

Ramy Hakam

# *Restful API in PHP*

*Build a powerful Web services with your favorite language*

## *What is An API ?!*

*An API (Application Programming Interface) is best thought of as a contract provided by one piece of computer software to another.*



## Can My App contact with others Apps without API

### **Without API:**

*An app finds the current weather in Cairo by opening Weather.com and reading the webpage like a human does, interpreting the content.*

### **With API:**

*An app finds the current weather in Cairo by sending a message to the Weather.com API (in a structured format like JSON). The Weather.com API then replies with a structured response.*

## IS That an API concept for web only or not ?!

No,

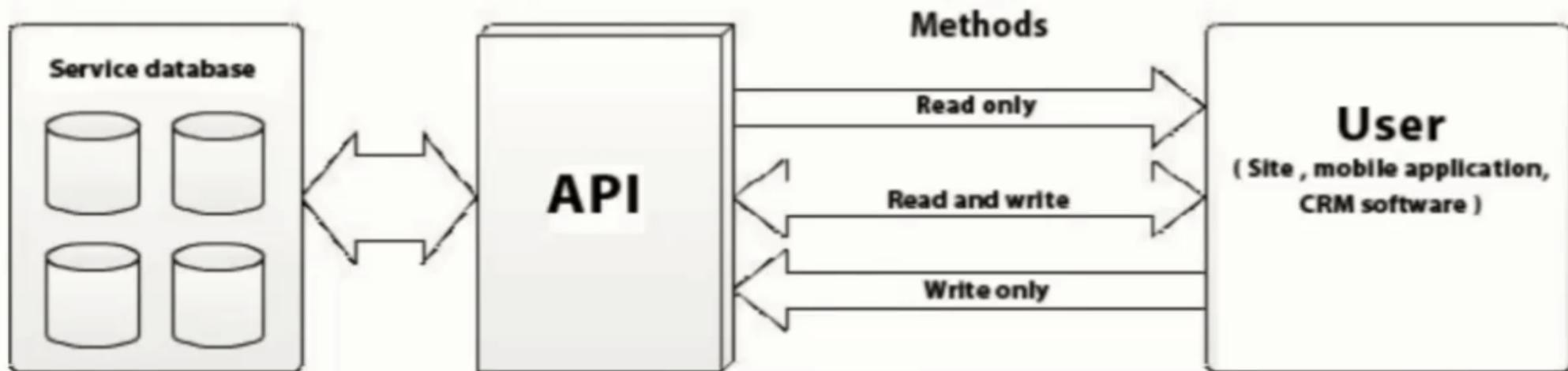
The API connect is not for web Only But every thing Around You in It world Contain APIs

Your Os Contain Enormous sets of APIS , Programming Languages Contain some Sets of APIs Softwares Contain API And others

## How API Works?!

*APIs Are Like User Interfaces--Just With Different Users in Mind  
The Users Are Others Softwares, Apps and Browsers.  
You Just want to build an UI for Theses to allow them to contact  
to your app in certain way as you built it.*

*In Web API example The app Send a request to your app and then  
You perform some work, then get the result and send the response  
To The sender app again.*



## Types of Web APIs

---

There are many classifications for the API  
But in this course we will focus on Web API that Used in the  
Interaction between web apps and servers

There are 3 different Types of These API

- 1- **SOAP** → Simple Object Access Protocol
- 2- **REST** → Representational State Transfer
- 3 **RPC** → Remote Procedure Call

## **What Restful API ?!**

*Rest (**R**epresentational **S**tate **T**ransfer) is the underlying architectural principle of the web.*

*which essentially refers to a style of web architecture that has many underlying characteristics and governs the behavior of clients and servers.*

*REST API lets you interact with anything that can send an HTTP request*



REST API

## **Client–server:**

*The client handles the front end and the server handles the backend and can both be replaced independently of each other.*

## **Stateless:**

*No client data is stored on the server between requests and session state is stored on the client.*

## **Cacheable:**



*Clients can cache response (just like browsers caching static elements of a web page) to improve performance.*

## **Resource-Based**

*Individual resources are identified in requests using URIs as resource identifiers. The resources themselves are conceptually separate from the representations that are*

## The “RESTful” Client Server Architecture

*HTTP, for example, has a very rich vocabulary in terms of verbs or methods, URIs, Internet media types, request and response codes, etc. REST uses these existing features of the HTTP protocol.*

*A RESTful web service (also called a RESTful web API) is a web service implemented using HTTP and the principles of REST. It is a collection of resources, with 3 defined aspects:*

- 1- The base URI for the web service, such as <http://example.com/resources/>*
- 2-The Internet media type of the data supported by the web service “XML,JSON,...”*
- 3-The set of operations supported by the web service using HTTP methods  
(e.g., POST, GET, PUT or DELETE).*

- 1- The base URI for the web service, such as `http://example.com/resources/`
- 2-The Internet media type of the data supported by the web service “`XML,JSON,...`”
- 3-The set of operations supported by the web service using HTTP methods  
(e.g., `POST, GET, PUT or DELETE`).



Ramy Hakam

# Restful API Design

## **1-Resource URL:**

*Provide the exact unique URL for this resource*

## **2-HTTP methods Support**

*Select which http this resource will support*

## **3- Media type Support:**

*Choose Which of media types can this resource send and receive*

## **4- WS internal Logic :**

*What will this Web service provide and what it its work on  
your Databases and logic*

## **5- API Authorization**

*Secure access for your API by adding a layer of Authorization*

# Restful API Design

## REST API Design

<b>GET</b>	/tasks - display all tasks
<b>POST</b>	/tasks - create a new task
<b>GET</b>	/tasks/{id} - display a task by ID
<b>PUT</b>	/tasks/{id} - update a task by ID
<b>DELETE</b>	/tasks/{id} - delete a task by ID

