

Prototyping

Mike Amundsen
@mamund

**Designing and Building
Great APIs**



Prototyping Great APIs

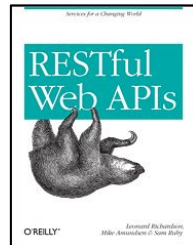
- What is a Prototype?
- OpenAPI Spec/Swagger



#mcaTravels

@mamund

#perth2018



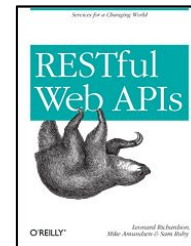
What is a Prototype?




#mcaTravels

@mamund

#perth2018



pro·to·type

/'prōdə,tīp/ 

noun

1. a first, typical or preliminary model of something, especially a machine, from which other forms are developed or copied.
"the firm is testing a prototype of the weapon"







Prototyping APIs

- Prototypes look like the real thing, but are not. They're "fakes."
- They let you work up something with all the details of a real API, but without the actual functionality behind it.
- They're an inexpensive way to work out the details
- Use them to discover challenges before you go into production.



Swagger Editor

The image shows the Swagger Editor web application. On the left, a dark-themed code editor displays a Swagger 2.0 specification for a 'TODO List API'. The specification includes metadata, a host, schemes, a base path, and a single endpoint `/todos` with a `GET` method. The endpoint has a query parameter `page` of type `integer` and a response of type `array` of `users`. On the right, the rendered API documentation is shown. It features a title 'TODO List API', a description 'The Power of TODOs in an API', and a version '1.0.1'. Below this, there's a 'Filter operations by a tag' section with a 'User' tag. The 'Paths' section shows the `/todos` path. A modal window is open for the `GET /todos` operation, showing a 'Summary' section with the description 'Use this call to get a list of all todo items', a 'Parameters' table, and a 'Responses' section.

```
1 {
2   "swagger": "2.0",
3   "info": {
4     "title": "TODO List API",
5     "description": "The Power of TODOs in an API",
6     "version": "1.0.1"
7   },
8   "host": "api.todos.com",
9   "schemes": [
10    "https"
11  ],
12  "basePath": "/v1.0.1/api",
13  "produces": [
14    "application/json",
15  ],
16  "paths": {
17    "/todos": {
18      "get": {
19        "summary": "Use this call to get a list of all todo
20        items",
21        "parameters": [
22          {
23            "name": "page",
24            "in": "query",
25            "description": "Requested page number.
26            Defaults to 1.",
27            "required": false,
28            "type": "integer",
29            "format": "int32"
30          }
31        ],
32        "tags": [
33          "User"
34        ],
35        "responses": {
36          "200": {
37            "description": "An array of users matching the
38            query parameters (if any).",
39            "schema": {
40              "$ref": "#/definitions/Users"
41            }
42          }
43        }
44      }
45    }
46  }
47 }
```

TODO List API

The Power of TODOs in an API

Version 1.0.1

Filter operations by a tag:

User

Paths

/todos

GET /todos

User

Summary

Use this call to get a list of all todo items

Parameters

Name	Located in	Description	Required	Schema
page	query	Requested page number. Defaults to 1.	No	integer (int32)

Responses

Code	Description	Schema
------	-------------	--------

Prototyping APIs

- Use tools like Live API Creator, NodeJS express, or other code-generating platforms.
- It's also a good idea to use service-virtualization frameworks to mock up the response data.
- If possible, include access-control checking when running tests against the prototype.
- If possible use existing production-level API consumers to test out the prototype.



Prototypes are made to be tested.



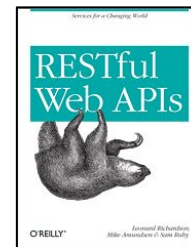
So...



#mcaTravels

@mamund

#perth2018



Prototyping Great APIs

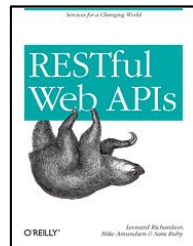
- Prototypes are for testing
- Use OpenAPI Spec/Swagger for prototyping



#mcaTravels

@mamund

#perth2018



Exercise

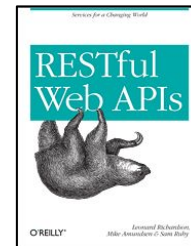
<http://g.mamund.com/todo-list-swagger>



#mcaTravels

@mamund

#perth2018





```
1 ---|
2 swagger: "2.0"
3 info:
4   description: From the todo list tutorial on goswagger.io
5   title: A Todo list application
6   version: 1.0.0
7 consumes:
8   - application/io.goswagger.examples.todo-list.v1+json
9 produces:
10  - application/io.goswagger.examples.todo-list.v1+json
11 schemes:
12  - http
13 paths:
14  /:
15    get:
16      tags:
17        - todos
18      parameters:
19        - name: since
20          in: query
21          type: integer
22          format: int64
23        - name: limit
24          in: query
25          type: integer
26          format: int32
27          default: 20
28      responses:
```



A Todo list application 1.0.0

From the todo list tutorial on goswagger.io

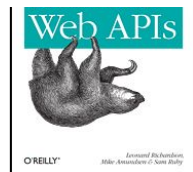
Schemes

HTTP ▾

todos ▾

GET /

Models ▾



Prototyping

Mike Amundsen
@mamund

**Designing and Building
Great APIs**

