Twilio Messaging API Review

Review Date: *February 2nd, 2024*
Reviewer: *Jon Harper*

This is a review of the Twilio Messaging API. This review intends to look at the surface area of the messaging API and provide a snapshot the teams producing APIs at Twilio can consider, but anyone else designing and delivering an API can put it to use as well.

What is Done Well..

The Twilio messagingAPI is obviously robust, and possesses many of the common elements expected of a REST API, providing industrial-grade messaging solutions delivered as a modern API.

OpenAPI

Twilio's providing an OpenAPI provides a machine-readable contract for the Message API, and usage of version 3.1.0 of OpenAPI reflects what the latest tooling will expect when importing the contract for integration.

- ✅ 3.1.0 Version of OpenAPI

Info

Twilio hits on all the notes when it comes to providing the meta data needed for an API, as defined using the info block of an API, providing everything you need to understand what is going on with an API.

- ✅ Info Title
- ✅ Info Description
- ✅ Contact Object
- ✅ Contact Name
- ✅ Contact URL
- ✅ Contact Email
- ✅ Terms of Service
- ✅ License Object
- ✅ Info Version
- ✅ Semantic Versioning

Paths

Twilio does not have a trailing slash on their paths, and provides descriptions of what each path delivers, helping enrich documentation for the API.

- ✅ Path Trailing Slash
- ✅ Path Description

Operations

The operations for the Twilio Messaging API provide rich descriptions of what they do, as well as unique operation identifiers, and tags to help with discovery and organization–providing a rich set of programmatic operations.

- ✅ Operation Description
- ✅ Operation ID.
- ✅ Operation Tags

Parameters

The Twilio Messaging APIs has the fundamentals when it comes to query and path parameters, properly identifying them, defining which are required, while providing descriptions and schema type.

- ✅ Parameters In
- ✅ Parameter Description
- ✅ Parameter has a required property.
- ✅ Parameter Schema Type

Request Bodies

The Twilio messaging API employs an x-ww-form-urlo-encoded media type, leveraging body payloads for POST commands with descriptions, and schema supporting request payloads.

- ✅ Request Body POST
- ✅ Request Body Application X WWW Form URL Encoded
- ✅ Request Body Schema
- ✅ Request Body Schema Required
- ✅ Schema Description
- ✅ Request Body Schema Property Array Items
- ✅ Request Body Schema Property Array MaxItems

Responses

The design of responses for the Twilio Messaging API reflects what you expect from a simple modern web API utilizing consistent success status codes, with application/json media type, and well-defined schema for each response.

- ✅ GET 200 Status Code
- ✅ POST 201 Status Code
- ✅ JSON Media Type GET
- ✅ JSON Media Type POST
- ✅ Schema GET
- ✅ Schema POST

Schema

Twilio does a good job defining the shapes of it’s payloads, providing type, format, required, enum, and descriptions for schema, while also strengthening properties for validation with string and array min and max ranges, and even use of regex to further tighten things down.

- ✅ Schema Type
- ✅ Schema Required
- ✅ Schema Enum
- ✅ Schema Properties Format
- ✅ Schema Description
- ✅ Schema Property String Maxlength
- ✅ Schema Property String Pattern
- ✅ Schema Property Array Items
- ✅ Schema Property Array MaxItems

What Can Use Improvement…

While the Twilio Messaging API possesses some common API design patterns it is pretty complex, non-standardized, and could use a number of structural refinements. The cognitive load for the API could be reduced for a version 2.0 with a handful of standardizations that you see at Twilio, as well as with other providers.

Paths

The paths for the Twilio Messaging are not very conformant with simple RESTful design–paths should not be pascal case, versioning should be avoided in the path, and acronyms are not intuitive in API design.

- ⚠ Path Segments Pascal Case
- ⚠ Version In The Path
- ⚠ Acronyms In The Path

Operations

All of the operations for the Twilio Messaging API lack the summary and description to properly onboard consumers with the value each one delivers, and does not have any unique identifiers, making code generation and automated integration more difficult.

- ⚠ Operation Summary
- ⚠ Operation Description
- ⚠ Operation ID Camel Case

Parameters

The parameters for the Twilio Messaging API have most of the details you need to put them to work, but it wasn't always clear which parameters are required when putting each operation to work in applications.

- ⚠ Parameters MUST have a required property.

Request Bodies

The request bodies for the Twilio Messaging API don't always possess a request body, and there isn't always a description to help understand what they contain, and the shape of the schema isn't always fully defined.

- ⚠ Request Body POST
- ⚠ Schema Description
- ⚠ Request Body Schema Property Array MaxItems

Responses

The responses for the Twilio Messaging API aren't always consistent with each other, let alone other common approaches to returning data, and do not possess a standard set of error codes that consumers will expect when encountering problems.

- ⚠ GET 200 Status Code
- ⚠ POST 201 Status Code
- ⚠ 404 Status Code for DELETE Responses
- ⚠ 500 Status Code for GET Responses
- ⚠ 500 Status Code for POST Responses
- ⚠ 500 Status Code for DELETE Responses

Schema

All of the schema for the Twilio Messaging API can use a lot more details to help reduce friction for consumers, helping further flesh out the shape and other details so that requests and responses can be validated, but allow for code generation and other aspects of operations to be more effective.

- ⚠ Schema Description
- ⚠ Schema Required
- ⚠ Schema Properties Type
- ⚠ Require schema property string maxlength.
- ⚠ Schema Property Array MaxItems

Conclusion

Twilio Messaging API is clearly robust, but without the proper design rigor and other work, it has gotten very busy and somewhat inconsistent. It is clear from the operations that it is robust and in heavy rotation by applications, but it is rough, not smooth like most REST APIs.

With a little more investment in the Twilio Messaging API it would be a first-class API. All the parts and pieces are there, they just need to be refined–smoothing out the rough edges and making sure it is consistent and standardized with what API consumers are expecting.