  type: string  
                user:  
                  type: string
```

paths

```
  /ratings/{session}:  
    get:  
      summary: get a session'rating  
      description: retrieve the rating of a given session  
      parameters:  
        - name: session  
          in: path  
          required: true  
          description: the title of the session  
          type: string  
      responses:  
        200:  
          description: a rating  
          schema:  
            required:  
              - rating  
              - session  
            properties:  
              rating:  
                type: integer  
              session:  
                type: string  
              user:  
                type: string  
        404:  
          description: Session not found
```

resources

methods

parameters

responses

API definition (discovery)

```
definitions:  
  Rating:  
    required:  
      - rating  
      - session  
    properties:  
      rating:  
        type: integer  
      session:  
        type: string  
      user:  
        type: string  
  
  Ratings:  
    type: array  
    items:  
      $ref: "#/definitions/Rating"
```

definitions

```
paths:  
  /ratings:  
    get:  
      summary: get some ratings  
      description: returns a list of all ratings unless differently requested (paging)  
      parameters:  
        - name: pageSize  
          in: query  
          description: number of ratings returned  
          type: integer  
        - name: pageNumber  
          in: query  
          description: page number  
          type: integer  
      responses:  
        200:  
          description: list of ratings  
          schema:  
            type: array  
            items:  
              required:  
                - rating  
                - session  
              properties:  
                rating:  
                  type: integer  
                session:  
                  type: string  
                user:  
                  type: string
```



```
paths:  
  /ratings:  
    get:  
      summary: get some ratings  
      description: returns a list of all ratings unless differently requested (paging)  
      parameters:  
        - name: pageSize  
          in: query  
          description: number of ratings returned  
          type: integer  
        - name: pageNumber  
          in: query  
          description: page number  
          type: integer  
      responses:  
        200:  
          description: list of ratings  
          schema:  
            $ref: "#/definitions/Ratings"
```

API definition (discovery)

```
responses:  
  DefaultErrorResponse:  
    description: generic error occurred  
    schema:  
      $ref: "#/definitions/Error"
```

responses

```
definitions:  
  Rating:  
    required:  
      - rating  
      - session  
    properties:  
      rating:  
        type: integer  
      session:  
        type: string  
      user:  
        type: string  
  
  Ratings:  
    type: array  
    items:  
      $ref: "#/definitions/Rating"  
  
  Error:  
    properties:  
      code:  
        type: string  
      message:  
        type: string
```

```
/ratings/{session}:  
get:  
  summary: get a session'rating  
  description: retrieve the rating of a given session  
  parameters:  
    - name: session  
      in: path  
      required: true  
      description: the title of the session  
      type: string  
  responses:  
    200:  
      description: a rating  
      schema:  
        required:  
          - rating  
          - session  
        properties:  
          rating:  
            type: integer  
          session:  
            type: string  
          user:  
            type: string  
    404:  
      description: Session not found
```



```
/ratings/{session}:  
get:  
  summary: get a session'rating  
  description: retrieve the rating of a given session  
  parameters:  
    - name: session  
      in: path  
      required: true  
      description: the title of the session  
      type: string  
  responses:  
    200:  
      description: a rating  
      schema:  
        $ref: "#/definitions/Rating"  
    404:  
      description: Session not found  
    500:  
      $ref: "#/responses/DefaultErrorResponse"
```

API definition (discovery)

```
parameters:  
  PageSize:  
    name: pageSize  
    in: query  
    description: number of ratings returned  
    type: integer  
  PageNumber:  
    name: pageNumber  
    in: query  
    description: page number  
    type: integer  
  Session:  
    name: session  
    in: path  
    required: true  
    description: the title of the session  
    type: string
```

parameters

```
paths:  
  /ratings:  
    get:  
      summary: get some ratings  
      description: returns a list of all ratings unless differently requested (paging)  
      parameters:  
        - name: pageSize  
          in: query  
          description: number of ratings returned  
          type: integer  
        - name: pageNumber  
          in: query  
          description: page number  
          type: integer  
      responses:  
        200:  
          description: list of ratings  
          schema:  
            type: array  
            items:  
              required:  
                - rating  
                - session  
            properties:  
              rating:  
                type: integer  
              session:  
                type: string  
              user:  
                type: string
```



```
paths:  
  /ratings:  
    get:  
      summary: get some ratings  
      description: returns a list of all ratings unless differently requested (paging)  
      parameters:  
        - $ref: "#/parameters/PageSize"  
        - $ref: "#/parameters/PageNumber"  
      responses:  
        200:  
          description: list of ratings  
          schema:  
            $ref: "#/definitions/Ratings"
```

API definition (discovery)

```
parameters:  
  PageSize:  
    name: pageSize  
    in: query  
    description: Number of persons returned  
    type: integer  
    format: int32  
    minimum: 0  
    exclusiveMinimum: true  
    maximum: 100  
    exclusiveMaximum: false  
    multipleOf: 10
```

Advanced type
definition

```
paths:  
  /ratings:  
    get:  
      summary: get some ratings  
      description: returns a list of all ratings unless differently requested (paging)  
      parameters:  
        - name: pageSize  
          in: query  
          description: number of ratings returned  
          type: integer  
        - name: pageNumber  
          in: query  
          description: page number  
          type: integer  
      responses:  
        200:  
          description: list of ratings  
          schema:  
            type: array  
            items:  
              required:  
                - rating  
                - session  
            properties:  
              rating:  
                type: integer  
              session:  
                type: string  
              user:  
                type: string
```



```
paths:  
  /ratings:  
    get:  
      summary: get some ratings  
      description: returns a list of all ratings unless differently requested (paging)  
      parameters:  
        - $ref: "#/parameters/PageSize"  
        - $ref: "#/parameters/PageNumber"  
      responses:  
        200:  
          description: list of ratings  
          schema:  
            $ref: "#/definitions/Ratings"
```

API definition (discovery)

```
securityDefinitions:
  BasicSecurity:
    type: basic
  KeySecurity:
    type: apiKey
    in: header
    name: apiKeyAdmin

  security:
    - BasicSecurity: □

paths: |
  /ratings:
    get:
      summary: get some ratings
      description: returns a list of all ratings unless differently requested (paging)
      parameters:
        - $ref: "#/parameters/PageSize"
        - $ref: "#/parameters/PageNumber"
      responses:
        200:
          description: list of ratings
          schema:
            $ref: "#/definitions/Ratings"
        500:
          $ref: "#/responses/DefaultErrorResponse"

    post:
      summary: create a rating
      description: add a new rating to the rating list
      security:
        - KeySecurity: □
      parameters:
        - name: rating
          in: body
          description: the rating to be created
          schema:
            $ref: "#/definitions/Rating"
      responses:
```

securityDefinitions

security

API definition (discovery)

```
x-ibm-configuration:  
assembly:  
execute:  
- map:  
    title: 'GetMortgagePayment: input'  
    inputs: {}  
    outputs:  
    body:  
        schema:  
            $ref: '#/x-ibm-configuration/targets/Mortgage/definitions/GetMortgagePaymentInput'  
            variable: message.body  
            content: application/xml  
    content-type:  
        schema:  
            type: string  
            variable: message.headers.content-type  
    SOAPAction:  
        schema:  
            type: string  
            variable: message.headers.SOAPAction  
    actions:  
    - set: content-type  
      default: text/xml  
    - set: SOAPAction  
      default: 'http://www.webserviceX.NET/GetMortgagePayment'  
- invoke:  
    title: 'GetMortgagePayment: invoke'  
    target-url: 'http://www.webserviceX.net/mortgage.asmx'  
    timeout: 60  
    verb: POST  
    cache-response: protocol  
    cache-ttl: 900  
    output: GetMortgagePaymentOut  
- map:  
    title: 'GetMortgagePayment: output'  
    inputs:  
    input:  
        schema:  
            $ref: '#/x-ibm-configuration/targets/Mortgage/definitions/GetMortgagePaymentOutput'  
            variable: GetMortgagePaymentOut.body  
            content: application/xml  
    outputs: {}  
    actions: []  
gateway: datapower-gateway
```

extensions

Remove Business – Development barriers

Bottom up

Marketing a product

Externalize IT assets

Leverage IT legacy
(or build new)



Top Down

Identify commercial products

Design the API

Implement the API logic

Leverage IT legacy
(or build new)



API readability (documentation)

The screenshot shows two main views of the Swagger tool. On the left is the 'editor.swagger.io' interface, displaying the Uber API specification in YAML. The right side shows the 'swagger' UI for the Petstore API.

Uber API (YAML Editor):

```
1 # this is an example of the Uber API
2 # as a demonstration of an API spec in YAML
3 swagger: '2.0'
4 info:
5   title: Uber API
6   description: Move your app forward with the Uber API
7   version: "1.0.0"
8   host: api.uber.com
9   schemes:
10      - https
11      # will be prefixed to all paths
12 basePath: /v1
13 produces:
14   - application/json
15 paths:
16   /products:
17     get:
18       summary: Product Types
19       description: |
20         The Products endpoint returns information about the *Uber* products offered at a given location. The response includes the display name and other details about each product, and lists the products in the proper display order.
21       parameters:
22         - name: latitude
23           in: query
24           description: Latitude component of location.
25           required: true
26           type: number
27           format: double
28         - name: longitude
29           in: query
30           description: Longitude component of location.
31           required: true
32           type: number
33           format: double
34       tags:
35         - Products
36       responses:
37         200:
38           description: An array of products
39           schema:
40             type: array
41             items:
42               $ref: '#/definitions/ProductType'
43             properties:
44               id:
45                 type: integer
```

Uber API (Swagger UI):

Move your app forward with the Uber API
Version 1.0.0
Filter operations by a tag:

Paths

- /products
 - GET /products**
 - Summary**
Product Types
 - Description**
The Products endpoint returns information about the *Uber* products offered at a given location. The response includes the display name and other details about each product, and lists the products in the proper display order.
 - Parameters**

Name	Located in	Description
latitude	query	Latitude component of location.
longitude	query	Longitude component of location.
 - Responses**

Swagger Petstore (UI):

This is a sample server Petstore server. You can find out more about Swagger at <http://swagger.io> or on [irc.freenode.net/#swagger](#). For this sample, you can use the api key `special-key` to test the authorization filters.

pet : Everything about your Pets

Method	Path	Description
POST	/pet	Add a new pet to the store
PUT	/pet	Update an existing pet
GET	/pet/findByStatus	Finds Pets by status
GET	/pet/findByTags	Finds Pets by tags
DELETE	/pet/{petId}	Deletes a pet
GET	/pet/{petId}	Find pet by ID

Implementation Notes
Returns a single pet

Response Class (Status 200)

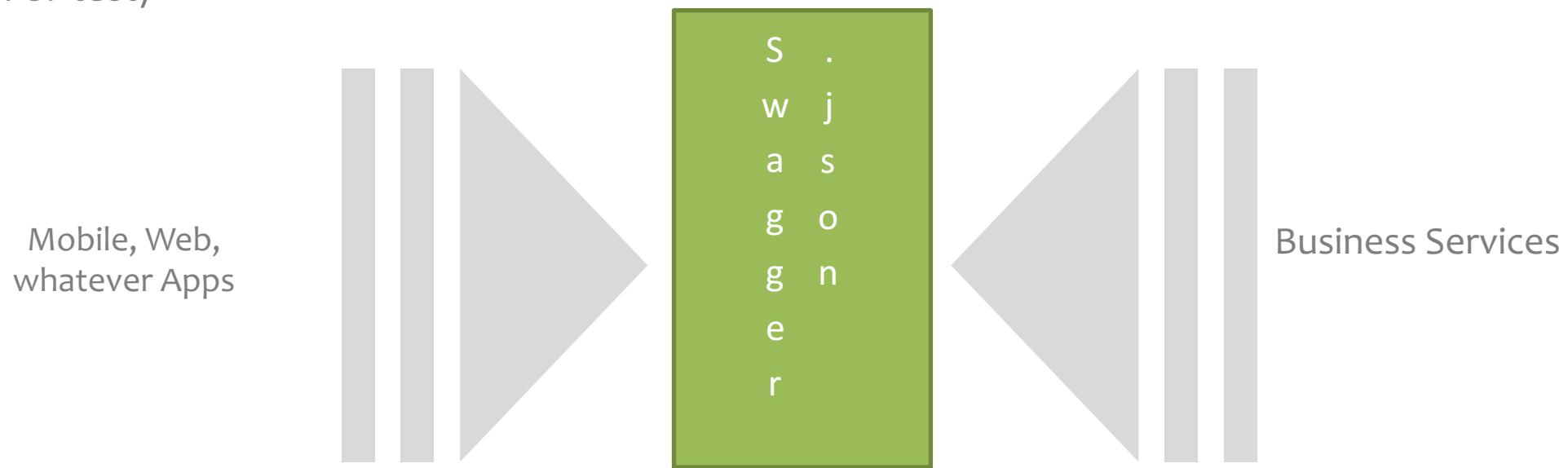
Curl

Swagger tools (editor, UI) help to define a standard for REST API documentation.

No more documentation cowboys !

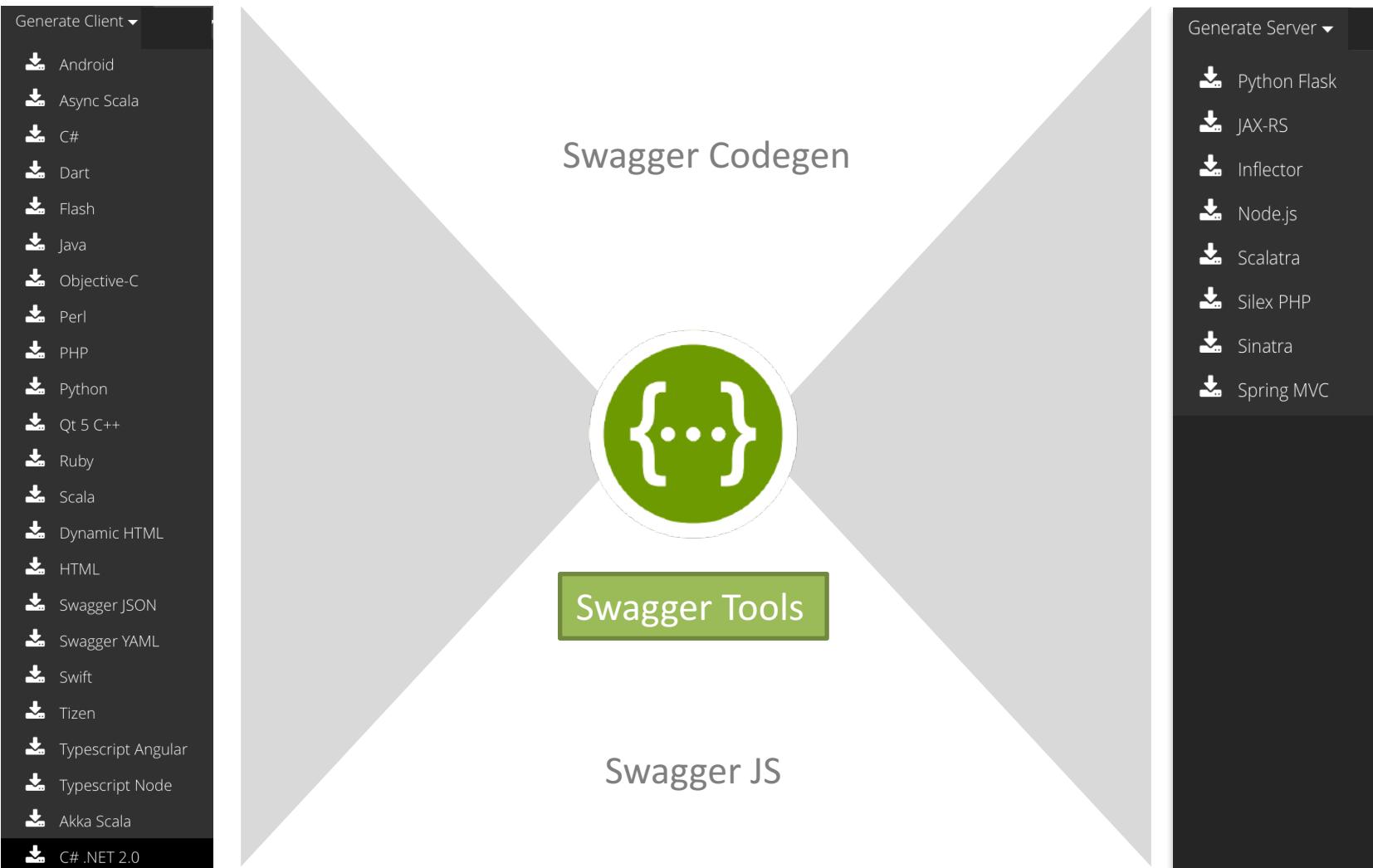
Dev. speed up – client and server decoupling

API definition allows **client** side developers
to create client against swagger definition
(codgen or test)

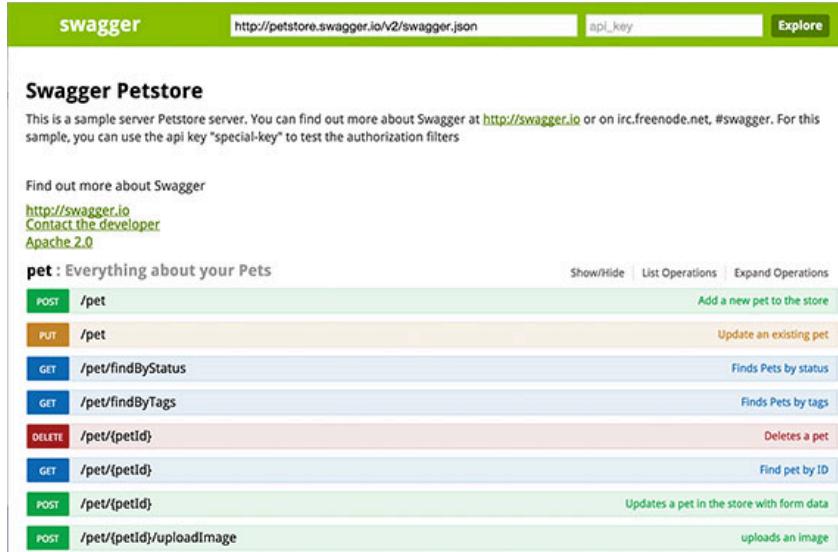


API definition allows **server** side
developers implement the logic to
server API resources

Dev. speed up – Code Generation



Dev. speed up – API Generation



The screenshot shows the Swagger Petstore API documentation. At the top, there's a navigation bar with tabs for 'swagger' (selected), 'http://petstore.swagger.io/v2/swagger.json', 'api_key', and 'Explore'. Below the navigation is a section titled 'Swagger Petstore' with a brief description. It includes links to 'Find out more about Swagger', 'http://swagger.io', 'Contact the developer', and 'Apache 2.0'. The main content area is titled 'pet : Everything about your Pets' and lists various API operations:

Method	Path	Description
POST	/pet	Add a new pet to the store
PUT	/pet	Update an existing pet
GET	/pet/findByStatus	Finds Pets by status
GET	/pet/findByTags	Finds Pets by tags
DELETE	/pet/{petId}	Deletes a pet
GET	/pet/{petId}	Find pet by ID
POST	/pet/{petId}	Updates a pet in the store with form data
POST	/pet/{petId}/uploadImage	uploads an image



JAX-RS

```
@Path("/pet")
@Api(value = "pet", authorizations = {
    @Authorization(value="sampleoauth", scopes = {})
})
@Produces({"application/json", "application/xml"})
public class PetResource {
    ...
}
```



100% open source, loved by big players



NETFLIX

intuit

akana



Restlet

IBM®

The 3scale logo features a circular pattern of dots in grey, orange, and black, followed by the word "3scale" in a grey sans-serif font.

WSO2

The Microsoft logo consists of four colored squares (red, green, blue, yellow) in a 2x2 grid, followed by the word "Microsoft" in a grey sans-serif font.

apinf

Google
apigee

The amazon.com logo features the word "amazon" in black with a yellow smiley arrow underneath, followed by ".com" in a smaller black font.

The ARTIK Cloud logo features a blue hexagonal icon above the text "ARTIK™ Cloud".

The mashape logo features a cartoon monkey head icon followed by the word "mashape" in a green sans-serif font.

100% open source, loved by big players



NETFLIX



Restlet



apiary

intuit

akana



Google

apigee



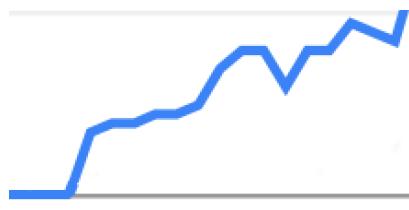
amazon.com®



Spec popularity ...



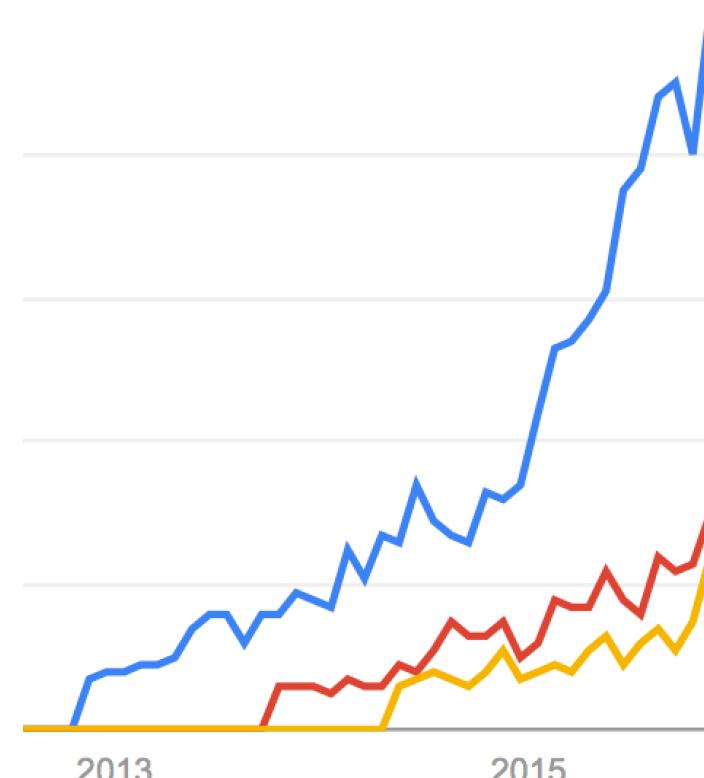
RAML api



Swagger api



Blueprint api



... 1 winner!
(if it was a race)

**ONE DOES NOT
SIMPLY**

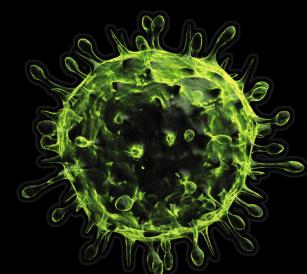
SWAGGER

memegenerator.net

RAML

RESTful API Modeling Language

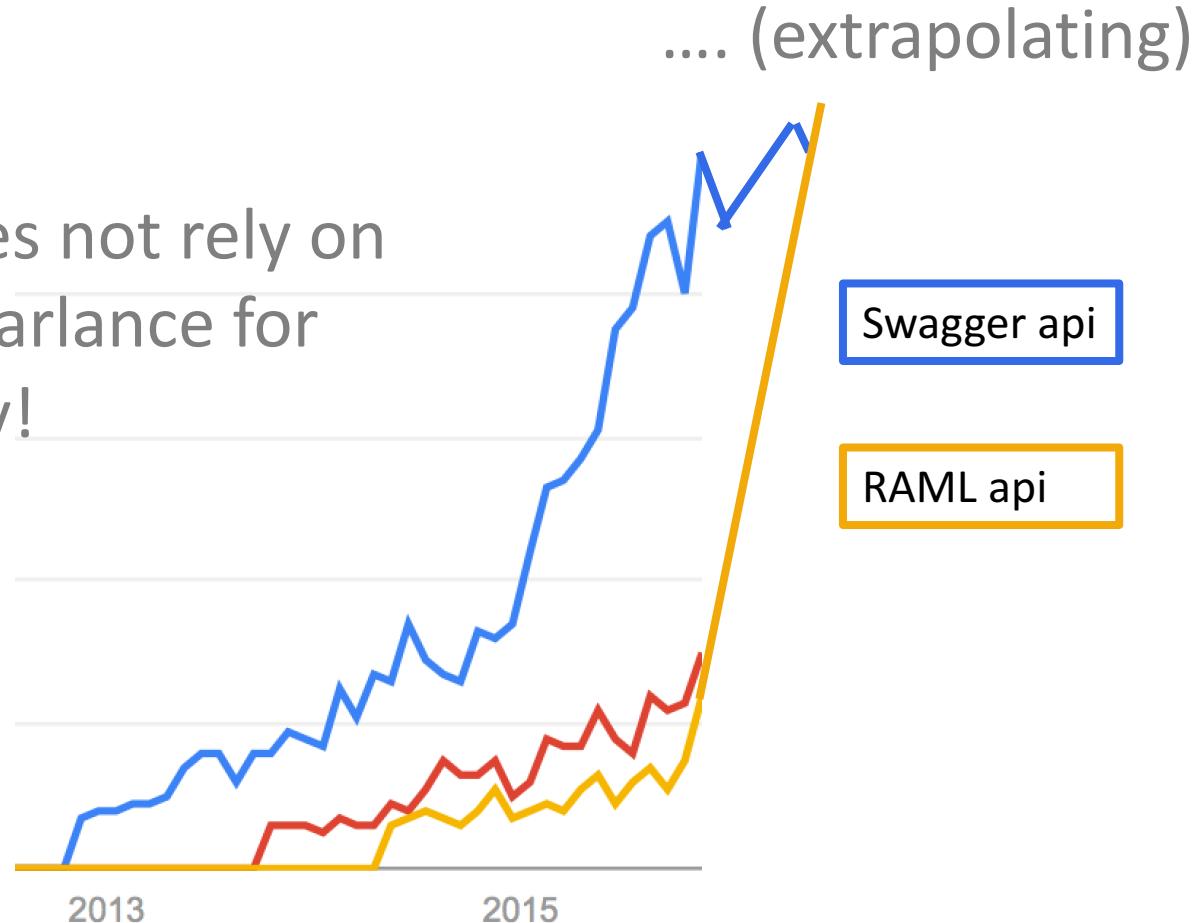
raml.org



Is RAML better?



Yes! ...
RAML does not rely on
hip-hop parlance for
popularity!



Is RAML better?

Jokes aside ...

it depends ...



RAML

DESIGN

BUILD

TEST

DOCUMENT

SHARE

API administration My Account - v1.0 Designer

+ New File New Folder Save Import Export files Help Mocking Service ON API is behind a firewall (?) Collapse All

examples schemas traits api.raml

```

1  #%RAML 0.8
2  baseUri: http://mocksvc.mulesoft.com/mocks/5265e198-45d4-4310-a3cc-0a7e8e6dedf1/{organisation}
3  title: My Account
4  version: v1.0
5  #commented this out because the mock services don't work with them but will use in the implementation
6  baseUriParameters:
7    organisation:
8      description: The short code for the organisation currently one 'GENE' for Genesis energy is supported
9
10 traits:
11   - !include traits/paging.yaml
12
13 accounts:
14   displayName: Billing Accounts
15   description: The account held in the billing systems
16   /{account_no}:
17     displayName: Account Summary
18     description: Gets basic/summary details for a users account. The account number is the billing record identifier (also known master consumer number) used for the given brand (Genesis or other brands like EOL). Currently only "GENE" for Genesis energy is supported.
19     get:
20       description: Gets key account information for the consumer
21       responses:
22         200:
23           body:
24             application/json:
25               example: !include examples/accountSummary.json
26               schema: !include schemas/accountSchema.json
27         400:
28           body:
29             application/json:
30               example: |
31                 {
32                   "status": "Failure",
33                   "code": 400,
34                   "message": "Bad request"
35                 }
36         403:
37           body:
38             application/json:
39               example: |

```

METHODS (9)

- connect
- delete
- get
- head
- options
- patch
- post
- put

PARAMETERS (2)

- baseUriParameters
- uriParameters

SECURITY (1)

- securedBy

RESOURCES (1)

- New Resource

TRAITS AND TYPES (2)

- is
- type

Resources

/accounts

The account held in the billing systems

/accounts/{account_no}

Gets basic/summary details for a users account. The account number is the billing record identifier (also known master consumer number) used for the given brand (Genesis or other brands like EOL). Currently only "GENE" for Genesis energy is supported.

GET

/accounts/{account_no}/contact

This provides the contact details for a given account.

GET

/accounts/{account_no}/invoices

List of invoices over the given period for the account.

GET

/accounts/{account_no}/payments

Payments made to an account

POST GET

/accounts/{account_no}/payments/calculate-fee

Calculates the 'convenience fee' for credit card payments

POST

/accounts/{account_no}/transactions

Transactions for the account

GET

/accounts/{account_no}/services/{service_id}

Services Related to an account

GET

/accounts/{account_no}/services/{service_id}/usage

Productive Dev Environment

API administration My Account - v1.0 Designer

+ New File New Folder Save Import Export files Help Mocking Service ON API is behind a firewall (?) Collapse All

examples schemas traits api.raml

Well Structured Code

```

1  #%RAML 0.8
2  baseUri: http://mocksvc.mulesoft.com/mocks/5265e198-45d4-4310-a3cc-0a7e8e6dedf1/{organisation}
3  title: My Account
4  version: v1.0
5  #commented this out because the mock services don't work with them but will use in the implementation
6  baseUriParameters:
7    organisation:
8      description: The short code for the organisation currently one 'GENE' for Genesis energy is supported
9
10 traits:
11   - !include traits/paging.yaml
12
13 /accounts:
14   displayName: Billing Accounts
15   description: The account held in the billing systems
16   /{account_no}:
17     displayName: Account Summary
18     description: Gets basic/summary details for a users account. The account number is the billing record
19     identifier (also known master consumer number) used for the given brand (Genesis or other brands like
20     EOL). Currently only "GENE" for Genesis energy is supported.
21     get:
22       description: Gets key account information for the consumer
23       responses:
24         200:
25           body:
26             application/json:
27               example: !include examples/accountSummary.json
28               schema: !include schemas/accountSchema.json
29         400:
29           body:
29             application/json:
30               example: |
31                 {
32                   "status": "Failure",
33                   "code": 400,
34                   "message": "Bad request"
35                 }
36         403:
37           body:
38             application/json:
39               example: |

```

METHODS (9) PARAMETERS (2) SECURITY (1) RESOURCES (1) TRAITS AND TYPES (2)

- connect delete
- baseUriParameters
- securedBy
- New Resource
- is

- get head
- uriParameters
-
-
- type

- options patch
-
-
-
-

- post put
-
-
-
-

Productive Dev Environment

Resources

/accounts

The account held in the billing systems

/accounts/{account_no}

Gets basic/summary details for a users account. The account number is the billing record identifier (also known master consumer number) used for the given brand (Genesis or other brands like EOL). Currently only "GENE" for Genesis energy is supported.

GET

/accounts/{account_no}/contact

This provides the contact details for a given account.

GET

/accounts/{account_no}/invoices

List of invoices over the given period for the account.

GET

/accounts/{account_no}/payments

Payments made to an account

POST GET

/accounts/{account_no}/payments/calculate-fee

Calculates the 'convenience fee' for credit card payments

POST

/accounts/{account_no}/transactions

Transactions for the account

GET

/accounts/{account_no}/services/{service_id}

Services Related to an account

GET

/accounts/{account_no}/services/{service_id}/usage

API administration My Account - v1.0 Designer

+ New File New Folder Save Import Export files Help Mocking Service ON API is behind a firewall (?) Collapse All

examples schemas traits api.raml

#%RAML 0.8
baseUri: http://mocksvc.mulesoft.com/mocks/5265e198-45d4-4310-a3cc-0a7e8e6dedf1/{organisation}
title: My Account
version: v1.0
#commented this out because the mock services don't work with them but will use in the implementation
baseUriParameters:
organisation:
description: The short code for the organisation currently one 'GENE' for Genesis energy is supported
traits:
- !include traits/paging.yaml
/accounts:
displayName: Billing Accounts
description: The account held in the billing systems
/account_no:
displayName: Account Summary
description: Gets basic/summary details for a users account. The account number is the billing record identifier (also known master consumer number) used for the given brand (Genesis or other brands like EOL). Currently only "GENE" for Genesis energy is supported.
get:
description: Gets key account information for the consumer
responses:
200:
body:
application/json:
example: !include examples/accountSummary.json
schema: !include schemas/accountSchema.json
400:
body:
application/json:
example: |
{
"status": "Failure",
"code": 400,
"message": "Bad request"
}
403:
body:
application/json:
example: |

Well Structured Code

Productive Dev Environment

Resources

/accounts
The account held in the billing systems

/accounts/{account_no}
Gets basic/summary details for a users account. The account number is the billing record identifier (also known master consumer number) used for the given brand (Genesis or other brands like EOL). Currently only "GENE" for Genesis energy is supported.

GET

/accounts/{account_no}/contact
This provides the contact details for the account.

GET

/accounts/{account_no}/invoices
List of invoices over the given period for the account.

GET

/accounts/{account_no}/payments
Payments made to an account

POST GET

/accounts/{account_no}/payments/calculate-fee
Calculates the 'convenience fee' for credit card payments

POST

/accounts/{account_no}/transactions
Transactions for the account

GET

/accounts/{account_no}/services/{service_id}
Services Related to an account

GET

/accounts/{account_no}/services/{service_id}/usage

Navigable Documentation

API administration My Account - v1.0 Designer

+ New File New Folder Save Import Export files Help Mocking Service ON API is behind a firewall (?)

examples schemas traits api.raml

```

1  #%RAML 0.8
2  baseUri: http://mocksvc.mulesoft.com/mocks/5265e198-45d4-4310-a3cc-0a7e8e6dedf1/{organisation}
3  title: My Account
4  version: v1.0
5  #commented this out because the mock services don't work with them but will use in the implementation
6  baseUriParameters:
7    organisation:
8      description: The short code for the organisation currently one 'GENE' for Genesis energy is supported
9
10 traits:
11   - !include traits/paging.yaml
12
13 /accounts:
14   displayName: Billing Accounts
15   description: The account held in the billing systems
16   /account_no:
17     displayName: Account Summary
18     description: Gets basic/summary details for a users account. The account number is the billing record identifier (also known master consumer number) used for the given brand (Genesis or other brands like EOL). Currently only "GENE" for Genesis energy is supported.
19     get:
20       description: Gets key account information for the consumer
21       responses:
22         200:
23           body:
24             application/json:
25               example: !include examples/accountSummary.json
26               schema: !include schemas/accountSchema.json
27         400:
28           body:
29             application/json:
30               example: |
31                 {
32                   "status": "Failure",
33                   "code": 400,
34                   "message": "Bad request"
35                 }
36         403:
37           body:
38             application/json:
39               example: |

```

METHODS (9) PARAMETERS (2) SECURITY (1) RESOURCES (1) TRAITS AND TYPES (2)

- connect delete
- baseUriParameters
- securedBy
- New Resource
- is

- get head
- uriParameters
-
-
- type

- options patch
-
-
-
-

- post put
-
-
-
-

Resources

/accounts The account held in the billing systems

/accounts/{account_no}

Gets basic/summary details for a users account. The account number is the billing record identifier (also known master consumer number) used for the given brand (Genesis or other brands like EOL). Currently only "GENE" for Genesis energy is supported.

GET

/accounts/{account_no}/contact

This provides the contact details for the account.

GET

/accounts/{account_no}/invoices

List of invoices over the given period for the account.

GET

/accounts/{account_no}/payments

Payments made to an account

POST GET

/accounts/{account_no}/payments/calculate-fee

Calculates the 'convenience fee' for credit card payments

POST

/accounts/{account_no}/transactions

Transactions for the account

GET

/accounts/{account_no}/services/{service_id}

Services Related to an account

GET

/accounts/{account_no}/services/{service_id}/usage

Usage details for the service

GET

Well Structured Code

Productive Dev Environment

Mocking

Navigable Documentation



```
1 {  
2     "swagger": "2.0",  
3     "info": {  
4         "version": "1.0.0",  
5         "title": "Swagger Petstore",  
6         "contact": {  
7             "name": "Swagger API Team",  
8             "url": "http://swagger.io"  
9         },  
10        "license": {  
11            "name": "Creative Commons 4.0 International",  
12            "url": "http://creativecommons.org/licenses/by/4.0/  
13        }  
14    },  
15    "host": "petstore.swagger.io",  
16    "basePath": "/api",  
17    "schemes": [  
18        "http"  
19    ],
```

```
1  #%RAML 0.8  
2  ---  
3  #=====-----  
4  # Products API - RAML example  
5  # References:  
6  #   - RAML Specification - http://raml.org/spec.html  
7  #   - RAML Projects - http://raml.org/projects.html  
8  #   - RAML Tools - http://www.apihub.com/raml-tools  
9  #=====-----  
10 title: Products API  
11 version: v0.1  
12 #baseUri: http://products.api.apievangelist.com  
13 baseUri: http://mocksvc.mulesoft.com/mocks/a86239e0-d3af-48c6-9fd5-f8edc7c3!  
14 #List of media type to support  
15 mediaType: application/json  
16 #List of protocols to support for baseUri  
17 protocols: [ HTTP, HTTPS ]  
18 #=====-----  
19 # API documentation  
20 #=====-----  
21 documentation:  
22     - title: Home  
23         content: I #This is a prototype product API.  
24
```

<https://github.com/swagger-api/swagger-spec/blob/master/examples/v2.0/json/petstore.json>

<http://apievangelist.com/2014/03/08/hello-world-product-api-with-blueprint-raml-and-swagger/>



```
1  swagger: "2.0"
2  info:
3    description: "This is a sample server Petstore server.
4    version: 1.0.0
5    title: Swagger Petstore
6    termsOfService: "http://helloreverb.com/terms/"
7    contact:
8      email: "apiteam@swagger.io"
9    license:
10      name: Apache 2.0
11      url: "http://www.apache.org/licenses/LICENSE-2.0.html"
12  host: petstore.swagger.io
13  basePath: /v2
14  tags:
15    - name: pet
16      description: Everything about your Pets
17      externalDocs:
18        description: Find out more
19        url: "http://swagger.io"
20    - name: store
21      description: Access to Petstore orders
```

<https://github.com/swagger-api/swagger-spec/blob/master/examples/v2.0/json/petstore.json>

<http://apievangelist.com/2014/03/08/hello-world-product-api-with-blueprint-raml-and-swagger/>



swagger

RAML

```
/pets/{petId}:
  get:
    summary: Info for a specific pet
    operationId: showPetById
    tags:
      - pets
    parameters:
      - name: petId
        in: path
        required: true
        description: The id of the pet to retrieve
        type: string
    responses:
      200:
        description: Expected response to a valid request
        schema:
          $ref: Pets
```

```
/books:
  /{bookTitle}:
    get:
      description: Retrieve a specific book title
      queryParameters:
        publish_date:
          description: The first date of publication
          type: date
          required: true
          example: 15-Jun2013
      responses:
        200:
          body:
            application/json:
              example: !include book.json
```

<https://github.com/swagger-api/swagger-spec/blob/master/examples/v2.0/json/petstore.json>
<http://apievangelist.com/2014/03/08/hello-world-product-api-with-blueprint-raml-and-swagger/>



swagger

RAML

```
/pets/{petId}:
  get:
    summary: Info for a specific pet
    operationId: showPetById
    tags:
      - pets
    parameters:
      - name: petId
        in: path
        required: true
        description: The id of the pet to retrieve
        type: string
    responses:
      200:
        description: Expected response to a valid request
        schema:
          $ref: Pets
```

```
/books:
  /{bookTitle}:
    get:
      description: Retrieve a specific book title
      queryParameters:
        publish_date:
          description: The first date of publication
          type: date
          required: true
          example: 15-Jun2013
      responses:
        200:
          body:
            application/json:
              example: !include book.json
```

<https://github.com/swagger-api/swagger-spec/blob/master/examples/v2.0/json/petstore.json>

<http://apievangelist.com/2014/03/08/hello-world-product-api-with-blueprint-raml-and-swagger/>



swagger

RAML

/pets/{petId}:

```
  get:  
    summary: Info for a specific pet  
    operationId: showPetById
```

```
      - pets
```

parameters:

```
  - name: petId  
    in: path  
    required: true  
    description: The id of the pet to retrieve  
    type: string
```

responses:

```
  200:
```

```
    description: Expected response to a valid request  
    schema:
```

```
      $ref: Pets
```

/books:

```
  /{bookTitle}:
```

```
    get:  
      description: Retrieve a specific book title  
      queryParameters:
```

```
        publication_date:
```

```
          description: The first date of publication  
          type: date  
          required: true  
          example: 15-Jun2013
```

responses:

```
  200:
```

body:

```
  application/json:  
    example: !include book.json
```

<https://github.com/swagger-api/swagger-spec/blob/master/examples/v2.0/json/petstore.json>

<http://apievangelist.com/2014/03/08/hello-world-product-api-with-blueprint-raml-and-swagger/>



swagger

RAML

```
/pets/{petId}:
  get:
    summary: Info for a specific pet
    operationId: showPetById
    tags:
      - ...
    parameters:
      - name: petId
        in: path
        required: true
        description: The id of the pet to retrieve
        type: string
    responses:
      200:
        description: Expected response to a valid request
        schema:
          $ref: Pets
```

```
/books:
  /{bookTitle}:
    get:
      ...
      queryParameters:
        publish_date:
          description: The first date of publication
          type: date
          required: true
          example: 15-Jun2013
      200:
        body:
          application/json:
            example: !include book.json
```

<https://github.com/swagger-api/swagger-spec/blob/master/examples/v2.0/json/petstore.json>
<http://apievangelist.com/2014/03/08/hello-world-product-api-with-blueprint-raml-and-swagger/>



swagger

RAML

```
/pets/{petId}:
  get:
    summary: Info for a specific pet
    operationId: showPetById
    tags:
      - pets
    parameters:
      - name: petId
        in: path
        required: true
        description: The id of the pet to retrieve
```

```
responses:
  200:
    description: Expected response to a valid request
    schema:
      $ref: Pets
```

```
/books:
  /{bookTitle}:
    get:
      description: Retrieve a specific book title
      queryParameters:
        publish_date:
          description: The first date of publication
          type: date
          required: true
```

```
responses:
  200:
    body:
      application/json:
        example: !include book.json
```

<https://github.com/swagger-api/swagger-spec/blob/master/examples/v2.0/json/petstore.json>
<http://apievangelist.com/2014/03/08/hello-world-product-api-with-blueprint-raml-and-swagger/>

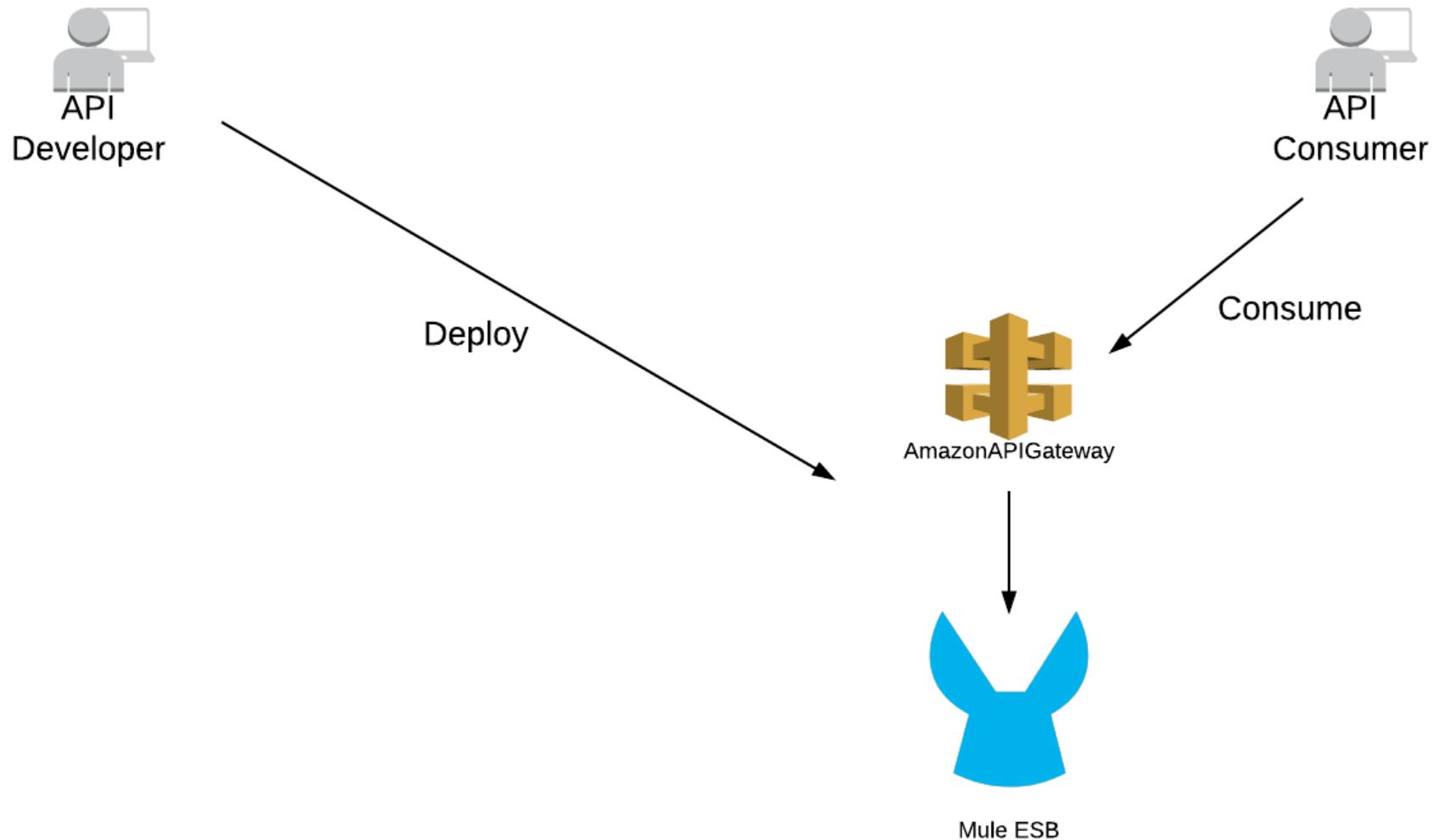
Choosing an API Design Language/Framework

1. Bottom up or top down?
2. What are my target technologies?
3. Do I need specific features?

- What is my team familiar with?
- How ‘Cloudy’ do I need to be?
- What support model do I need?
- Test the tools?

... but remember a **good REST service design is what you are really trying to achieve** regardless of what language you choose to express it in.

Product Driven Spec Choice

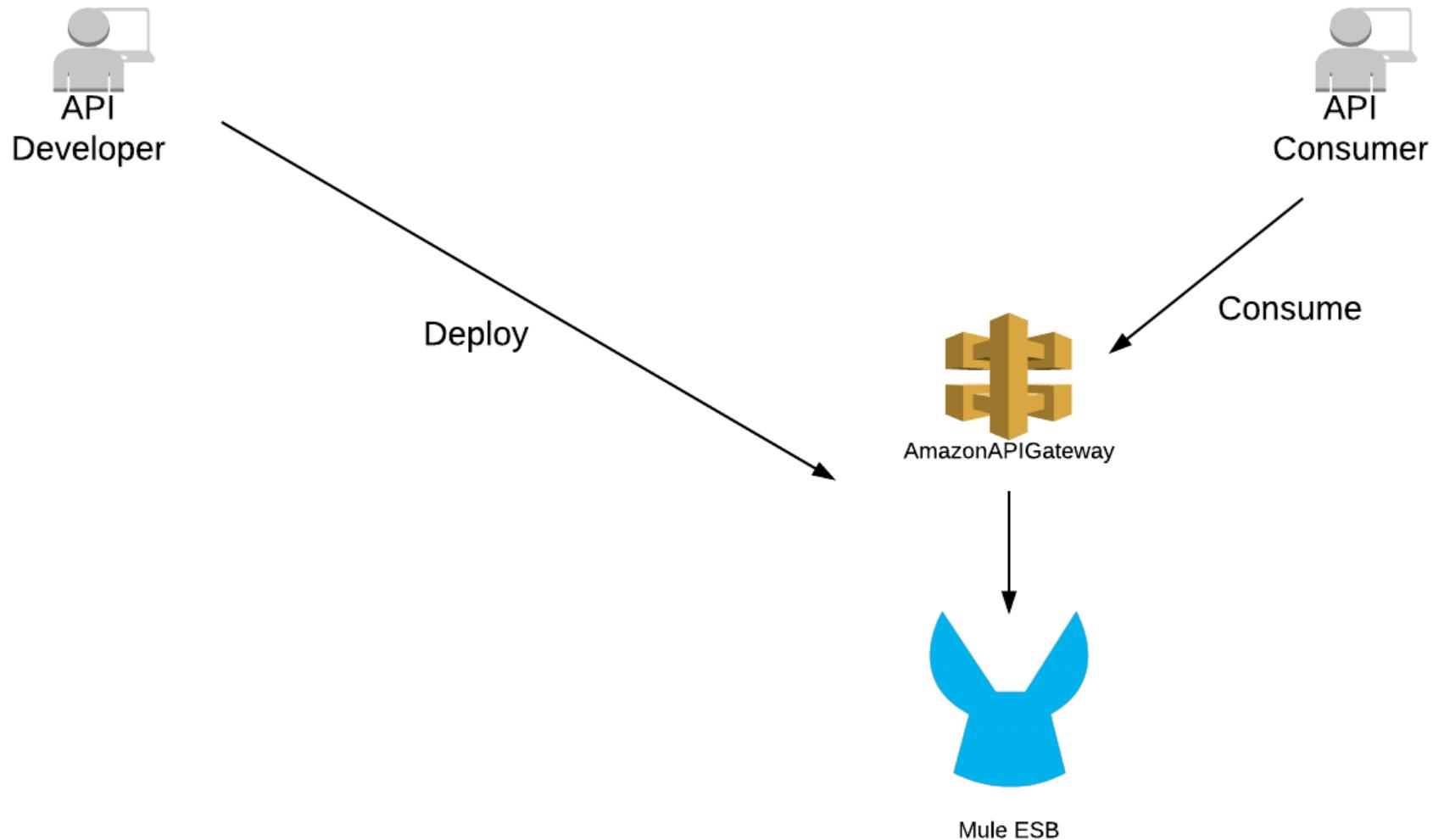


Seek technical advice ...

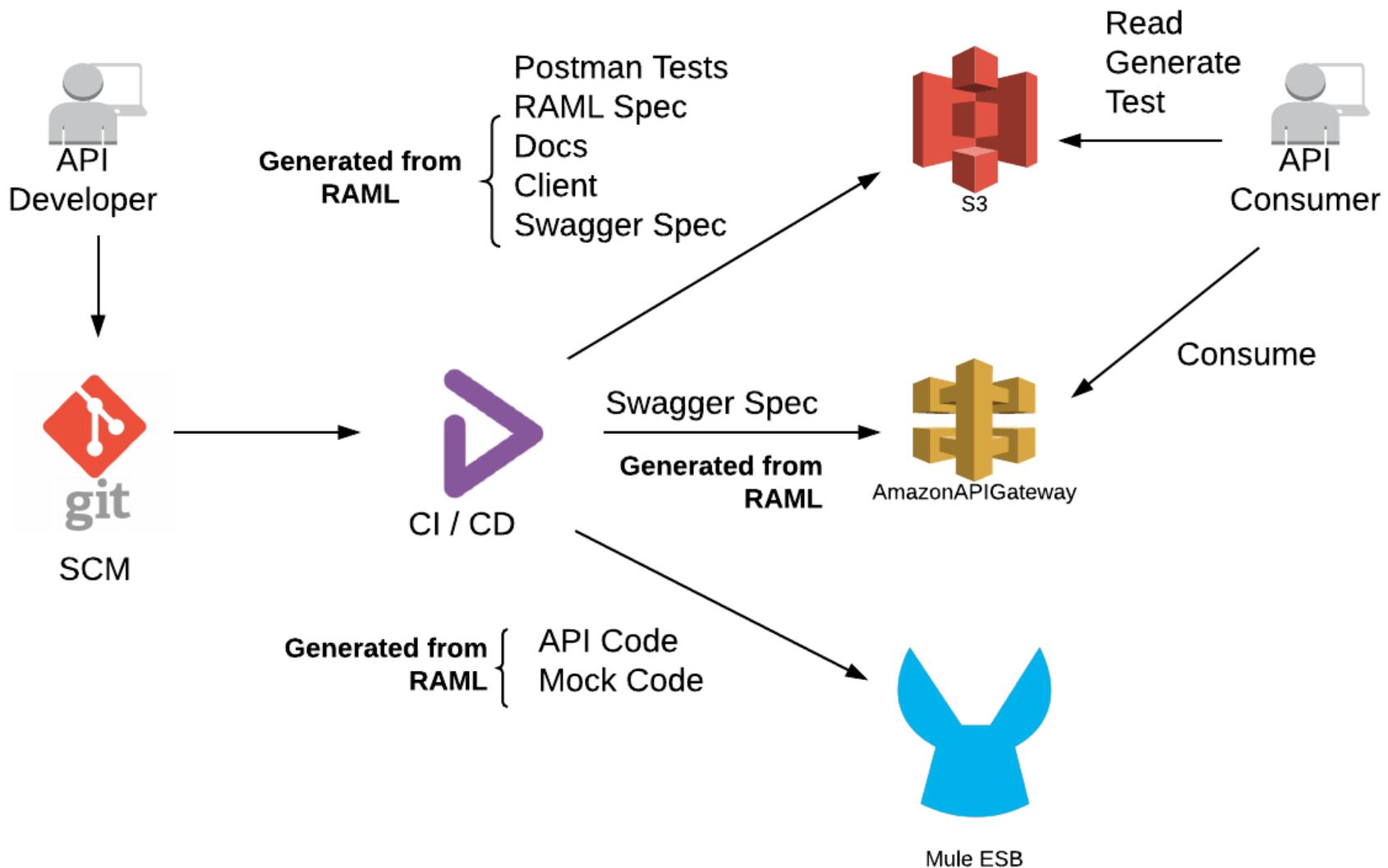
Seek technical advice ...



Product Driven Spec Choice

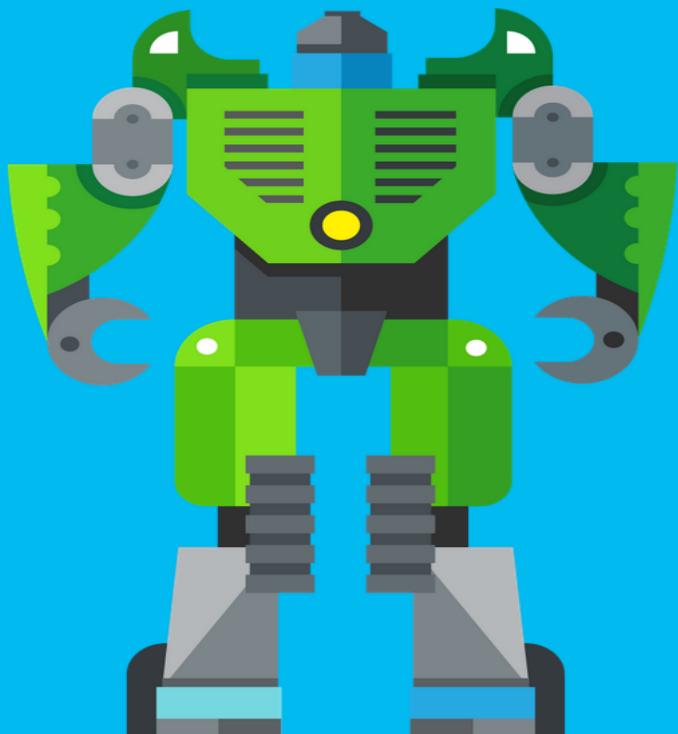


Product Driven Spec Choice



Polyglot!

API TRANSFORMER



Transform API Descriptions from/to various formats e.g., **Swagger, API Blueprint, RAML, WADL, Google Discovery, I/O Docs.**

[Supported Formats](#)

Powered by [APIMATIC](#)

Images credit:

<http://swagger.io>

<http://www.miramax.com/subscript/pulp-fiction-celebrating-20-years/>

https://en.wikipedia.org/wiki/Night_vision_device

<https://maxpreme.wordpress.com/2008/12/08/the-north-face-x-swagger/>

<https://www.google.co.nz/trends/>

<http://www.telestar.fr/2014/photos/zach-galifianakis-l-incroyable-perte-de-poids-de-l-acteur-de-very-bad-trip-diaporama-49531>

<http://mic.com/articles/77505/why-everyone-is-so-mad-about-the-wolf-of-wall-street#.GL46BT50a>