|  |  |  |  |
| --- | --- | --- | --- |
|  | **Quota** | **Spike Arrest** | **Concurrent Rate Limit** |
| Use it to: | Limit the number of connections apps can make to your API proxy's target backend over a specific period of time. | Protect your API proxy's target backend against severe traffic spikes and denial of service attacks. | Limit the number of concurrent connections apps can make to your API proxy's target backend. |
| Don't use it to: | Don't use it to protect your API proxy against traffic spikes.  For that, use the Spike Arrest policy or Concurrent Rate Limit policy. | Don't use it to count and limit the number of connections apps can make to your API proxy's target backend over a specific period of time.  For that, use the Quota policy. | Don't use it to limit the number of connections apps can make to your API proxy's target backend over a specific period of time.  For that, use the Quota policy. |
| Stores a count? | Yes | No | Yes |
| Best practices for attaching the policy: | Attach it to the **ProxyEndpoint Request PreFlow**, generally after the authentication of the user.  This enables the policy to check the quota counter at the entry point of your API proxy. | Attach it to the **ProxyEndpoint Request PreFlow**, generally at the very beginning of the flow.  This provides spike protection at the entry point of your API proxy. | This policy must be attached in these three locations:   * **TargetEndpoint Request PreFlow** * **TargetEndpoint Response PreFlow** * **TargetEndpoint DefaultFaultRule** |
| HTTP status code when limit has been reached: | 500 (Internal Server Error) \* | 500 (Internal Server Error) \* | 503 (Service Unavailable) |
| Good to know: | * Quota counter is stored in Cassandra. * Configure the policy to synchronize the counter asynchronously to save resources. * Asynchronous counter synchronization may cause a delay in the rate limiting response, which may allow calls slightly in excess of the limit you've set. | * Performs throttling based on the time at which the last traffic was received. This time is stored per message processor. * If you specify a rate limit of 100 calls per second, only 1 call every 1/100 second (10 ms) will be allowed on the message processor. A second call within 10 ms will be rejected. * Even with a high rate limit per second, nearly simultaneous requests may result in rejections. | * Keeps a count of concurrent connections per message processor. * While an individual API proxy may be handling just a few connections, collectively, the connections to a set of replicated API proxies pointing to the same backend service may swamp the capacity of the service. Use this policy to limit this traffic to a manageable number of connections. |
| Get more details: | [Quota policy](http://docs.apigee.com/api-services/reference/quota-policy) | [Spike Arrest policy](http://docs.apigee.com/api-services/reference/spike-arrest-policy) | [Concurrent Rate Limit policy](http://docs.apigee.com/api-services/reference/concurrent-rate-limit-policy) |