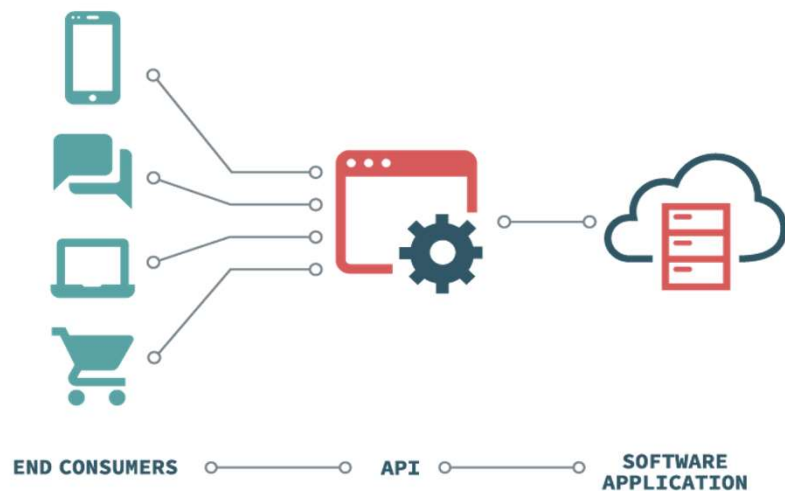




SECCION VI: PRUEBAS DE API



¿QUÉ NECESITAMOS SABER PARA PROBAR A NIVEL DE BACKEND (APIS)? EJECUCIÓN DE ENDPOINTS

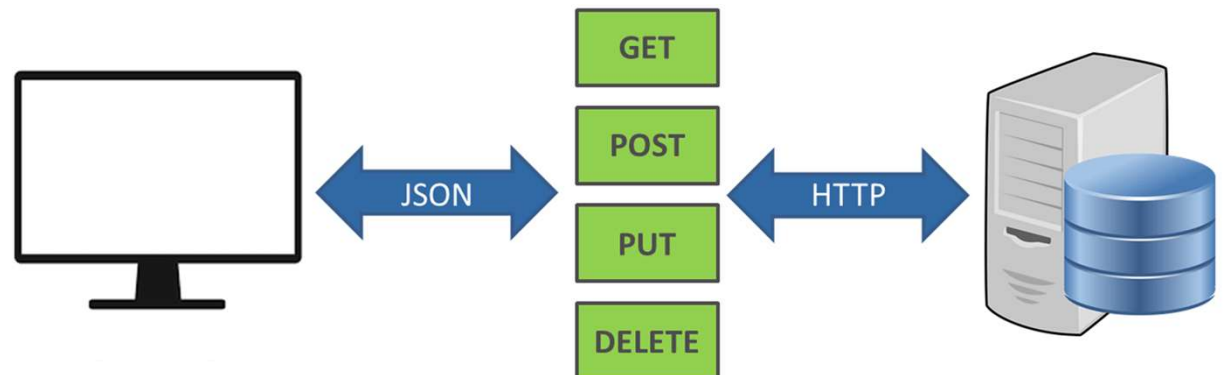


- Un servicio web es una funcionalidad que necesita estar publicada en la web porque es utilizada por diferentes sistemas / empresas o porque actúa como transporte de datos de un lugar a otro. Es por ello que debe escribir por medio de un protocolo estándar como SOAP o REST. Sus elementos principales para probarlo son:
 - URL
 - Request: Datos de Entrada
 - Mecanismo de Autenticación
 - Response: Datos de Salida
 - Código de estatus

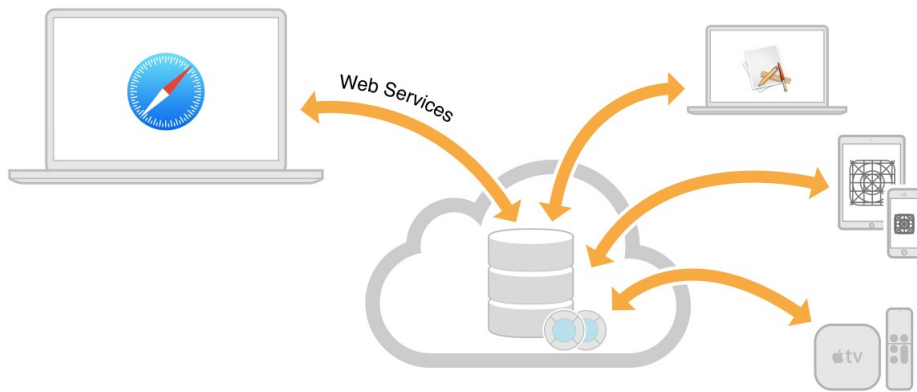
¿QUÉ NECESITAMOS SABER PARA PROBAR A NIVEL DE BACKEND (APIS)? EJECUCIÓN DE ENDPOINTS

- Para hacer pruebas de un endpoint debemos contemplar:
 - Datos de entrada positivos y negativos
 - Posibles valores de salida “no comunes”
 - Si es una funcionalidad crítica
 - Si es una funcionalidad muy utilizada
- Nuestro alcance abarcará servicios bajo el protocolo REST

Ejemplo: [Public Holiday Api](#) - [Nager.Date](#)



SERVICIOS WEB



Como parte del funcionamiento de las aplicaciones web es muy común que las empresas utilicen servicios web para implementar funcionalidades que serán utilizadas por otros sistemas internamente o incluso, por otras empresas.

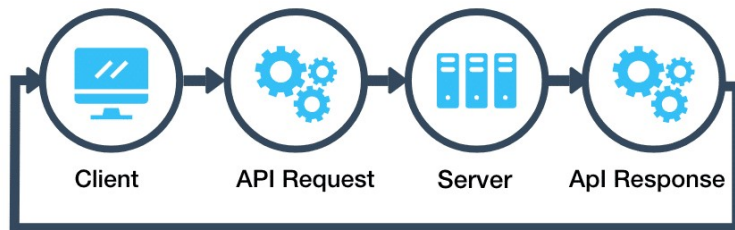
Por ejemplo: la funcionalidad Transferir Dinero está disponible en la Banca en Línea (WEB), llamando por teléfono (IVR), desde el celular (APP) y por cajero automático (ATM). En todos los casos la interfaz es diferente pero al ingresar los datos y hacer clic en Transferir el funcionamiento será el mismo.

¡Ese será un servicio web!

SERVICIO WEB

API

Application Programming Interface



Como definición puede decirse que un servicio web es una funcionalidad publicada en la web y disponible para ser utilizada.

En general, los servicios web tienen:

- Una URL para ser accedida. Por ejemplo:
<https://api.miempresa.com/transferir>
- Datos de entrada para ser procesados. Por ejemplo: Número de cuenta origen, Número de cuenta destino y monto a transferir
- Datos de salida producto del procesamiento de la funcionalidad. Por ejemplo: “Operación exitosa”, Número de operación 1234
- Código de estatus que sería un mecanismo interno para indicar si los datos de entrada pudieron ser procesados o no. Por ejemplo: 200 OK

SERVICIO WEB EN LAS PRUEBAS

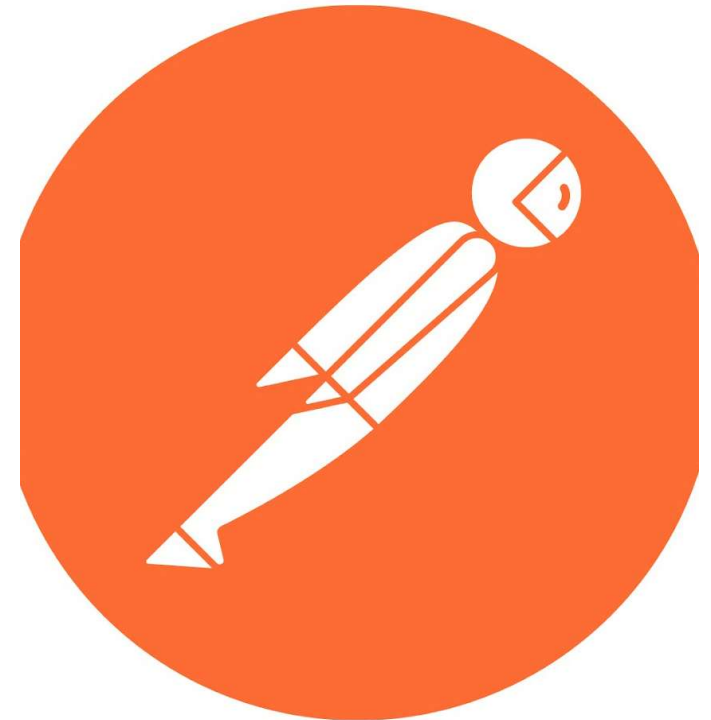
En el ámbito de las pruebas, los servicios web representan una estupenda oportunidad para hacer “pruebas tempranas” porque suelen ser los primeros entregables de Desarrollo.

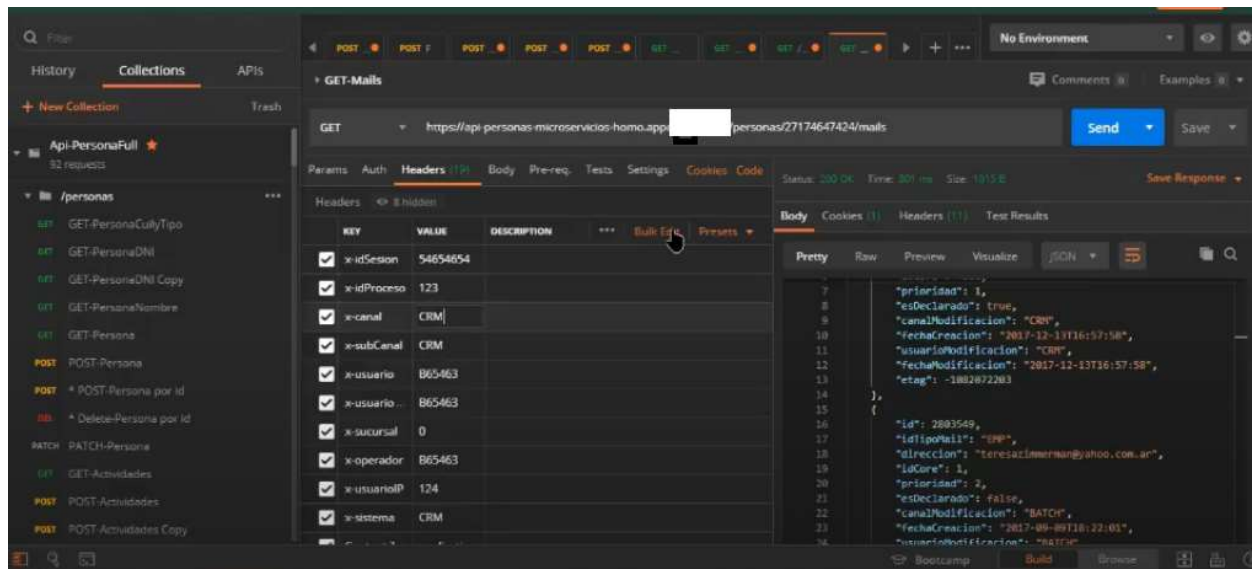
- Si un servicio web no procesa correctamente ciertos datos de entrada y devuelve un resultado que no esperamos, es muy ventajoso reportarlo incluso antes que esté listo el sistema para probar la funcionalidad desde la interfaz.
- Por otro lado, comprender sobre los servicios web que son invocados por un sistema ayuda al tester a desarrollar un criterio más amplio al momento de decidir qué probar.



POSTMAN PARA LAS PRUEBAS DE SERVICIOS WEB

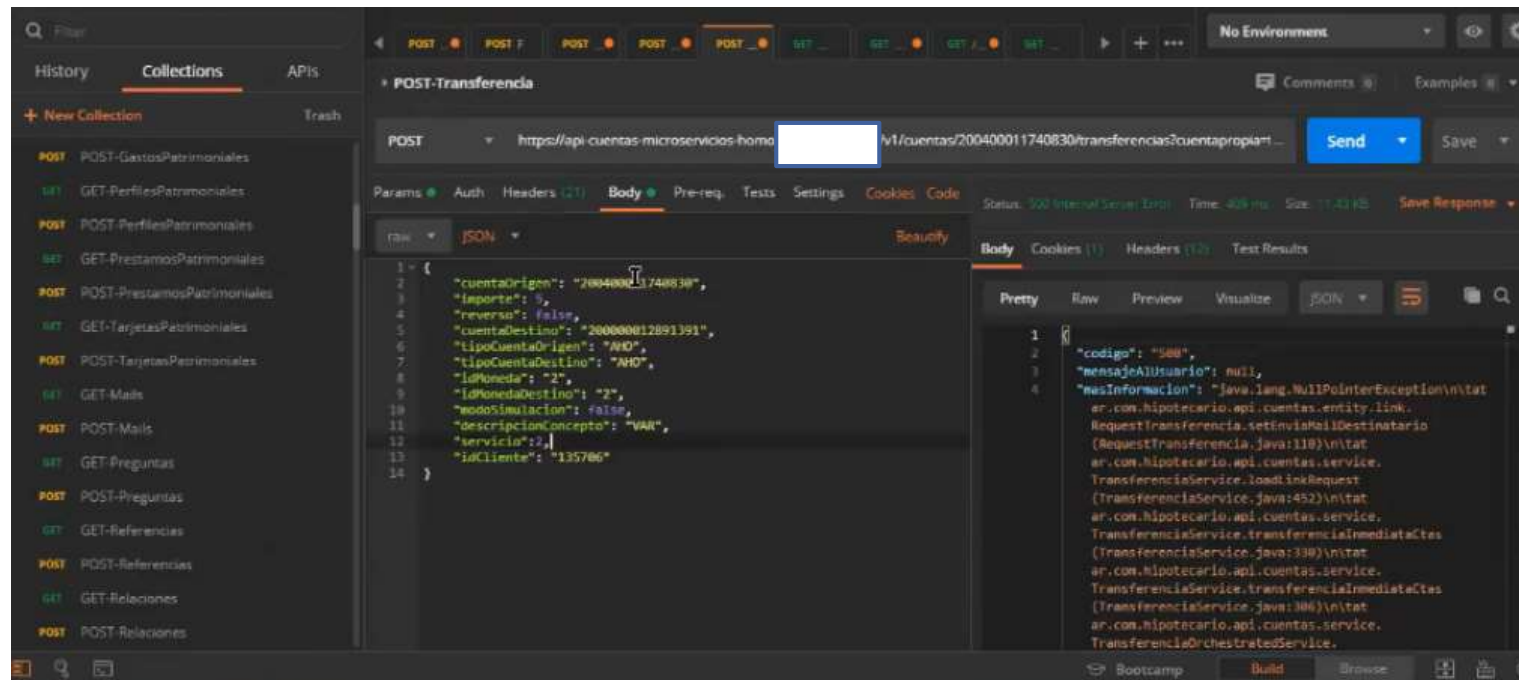
- Ya que un servicio web no tiene una interfaz, es obligatorio utilizar alguna herramienta para poder evaluarlos en forma preliminar.
- Postman es una de las herramientas más conocidas para probar servicios web. Permite:
 - Configurar un endpoint de una API (Servicio Web)
 - Ejecutar un endpoint con ciertos datos de entrada para visualizar la salida que genera: datos de salida, código de estatus y tiempo de respuesta
 - Implementar chequeos semi-automatizados
 - Ejecutar un endpoint con un conjunto de datos de entrada por medio de un archivo CSV
 - Definir variables de ambiente para evitar la duplicidad de configuraciones cuando probamos en diferentes entornos (DEV, QA, PRE, etc.)

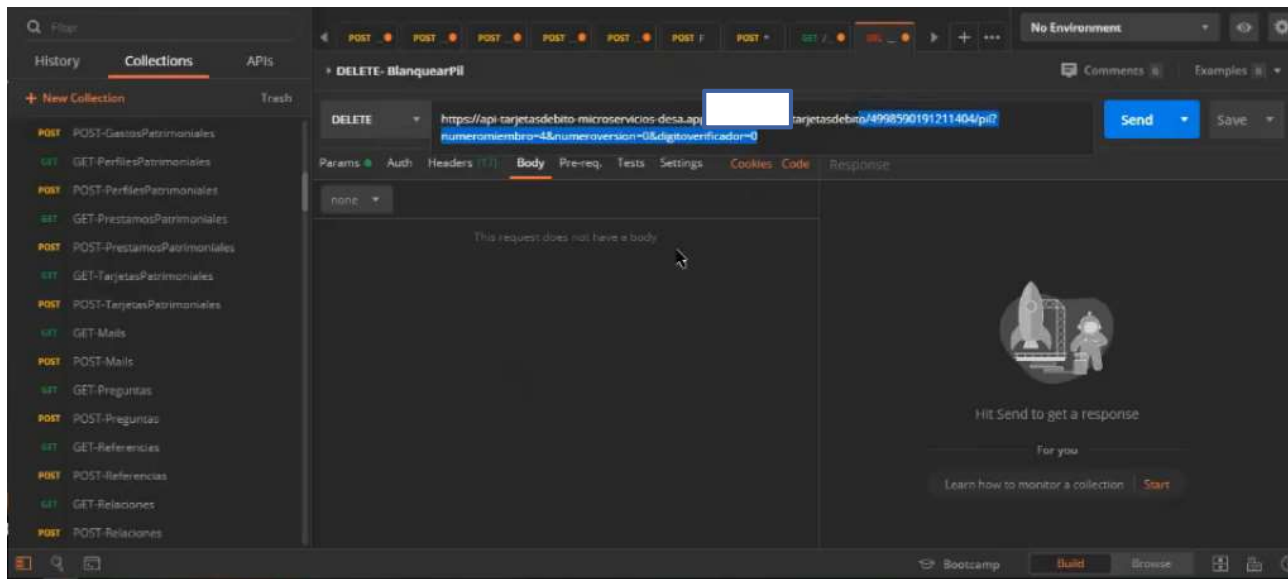




EJEMPLO SERVICIO (GET)

EJEMPLO SERVICIO (POST)





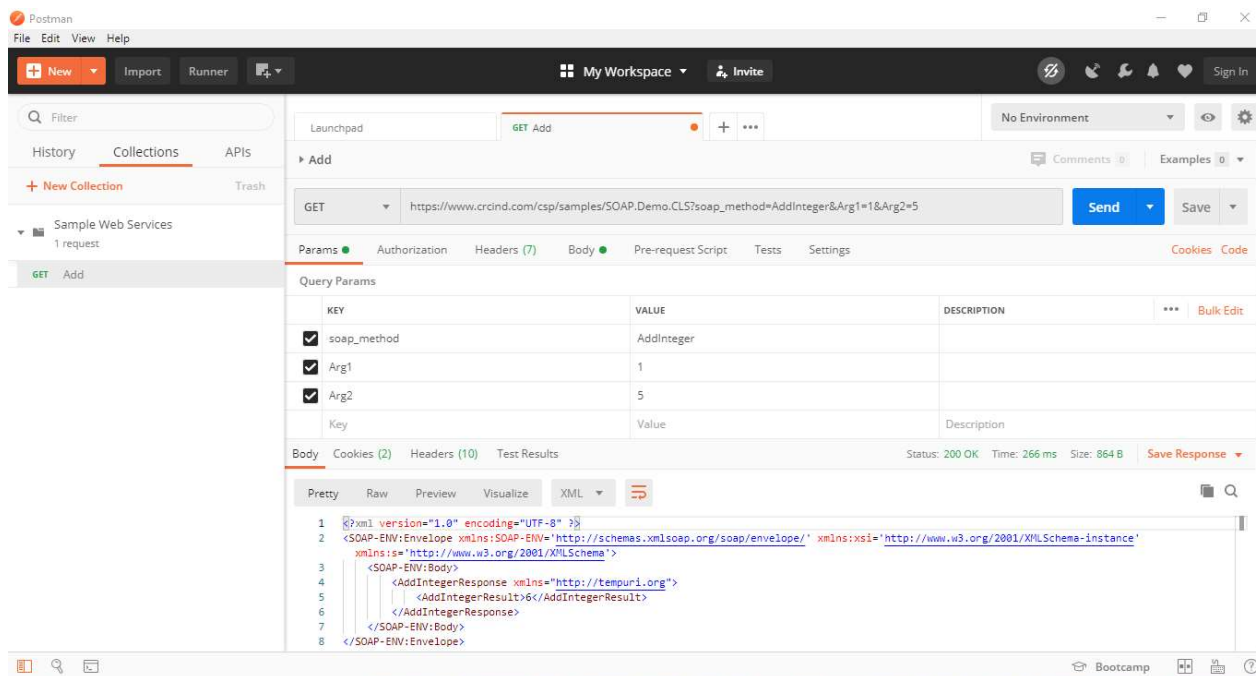
EJEMPLO
SERVICIO
(DELETE)

CONFIGURACIÓN DE UN SERVICIO WEB EN POSTMAN

- Servicio web a probar (en este caso la operación de suma: AddInteger del servicio <https://www.crcind.com/csp/samples/SOAP.Demo.CLS?WSDL>)
 - El método se llama AddInteger
 - Recibe dos números como parámetro Arg1 y Arg2
 - Devuelve un número como resultado

CONFIGURACIÓN Y EJECUCIÓN DE UN SERVICIO WEB EN POSTMAN

1. Abrimos Postman haciendo doble clic sobre el ícono que fue creado producto de la instalación de la herramienta
2. Hacemos clic en New > Collection
3. Ingresamos el nombre de la colección
4. En la colección, hacemos clic en el ícono de “tres puntos” y luego en la opción New Request
5. Ingresamos el nombre del request (es un nombre que fácilmente nos permita identificar el servicio)
6. Hacemos clic en el botón Create
7. Ingresamos la URL del servicio en la barra correspondiente estableciéndolo como un servicio GET (siendo que devuelve resultados). En este caso <https://www.crcind.com/csp/samples/SOAP.Demo.CLS>
8. Vamos a la pestaña Params (en ocasiones también se utiliza la pestaña Body para esto) e ingresamos tres parámetros con su nombre y valor
 - soap_method con el nombre de la operación a realizar (AddInteger)
 - Arg1 con el primer número a sumar
 - Arg2 con el segundo número a sumar
9. Hacemos clic en el botón Send
10. Visualizamos los resultados en la sección inferior de Postman



RESULTADO DE
LA EJECUCIÓN
DE UN SERVICIO
WEB EN
POSTMAN

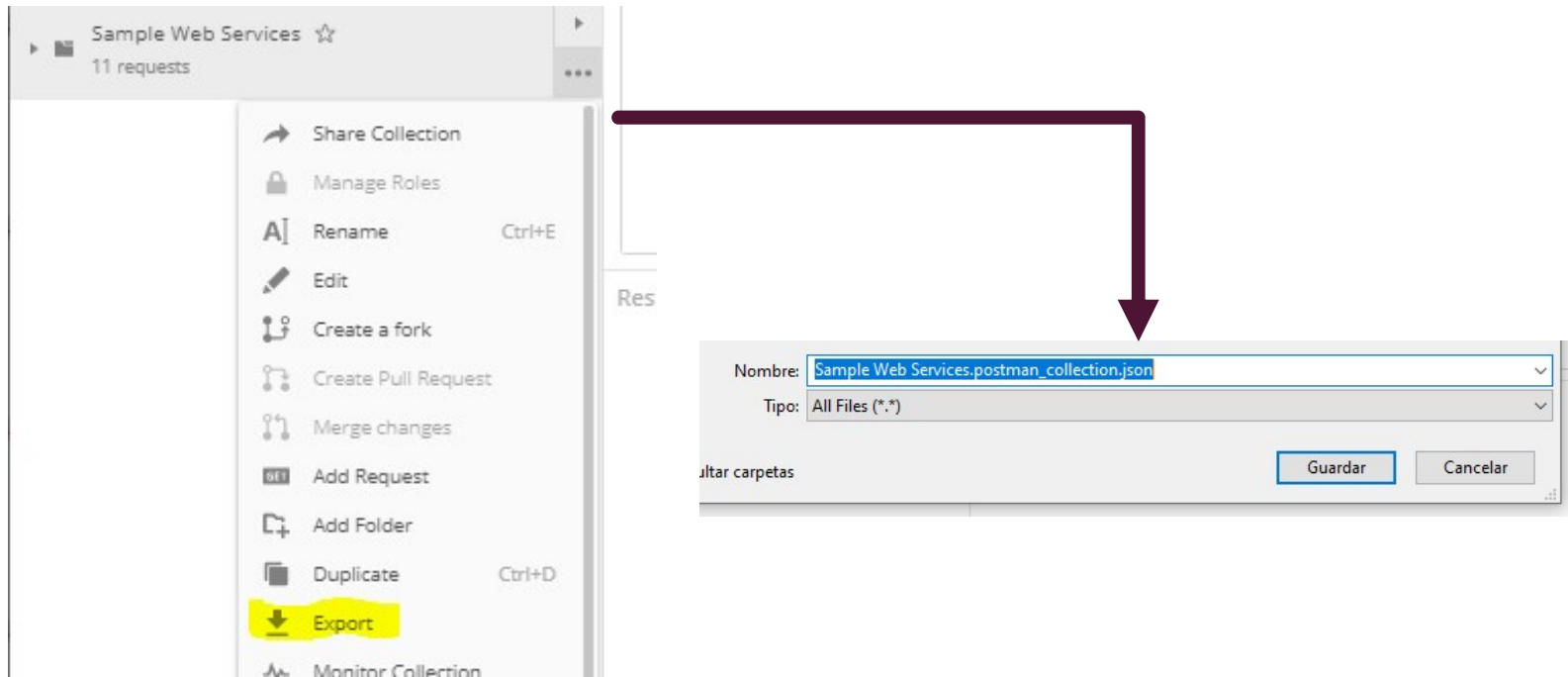
GUARDAR UNA COLLECTION EN TU COMPUTADOR

Cuando se necesite respaldar o compartir el trabajo realizado en Postman, es factible guardar la collection en tu computador.

Para ello puedes hacer:

- Clic en los “tres puntos” que están a la derecha de la collection
- Clic en la opción Exportar
- Elige el directorio donde desees respaldar la collection
- Es posible modificar el nombre si así lo desees (el formato si debe quedar como .json)
- Clic en el botón Guardar

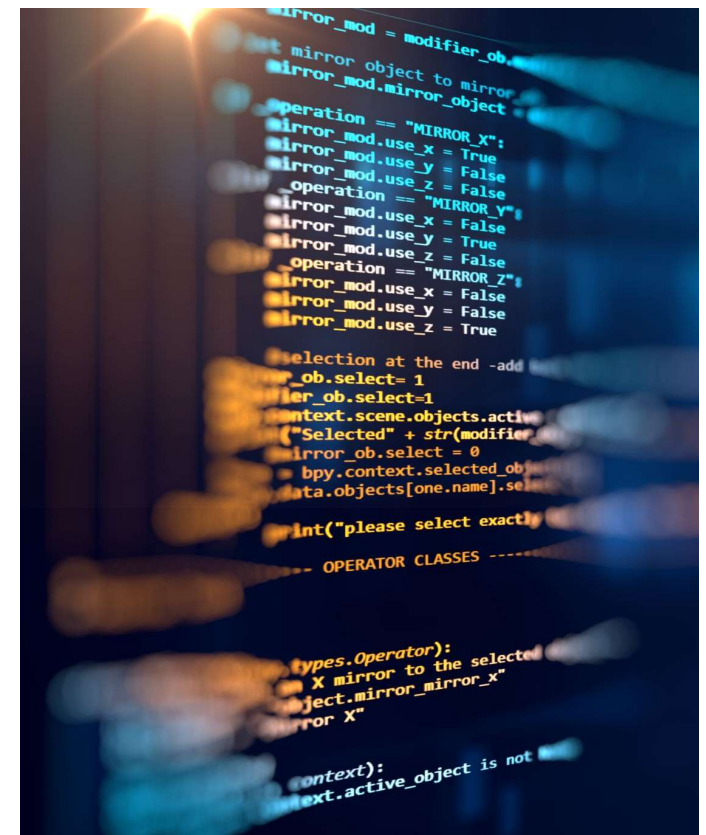
EXPORTAR COLECCIONES



EVIDENCIAS DE LAS PRUEBAS DE SERVICIOS WEB

Algunas de las evidencias recomendadas como parte de la ejecución de pruebas de Servicios Web

- Definición del servicio: URL, Request y Response
- Parámetros de entrada en la ejecución a realizar
- Resultado de la ejecución (Por ejemplo 200 OK)

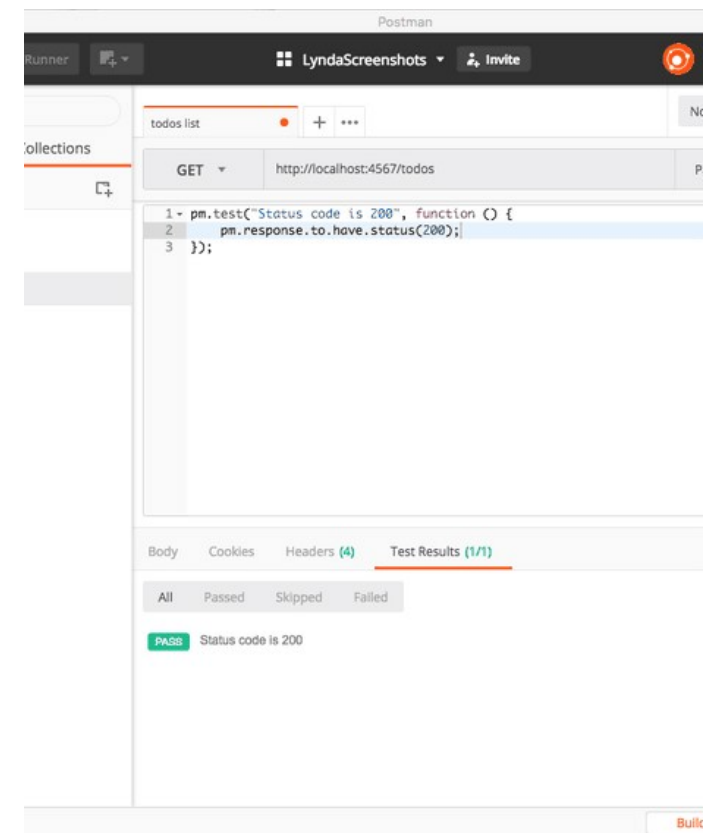


COMPROBACIÓN Y GESTIÓN DE RESPUESTAS

Al recibir una respuesta como parte de la ejecución de un servicio web será valioso:

- Hacer una comprobación de que la respuesta es la esperada
- Guardar el resultado de la respuesta para ser usado (probablemente) en nuevas peticiones

Todo esto será posible usando la pestaña Tests para hacer operaciones con los resultados de la respuesta por medio de Javascript.



COMPROBACIÓN Y GESTIÓN DE RESPUESTAS

```
pm.test("CHK01 - Validar codigo de estatus 200", function () {  
    pm.response.to.have.status(200);  
});
```

Postman almacena automáticamente el código devuelto en una petición en `responseCode.code`. Así podremos comprobar que el código devuelto es 200 (es decir, la petición se realizó exitosamente) y en ese caso se mostrará el mensaje en la pantalla de resultados. Si se cumple se mostrará un PASS y si no es así, se mostrará un FAIL.

HTTP Status



COMPROBACIÓN Y GESTIÓN DE RESPUESTAS - EJEMPLOS

Para chequear si el body del response contiene cierta cadena de caracteres:

```
pm.test("CHK02 - Validar que el response contenga Dinosaur", function () {  
    pm.expect(pm.response.text()).to.include("Dinosaur");  
});
```

COMPROBACIÓN Y GESTIÓN DE RESPUESTAS - EJEMPLOS

Para chequear si el tiempo de respuesta es menor a 200 ms.:

```
pm.test("CHK03 - Validar que el tiempo de respuesta sea hasta 500ms", function () {  
    pm.expect(pm.response.responseTime).to.be.below(500);  
});
```

COMPROBACIÓN Y GESTIÓN DE RESPUESTAS - EJEMPLOS

En pruebas más avanzadas también podríamos chequear el resultado que nosotros esperamos:

```
pm.test("Status code OK", function () {  
    pm.response.to.status(pm.variables.get("expectedResult"));  
});
```

Donde “expectedResult” es una variable por ejemplo leída de un archivo CSV.

COMPROBACIÓN Y GESTIÓN DE RESPUESTAS - EJEMPLOS

Para enviar un request asíncrono:

```
pm.sendRequest("https://postman-echo.com/get", function (err, response) {  
    console.log(response.json());  
});
```

EJEMPLO MÁS COMPLEJO

```
let confirmRequest = {
  url: pm.environment.get("CNP-DEV-API") + '/assurplan/cotizar',
  method: 'POST',
  header: {
    'Content-Type': 'application/json',
    'Accept': '*/*',
    'Accept-Encoding': 'gzip, deflate, br',
    'Connection': 'keep-alive',
    'Ocp-Apim-Subscription-Key': pm.environment.get("API_KEY")
  },
  body: {
    mode: 'raw',
    raw: JSON.stringify({
      codigoProductor: pm.variables.get("codigoProductor"),
      fechaNacimiento: pm.variables.get("fechaNacimiento"),
      cantidadModulos: pm.variables.get("cantidadModulos")
    })
  }
}

console.log(confirmRequest);
pm.sendRequest(confirmRequest, function (err, response) {
  console.log(response);
  var jsonData = response.json();
  pm.environment.set("cotizacionId", jsonData.id);
});
```

EJECUCIÓN DEL TEST

Con las configuraciones previas de variables de entorno y script's en las pestañas Tests y Pre-request script es posible crear una colección de peticiones concatenadas en las que resultados de unas pueden ser usadas en otras.

Esto, junto con la posibilidad de creación de datos de manera dinámica, nos permite realizar un test de regresión completo que comprueba la funcionalidad principal de una API.

Una vez terminada la colección, para ejecutar el test, hay que abrir la colección, hacer clic en la carpeta del test y luego en el botón Run. Una vez hecho esto, aparecerá una nueva pantalla.

PD: Si tenemos código (validaciones, instrucciones) en las pestañas Tests y Pre-request script es posible monitorear/leer los resultados en consola (Ctrl + Alt + C).

Pruebas de rendimiento de servicios web REST



Las pruebas de rendimiento son **cruciales para simular condiciones reales de uso y medir cómo los servicios web manejan diversas cargas.**



Estas pruebas ayudan a predecir el comportamiento del sistema en situaciones de alta demanda y a identificar potenciales puntos de falla antes de que el sistema esté en producción.

Capacidades de Postman en pruebas de rendimiento

Postman es una herramienta versátil que soporta varios tipos de pruebas de rendimiento:

- **Pruebas de carga:** evalúan cómo el sistema maneja grandes volúmenes de solicitudes.
- **Pruebas de estrés:** determinan los límites del sistema, incrementando gradualmente la carga hasta que el sistema falla.
- **Pruebas de pico y rampa:** simulan aumentos repentinos o graduales en el uso para ver cómo el sistema responde y se recupera.

¿Cuándo realizar pruebas de rendimiento?

La necesidad de realizar pruebas de rendimiento puede surgir en varias situaciones, como:

- **Servicios web con alta demanda prevista:** es crucial para sistemas que se espera manejen un gran número de usuarios simultáneos, como aplicaciones comerciales en línea o servicios de medios digitales.
- **Servicios web con cambios recientes en el código:** cualquier optimización o modificación sustancial en el código debe ser seguida por pruebas de rendimiento, para asegurar que las mejoras no afecten negativamente la funcionalidad o la capacidad de respuesta del servicio.



Implementación de pruebas de rendimiento

Implementar pruebas de rendimiento en Postman permite a los equipos de desarrollo y QA **obtener métricas valiosas sobre la capacidad de respuesta, la estabilidad y la escalabilidad de los servicios web bajo diferentes escenarios de uso**. Esto facilita la toma de decisiones informadas sobre optimizaciones y mejoras necesarias.

Pre-requisitos para la prueba de rendimiento en Postman

Antes de iniciar pruebas de rendimiento, **es crucial definir claramente la necesidad y los objetivos específicos asociados con el servicio web que se va a evaluar**. Estos preparativos son esenciales para asegurar que las pruebas proporcionen datos útiles y relevantes.

Información necesaria para configurar la prueba

- Parámetros de carga esperada.
- Configuración previa en Postman.

- **Parámetros de carga esperada:**

- **Descripción:** entender el volumen de concurrencia esperado es fundamental para configurar las pruebas de manera adecuada. Esto incluye definir el número de usuarios virtuales que interactuarán con el servicio web durante la prueba.

Ejemplo de configuración: determinar si el servicio debe manejar cientos, miles, o más solicitudes simultáneas.

- **Configuración previa en Postman:**

- **Preparación del servicio:** asegúrate de que el servicio web está completamente configurado en Postman antes de iniciar las pruebas. Esto incluye tener todos los endpoints correctamente establecidos con los parámetros necesarios, headers, y cuerpos de solicitud.
- **Validación de la configuración:** realiza pruebas básicas para verificar que todos los endpoints respondan como se espera sin la carga de prueba, asegurándote de que cualquier fallo básico sea corregido antes de proceder a pruebas más intensivas.

Recomendaciones para una preparación efectiva

Preparar adecuadamente los pre-requisitos no solo facilita la ejecución de las pruebas de rendimiento, sino que también mejora la calidad de los resultados obtenidos.

Establecer claras expectativas y preparar detalladamente el entorno de prueba ayuda a identificar más efectivamente problemas potenciales y áreas de mejora en el servicio web.

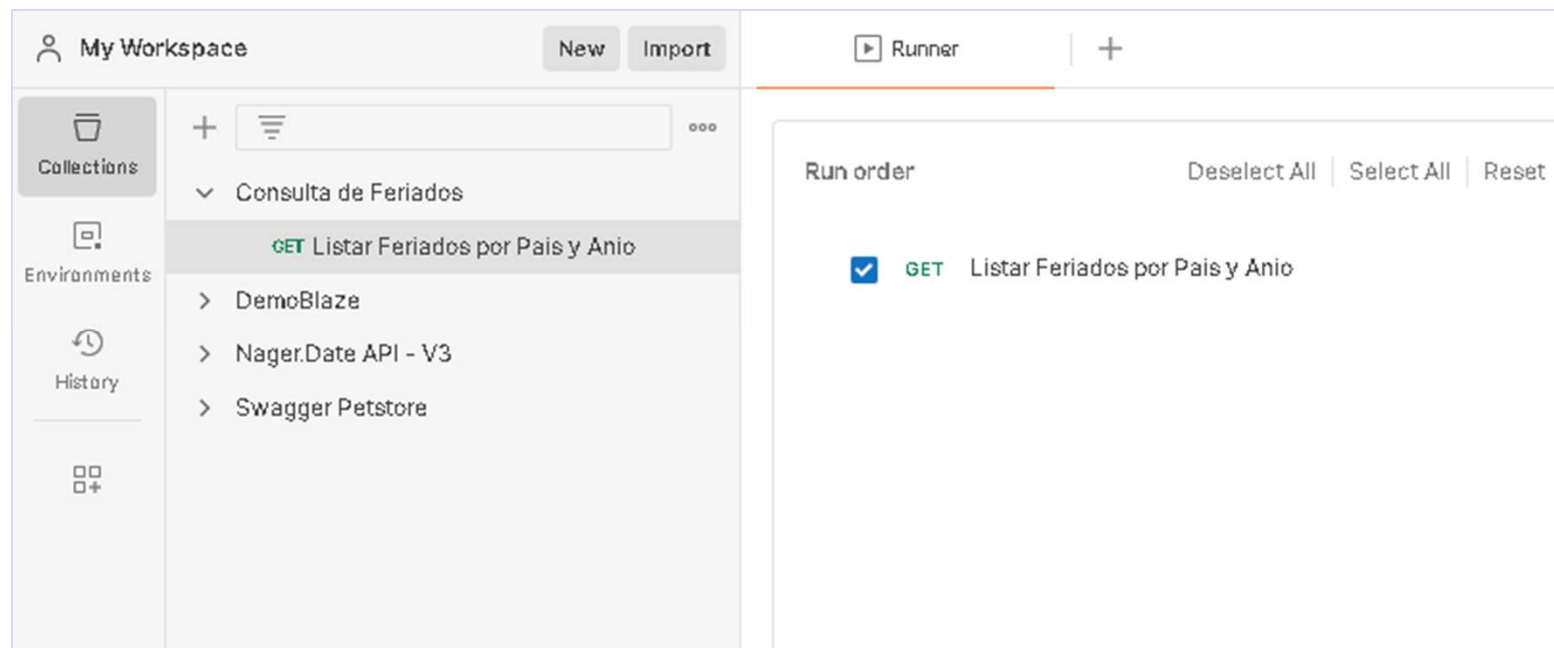
Configuración de una prueba de rendimiento en Postman


Pasos a seguir

1. Hacer clic en la opción **Runner**.
2. Hacer clic y arrastrar el mouse con el *endpoint* que se desea probar.
3. Se hace clic en la sección **Performance**.
4. Se configuran los datos requeridos para la prueba de rendimiento.

1. Hacer clic en la opción **Runner**.
2. Hacer clic y arrastrar el mouse con el **endpoint** que se desea probar.

Si la *collection* tiene más de un *endpoint* se puede hacer una prueba múltiple o se puede desmarcar los que no se deseen probar.



- 
3. Se hace clic en la sección ***Performance***.
 4. Se configuran los datos requeridos para la prueba de rendimiento:
 - **Usuarios virtuales:** concurrencia en las invocaciones al *endpoint*.
 - **Duración:** minutos de duración de la prueba.
 - **Perfil de carga:** tipo de prueba a ejecutar. Usualmente ***fixed***.

Runner

+

No environment

Run order

Deselect All

Select All

Reset

☒ GET Listar Feriados por Pais y Anio

Functional

Performance

Test how your APIs perform under load

×

Simulate real-world traffic from your local machine and observe the performance of your APIs. Learn more about [performance testing](#)

Set up your performance test

Load profile ⓘ

Virtual users ⓘ

Test duration

Fixed

20

1 mins

Perfiles de carga en Postman

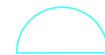
Tipo	Descripción
<i>Fixed</i>	El número de invocaciones simultáneas se mantiene constante durante toda la prueba.
<i>Ramp up</i>	El número de invocaciones simultáneas incrementa hasta llegar al nivel deseado.
<i>Spike</i>	El número de invocaciones simultáneas aumenta repentinamente y luego comienza a descender.
<i>Peak</i>	El número de invocaciones simultáneas aumenta y se mantiene así para luego disminuir.

Ejecución de prueba de rendimiento en Postman

Una vez que has configurado correctamente el servicio web en Postman y definido los parámetros de la prueba, **el siguiente paso es iniciar la prueba de rendimiento.**

Pasos para ejecutar la prueba

1. Selección del *Endpoint*.
2. Configuración final de la prueba.
3. Inicio de la prueba.



-
1. **Selección del *Endpoint*:** asegúrate de que el *endpoint* que se va a probar esté correctamente seleccionado en Postman. Verifica que todos los parámetros y configuraciones estén ajustados según las necesidades de la prueba.
 2. **Configuración final de la prueba:** revisa las configuraciones de rendimiento establecidas, como la concurrencia de usuarios virtuales, la duración de la prueba, y el perfil de carga. Asegúrate de que todos los ajustes reflejen el escenario de carga que deseas simular.
 3. **Inicio de la prueba:** Haz clic en el botón ***Run*** para comenzar la prueba de rendimiento. Este botón generalmente se encuentra en la interfaz del ***Runner*** de Postman, donde puedes ejecutar colecciones y pruebas específicas.

Consulta de Feriados (#1) IN PROGRESS

1 min left

Stop

Consulta de Feriados - 20 VU - Jun 29, 2024 20:07:47 (1 min) - Fixed

Summary Errors

Total requests sent ⓘ

91

Requests/second ⓘ

11.69

Avg. response time ⓘ

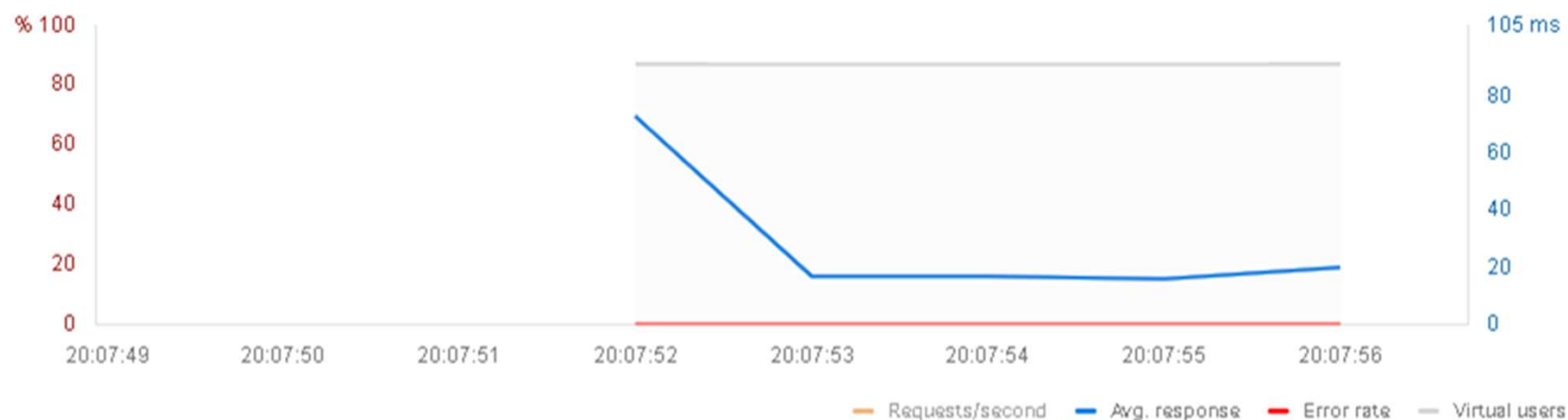
29 ms

Error rate ⓘ

0.00 %

Filter by requests ▾

Avg. response ▾



Interpretación de la prueba de rendimiento en Postman

Métrica	Descripción
<i>Total request sent</i>	Total de invocaciones realizadas en el tiempo de la prueba.
<i>Request/second</i>	Número de invocaciones por segundo.
<i>Avg. Response time</i>	Tiempo de respuesta promedio de las invocaciones.
<i>Error rate</i>	Porcentaje de invocaciones que resultaron con errores.



Monitoreo de la prueba

- **Visualización del progreso:** durante la ejecución de la prueba, Postman proporcionará un gráfico en tiempo real que muestra el progreso de la prueba. Este gráfico incluirá métricas como el número de solicitudes enviadas, la tasa de solicitudes por segundo, y los tiempos de respuesta.
- **Interpretación de resultados:** observa el gráfico cuidadosamente para entender cómo el servicio web está manejando la carga aplicada. Presta atención especial a picos en los tiempos de respuesta o a un aumento en los errores, lo que podría indicar problemas de rendimiento.

Al finalizar la prueba, revisa los resultados acumulados para evaluar el **rendimiento general del servicio web**. Utiliza estos datos para identificar áreas de mejora o para validar que el servicio cumple con los requisitos de rendimiento establecidos.



Reporte de resultados de la prueba de rendimiento

Al hacer clic en “...” > **Export PDF report** o en la opción **Export HTML report** se generará un archivo con los resultados de la prueba de rendimiento.

Además del gráfico principal que se mostraba en pantalla, se detallará información sobre los errores arrojados.

Cuando la prueba involucra varios *endpoints*, este reporte **muestra una comparación en las métricas de cada uno**, lo cual facilita determinar dónde podría necesitarse una optimización.

Test setup

Virtual users
20 VU

Duration
1 minute

Start time
Jun 29, 20:07:48 (GMT-3)

End time
Jun 29, 20:08:55 (GMT-3)

Load profile
Fixed

Environment
-

1. Summary

Total requests sent
1,130

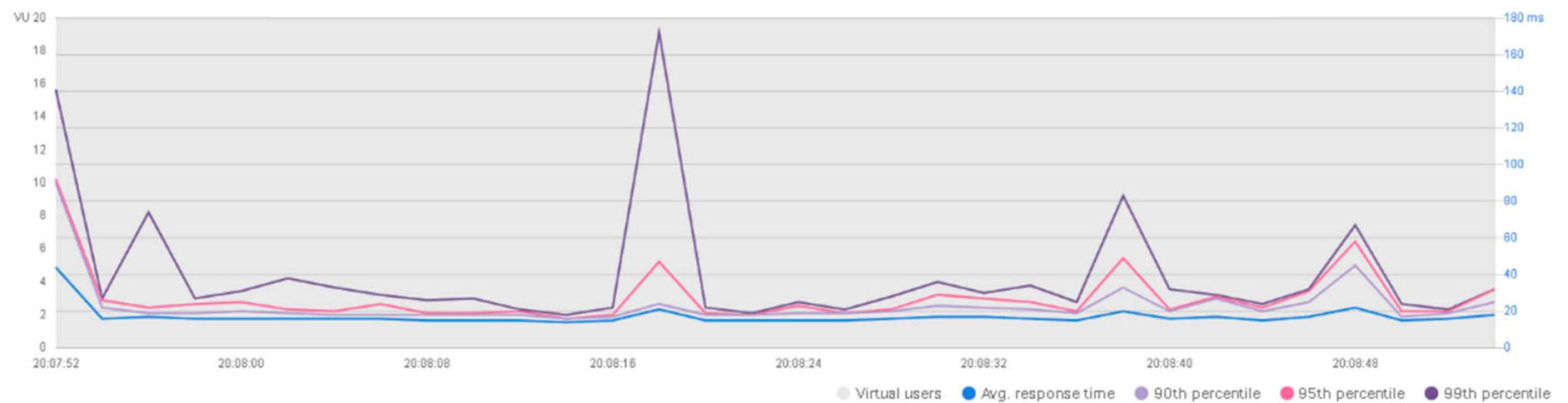
Throughput
16.95 requests/second

Average response time
17 ms

Error rate
0.00 %

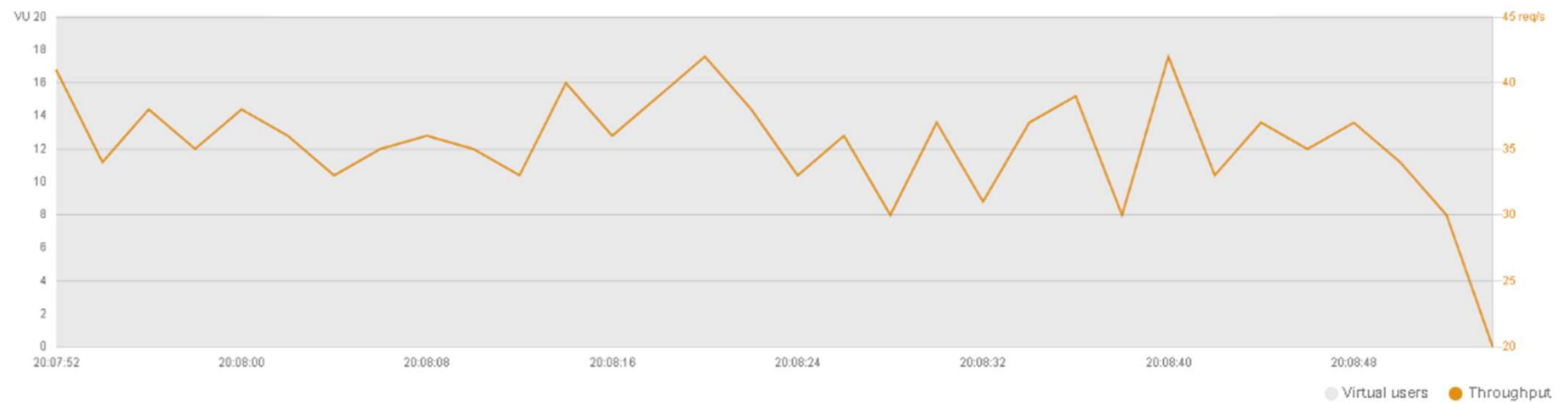
1.1 Response time

Response time trends during the test duration.



1.2 Throughput

Rate of requests sent per second during the test duration.



1.3 Requests with slowest response times

Top 5 slowest requests based on their average response times.

Request	Resp. time (Avg ms)	90th (ms)	95th (ms)	99th (ms)	Min (ms)	Max (ms)
GET Listar Feriados por Pais y Anio https://date.nager.at/api/v3/publicholidays/2024/AT	17	20	27	75	11	172

2. Metrics for each request

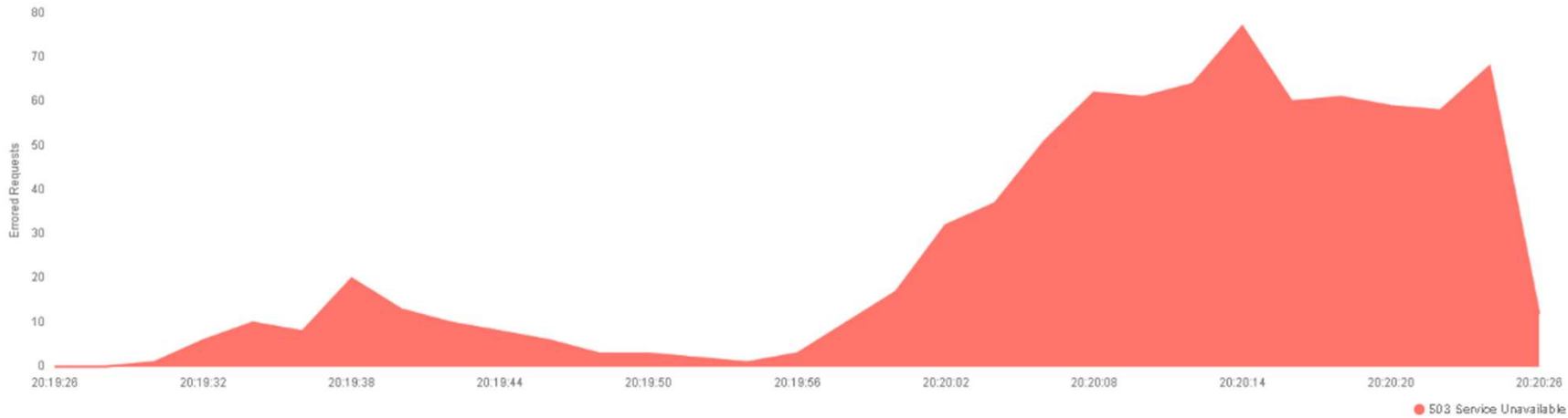
The requests are shown in the order they were sent by virtual users.

Request	Total requests	Requests/s	Min (ms)	Avg (ms)	90th (ms)	Max (ms)	Error %
GET Listar Feriados por Pais y Anio https://date.nager.at/api/v3/publicholidays/2024/AT	1,130	16.95	11	17	20	172	0

3. Errors

3.1 Error distribution over time

Top 5 error classes observed during the test duration.



3.2 Error distribution for requests

Errored requests grouped by error class, along with the error count for each class.

Error class	Total counts
503 Service Unavailable	823
GET Suma	823