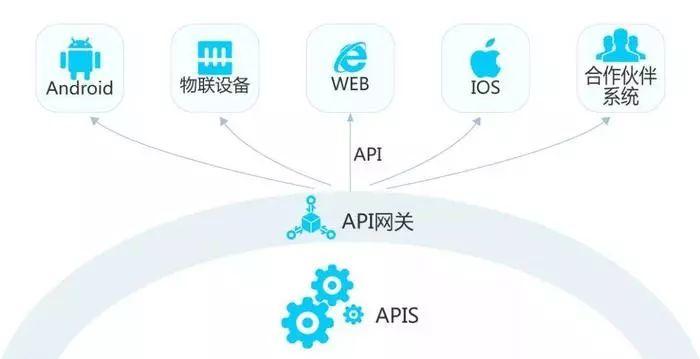
随着软件规模的日益庞大，常常需要把复杂的系统划分成小的组成部分，编程接口的设计十分重要，程序设计的实践中，编程接口的设计首先要使软件系统的职责得到合理划分，良好的接口设计可以降低系统各部分的相互依赖，提高组成单元的内聚性，降低组成单元间的耦合程度，从而提高系统的维护性和扩展性。API是接口的一种，在程序交互中具有重要的作用，而SDK与API有着密不可分的关系。

API

**（1）API的概念**

API即“应用程序编程接口”，是一些预先定义的函数，目的是作为“介面”沟通两个不同的东西，提供应用程序与开发人员基于某软件或硬件得以访问一组例程的能力，而又无需访问源码，或理解内部工作机制的细节。

其实就是别人已经写好的可以实现特定功能的函数，而你只需要根据他提供好的接口，也就是调用他的方法，传入他规定的参数，然后这个函数就会帮你实现这些功能。



从接口interface来说，在计算机领域是指两个不同事物之间交互的地方，大可以到两个完整的不同系统，小可以到两段程序。所以这个I就这么理解。在这个基础上，人和程序交互的地方，叫做UI，user interface,所有人输入的包括鼠标键盘触摸屏声音输入都算。那么程序和程序交互的就叫做API，所有非人对非人交互都通过API进行交互，所谓交互，其实就是传递数据，触发功能。

**（2）API应用案例**

示例场景：假如你是一家小企业，公司网站上有一个表格是用来给客户注册预约的。你想要凭借这些预约细节信息，让客户能够自动在谷歌日程上创建活动。

API使用：这就意味着，你的网站服务器需要直接与谷歌服务器进行对话，在掌握既定细节信息的情况下，申请创建活动。之后，你的服务器就会接收到谷歌的响应并进行处理，然后将相关信息发送回浏览器，比如说向用户发送一个确

认信息。

**（3）API产品——现在也有公司将API包装成产品**

案例：Weather Underground出售其天气数据API的访问权限给其他人。

**（4）API的分类**

API又分为（Windows、Linux、Unix等系统的）系统级API，及非操作系统级的自定义API。作为一种有效的代码封装模式，微软Windows的API开发模式已经为许多商业应用开发的公司所借鉴，并开发出某些商业应用系统的API函数予以发布，方便第三方进行功能扩展。如Google、苹果电脑公司，以及诺基亚等手机开发的API等等。

API又分为开放式API和私有API。顾名思义，开放式API即是向所有人公开的接口，允许任何人调用它并获取到它背后的数据，有时公司会将 API 作为其公共开放系统，也就是说，公司制定自己的系统接口标准，当需要执行系统整合、自定义和程序应用等操作时，公司所有成员都可以通过该接口标准调用源代码，该接口标准被称之为开放式API。私有API即接口未对外开放。



**SDK**

**（1）SDK的概念**

SDK即“软体开发工具包”，一般是一些被软件工程师用于为特定的软件包、软件框架、硬件平台、操作系统等建立应用软件的开发工具的集合。通俗点是指由第三方服务商提供的实现软件产品某项功能的工具包。

通常SDK是由专业性质的公司提供专业服务的集合，比如提供安卓开发工具、或者基于硬件开发的服务等。也有针对某项软件功能的SDK，如推送技术、图像识别技术、移动支付技术、语音识别分析技术等，在互联网开放的大趋势下，一些功能性的SDK已经被当作一个产品来运营。

开发者不需要再对产品的每个功能进行开发，选择合适稳定的SDK服务并花费很少的经历就可以在产品中集成某项功能。

**（2）SDK和API的区别**

SDK相当于开发集成工具环境，API就是数据接口。在SDK环境下调用API数据。



实际上SDK包含了API的定义，API定义一种能力，一种接口的规范，而SDK可以包含这种能力、包含这种规范。但是SDK又不完完全全只包含API以及API的实现，它是一个软件工具包，它还有很多其他辅助性的功能。

SDK 包含了使用 API 的必需资料，所以人们也常把仅使用 API 来编写 Windows 应用程序的开发方式叫做“SDK编程”。

**通俗语言解释**

**API**

前端调用后端数据的一个通道，就是我们俗说的接口，通过这个通道，可以访问到后端的数据，但是又无需调用源代码。

**SDK**

工程师为辅助开发某类软件的相关文档、范例和工具的集合，使用SDK可以提高开发效率，更简单的接入某个功能。

举例说明：一个产品想实现某个功能，可以找到相关的SDK，工程师直接接入SDK，就不用再重新开发了。