

# JS-JSON- STORAGE-API

ANECO ACADEMY

# JSON

- JSON stands for JavaScript Object Notation.
- JSON used for store and exchange data between servers or remote servers.
- Its lightweight and self describing.
- JSON language independent.
- JSON object starts with '{' JSON array starts with '['.
- Easy to parse.
- JSON is purely a data format — it contains only properties, no methods.

# JSON

## Rules:

- Key must be a string which is denoted by “”.
- JSON supports the following data types,

- I. string
- li. number
- lii. Array.
- Iv. Object
- V. Boolean
- Vi. null

## **JSON doesn't support**

- I. Function and inbuilt methods.
- li. Undefined

# JSON

```
{
  "menu": {
    "id": "file",
    "value": "File",
    "popup": {
      "menuitem": [
        {
          "value": "New",
          "onclick": "CreateDoc()"
        },
        {
          "value": "Open",
          "onclick": "OpenDoc()"
        },
        {
          "value": "Save",
          "onclick": "SaveDoc()"
        }
      ]
    }
  }
}
```



# STORAGE

Local Storage	Session Storage	Cookies
It allows <b>10MB</b> of data to be stored.	It allows <b>5MB</b> of data to be stored.	The storage capacity is limited to 4KB of data.
The stored data is not deleted when the browser is closed.	The data is stored only for the session and will be deleted when the browser is closed.	The data can be set to expire at a certain time.
Local storage is useful for storing data that the user will need to access later, such as offline data.	Session storage is a great way to improve the performance of your web applications.	Cookies are a good choice for storing data that should not be persisted for a long time, such as session IDs.
This is especially useful for storing data that you want to persist even if the user closes the browser, such as preferences or settings.	Session storage is useful for storing data that is sensitive, such as login credentials.	Cookies are often used to store data that is not sensitive, such as user preferences
Local Storage is a new feature introduced in <b>HTML5</b>	Session Storage is a new feature introduced in <b>HTML5</b>	Cookies are the oldest ( <b>HTML4</b> ) and most wellknown mechanism.
The data is not sent with the request from the client to the server.	The data is not sent with the request from the client to the server	The data is sent with the request from the client to the server
The data is stored on the browser and system.	The data is stored on the browser only.	The data is stored on the browser only.

# LOCAL STORAGE

To store data, you use the **setItem()** method again. This time, you pass in the key as the first argument and the data you want to store as the second argument.

```
localStorage.setItem('key', 'value');
```

To retrieve data from Local Storage, you use the **getItem()** method. This method takes the key as an argument and returns the data that is stored under that key.

```
localStorage.getItem('key');
```

If you want to remove an item from Local Storage, you use the **removeItem()** method. This method takes the key as an argument and removes the data that is stored under that key.

```
localStorage.removeItem('key');
```



# SESSION STORAGE

To use Session Storage in your web applications, you'll need to use the **sessionStorage** object. This object provides access to the current session's storage.

The sessionStorage object has two methods

**setItem()** – This method sets a key/value pair in the session storage.

```
sessionStorage.setItem("name", "tutorialsPoint");
```

**getItem()** – This method retrieves the value of a key from the session storage.

```
var name = sessionStorage.getItem("name");
```

The sessionStorage object also has a couple of other properties –

- **length** – This property returns the number of key/value pairs in the session storage.
- **key()** – This method accepts an index as a parameter and returns the key at that index in the session storage



# COOKIE

## Setting a Cookie

A cookie is set using the **setItem()** method. This method accepts two arguments, the name of the cookie and the value of the cookie.

The name of the cookie is used to identify the cookie, and the value is the information that is to be stored in the cookie.

The following code sets a cookie with the name "name" and the value "tutorialsPoint".

```
document.cookie = "name=tutorialsPoint";
```

## Reading a Cookie

A cookie is read using the **getItem()** method. This method accepts the name of the cookie as an argument and returns the value of the cookie.

If the cookie does not exist, the *getItem()* method will return null.

The following code reads the "name" cookie and stores the value in the "user" variable.

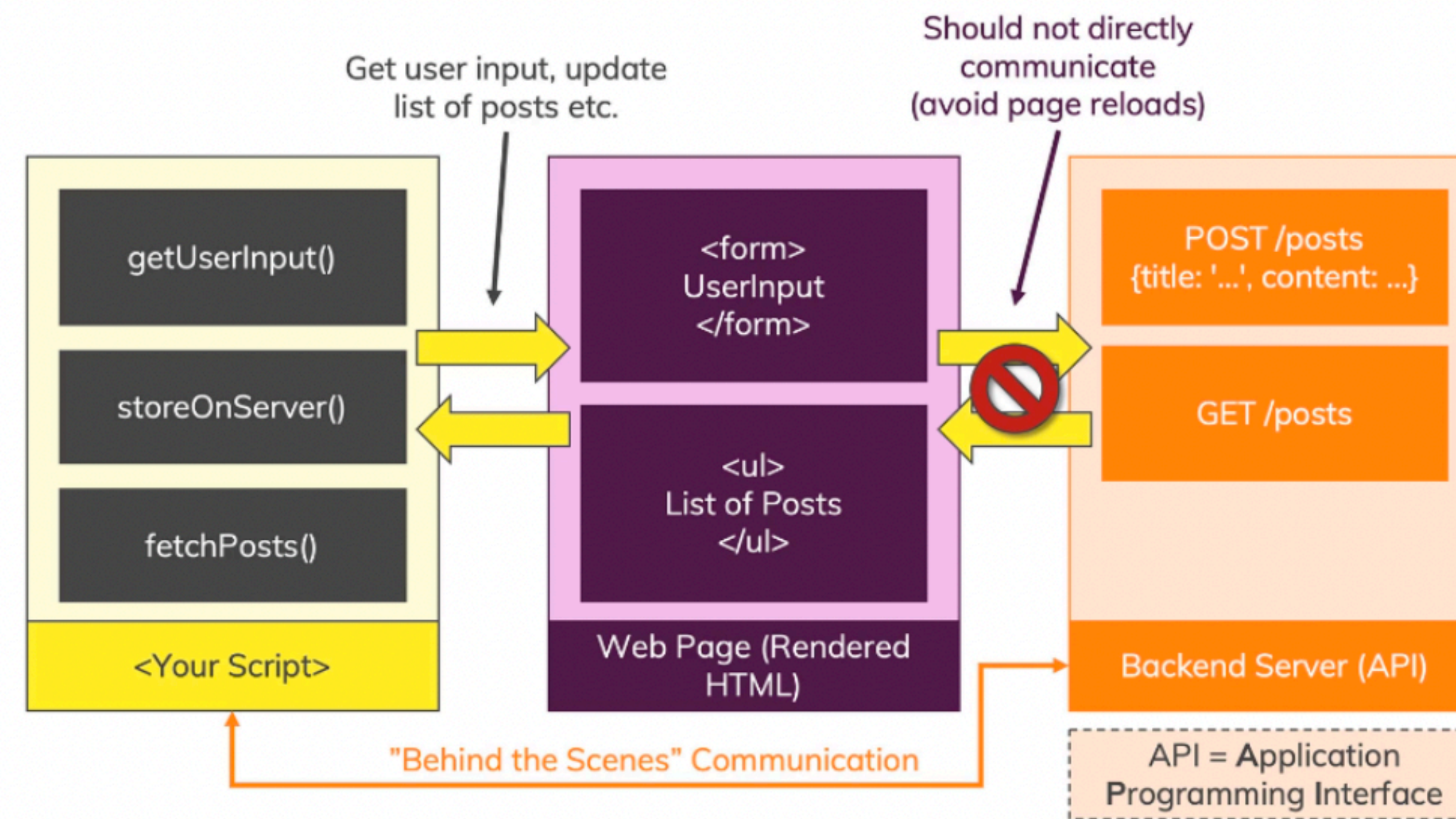
```
var user = document.cookie.getItem("name");
```

One advantage of cookies over local Storage and session Storage is that they can be set to expire at a certain time, which makes them a good choice for storing data that should not be persisted for a long time, such as session IDs.



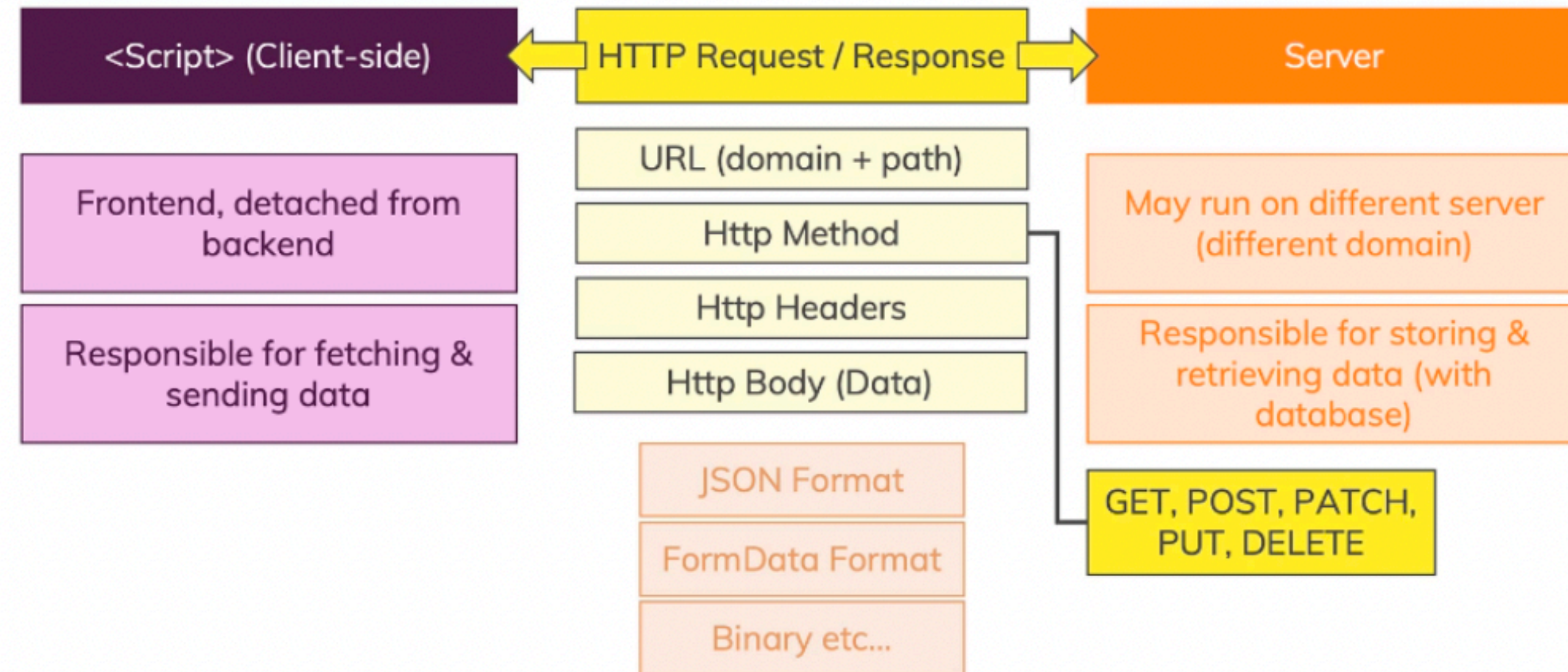
# AJAX

## Working with Http Requests



# AJAX

## HTTP Overview





# AJAX

## AJAX

AJAX stands for Asynchronous JavaScript And XML.

Ajax used for get / post data from/to server **without reloading** the page.

Ajax uses, Browser's inbuilt Object **XMLHttpRequest**.

- The XMLHttpRequest object is used to exchange data with a server.
- Use XMLHttpRequest (XHR) objects to interact with servers.
- You can retrieve data from a URL without having to do a full page refresh.
- This enables a Web page to update just part of a page without disrupting what the user is doing.

Ajax send / receive data to / from the server by asynchronously.