Quality of Service

Requests made to the World-Check One API are rate limited to ensure a consistent experience for all customers, however exact throttling limits may vary over time to adjust for available capacity on the platform. When making requests to the World-Check One API, it is advisable to check for `HTTP 429 Too Many Requests` response codes and attempt to retry the affected requests after a brief delay.

Please note that throttling limits are applied to all requests made by a user of the API, so a HTTP 429 response for one method would require a delay waiting period for all other API requests that may be processing during the same time period, for the same API user. Appropriate error handling that can check for errors such as HTTP 429 and retry requests using a policy such as exponential backoff should therefore be added to all World-Check One API calls.

More advanced integrations can use the occurrence of HTTP 429 responses to dynamically scale the rate of API requests being made. For example, if 429 responses are regularly being encountered, overall request rates can be slowed, whereas if a period of time with no 429 responses is observed, overall request rates can be increased to make use of additional capacity.

An example of how to achieve this dynamic throttling is to process API requests asynchronously, by queueing them in some form of messaging middleware, rather than invoking them directly from procedural business logic. A request processor task could be implemented that dequeues API requests and sends them through to the World-Check One API. A business process that is waiting for a response from a queued request can be marked in a 'Pending' state until a successful response is obtained. The request processor can then detect if a 429 response is received from the World-Check One API, and upon receipt, scale down the dequeue rate as well as re-queue the failed message for later processing with an appropriate retry policy. Upon receipt of a successful API response, some form of callback attached to the API request can be invoked to asynchronously continue any dependant business processing and transition out of the 'Pending' state. For any other failure response, or after an appropriate global timeout period has elapsed while waiting for an API request to process, an error callback could be invoked to transition the business process from a 'Pending' to a 'Failed' state, potentially allowing for later reprocessing as part of an end-of-day batch job.

Please note that the performance characteristics of the **Pilot** environment are substantially reduced compared to **Production**, so should not be taken as indication of how integration with **Production** will perform. **Pilot** is purely intended for functional acceptance testing with sample, non-sensitive data and highly limited API request volumes. The occurrence of 429 responses within the **Pilot** environment can be used to validate the error and retry handling of an integration project, as well as any dynamic request rate processing that may be put in place.