**Statelessness in RESTful APIs**

One of the most important ideas in RESTful design is *statelessness*. In simple terms, statelessness means every client request to a server has to be self-contained. The server doesn’t keep track of previous interactions or store session data. Each request includes all the details the server needs to understand and respond. This approach goes back to the design of HTTP itself, which was built as a stateless protocol so the web could stay fast and scalable.

In practice, this means if your RESTful service requires authentication, the client has to send credentials (like an API key or token) with each request. The server doesn’t assume the client is still logged in from before. For example, here’s a quick Java snippet showing a DELETE request:

URL url = new URL("https://api.myapp.com/items/42");

HttpURLConnection conn = (HttpURLConnection) url.openConnection();

conn.setRequestMethod("DELETE");

conn.setRequestProperty("Authorization", "Bearer <token>");

int resp = conn.getResponseCode();

// handle response …

*Here, the server doesn’t remember past requests—it only looks at what comes in this one.*

Statelessness matters because it makes systems easier to scale, since any server in a cluster can handle any request. It also simplifies reliability: if one server goes down, another can take over without worrying about “session memory.” Developers benefit too, since debugging is easier when requests don’t rely on hidden state. The tradeoff is a bit of extra work—like always sending tokens—but the gains in performance and simplicity far outweigh the costs.

**References**

Fadatare, R. (2025, March 12). *Statelessness in REST APIs*. Medium. <https://rameshfadatare.medium.com/statelessness-in-rest-apis-e44e4587db00>

RestfulAPI.net. (n.d.). *Stateless REST API advantages and constraints*. <https://restfulapi.net/statelessness/>

Sahane, B. (2024, July 10). *Mastering REST APIs: Statelessness in REST*. Medium. <https://medium.com/@bvsahane89/mastering-rest-apis-statelessness-in-rest-f2a7fe05e73c>