In this thread I want to compare and contrast Stateful and Stateless API’s. To do so I will define both and then provide a table(s) below that will compare the two. To understand statelessness, one must understand **statefulness.** When we talk about computer systems, a “**state**” is simply the condition or quality of an entity at an instant in time, and to be stateful is to rely on these moments in time and to change the output given the determined inputs and state. (Sandoval, 2017). Consider the example of a phone call, “the connection is maintained from the beginning to the end of the call to ensure continuous communication. The connection is validated first, and only then will the session (phone call) be established indefinitely until the end of the call. This is what a stateful protocol is.” (Stateful Vs Stateless: Full Difference, 2022).

|  |  |
| --- | --- |
| **Advantages of Stateful** | **Disadvantages of Stateful** |
| Stateful protocol keeps track of the connection information, and as a result, delivers superior performance because of continually keeping track of information. | Stateful protocol requires memory allocation in order to store data. |
| Stateful protocols are more intuitive because they can maintain data on the server between two requests. | In the event of inefficient maintenance of session storage, there can be a decrease in the performance. It requires continuous management of the service’s full lifecycle. |
| They can improve performance when data retrieval is required only once. | These protocols are highly dependent on the server-side state. |
|  | Usually, stateful protocols require backing storage. |
|  | Since the state is maintained, stateful is not very secure. |

(Stateful Vs Stateless: Full Difference, 2022).

Now that we understand what Stateful looks like, let’s examine Stateless. Consider the example of sending a text message to someone. “Here the recipient’s availability is not confirmed, and the sender just sends the text to the recipient. There is no confirmation from the receiving device to the sending device that the message has been received. Despite being transmitted, the communication may or may not be received. There can be no cross-verification of status or retries. This is what stateless is all about. A stateless protocol is one in which the receiver is not required to keep session state from previous requests. The sender sends relevant session state to the receiver in such a way that each request may be interpreted without reference to prior requests’ session state, which the receiver retains. HTTP (HyperText Transfer Protocol) is an example of Stateless Protocol because each request is executed independently of the requests that came before it. This implies that once a transaction is completed, the connection between the browser and the server is also terminated.” (Stateful Vs Stateless: Full Difference, 2022).

|  |  |
| --- | --- |
| **Advantages of Stateless** | **Disadvantages of Stateless** |
| Since the monitoring of the system does not have to look beyond a single request to determine its whole nature, visibility of the protocol is improved. | It may be essential to include additional information in each request, and as a result, the server will need to interpret this new info. |
| It is easier to recover from partial failures like crashes since no state is maintained, which improves reliability. | They may degrade network performance by increasing the amount of repetitive data delivered in a series of requests, which cannot be saved and reused. |
| The server does not have to store session state between requests; hence, scalability is enhanced as deploying the services to any number of servers is possible, and implementation is simplified even more. | They are inherently less capable as they do not store information about a particular session. |
| It only necessitates a small number of resources because the system doesn’t need to keep track of communication over numerous lines, as well as session information. |  |
| In Stateless protocols, each individual communication is unconnected and distinct from the ones that come before or after it. |  |
| Here, each packet of data travels on its own. There is no need to refer to another packet in these packets. |  |

(Stateful Vs Stateless: Full Difference, 2022).

References:

Kanjilal, J. (2013). *ASP.Net Web API: Build Restful Web Applications and Services on the .Net Framework*. Packt Publishing.

Sandoval, K. (2017, December 7). *Smarter Tech Decisions Using APIs*. Nordic APIs. Retrieved November 9, 2022, from <https://nordicapis.com/defining-stateful-vs-stateless-web-services/>

*Stateful vs Stateless: Full Difference*. (2022, February 23). InterviewBit. Retrieved November 9, 2022, from <https://www.interviewbit.com/blog/stateful-vs-stateless/>