

Mocking as a service with Apiary.io

**Impress your clients
with this one simple
trick!**

Why I love Apiary.io

- API Documentation
- API Mocking
- API Testing
- API Collaboration

API DESIGN IS HARD

Build beautiful APIs



See how Apiary works



Quick start with GitHub



Documentation

Yaaaaay... Documentation...

API Blueprint

(Essentially, Markdown for APIs)

API Blueprint (Step 1) - Give your API a name

Polls

Polls is a simple API allowing consumers to view polls and vote in them.

API Blueprint (Step 2) - Create a group of resources

Group Questions

Resource related to questions in the API.

API Blueprint (Step 3a) - Add a resource!

```
## Question Collection [/questions]
```

API Blueprint (Step 3b) - With parameters?

```
## Question [/questions/{question_id}]
```

```
+ Parameters
```

```
  + question_id (number) ... ID of the Question in form of an integer
```

API Blueprint (Step 4a) - Add actions!

```
### List All Questions [GET]
```

API Blueprint (Step 4b) - With a response?

+ Response 200 (application/json)

```
[
  {
    "question": "Favourite programming language?",
    "published_at": "2014-11-11T08:40:51.620Z",
    "url": "/questions/1",
    "choices": [
      ...
    ]
  },
  ...
]
```

API Blueprint (Step 4c) - With a request?

Create a New Question [POST]

```
+ Request (application/json)
  {
    "question": "Favourite programming language?",
    "choices": [
      "Swift",
      "Python",
      "Objective-C",
      "Ruby"
    ]
  }
```

API Blueprint (Step 4d) - With practically nothing?

```
### Delete [DELETE]
```

```
+ Response 204
```



INTRODUCTION

REFERENCE

Polls API Root

Question

Question

Choice

Questions Collection

Resources related to questions in the API.

Question

A Question object has the following attributes:

- question
- published_at - An ISO8601 date when the question was published.
- url
- choices - An array of Choice objects.

View a Questions Detail



Choice

Vote on a Choice



This action allows you to vote on a question's choice.

Questions Collection



Switch to Console

Parameters

question_id ● ID of the Question in form of an integer Example: **1**. Number

Request

Production



Raw



Try

Response

200

Content-Type: application/json

```
1  {
2    "question": "Favourite programming language?",
3    "published_at": "2014-11-11T08:40:51.620Z",
4    "url": "/questions/1",
5    "choices": [
6      {
7        "choice": "Swift",
8        "url": "/questions/1/choices/1",
9        "votes": 2048
10     },
11     {
12       "choice": "Python",
13       "url": "/questions/1/choices/2",
14       "votes": 1024
15     }
16   ]
17 }
```


Mocking

Mocking

- Apiary.io creates a mock server using your API Blueprint.
- This means you instantly get a server against which you can make calls and get responses.
- **<http://private-088936-gtdtestapi.apiary-mock.com>**

Testing

Testing

- You can get Feedback via the Inspector
- Easy to see what worked, what didn't work, and what ehhehhh kinda worked



Listening at http://private-088936-gtdtestapi.apiary-mock.com

Show Missed Request

On

Resource	Method	Status	Source	Time
 /specialquestions	GET	404	206.108.168.134	a few seconds ago
 /questions	POST	201	206.108.168.134	a minute ago
 /questions	POST	201	206.108.168.134	2 minutes ago
 /questions	GET	200	206.108.168.134	2 minutes ago
 /questions/%7Bquestion_id%7D/choices/%7Bchoice_id%7D	POST	201	206.108.168.134	3 minutes ago
 /questions/%7Bquestion_id%7D	GET	200	206.108.168.134	3 minutes ago



Listening at http://private-088936-gtdtestapi.apiary-mock.com

Show Missed Request

On

Resource

Method

Status

Source

Time

```
+ connect-time: 3
+ x-request-start: 1429037231686
+ total-route-time: 2
+ content-length: 46
```

Call Body

Show Expected

At '/choices' Missing required property: choices

```
{
  "question": "Favourite programming language?"
}
```

Response Header

Show Expected

```
Content-Type: application/json
Location: /questions/2
```

Collaboration

Collaboration

- Apiary is built for collaboration
- 2 types of users: viewers and editors
- Viewers can leave comments on the documentation
- Editors can straight up change the spec

Other cool stuff

Other cool stuff

- Support for publicly available API specs
- A bunch of things integrate with API Blueprint (like Paw!)
- Auto push/pull changes to Github
- And all this stuff is free!

Thanks!

@frankacy

Questions?

