In this report we will discuss the workflow of api with various use cases. Let’s start:

First of all, we have only one api endpoint, which is /api/smart/<timeout\_parameter> which performs up to 3 HTTP requests to Exponea Testing HTTP Server (<https://exponea-engineering-assignment.appspot.com/api/work>) and returns first successful response. Note: if no successful request returned within the given timeout, api responds with error messages for each of 3 requests that failed.

Algorithm is built in this way:

First it sends one request to the testing server. If there is a successful response within 300 milliseconds, then the endpoint returns this response and doesn’t fire other requests.

But if the time of first request is bigger than 300 milliseconds, it fires concurrently (asynchronously) two other requests, and then returns the first successful request from any of the 3 requests, including the first one.

Example №1:

İf first request to testing server returns time equal to 270, then api returns answer like this {“time”: 270}

Example №2:

İf first request to testing server returns time equal to 350, then immediately it fires another 2 requests, which are 280 and 528 respectively, and returns the fastest one, means {“time”: 280}

Concurrent requests are made with help of httpx and asyncio, one of which helps to send the requests asynchronously, other one to gather data from the responses of requests.

But there are other cases, when there appear errors, such as Server timeout, Connection timeout, Internal server error (which comes from testing server), which require to be handled. In this case, if one request returns an exception, it just does nothing but writes it down and deals with other requests. For example, if second request returned an error, it just skips it and returns first successful response from first and third request, and so on. But in case if all requests are failed, it returns json response with the error message for each of the request.

Let me show some use cases:

Example №3:

First request to test server returned 350, then second and third are fired, second returns an error, and third returns 420. The response time will be 350

Example №4:

First request to test server returned 400, then second and third are fired, second returns an error, and third also returns an error. The response time will be 400

Example №5:

First request to test server returned an error, then second and third are fired, second returns 340, and third returns 450. The response time will be 340

Example №6:

First request to test server returned a server error, then second and third are fired, second returns a timeout error, and third also returns a timeout error. The response will be like that {1: Server error, 2: Timeout error, 3: Timeout error}

During the testing period, the testing server behaved differently. Sometimes it worked perfectly without errors, sometimes it returned almost all errors each time. It is advised to test Api regularly to avoid misunderstandings in app workflow.