Cheatography

API Hierarchy of Needs Cheat Sheet

by [deleted] via cheatography.com/2754/cs/17350/

Introduction

The API hierarchy of needs is inspired by the work of Abraham Maslow, a psychologist who created a theory that explains how human needs are fulfilled — the Maslow's hierarchy of needs.
While Maslow's goal was to understand and explain the priorities of human needs, from breathing and feeding to self-esteem and morality, the API hierarchy of needs explains different characteristics that make an API usable.
The hierarchy of needs is a pyramid divided into five different layers that represent different characteristics that you should consider when launching and maintaining an API.

Source: Bruno Pedro is a Web and Business developer with over fifteen years' experience in both startups and large corporations.
http://apiux.com/2013/05/29/api-hierarchy-needs/#more-454

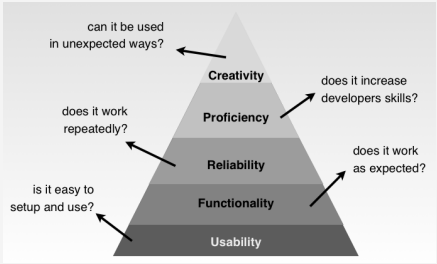
Usability

Usability is at the bottom of the pyramid because it's the most important characteristic that will dictate the success of your API among developers.
Is your API easy to set up and use by developers? You should make sure that any developer doesn't take more than 3 seconds to understand what your API does, 30 seconds to find the correct endpoint and 3 minutes to start consuming it (the 3:30:3 Rule, by Ori Pekelman).
Other usability aspects involve, among other things, using an appropriate authentication scheme, offering easy to use endpoint URLs, and responding standard HTTP error codes.

Functionality

Does your API work as expected? Your API endpoints should behave exactly as announced on the documentation. In situations where unexpected things happen, you should offer easy to consume error codes.
There are tools that help you test your API thoroughly, so you don't have an excuse not to make sure what you ship really works. SmartBear and Runscope should be under your radar, if you really care about your API.

API Hierarchy of Needs



Reliability

Does your API work flawlessly repeatedly? You should care about uptime, rate limiting and throttling, to make sure that every API call is handled properly according to your usage policies. If you have a business model around your API, then you should pay extra attention to this reliability.
Again, you don't have to reinvent the wheel. Products like 3scale offer all these tools so you can sleep better without worrying if your API is down.

Proficiency

Does your API increase developers skills? After fulfilling all the previous needs, you should provide ready to consume documentation and libraries that educate developers, amplifying their skills. If possible, offer tools that encourage discussion and participation.
You should take a look at tools like apiary.io and Swagger, that let you describe and document your API programmatically.

Creativity

The last layer of the pyramid is related with all the new things developers can build with your API. Can your API be used in unexpected ways? You should foster innovation among your API developer community so new products can be built by mixing different APIs together. If you're too rigid about what developers can or cannot do, your API popularity will certainly decrease.