Is there still a difference between a library and an API?

Asked 9 years, 9 months ago Active 2 years, 2 months ago Viewed 29k times

53

33

Whenever I ask people about the difference between an API and a library, I get different opinions. Some give this kind of definition, saying that an API is a spec and a library is an implementation...

Some will tell you this type of definition, that an API is a bunch of mapped out functions, and a Library is just the distribution in compiled form.

All this makes me wonder, in a world of web code, frameworks and open-source, is there really a practical difference anymore? Could a library like jQuery or cURL crossover into the definition of an API?

Also, do frameworks cross over into this category at all? Is there part of Rails or Zend that could be more "API-like," or "libraryesque"?

Really looking forward to some enlightening thoughts :)

api frameworks paradigms

asked Sep 9 '10 at 16:45

Trafalmadorian
2,332 2 18 13

"Is there still a difference between a library and an API?" Answer: Very much so! – AnBisw Aug 9 '14 at 9:18

7 Answers

Active | Oldest | Votes

57

My view is that when I speak of an API, it means only the parts that are exposed to the programmer. If I speak of a 'library' then I also mean everything that is working "under the hood", though part of the library nevertheless.

answered Sep 9 '10 at 16:48

teukkam
4,107 1 22 34

Excellent distinction and a great point in how we think about writing reusable code. – Trafalmadorian Sep 9 '10 at 16:49

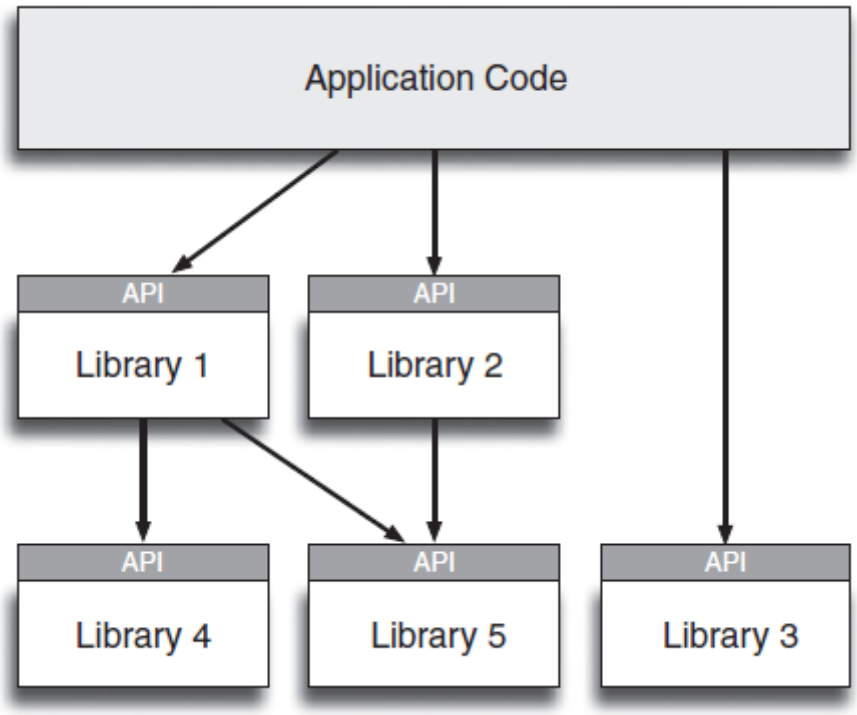
3 Today not only libraries have APIs, i think that's why this confusion happen. – Andrey Sep 9 '10 at 16:52

I dismissed this answer at first because I thought APIs were only a communication interface. I didn't know the definition includes ANY interface, even a "code interface", so to speak. Knowing that, this

21

A **library** contains re-usable chunks of code (a software program).

These re-usable codes of library is linked to your program through **APIs** (Application Programming Interfaces). That is, API is **interface to library** through which re-usable codes are linked to an application program.
In simple term it can be said that *an API is an interface between two software programs which facilitates the interaction between them*.



For example, in procedural languages like C, the library `math.c` contains the implementations of mathematical function, such as `sqrt`, `exp`, `log` etc. It contains the definition of all these functions.

These function can be referenced by using the API `math.h` which *describes and prescribes the expected behavior*.

edited Apr 5 '18 at 7:04

answered Jan 19 '15 at 15:32

haccks

93.3k 22 143 233

2 After a lot of searching, this answer makes the most sense. There are a lot of opinions out there, with very few backed up by examples. – Drenai Aug 15 '17 at 7:04

15

API is part of library that defines how it will interact with external code. Every library has API, API is sum of all public/exported stuff. Nowadays meaning of API is widened. we might call the way web site/service interact with code as API also. You can also tell that some device has API - the set of commands you can call.

Sometimes this terms can be mixed together. For example you have some server app (like TFS for example). It has API with it, and this API is implemented as a library. But this library is just a middle layer between you and not the one who executes your calls. But if library itself

edited Sep 9 '10 at 16:57

answered Sep 9 '10 at 16:48

Andrey

54.8k 9 106 152

11

I think that `Library` is a set of all classes and functions that can be used from our code to do our task easily. But the library can contain some of its private functions for its usage which it does not want to expose.

`API` is a part of library which is exposed to the user. So whatever documentation we have regarding a library, we call it an `API Documentation` because it contains only those classes and functions to which we have access.

edited Aug 9 '14 at 9:17

answered Sep 6 '12 at 12:04

Shashwat

2,228 6 31 51

3

we have first to define an interface ...

Interface :is the means by which 2 "things" talk to each other and exchange information. "things" could be a (1) human or (2) a running code of any sort (e.g. library ,desktop application , OS , web service ... etc).

if a human want to talks to a program he need Graphical user **interface** (GUI) or command line **interface** (CLI). both are types of interfaces that humans (but not programs) would like to use.

if however a running code (of any sort) want to talk to another running code (of any sort) it doesn't need or want a GUI or CLI ,it rather need an Application Programming **Interface** (API).

so to answer the original poster question : library is a type of running code and the API is the means by which this running code talk to other running codes.

answered Nov 20 '15 at 17:24

EKanadily

2,641 2 25 29

0

In Clear and concise language

Library: Collection of all classes and methods stored for re-usability

API: Part of **library** classes and methods which can be used by a user in his/her code.

answered Oct 12 '13 at 13:55

Devaarth

156 1 5

According to my perspective, whatever the functions are accessible to invoker , we can called as api in library file. library file having some of the functions which is private , we cannot

answered Dec 29 '14 at 6:41

nirmal

49 5