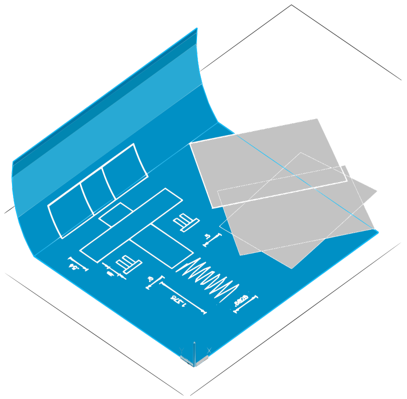
API Description Languages

Laura Heritage
Director of API Strategy

SOA software™
Powering the API Economy

What is an API Description Language (API DL)?



Blueprint

Contract

Metadata

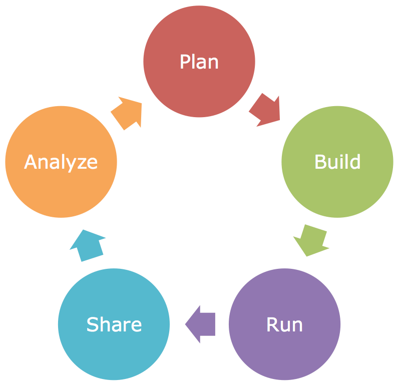
Human Docs

SOA software™
Powering the API Economy

What About WSDL2.0 or WADL?

- For REST, they are not widely adopted
- Both are not very “humanly readable”
- Both are typically auto-generate from code – wouldn’t use them as a “blueprint” amongst non-technical types
- WADL doesn’t contain enough information to adequately describe a RESTful API. Though does have extension points which are seldom used.
- WSDL contains almost everything you need but is quite brittle. If it changes the clients must change too

API DL Brings REST to the Enterprise



Governable

Shareable

Readable

"Lack of a way to describe a RESTful services was one of the largest barriers to REST adoption in the enterprise."

SOA software™
Powering the API Economy

Many API DL Are Available Today

WSDL

WADL

Swagger

idDocs

RAML

Hypermedia

API Blueprint

SOA software™
Powering the API Economy

Most Active API DL Communities

	API-Blueprint	RAML*	SWAGGER*
Format	Markdown	YAML	JSON
Available at	GitHub	GitHub	GitHub
Sponsored by	Apiary	Mulesoft	Reverb
Current Version	1A3	0.8	1.2
Workgroup	No	Yes	Yes
Initial commit	April, 2013	Sep, 2013	July, 2011
API Design Approach	Top-down	Top-down	**Bottom-up

** Most Widely Adopted by Enterprises*

SOA software™
Powering the API Economy

How Do You Choose?

Define Key Purpose Behind Using an API DL

	SWAGGER	RAML
I need the ability to design the API with technically limited API stakeholders	- (new top down tooling being developed)	X
I want a way to describe and design an API with my technical team	X	X (has more advanced meta data constructs (includes, inheritance, traits))
I need to validate the requests and responses at runtime	X (swagger-node-express, swagger - Play)	X (Osprey)
I need to easily consume the API specification between two or more systems	X	X
I need documentation for audit and compliance	X	X
I need exceptional external API developer experiences	X	X
I need to generate server code	X (node.js, Java)	X (node.js, Java)
I need to generate client code	X	limited

SOA software™
Powering the API Economy

Understand How Your Team Works

	SWAGGER	RAML
Do you design before you code?	-	X
Do you generate documents after you code from your code?	X	X (early release of JAXRS-to-RAML)
Do you want docs embedded in your server code?	X	-

SOA software™
Powering the API Economy

Look At Their Community

- **Swagger** by far has the largest community, since its been around since 2011
- **RAML** has gained traction in the enterprise due to the richness of its modeling capability; API version metatdata, nested resources, composition and inheritance, file inclusions and top down approach

Generators	SWAGGER	RAML
Documentation From Code	Clojure, ColdFusion/CFML, Eiffel, Go, Java, .Net, Node.js, PHP, Python, Ruby, Scala	JAX-RS
Spec Parsers	Java, js	PHP, Ruby, Phython, Java, Javascript
API Interfaces	Java, Node.js	Java, Node.js
Client Code	Several	Developing
Editor Tooling	*new in the works based on YAML, demo'd in May	API Designer, Sublime plugin, Atom

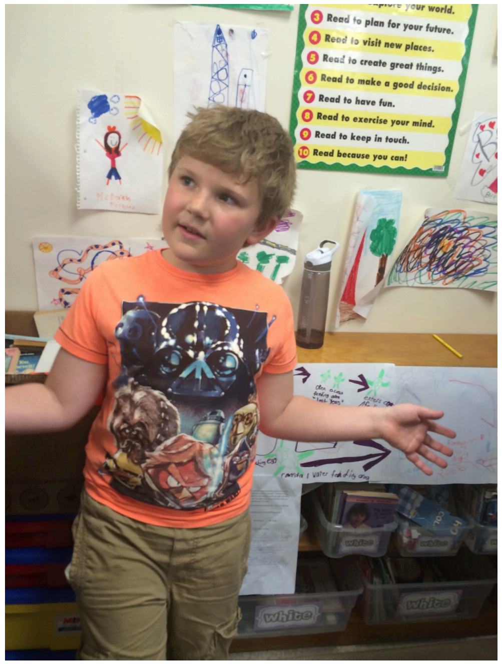
SOA software™
Powering the API Economy

In Action – What Is the Experience?

- WishList Application
  - Add a User
  - Get Users
  - Add a wish
  - Get all wishes in the system
  - Get wishes for particular user
    - Filter by price range
    - Filter by merchant

- Platform : Node.js, Express and Mongo

- Test
  - Build with RAML
  - Build with SWAGGER



Reese's Birthday Soon

SOA software™
Powering the API Economy

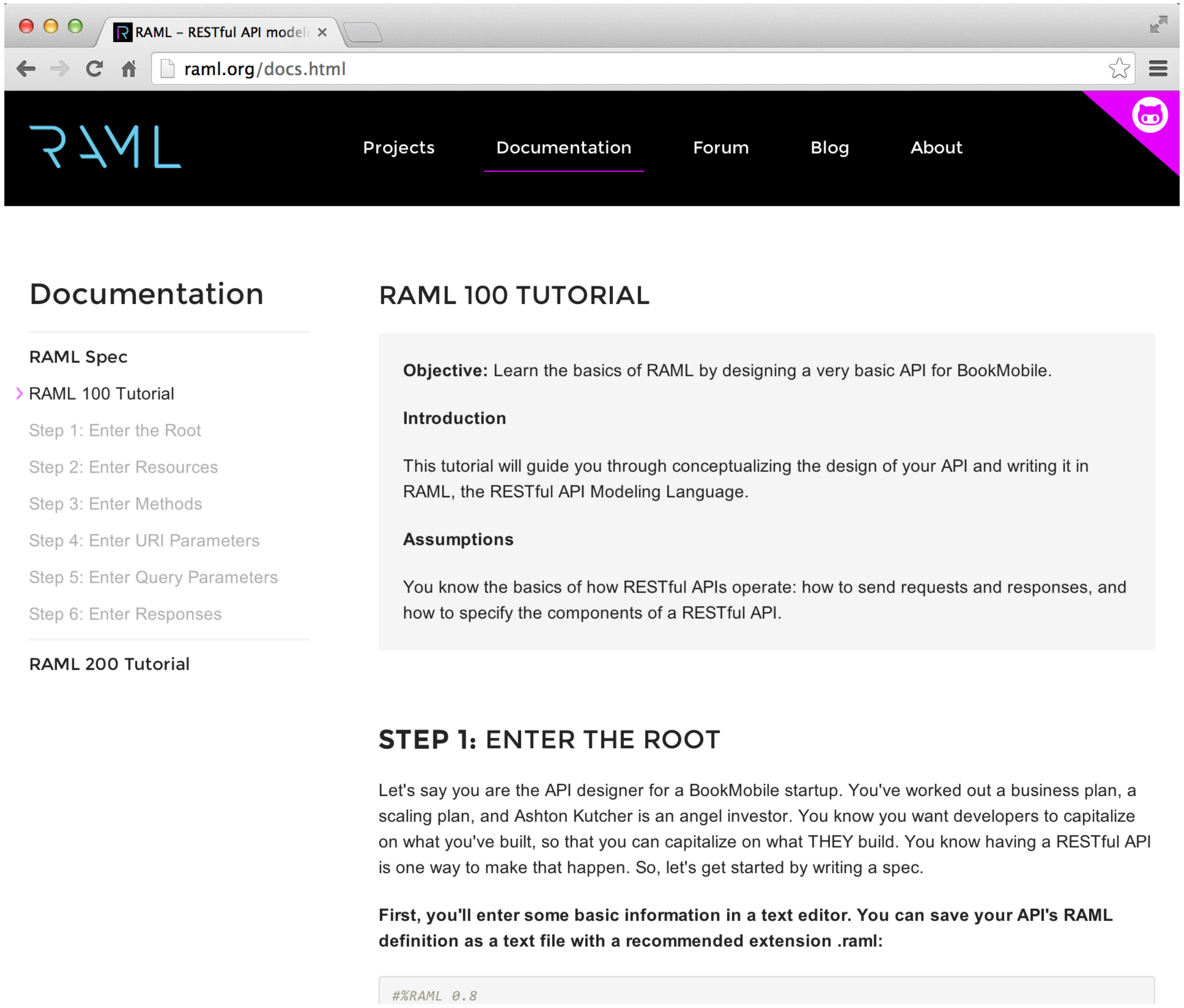
My Background

- Degree in Computer Science & Mathematic
- 16 years @IBM and 5 Months at @SOASoftwareInc
- 14+ years working with SOAP/REST, SOA, API
- Over the years I have been a developer, an architect, a competitive technical sales, a product manager and an evangelist
- I always liked to be on the bleeding of edge technology
- Very hands on
- I would consider myself a “Sunday” developer.   I don’t get to code as often as I use to.   Now I mostly configure products and do scripting.
- Spend most of my time working with customer.

SOA software™
Powering the API Economy

RAML

Start with tutorial at RAML.org



Documentation

RAML Spec
RAML 100 Tutorial
Step 1: Enter the Root
Step 2: Enter Resources
Step 3: Enter Methods
Step 4: Enter URI Parameters
Step 5: Enter Query Parameters
Step 6: Enter Responses
RAML 200 Tutorial

RAML 100 TUTORIAL

Objective: Learn the basics of RAML by designing a very basic API for BookMobile.

Introduction

This tutorial will guide you through conceptualizing the design of your API and writing it in RAML, the RESTful API Modeling Language.

Assumptions

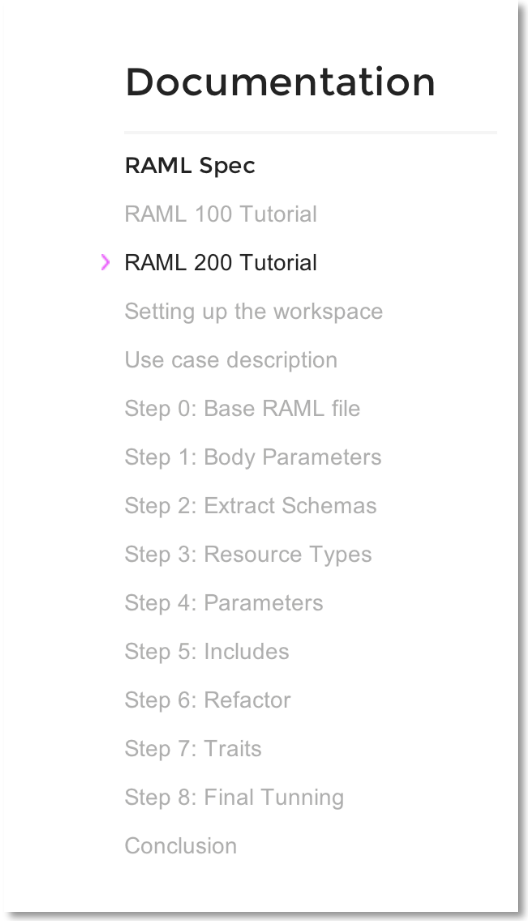
You know the basics of how RESTful APIs operate: how to send requests and responses, and how to specify the components of a RESTful API.

STEP 1: ENTER THE ROOT

Let's say you are the API designer for a BookMobile startup. You've worked out a business plan, a scaling plan, and Ashton Kutcher is an angel investor. You know you want developers to capitalize on what you've built, so that you can capitalize on what THEY build. You know having a RESTful API is one way to make that happen. So, let's get started by writing a spec.

First, you'll enter some basic information in a text editor. You can save your API's RAML definition as a text file with a recommended extension .raml:

```
#%RAML 0.8
```

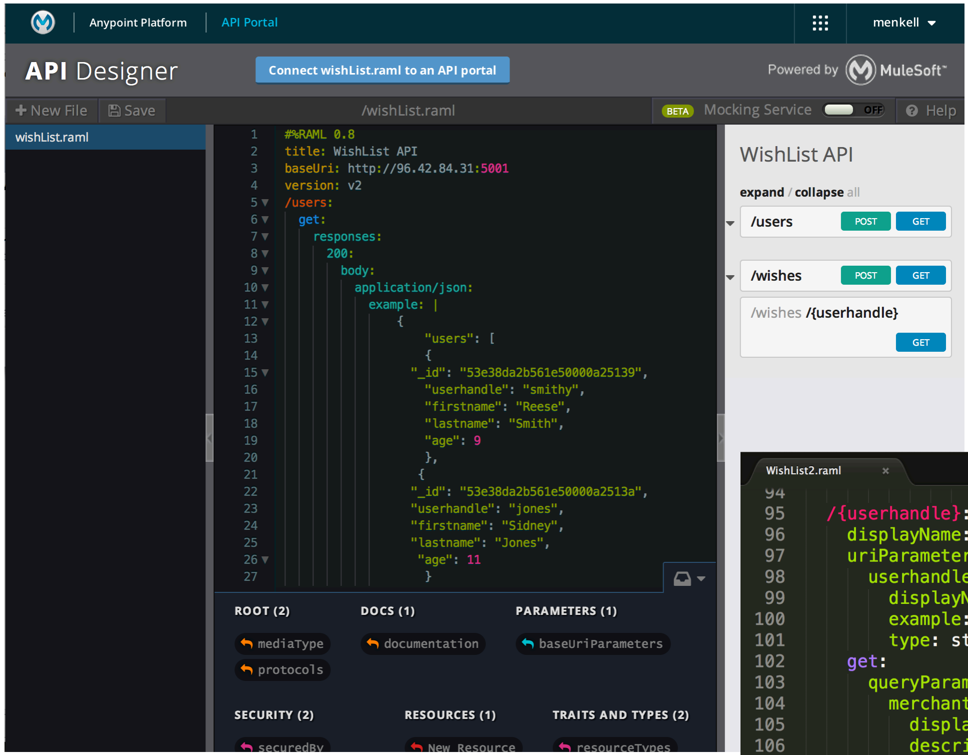


Documentation

RAML Spec
RAML 100 Tutorial
RAML 200 Tutorial
Setting up the workspace
Use case description
Step 0: Base RAML file
Step 1: Body Parameters
Step 2: Extract Schemas
Step 3: Resource Types
Step 4: Parameters
Step 5: Includes
Step 6: Refactor
Step 7: Traits
Step 8: Final Tunning
Conclusion

SOA software™
Powering the API Economy

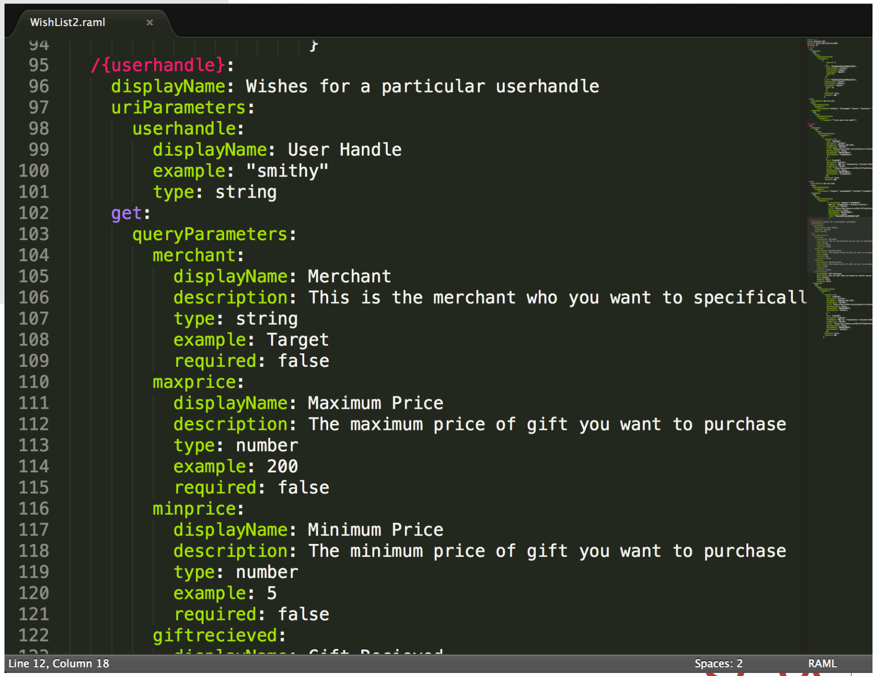
RAML Editors



```
#%RAML 0.8
title: WishList API
baseUri: http://96.42.84.31:5001
version: v2
/users:
  get:
    responses:
      200:
        body:
          application/json:
            example: |
              {
                  "users": [
                  {
                "_id": "53e38da2b561e50000a25139",
                  "userhandle": "smithy",
                  "firstname": "Reese",
                  "lastname": "Smith",
                  "age": 9
                  },
                 {
                "_id": "53e38da2b561e50000a2513a",
                "userhandle": "jones",
                "firstname": "Sidney",
                "lastname": "Jones",
                 "age": 11
                  }
```

- API Designer - http://api-portal.anypoint.mulesoft.com/raml/api-designer
  - (allows for mocking)
- Sublime Editor https://github.com/mulesoft/raml-sublime-plugin

NOTE: Indentation matters in both

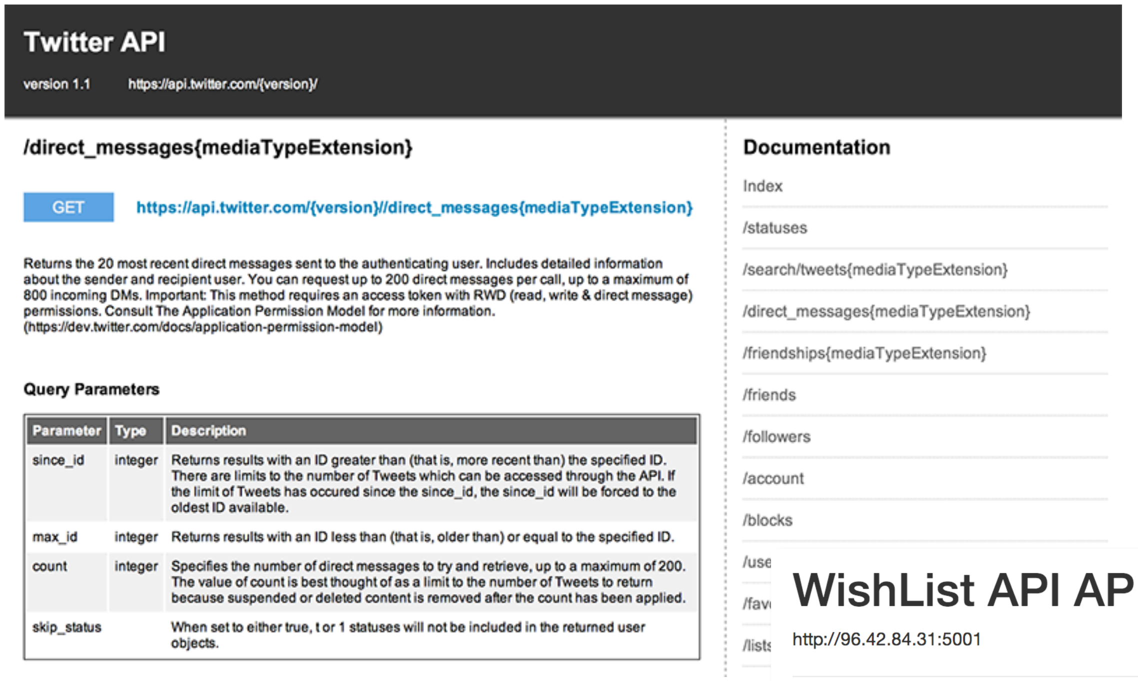


```
        }
/{userhandle}:
  displayName: Wishes for a particular userhandle
  uriParameters:
    userhandle:
      displayName: User Handle
      example: "smithy"
      type: string
  get:
    queryParameters:
      merchant:
        displayName: Merchant
        description: This is the merchant who you want to specificall
        type: string
        example: Target
        required: false
      maxprice:
        displayName: Maximum Price
        description: The maximum price of gift you want to purchase
        type: number
        example: 200
        required: false
      minprice:
        displayName: Minimum Price
        description: The minimum price of gift you want to purchase
        type: number
        example: 5
        required: false
      giftrecieved:
```

- *Can have 1 to many files*
- *I have a very simple API. I kept mine in 1. Didn't use includes, but did play with them*
- *No Security*

SOA software™
Powering the API Economy

Document Generation



Twitter API

version 1.1 https://api.twitter.com/{version}/

/direct_messages{mediaTypeExtension}

GET https://api.twitter.com/{version}//direct_messages{mediaTypeExtension}

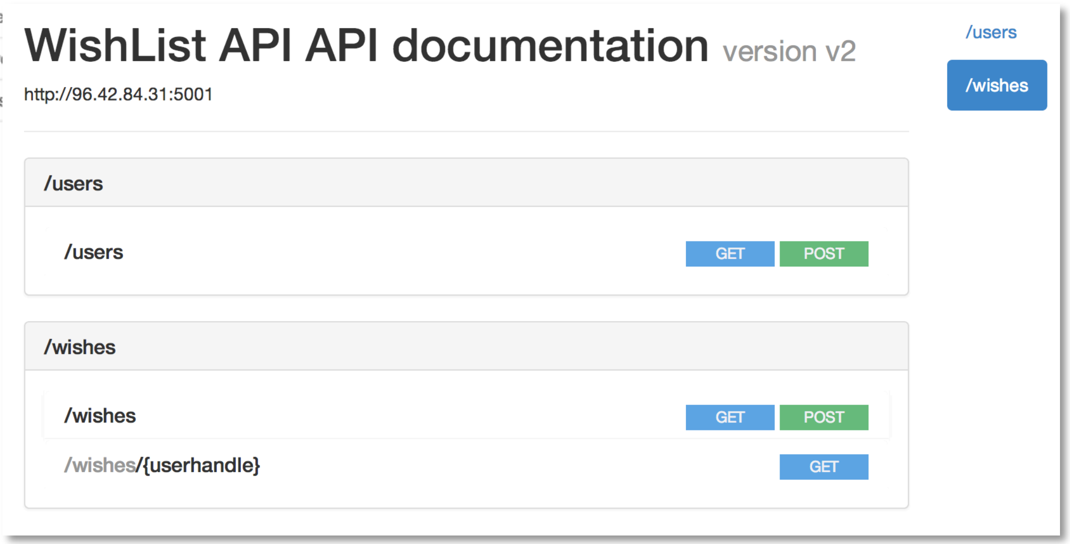
Returns the 20 most recent direct messages sent to the authenticating user. Includes detailed information about the sender and recipient user. You can request up to 200 direct messages per call, up to a maximum of 800 incoming DMs. Important: This method requires an access token with RWD (read, write & direct message) permissions. Consult The Application Permission Model for more information. (https://dev.twitter.com/docs/application-permission-model)

Query Parameters

Parameter	Type	Description
since_id	integer	Returns results with an ID greater than (that is, more recent than) the specified ID. There are limits to the number of Tweets which can be accessed through the API. If the limit of Tweets has occured since the since_id, the since_id will be forced to the oldest ID available.
max_id	integer	Returns results with an ID less than (that is, older than) or equal to the specified ID.
count	integer	Specifies the number of direct messages to try and retrieve, up to a maximum of 200. The value of count is best thought of as a limit to the number of Tweets to return because suspended or deleted content is removed after the count has been applied.
skip_status		When set to either true, t or 1 statuses will not be included in the returned user objects.

Documentation

- Index
- /statuses
- /search/tweets{mediaTypeExtension}
- /direct_messages{mediaTypeExtension}
- /friendships{mediaTypeExtension}
- /friends
- /followers
- /account
- /blocks
- /use
- /fav
- /lists



WishList API API documentation version v2

http://96.42.84.31:5001

/users

/users GET POST

/wishes

/wishes GET POST

/wishes/{userhandle} GET

- RAML to HTML
- RAML to HTML - PHP

SOA software™ Powering the API Economy

Thoughts on RAML Experience

- Great documentation, samples and tutorial
- Good way to model your API
  - When writing my server code, I did find I went back to my RAML model to remember what I was suppose to be doing.
- It is easy to understand and write, from the basic API perspective.
  - When you get into Includes, Traits, it becomes a little more technical.
- Community tools
  - Are easy to understand, install and work with.
  - I played with:
    - API Designer
    - RAML Sublime Plugin
    - RAML to HTML
    - Swagger2raml
    - Osprey / Osprey CLI

SOA software™
Powering the API Economy

SWAGGER

Swagger Community

- https://github.com/wordnik/swagger-spec

Swagger Getting Started

- https://github.com/wordnik/swagger-spec/wiki

Three Ways To Create Swagger

1. Codegen: Traditional way of creating a Swagger Specification. The swagger codegen converts annotation in your code to Swagger Specification

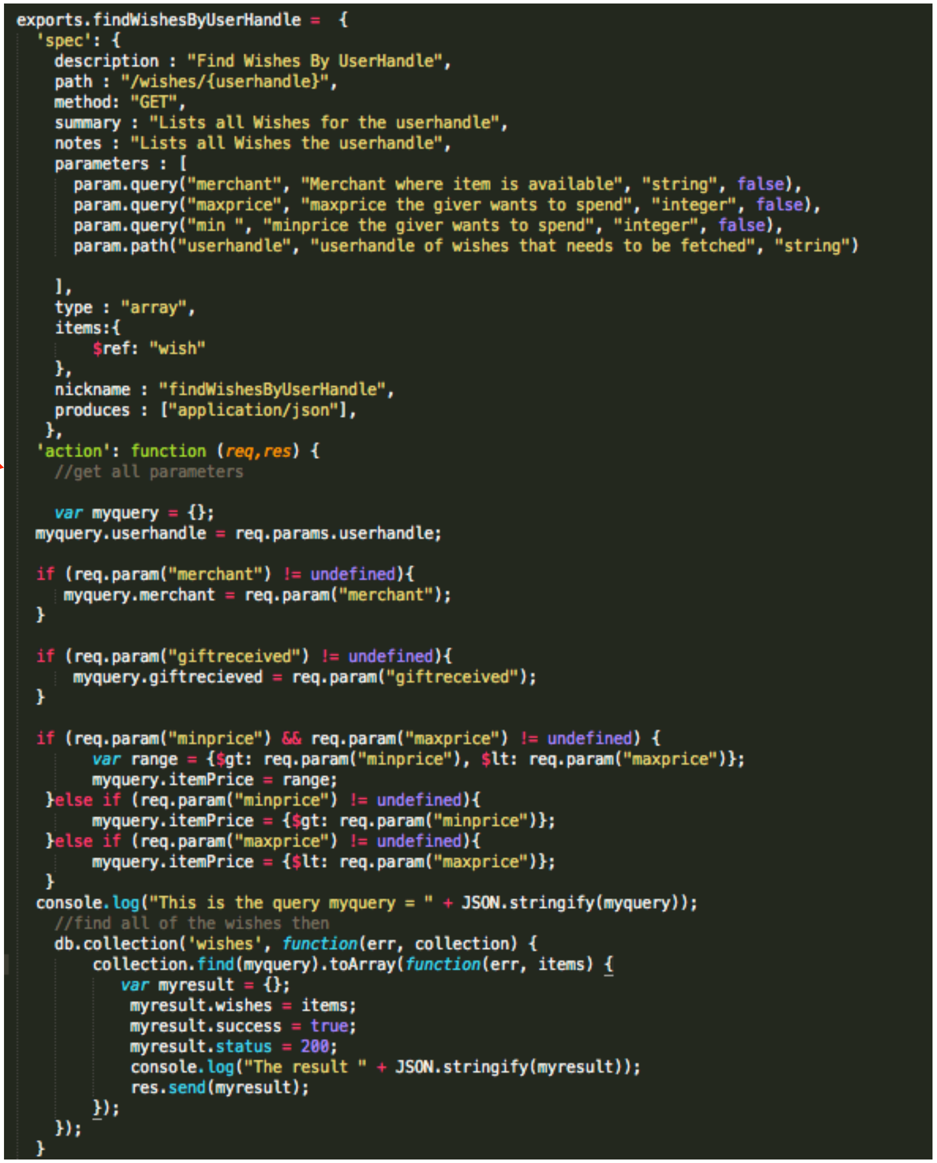
2. Automatically: swagger-node-express and swagger-play will create both your REST APIs and your Swagger Specification for you at the same time
   - https://www.npmjs.org/package/swagger-node-express

3. Manually: Write the json by hand.

Used swagger-node-express

Spec

Action

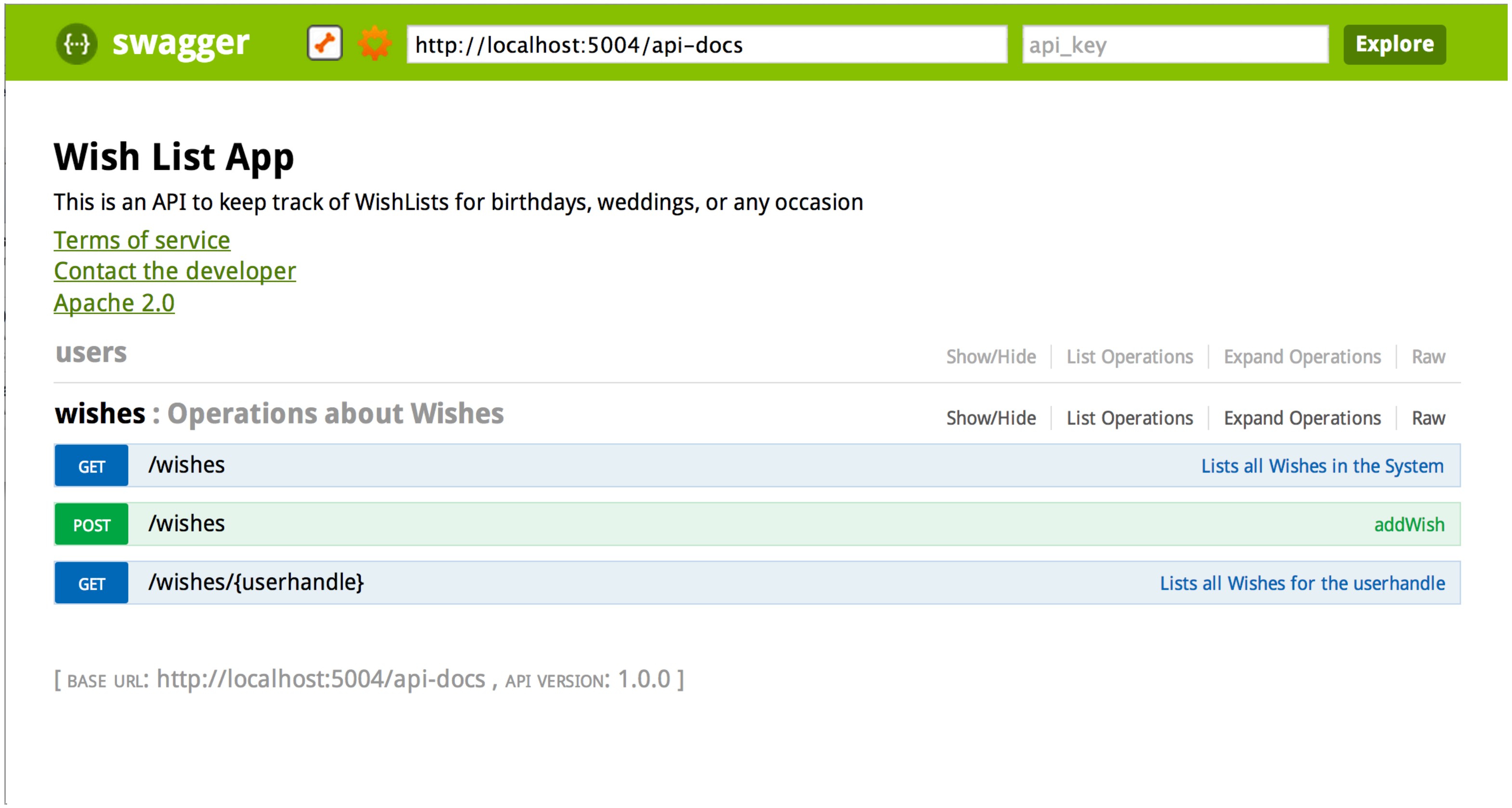


```
exports.findWishesByUserHandle =  {
  'spec': {
    description : "Find Wishes By UserHandle",
    path : "/wishes/{userhandle}",
    method: "GET",
    summary : "Lists all Wishes for the userhandle",
    notes : "Lists all Wishes the userhandle",
    parameters : [
      param.query("merchant", "Merchant where item is available", "string", false),
      param.query("maxprice", "maxprice the giver wants to spend", "integer", false),
      param.query("min ", "minprice the giver wants to spend", "integer", false),
      param.path("userhandle", "userhandle of wishes that needs to be fetched", "string")

    ],
    type : "array",
    items:{
        $ref: "wish"
    },
    nickname : "findWishesByUserHandle",
    produces : ["application/json"],
   },
  'action': function (req,res) {
    //get all parameters

    var myquery = {};
  myquery.userhandle = req.params.userhandle;

  if (req.param("merchant") != undefined){
     myquery.merchant = req.param("merchant");
  }

  if (req.param("giftreceived") != undefined){
      myquery.giftrecieved = req.param("giftreceived");
  }

  if (req.param("minprice") && req.param("maxprice") != undefined) {
        var range = {$gt: req.param("minprice"), $lt: req.param("maxprice")};
        myquery.itemPrice = range;
   }else if (req.param("minprice") != undefined){
        myquery.itemPrice = {$gt: req.param("minprice")};
   }else if (req.param("maxprice") != undefined){
        myquery.itemPrice = {$lt: req.param("maxprice")};
   }
  console.log("This is the query myquery = " + JSON.stringify(myquery));
    //find all of the wishes then
    db.collection('wishes', function(err, collection) {
        collection.find(myquery).toArray(function(err, items) {
           var myresult = {};
            myresult.wishes = items;
            myresult.success = true;
            myresult.status = 200;
            console.log("The result " + JSON.stringify(myresult));
            res.send(myresult);
        });
    });
  }
```

- Server.js – the node-express server
- Model.js - describes the resources (User, Wish)
- Resource.js – defines the actions for the resources

SOA software™
Powering the API Economy

Swagger Documentation



swagger http://localhost:5004/api-docs api_key Explore

Wish List App

This is an API to keep track of WishLists for birthdays, weddings, or any occasion

Terms of service
Contact the developer
Apache 2.0

users — Show/Hide | List Operations | Expand Operations | Raw

wishes : Operations about Wishes — Show/Hide | List Operations | Expand Operations | Raw

GET	/wishes	Lists all Wishes in the System
POST	/wishes	addWish
GET	/wishes/{userhandle}	Lists all Wishes for the userhandle

[BASE URL: http://localhost:5004/api-docs , API VERSION: 1.0.0]

Composed of two files:

- **Resource Listing:** Lists the APIs that are available and gives a brief description of the them.
- **API Description:** Detailed description of each API in the Resource Listing.

SOA software™ Powering the API Economy

Thoughts on SWAGGER Experience

- Most widely used API DL to date
- On working with swagger-node-express:
  - Once you get the hang of it, went smooth and very fun to see results.
  - Key is to get your model.js (resource definitions) and your routes.js for your data access specified correctly.
  - Definitely not top down. I ended up using my RAML spec to keep me on track.
  - Con - you are really embedding swagger throughout your code. Code which may live forever, but will swagger?
- On Codegen creation of swagger:
  - Easy self-explanatory
  - Is still a bottom up approach. Even though you are stubbing out your code as you are modeling it, it is hard to share and express thoughts with out a lot of effort up front.
- Swagger-UI is very useful to help visualize and test your API
- If you have complex APIs, swagger probably won't have the the constructs you need to fully express them.
- Writing swagger manually in JSON is not fun. It is not very human readable. A new Swagger Editor project was launched in May.

Where to do documentation?

	Pros	Cons
Document in design and generate interface and docs from design	• Allows for changing docs without disturbing production code. • Allows for NLS translation outside of production cycles • If backend implementation changes but interface still the same, you are okay.	Documentation can get out of sync with what is running in production.
Document in code and generate docs from code	• Documentation is more likely to stay in sync with what is in production	• If issues with documentation, need to edit production code to regenerate
Embed documentation in server code and have server dynamically serve up docs	• Documentation is more likely to stay in sync because it is embedded and running as part of the application	• You are now tied to that particular API DL implementation running your production code • If there is user documentation errors, production changes are a must

SOA software™
Powering the API Economy

Thoughts On the API DL Focused on Today

- All the API DL are starting to provide similar features and functions. They will continue to get closer and closer.
- Neither specifications that we focused on today can describe anything other then a RESTful API/Service.
  - For SOAP based APIs WSDL is still king. An API Platform you choose should support SOAP based APIs via WSDL as well.
- Neither specification provides ability for extension, for example: describing testing and monitoring of the operations
- Neither specification handles National Language (NLS) of documentation
- System to System interactions – Today mostly focused on API creation and developer consumption. Next step is for system-to-system integrations

SOA software™
Powering the API Economy

References

- Another API-Blueprint, RAML, Swagger Comparison – Ole Lensmar
- Investigating API Developer Tooling - @DanMayer

Questions

SOA software™
Powering the API Economy