**Spike Arrest vs. Quota Policy in Apigee**

**Spike Arrest vs. Quota Policy in Apigee**

| **Feature** | **Spike Arrest** | **Quota** |
| --- | --- | --- |
| **Purpose** | Throttle sudden bursts of traffic (burst control) | Enforce a fixed number of allowed requests (rate limit) |
| **Limits** | Limits **rate per second or minute** | Limits **number of requests per time interval** |
| **Example Usage** | Allow 2 requests per second | Allow 1000 requests per day |
| **Behavior** | Smooths traffic flow by spacing out requests | Tracks and enforces total request count over time |
| **Reset Mechanism** | No reset — always applies per second/minute | Resets after the specified interval (hour, day, etc.) |
| **Best Used For** | Preventing backend overload during sudden traffic spikes | Enforcing business/API usage limits per client or app |
| **Identifier Support** | Not typically used with client-based identifiers | Can enforce quotas **per app**, **per user**, or **per API key** using <Identifier> |
| **Response on Limit Exceeded** | Returns **429 Too Many Requests** | Returns **429 Too Many Requests** |
| **Common Policy Element** | <SpikeArrest> | <Quota> |
| **Stateful/Stateless** | **Stateless** (no storage of past request count) | **Stateful** (tracks usage in memory or persistent storage) |

**Example Scenarios**

* **Spike Arrest:**  
  To allow up to 5 requests per second regardless of who is sending them.  
  *Use case:* Protect backend from sudden spikes from a large client.
* **Quota:**  
  To allow each user to send only 100 requests per hour.  
  *Use case:* Enforce rate-limiting in a freemium API model.

**Summary**

* Use **Spike Arrest** to **smooth traffic flow** and prevent backend crashing during **bursts**.
* Use **Quota** to **enforce usage limits** over time and implement **API access plans**.