

API and Push Notifications

Arnav Gupta

December 7, 2024

Contents

1	REST API	1
---	----------	---

1 REST API

API defines how software can be used by other software.

API for a library is functions/classes it provides. For software services, API is from network remote procedure calls.

Network-level APIs can be REST, SOAP, Thrift, Protobufs, or GraphQL. These usually require some form of auth.

HTTP methods: GET (request data), POST (post data), PUT (create or replace document), DELETE (delete document), HEAD (just return headers)

Response codes: 200 OK, 301 Moved Permanently, 403 Forbidden, 404 Not Found, 500 Internal Server Error

Idempotent request can be repeated without changing the result.

In HTTP, every method except POST should be idempotent.

Repeating PUT should just replace with the same and repeating DELETE should only do one deletion.

REST API has inputs:

- method
- path

- query params
- headers
- body

REST API has response

- status code
- headers
- body

Custom HTTP headers are undesired (better to build on top of HTTP rather than alter).

APIs require API key or access token for authentication.

Paths represent resources (data or system objects).

JSON is data format returned by most REST APIs. It allows arbitrary nesting and spaces are ignored, except withing quotes.

XML is older than JSON and less common (more complex). HTML is XML that defines a web page. Components are text, tags, and attributes.

Data files are byte arrays and messages sent over the network are serial streams of bytes.

Serialization: convert data object into sequence of bytes

References (pointers) make serialization non-trivial, like deciding how to repeat referenced objects and circular references.

Using references requires more than 1 pass through the data:

- **producer** must find and store all referenced objects before printing
- **consumer** may need to read more before finding the data being referred to

Using HTTP is helpful since it is already so used:

- encryption
- compression
- every programming language has HTTP client libraries

- many server frameworks to choose from that handle encryption, queuing, DB connection pooling
- web proxies and caches can be reused
- HTTP response codes are generic enough to be adapted to other services

HTTP has some unneeded complexities and human-readable headers have some overhead.

Thrift and Protobufs are alternatives for network APIs, with more space-efficient (less human-readable messages). Without HTTP overhead, less processing on both sides. Less processing on both sides.

To use these, specify a list of functions for the API, tools generate libraries to easily use the API in the language of choice.

Message complexity is usually not a concern, so REST remains the most popular network API format.