Q1. What is an API? Give an example, where an API is used in real life.

* An API is the interface that allows two independent software components to exchange information. An API acts as an intermediary between internal software functions and external ones, creating an exchange of information so seamless that it often goes unnoticed by the end user.
* Example
* Sharing flight information between airlines and [travel sites](https://www.kayak.com/horizon/sem/hotels/general?lang=en&skipapp=true&kw=-1&gclid=EAIaIQobChMI8MD8rNmt4gIVB9VkCh1DyAHkEAAYAiAAEgKYdPD_BwE&g_kw=%2Bkayak&aid=55963122471)
* Using [Google Maps](https://developers.google.com/maps/documentation/) in a rideshare app

Q2. Give advantages and disadvantages of using API.

Advantages

* **Applications**: Access to APIs ensures more flexibility in information-transfer processes.
* **Reach:** APIs let you create layers in apps in order to distribute information to different audiences.
* **Customization:** Furthermore, it can serve as a solution to create different experiences for users, letting protocols, functions, and commands be adapted according to specific demands.
* **Efficiency:** When you have content that is automatically published and made available on different channels simultaneously, APIs allow for more efficient data distribution.

**Disadvantages**

* As a single point of entry, an API is a gateway and can become a hacker's primary target. Once the API is compromised, all other applications and systems become vulnerable.

Q3. What is a Web API? Differentiate between API and Web API.

* A Web API stands for Application Programming Interface. It is a software application that allows two different applications or machines to interact with each other without any user interference. The API contains a complete set of rules and specifications used when interacting with any web application.

|  |  |
| --- | --- |
| **Web Service - API** | **Web API** |
| Web service is used to communicate between two machines on a network. | Web API is used as an interface between two different applications for communicating with each other. |
| It uses HTML requests that can be compressed, but XML data cannot be compressed. | Data can be compressed. |
| A web service is just an API wrapped in HTTP. | It's not always be a web-based |
| All Web Services are APIs. | All APIs are not web services. |
| It uses XML as structured data for exchanging information and communication. | It uses XML, JSON or plain data as structured data. |

Q4. Explain REST and SOAP Architecture. Mention shortcomings of SOAP.

## REST: representational state transfer

* REST is a set of architectural principles attuned to the needs of lightweight [web services](https://www.redhat.com/en/topics/cloud-computing/what-are-cloud-services) and [mobile applications](https://www.redhat.com/en/topics/mobile). Because it's a set of guidelines, it leaves the implementation of these recommendations to developers.
* When a request for data is sent to a REST API, it’s usually done through hypertext transfer protocol (commonly referred to as HTTP). Once a request is received, APIs designed for REST (called RESTful APIs or RESTful web services) can return messages in a variety of formats: HTML, XML, plain text, and JSON. JSON (JavaScript object notation) is favored as a message format because it can be read by any programming language (despite the name), is human- and machine-readable, and is lightweight. In this way, RESTful APIs are more flexible and can be easier to set up.

**SOAP: simple object access protocol**

* SOAP is a standard protocol that was first designed so that applications built with different languages and on different platforms could communicate. Because it is a protocol, it imposes built-in rules that increase its complexity and overhead, which can lead to longer page load times. However, these standards also offer built-in compliances that can make it preferable for enterprise scenarios. The built-in compliance standards include [security](https://www.redhat.com/en/topics/security/api-security), atomicity, consistency, isolation, and durability (ACID), which is a set of properties for ensuring reliable database transactions.

Q5. Differentiate between REST and SOAP.

These standards, called web service protocols, are sets of practices that dictate how data is communicated and how APIs are accessed. The two most popular protocols, REST and SOAP, dominate the competition, with the vast majority of open APIs [using one of the two](https://stormpath.com/blog/rest-vs-soap).

* **SOAP**, which stands for Simple Object Access Protocol, was the hands-down favorite for API developers until recently. Now 70% of public APIs follow REST protocols. That said, SOAP is still used in many major tech companies, offering support for legacy systems that may be compatible only with it.
* **REST**, which stands for Representational State Transfer, is the newcomer to web service protocols and allows for a greater number of data formats. Additionally, REST tends to be easier for developers to access while offering faster load times and better performance.

.