

Build An API Your Users Will Love

PRESENTED BY:

Chris Tankersley


Developer Relations Tooling
Manager

Vonage Developer Center

Create a free... →

Login →


Communications & Network APIs


Voice

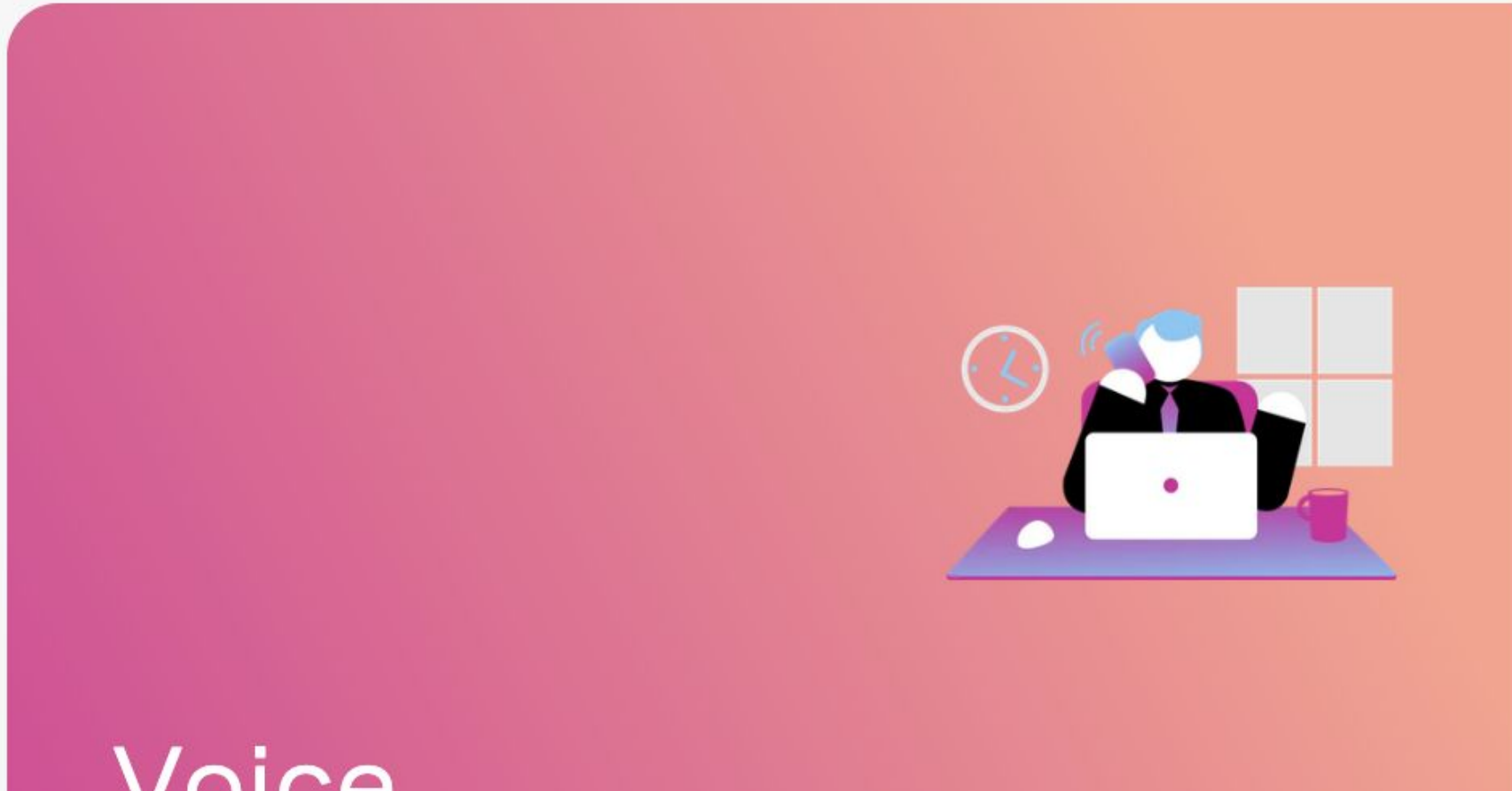

Video


Messaging


Network APIs


Low Code and AI


Privacy and Security



- ✓ Send smart voice notifications
- ✓ Create IVRs and Voice Bots
- ✓ Speech to text and WebSockets
- ✓ Text to Speech with over 50+ languages
- ✓ Embed calls in web and mobile

OVERVIEW

HTTP CODES



CALLBACKS



PROPERTIES



REQUEST



RESPONSE



URI



VERSIONING



🔗 Vonage API Standards

This is a living document containing Vonage's API standards. We'll be discussing new ones in [#api-standards](#) on Slack, so please join us if you're interested in contributing. Standards themselves are still a work in progress, and can be [viewed online](#).

Open API Spec

We document APIs using [Open API 3.0](#) (formerly known as Swagger). These documents are in the repo that powers this website.

General API Guidelines



These are general API guidelines:

- Updating resources will fail if required parameters are not set.
- Unsent optional parameters will not be overwritten with default values.
- Response for a POST or GET will be the same resource.
- All APIs that allow updating a resource will also allow fetching a resource.
- In each context properties that are always unknown may be omitted. For example,

Building APIs is Easy

Making a Good API is Hard

Information wants to be free

Your Customers Dictate the API

Developers can tell when you cut corners

- Your API is not just a layer over your database
 - If it is, just use GraphQL
- Think about Resources in terms of how your users will use your API
- You are paid to write a useful API for users, not shift work to them

Methods and Status Codes

What are you doing and what happened as a result?

Methods and Codes

- **Methods** - Describe what I am doing
- **Status Codes** - Describe what happened

Methods

What do I want to do?

HTTP Methods

Also call Verbs

- **GET** - Return a Resource
- **POST** - Create a Resource
- **PUT** - Update a Resource
- **DELETE** - Remote a Resource

More HTTP Methods

- **PATCH** - Partially update a Resource
- **HEAD** - Return only Headers for a Resource
- **OPTIONS** - Discover methods allowed on a Resource
- **TRACE** - Echo the Request Back

Status Codes

What Happened?

HTTP Response Codes

- **1xx** - Informational responses
- **2xx** - Success responses
- **3xx** - Redirection messages
- **4xx** - Client error responses
- **5xx** Server error responses

2xx Codes - Everything Worked

200 OK

The request succeed, here's the information

- GET - Return the resource
- HEAD - Return the headers for the resource without the body
- PUT/POST - The creation or update succeed and the new resource is returned
- TRACE - Return the received request

201 Created

The resource was created, but we don't need to return the resource itself

- POST - The new resource was created
- PUT - Indicates a new resource was created at the route

2xx Codes - Everything Worked

202 Accepted

Accepted the request and will act upon it later

204 No Content

The request succeeded but there is no content to display.

Useful for DELETE requests

4xx Codes - It's not Me, it's You

400 Bad Request

Generic error. The client can fix something and try again.

401 Unauthorized

There was a failure with authentication.

4xx Codes - It's not Me, it's You

403 Forbidden

You authenticated, but aren't allowed to be here.

Many systems will mask this with a 404 to avoid information leakage

404 Not Found

You are allowed to be here, but the resource does not exist

4xx Codes - It's not Me, it's You

405 Method Not Allowed

The URI is valid, but the requested method cannot be performed.

- GET a resource, but cannot be DELETE-d

409 Conflict

Your request was valid but conflicts with the state of something else.

4xx Codes - It's not Me, it's You

410 Gone

The resource used to exist, but no longer does.

415 Unsupported Media Type

The requested media type cannot be supported.

4xx Codes - It's not Me, it's You

429 Too Many Requests

You have sent too many requests - further requests are throttled.

Requests

Requests

- Consistency provides comfortability
- Standards provides confidence
- Be flexible

Responses

Responses

- Consistency provides comfortability
- Structure drives discoverability
- Standards provides confidence

HATEOS

Hypermedia as the Engine of Application State

HAL

Hypertext Application Language

https://stateless.group/hal_specification.html

```
{
  "_links": {
    "self": {
      "href": "/calls/63f61863-4a51-4f6b-86e1-46edebcf9356"
    }
  },
  "uuid": "63f61863-4a51-4f6b-86e1-46edebcf9356",
  "conversation_uuid": "CON-f972836a-550f-45fa-956c-12a2ab5b7d22",
  "to": {
    "type": "phone",
    "number": "14155550100",
    "dtmfAnswer": "p*123#"
  },
  "from": {
    "type": "phone",
    "number": "447700900001"
  },
  "status": "completed",
  "direction": "outbound",
  "rate": "0.39",
  "price": "23.40",
  "duration": "60",
  "start_time": "2020-01-01 12:00:00",
  "end_time": "2020-01-01 12:00:00",
  "network": "65512"
}
```

```
{
  "_links": {
    "self": {
      "href": "/calls/63f61863-4a51-4f6b-86e1-46edebcf9356"
    }
  },
  // ...
}
```

```
{
  "_links": {
    "self": {
      "href": "/calls/{uuid}",
      "templateRequired": ["uuid"]
    }
  },
  "uuid": "63f61863-4a51-4f6b-86e1-46edebcf9356",
  // ...
}
```

```
{
  "_links": {
    "self": {
      "href": "/calls/63f61863-4a51-4f6b-86e1-46edebcf9356"
    },
    "conversation": {
      "href": "/conversation/CON-63f61863-4a51-4f6b-86e1-46edebcf9356"
    },
    "recordings": {
      "href": "/recordings/63f61863-4a51-4f6b-86e1-46edebcf9356"
    }
  },
  // ...
}
```

Errors

[RFC 9457](#)

```
{
  "type": "https://example.com/probs/out-of-credit",
  "title": "You do not have enough credit.",
  "detail": "Your current balance is 30, but that costs 50.",
  "instance": "/account/12345/messages/abc",
  "balance": 30,
  "accounts": ["/account/12345",
               "/account/67890"]
}
```

API Versioning

APIs Change Over Time

- Global API Versioning
 - /v2/verify
 - /v2/calls
- Resource API Versioning
 - /v2/verify
 - /v1/calls
- Mime-Based Versioning
 - Accept: application/vnd.vonage.verify.v2+json
- Method-Based Versioning
 - GET /v2/verify
 - POST /v1/verify

What Happens when an API Changes?

- Is there a Backwards Compatibility break?
- How do you notify customers when things are being sunset?
- How do you notify customers when things are being added?
- Evolution vs Versioning

Documentation

Multiple Ways

- OpenAPI to document the structure
- HAL to document relationships
- Diataxis for developer documentation

Resources

- [Mozilla HTTP Response Codes](#)
- [Vonage API Standards](#)
- [HAL Writeup](#)
- [APIs You Won't Hate](#)
- [OpenAPI Initiative](#)
- [Diataxis](#)

Exceptions to the Rule

It's OK to be different

It's all about servicing the customer

- Determine what the API is doing
- Make the API route clear in its intentions
- Stick to standards as much as possible

<https://joind.in/talk/b36b9>

Chris Tankersley

**Developer Relations Tooling
Manager**

@dragonmantank.bsky.social
@dragonmantank@phpc.social

chris.tankersley@vonage.com
chris@ctankersley.com

