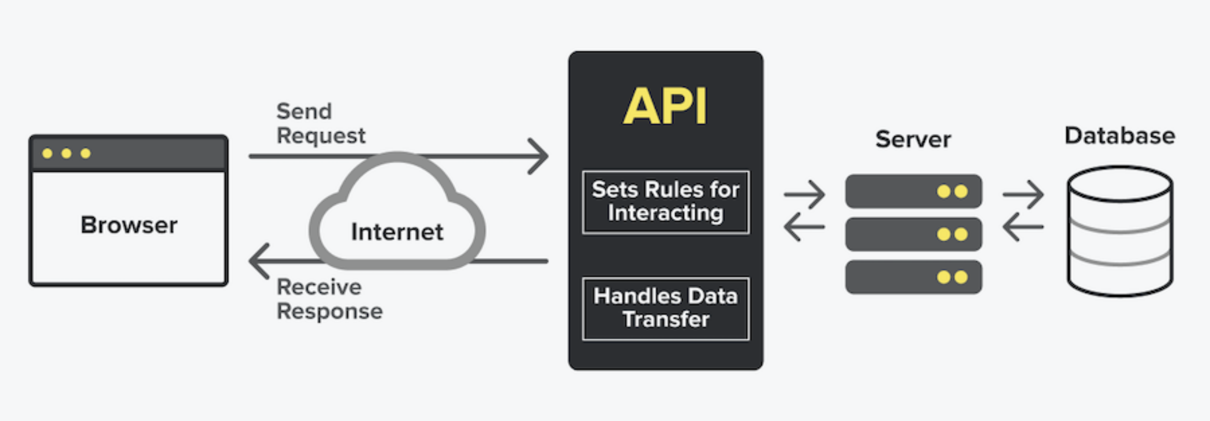
JAVASCRIPT: CONSOLE APIs

By Sakshi Vinayak Karande
TE/AI-DS/26

Console APIs in JavaScript provide developers with a way to interact with the browser's developer console. The console is a powerful tool for debugging, logging information, and testing code. You can use various console methods to display messages, inspect variables, and track the flow of your code.



Before knowing more about this, let's understand what an API is -

- **API stands for Application Programming Interface**. It is like a bridge that allows different software applications to communicate and interact with each other.
- APIs define a set of rules and protocols that enable developers to access the functionality or data of another application, service, or platform without needing to understand the internal details of how that functionality is implemented.
- Let's imagine you're at a restaurant. The menu is like an API – it lists the dishes you can order, along with their descriptions and prices. You, as a customer, don't need to know how the chef prepares the food; you just need to know what dishes are available, how to order them and through whom to place the order.

Console :

In the context of computing, it is a text-based interface or tool that allows users and developers to interact with a computer system or software application. It provides a way to input commands, receive text-based outputs, and perform various tasks, such as running programs, managing files, and troubleshooting issues. Consoles are commonly used for tasks like system administration, programming, and debugging.

How to access the console?

In most current browsers, the javascript console is integrated with dev tools. The shortcut keys help us open the console more easily.

For Chrome, you can access the console by-

1. Right click(on any element on a webpage)>Inspect Element> Console.
2. Shortcut key viz ,
   -'Ctrl + Shift + J' or 'Ctrl + Shift + I //then click to the console tab.(Windows)
   -'Cmd + Opt + J' (Mac OS)

Here are some common console methods you can use:
(These are just a few examples of the many console methods available in JavaScript)

1. *console.log()*: This is the most basic method and is used to print messages to the console. It's great for logging information for debugging purposes.

2. *console.error()*: Use this method to log error messages. Errors logged with this method often appear with a distinctive visual indication in the console, making them easy to spot.

3. *console.warn()*: This method is used to log warning messages. It's similar to console.error(), but it's typically used for non-critical issues.

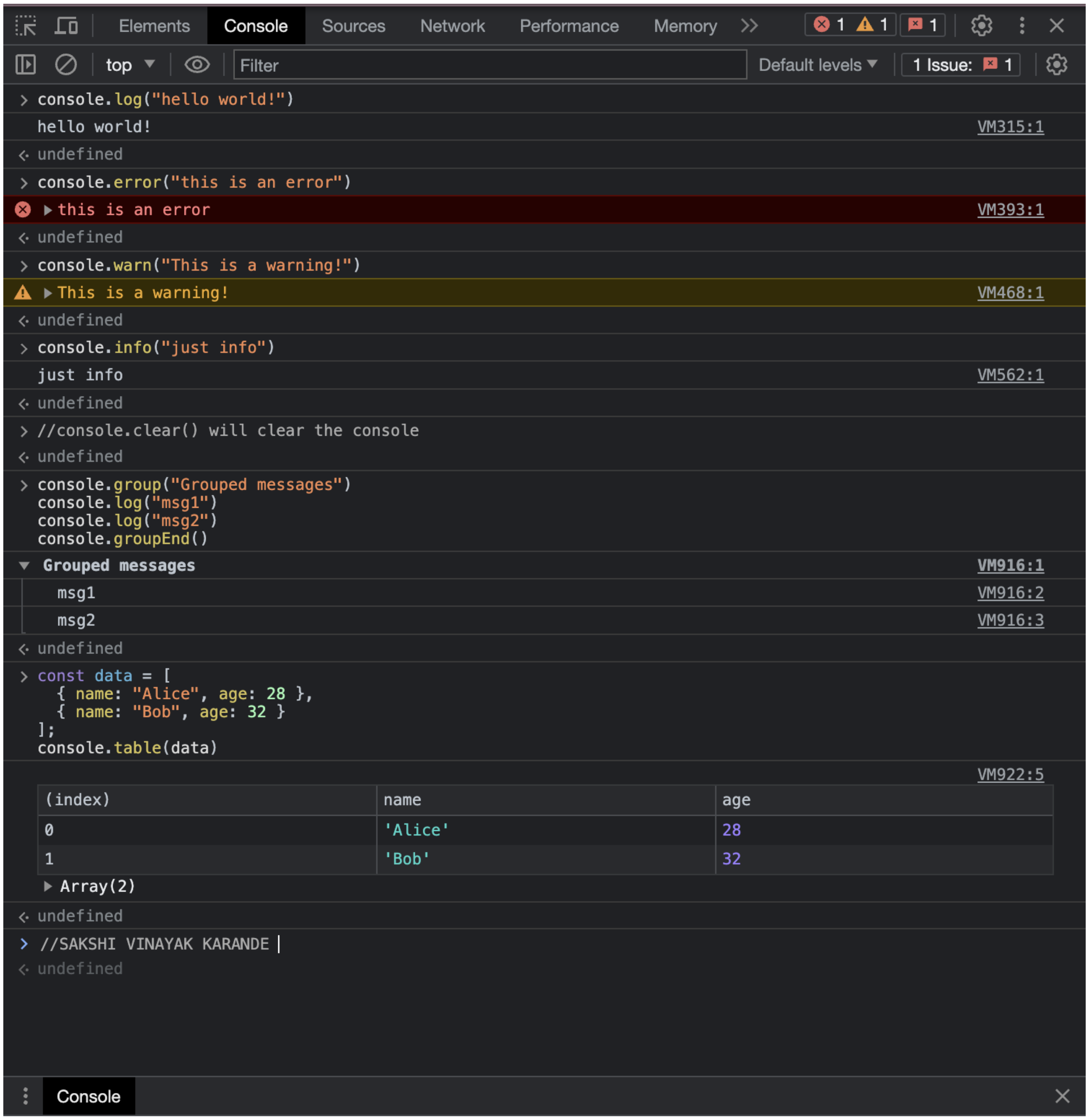
4. *console.info()*: Use this method to log informational messages. It's similar to console.log(), but it's often used to provide context or details.

5. *console.clear()*: This method clears the console, removing all logged messages and any other content.

6. *console.group()* and *console.groupEnd()*: These methods allow you to group related console messages together for better organization. group() marks the beginning of a grouped section, and groupEnd() marks the end.

7. *console.table()*: This method displays tabular data in a well-structured format, which can be useful for visualizing arrays or objects.

IMAGE ATTACHED SHOWS ITS IMPLEMENTATION:



```
> console.log("hello world!")
  hello world!                                                    VM315:1
<· undefined
> console.error("this is an error")
⊗ ▶ this is an error                                              VM393:1
<· undefined
> console.warn("This is a warning!")
⚠ ▶ This is a warning!                                            VM468:1
<· undefined
> console.info("just info")
  just info                                                       VM562:1
<· undefined
> //console.clear() will clear the console
<· undefined
> console.group("Grouped messages")
  console.log("msg1")
  console.log("msg2")
  console.groupEnd()
▼ Grouped messages                                                VM916:1
    msg1                                                          VM916:2
    msg2                                                          VM916:3
<· undefined
> const data = [
    { name: "Alice", age: 28 },
    { name: "Bob", age: 32 }
  ];
  console.table(data)
                                                                  VM922:5
  (index)            name               age
  0                  'Alice'            28
  1                  'Bob'              32
  ▶ Array(2)
<· undefined
> //SAKSHI VINAYAK KARANDE
<· undefined
```