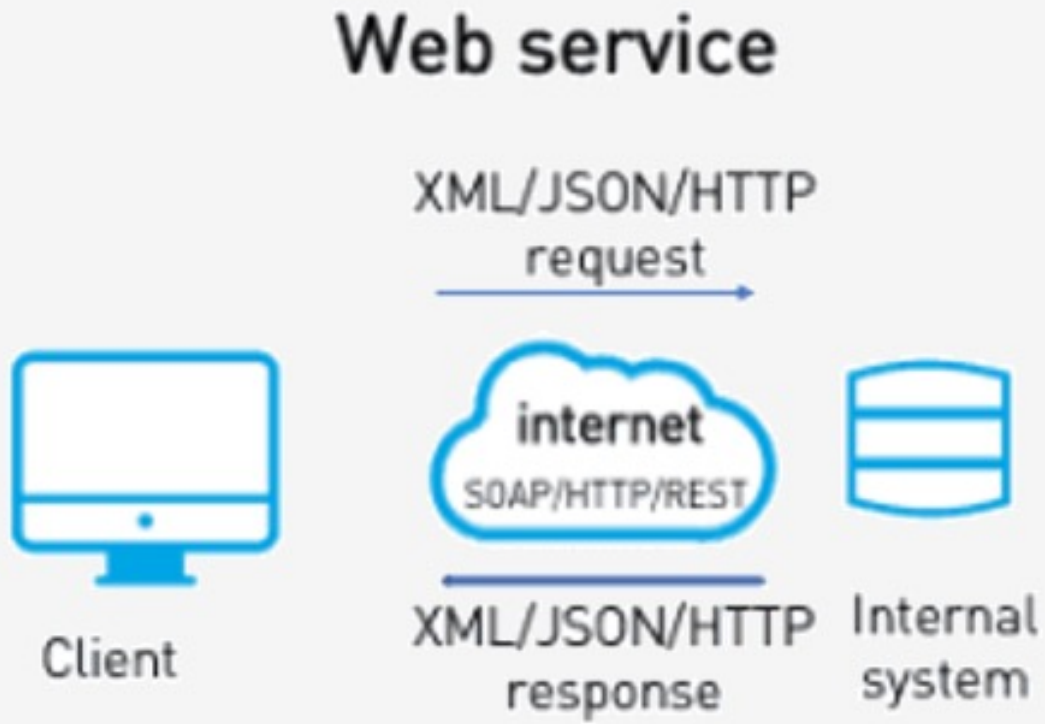
Web Services

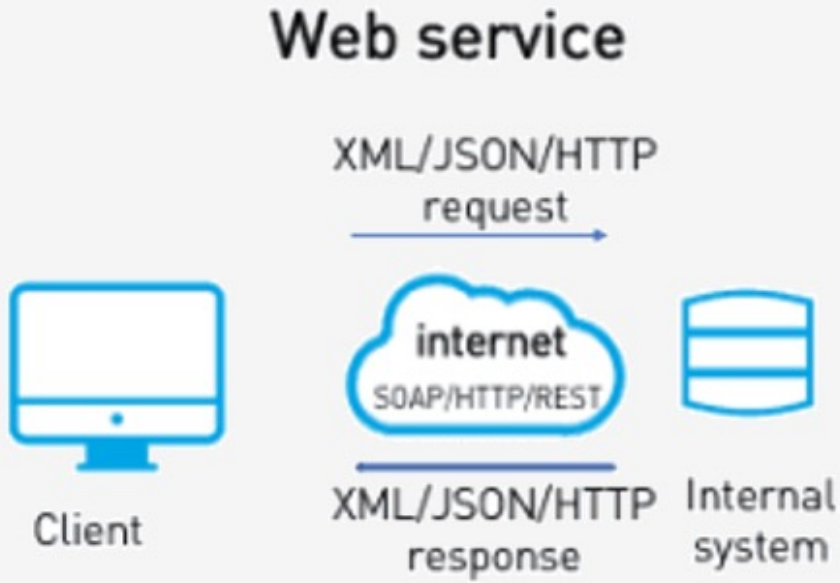


Source: https://www.w3.org/wiki/WebServices

Web Services

Web services provide a standard means of interoperating between different software applications, running on a variety of platforms and/or frameworks.

Web services are characterized by their great interoperability and extensibility, as well as their machine-processable descriptions thanks to the use of XML.



Source: https://www.w3.org/wiki/WebServices

Advantages of Web services

Interoperability: One of the advantages of web service is interoperability. Web services allow applications to communicate, exchange data and share services among themselves.

Usability: Web services are designed to be used like a web page request and receive data.

Reusability: Web Services are designed to be combined to deliver more added-value services.

Deployability: Web Services are deployed over Internet standards such as standard Apache, Axis2 to provide HTTP, WSDL driven services.

Types of Web Services

I. **SOAP Web Services:** SOAP (Simple Object Access Protocol) is an XML-based protocol for accessing web services. Its interface is described in a machine-processable format called **WSDL** (Web Service Definition Language) document. A web service is described by using a standard, formal XML notion that provides all necessary details like message format, transport protocols, and location to interact with the web service.

II. **REST Web Services:** REST (Representational State Transfer) is a style of software architecture. The data format is described by using JSON schema notation, and it requires the use of the HTTP transport protocol.

SOAP vs REST

- Social Media
- Web Chat
- Mobile

- Financial
- Telecommunication
- Payment Gateways

Differences between SOAP and REST

1

SOAP	REST
SOAP is a protocol.	REST is an architectural style.
SOAP **can't use REST** because it is a protocol.	REST **can use SOAP** web services because it is a concept and can use any protocol like HTTP, SOAP.
SOAP only permits XML.	REST permits many different data formats including plain text, HTML, XML, and JSON…
SOAP requires **more bandwidth** and more resources.	REST requires **less bandwidth** and less resources.

Differences between SOAP and REST

2

SOAP	REST
SOAP supports both **SMTP** and **HTTP** protocols.	REST requires the use of HTTP only.
SOAP is more reliable than REST.	REST is less secure than SOAP.
In most cases, SOAP is faster than REST.	REST is slower than SOAP.
SOAP defines its own security.	RESTful web services inherit security measures from the underlying transport.

HTTP methods supported by REST

GET - The GET method retrieves specific information from the server as identified by the request URI.

PUT - The PUT method requests that the message body sent with the request be stored under the location provided in the HTTP message.

/books

GET	/books	Lists all the books in the database
DELETE	/books/{bookId}	Deletes a book based on their id
POST	/books	Creates a Book
PUT	/books/{bookId}	Method to update a book
GET	/books/{bookId}	Retrieves a book based on their id

Source: https://docs.oracle.com/cd/E19182-01/821-0540/ug_restbd-methods_c/index.html

HTTP methods supported by REST

DELETE - The DELETE method deletes the specified resources.

POST - The POST method modifies data on the server from which a request was sent.

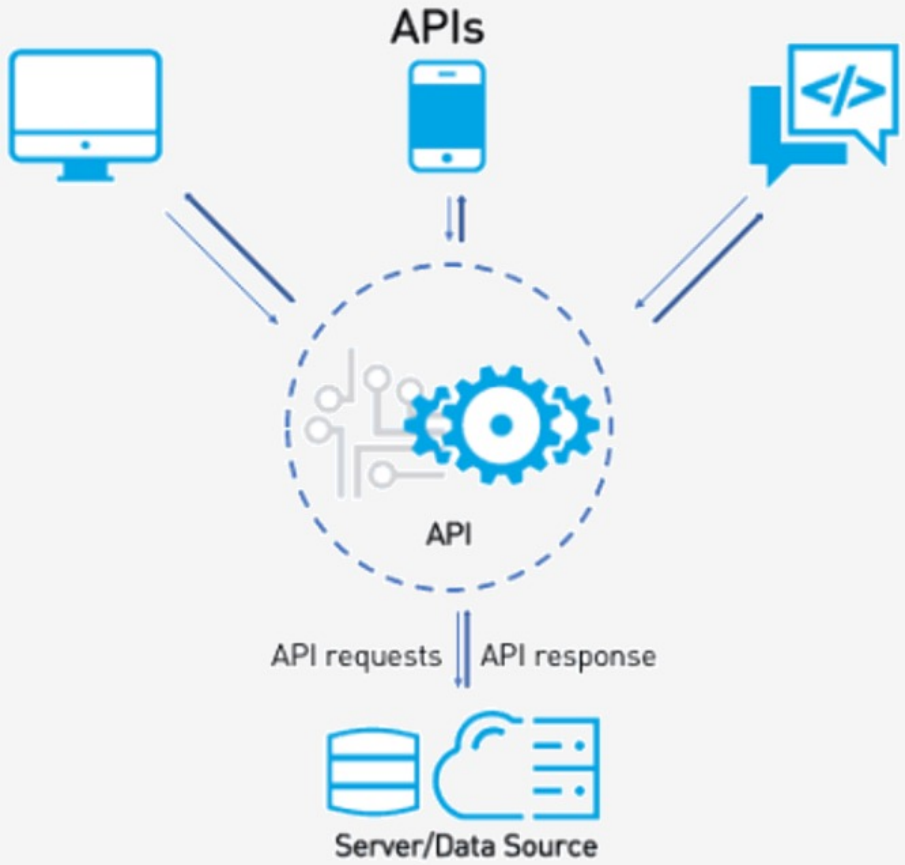
HEAD - The HEAD method is similar to the GET method except the message body is not returned in the response. The response only includes **meta-information**, such as a response code or corresponding headers.

/books

GET	/books	Lists all the books in the database
DELETE	/books/{bookId}	Deletes a book based on their id
POST	/books	Creates a Book
PUT	/books/{bookId}	Method to update a book
GET	/books/{bookId}	Retrieves a book based on their id

Source: https://docs.oracle.com/cd/E19182-01/821-0540/ug_restbd-methods_c/index.html

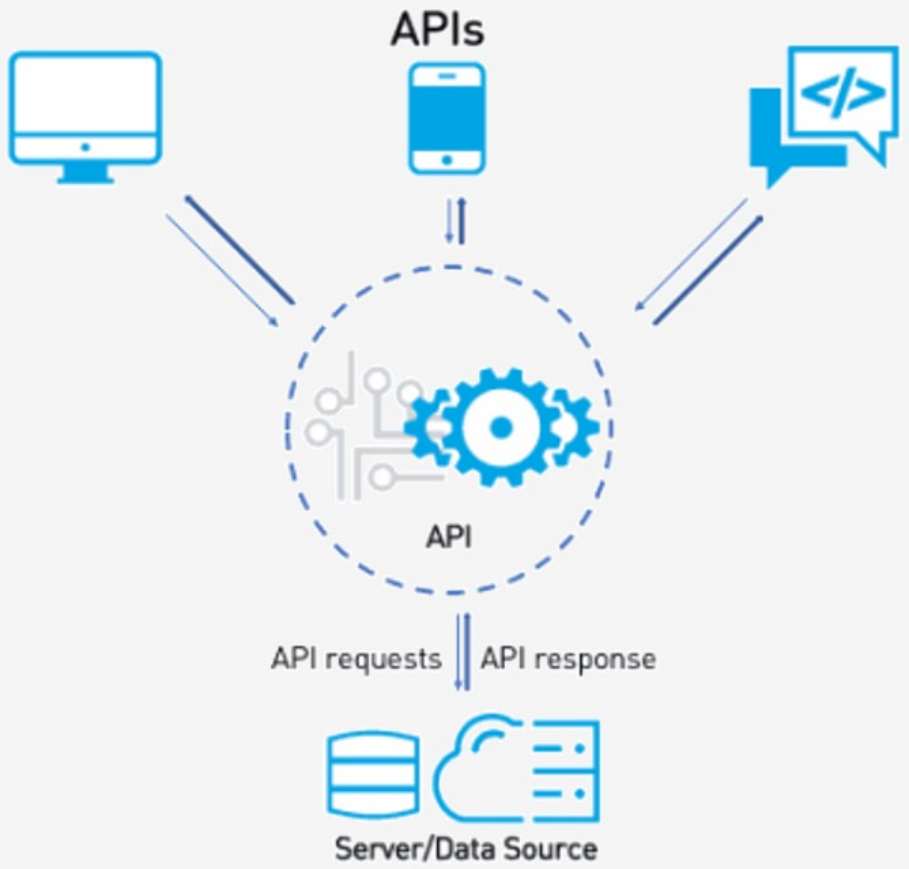
API (Application Programming Interface)



API (Application Programming Interface)

API is a protocol used as an interface by software components to communicate with each other.

An API serves as an interface between two different applications so that they can communicate with each other.



The differences between Web Services vs API

1

WEB SERVICE	API
All web services are APIs.	All APIs are not web services.
It can only be hosted on IIS.	It can be hosted within an application or IIS.
It is not open source but can be used by any client that understands XML.	It is open source and it can be used by any client that understands JSON or XML.

The differences between Web Services vs API

2

WEB SERVICE	API
It requires a SOAP protocol to receive and send data over the network, so it is not a light-weight architecture.	It is light-weight architecture and good for devices which have limited bandwidth, like mobile devices.
A Web service uses only three styles of use: **SOAP**, **REST** and **XML-RPC** for communication.	API may use any style of communication.
It only supports the HTTP protocol.	It supports the HTTP protocol: URL, Request/Response Headers, caching, versioning, content formats.

Top 5 API Testing Tools for 2018

Product	SoapUI	Katalon	POSTMAN	TRICENTIS	apigee
Application Under Test	API	Web (UI & API), Mobile apps	API	Web (UI & API), Mobile apps, SAP	API
Pricing	Paid + Free	Free	Paid + Free	Paid + Free	Paid + Free
Supported Platform	Windows Linux MacOS	Windows Linux MacOS	Windows Linux MacOS	Windows	Windows Linux MacOS
Ease of installing and use	Easy to setup & use	Easy to setup & use	Easy to setup & use	Easy to setup. Need training to properly use the tool	Require end-points management knowledge to use

Source: https://medium.com/@alicealdaine/top-10-api-testing-tools-rest-soap-services-5395cb03cfa9

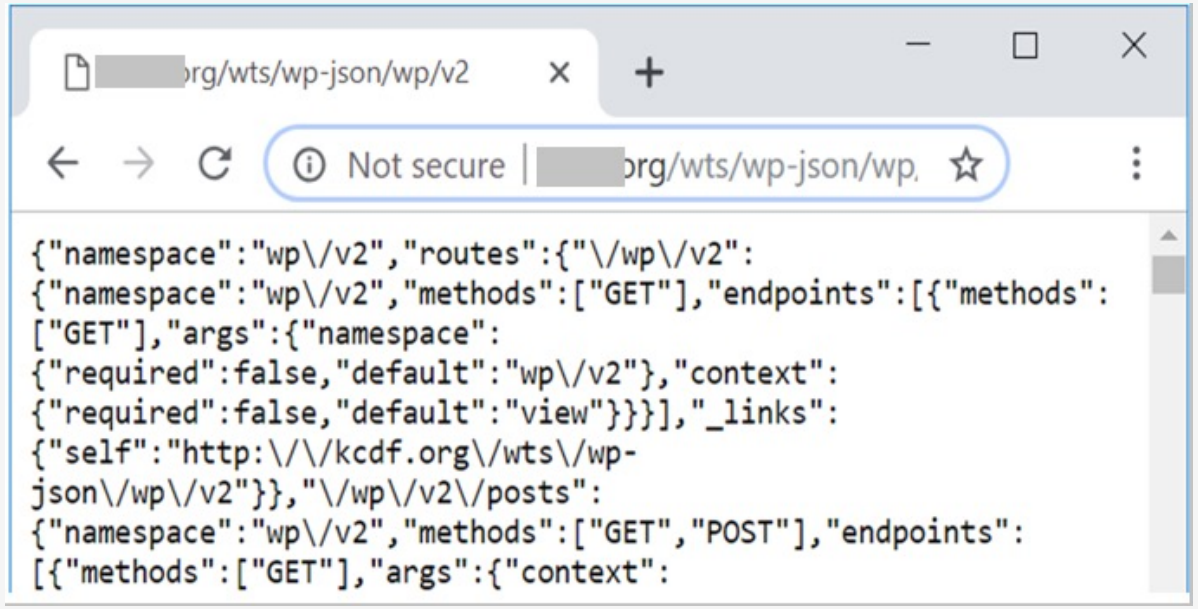
Web Technology and Services

Understanding and Using **REST APIs**

WordPress REST API

To access the WordPress REST API, you'll need to start with the following route:

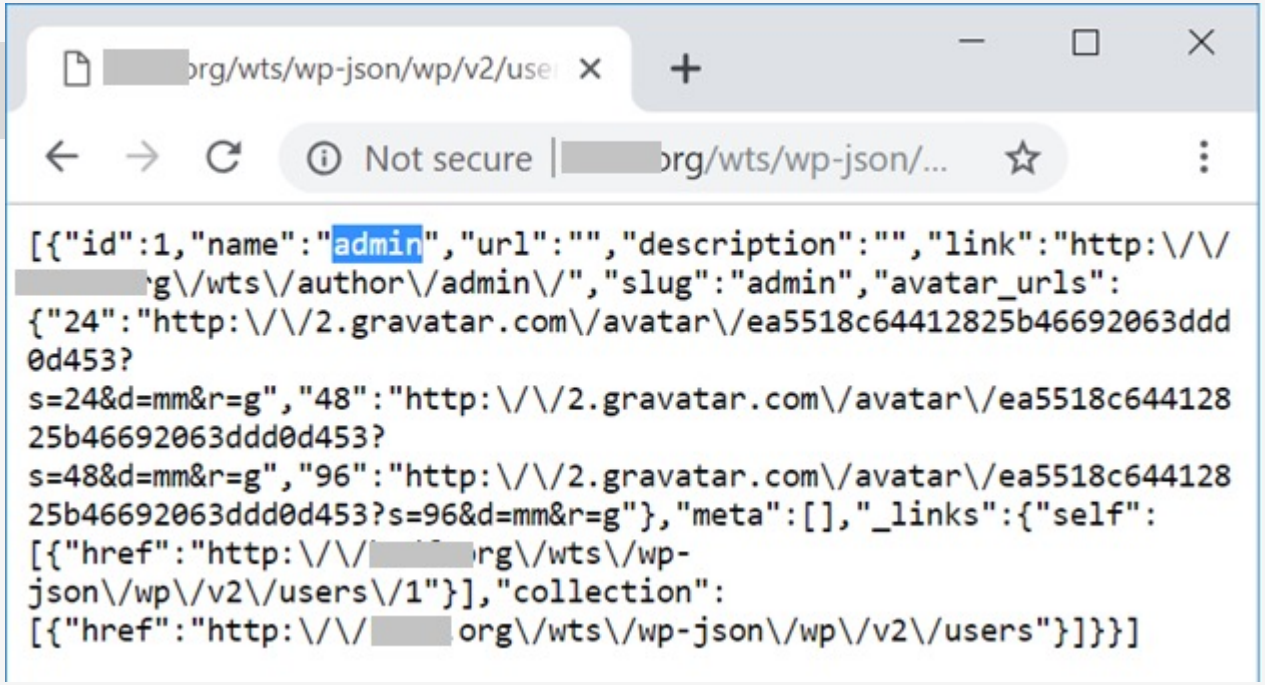
`<yoursite.com>/wp-json/wp/v2`



WordPress REST API

You can add onto this URL to access various types of data.

```
<yoursite.com>/wp-json/wp/v2/users
```

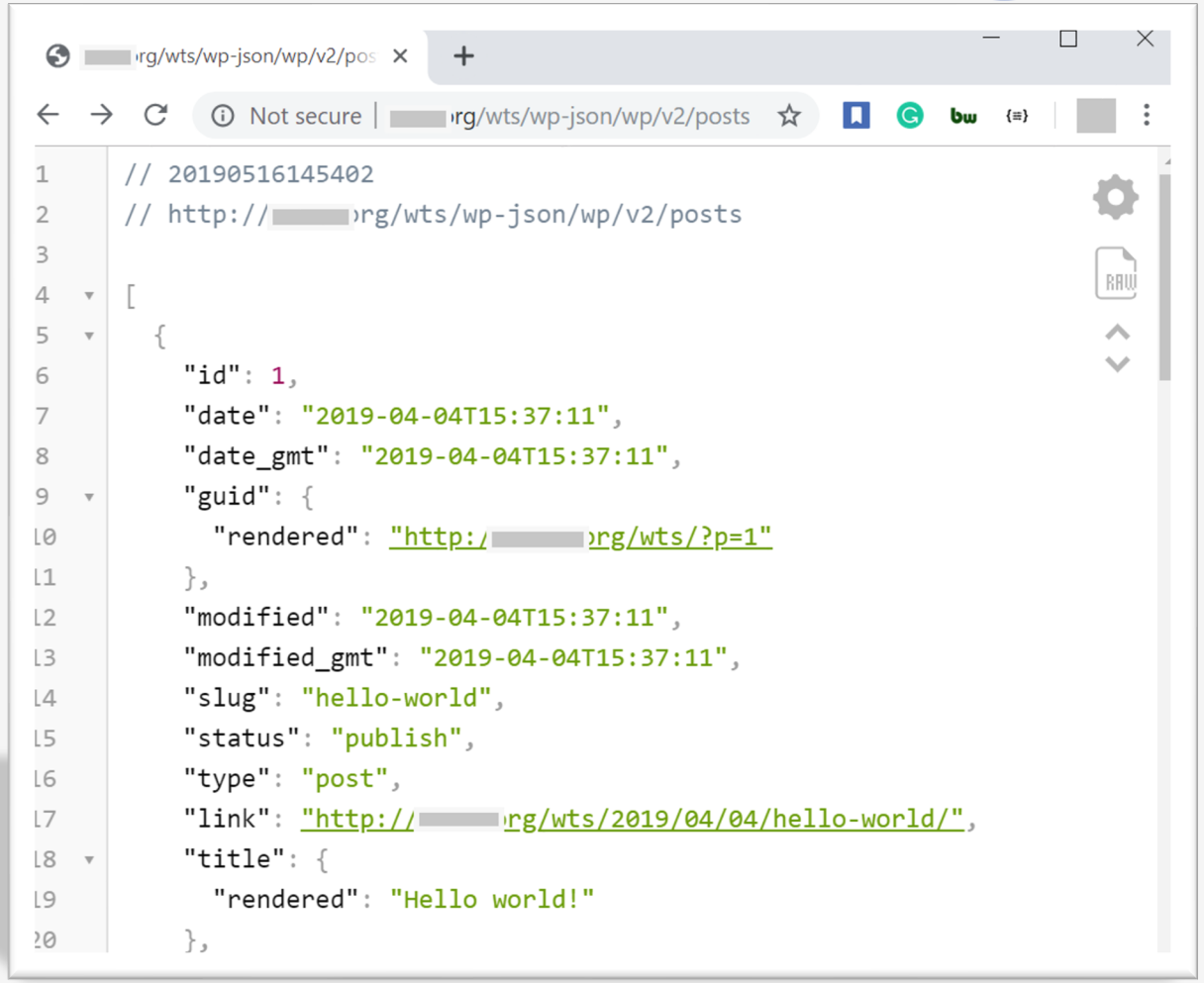


WordPress REST API

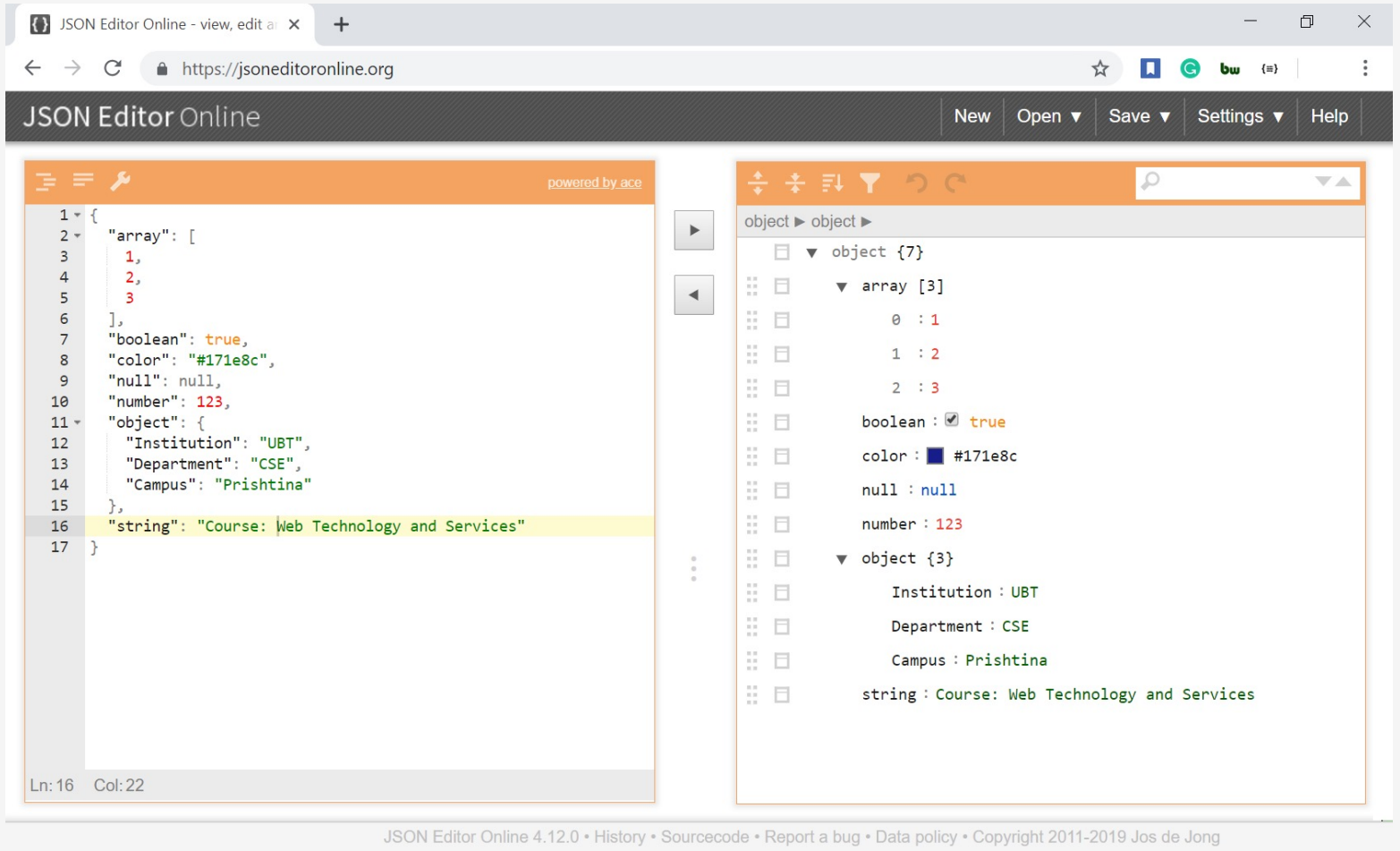
You can use the same basic route to view other types of data, such as your posts or pages. You can even search for subsets of the data that meet certain criteria. For example, you could retrieve all posts that include a specific term using this URL:

```
<yoursite.com>/wp-json/wp/v2/posts?=search[keyword]
```

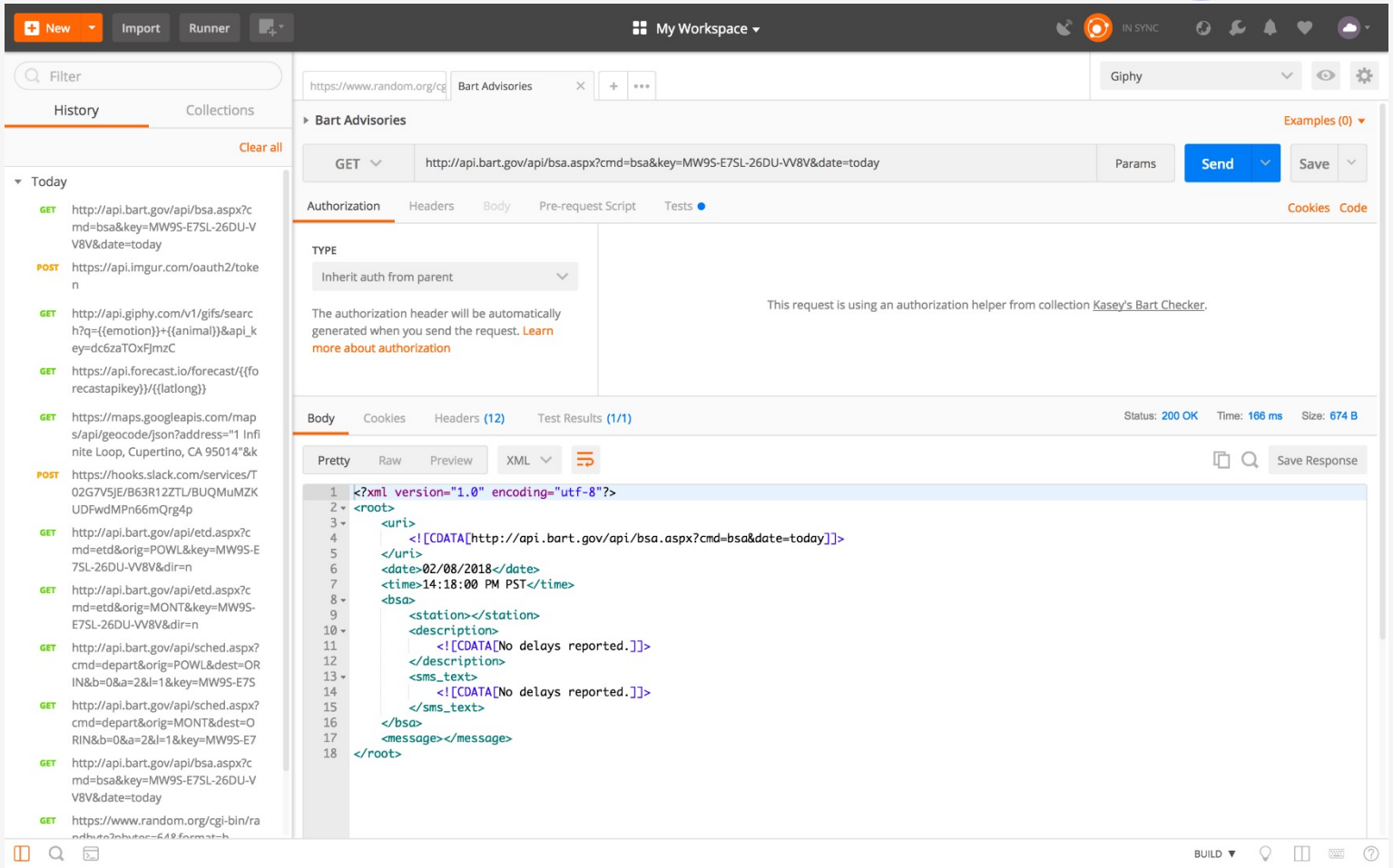
Browser Extension



JSON Online Editor



Postman Application



POSTMAN

JSON Values

Values must be one of the following data types

- an array
- a boolean
- a number
- a string
- an object (JSON object)
- null


```
{
  "array": [
    1,
    2,
    3
  ],
  "boolean": true,
  "color": "#171e8c",
  "null": null,
  "number": 123,
  "object": {
    "Institution": "UBT",
    "Department": "CSE",
    "Campus": "Prishtina"
  },
  "string": "Course: Web Technology and Services"
}
```

JSON Values

Values must be one of the following data types

- an array
- a boolean
- a number
- a string
- an object (JSON object)
- null


```
{
  "array": [
    1,
    2,
    3
  ],
  "boolean": true,
  "color": "#171e8c",
  "null": null,
  "number": 123,
  "object": {
    "Institution": "UBT",
    "Department": "CSE",
    "Campus": "Prishtina"
  },
  "string": "Course: Web Technology and Services"
}
```

Access Nested JSON objects

powered by ace

```
{
  "name":"Xhelal Jashari",
  "age":33,
  "Courses": {
  "Course1":"Computer Science 1",
  "Course2":"Web Programing",
  "Course3":"Web Technology and Services"
  }
}
```

Ln: 9 Col: 2

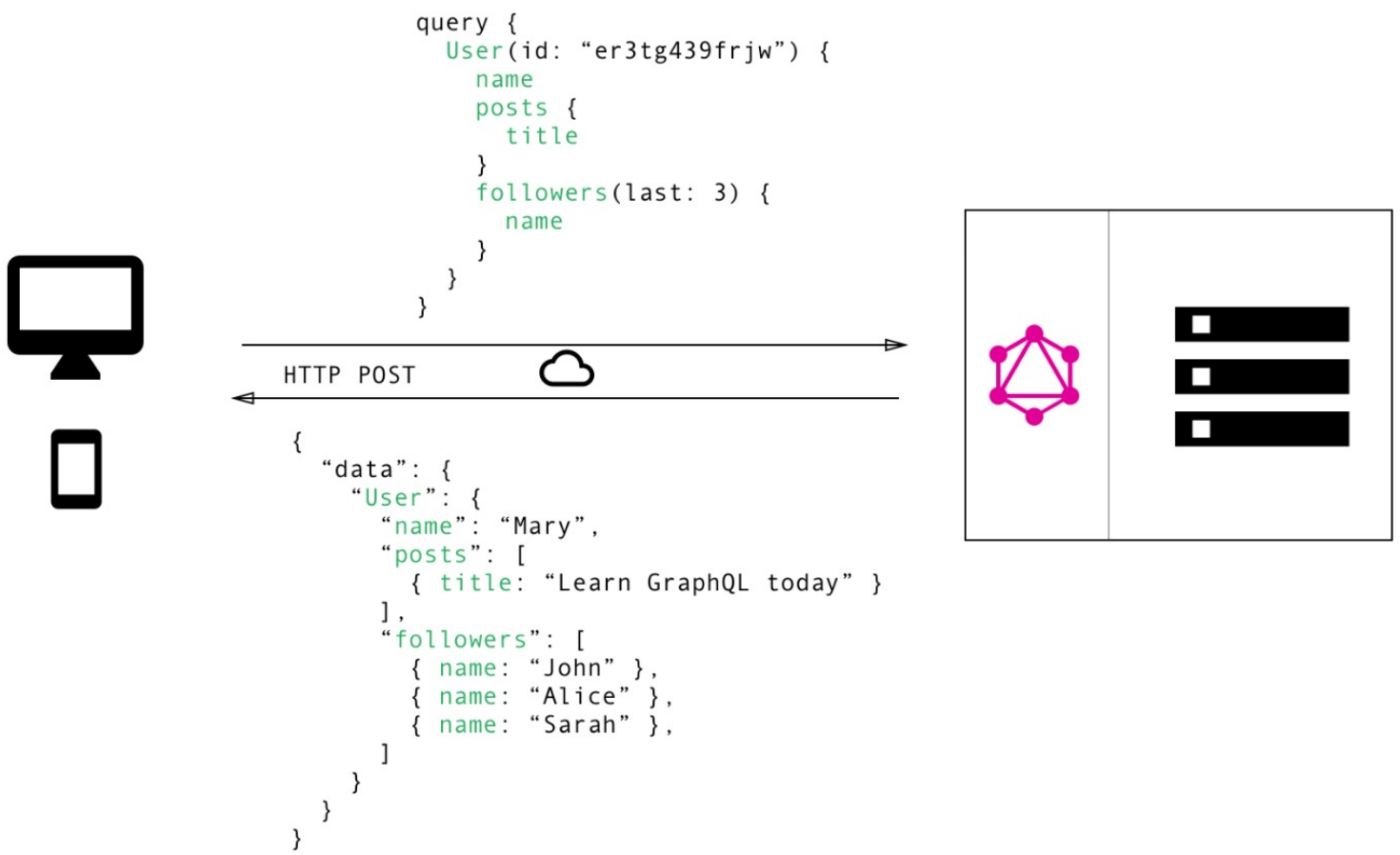
```
<!DOCTYPE html>
<html>
<title> Nested JSON objects </title>
<body>

<p>Nested JSON objects</p>

<p id="test"></p>

    <script>
    var myObj = {
      "name":"Xhelal Jashari",
      "age":33,
      "Courses": {
      "Course1":"Computer Science 1",
      "Course2":"Web Programing",
      "Course3":"Web Technology and Services"
      }
    }
    document.getElementById("test").innerHTML += myObj.name + "<br>";
    document.getElementById("test").innerHTML += myObj.Courses.Course3 + "<br>";
    //or:
    document.getElementById("test").innerHTML += myObj.Courses["Course3"];
    </script>

</body>
</html>
```

Modify values in a JSON object

```
<!DOCTYPE html>
<html>
<title> Mdify values in a JSON object </title>
<body>

<p>Modify values in a JSON object</p>

<p id="test"></p>

    <script>
    var myObj, i, x = "";
    var myObj = {
      "name":"Xhelal Jashari",
      "age":33,
      "Courses": {
      "Course1":"Computer Science 1",
      "Course2":"Web Programing",
      "Course3":"Web Technology and Services"
      }
    }
    myObj.Courses.Course1 = "Softare Architecture";

    for (i in myObj.Courses) {
      x += myObj.Courses[i] + "<br>";
    }

    document.getElementById("test").innerHTML = x;
    </script>

</body>
```

GraphQL

With REST, you have to make three requests to different endpoints to fetch the required data.
You're also *overfetching* since the endpoints return additional information that's not needed.



Using GraphQL, the client can specify exactly the data it needs in a *query*. Notice that the *structure* of the server's response follows precisely the nested structure defined in the query.

PHP JSON


```php
<?php
    $arr = array ('institution' => 'UBT',
                  'department' => 'CSE',
                  'course' => 'Web Technology and Services');
    echo json_encode($arr);
?>
```

{"institution":"UBT","department":"CSE","course":"Web Technology and Services"}


```php
<?php

    $json = file_get_contents('http://localhost/json_example1.php');
    $data = json_decode($json);


    echo "Copyright 2020" . $data -> institution . " - Department " . $data -> department . " - ". $data -> course .", All rights Reserved. "


?>
```

Create dynamic JSON File in PHP and MySql