

# Performance Test Report - Dec 5, 2024 (#1)

[Open in Postman](#)

Postman collection: Teste\_viaccep

Report exported on: Dec 5, 2024, 2:00:42 (GMT-3)

## Test setup

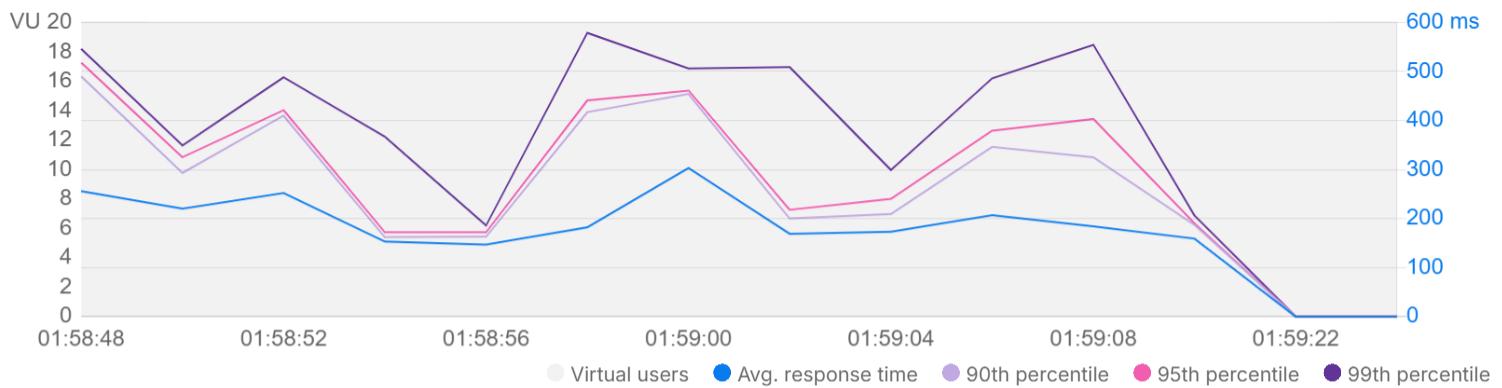
Virtual users	Start time	Load profile
20 VU	Dec 5, 1:58:42 (GMT-3)	Fixed
Duration	End time	Environment
1 minute	Dec 5, 1:59:49 (GMT-3)	-

## 1. Summary

Total requests sent	Throughput	Average response time	Error rate
1,546	23.29 requests/second	193 ms	56.02 %

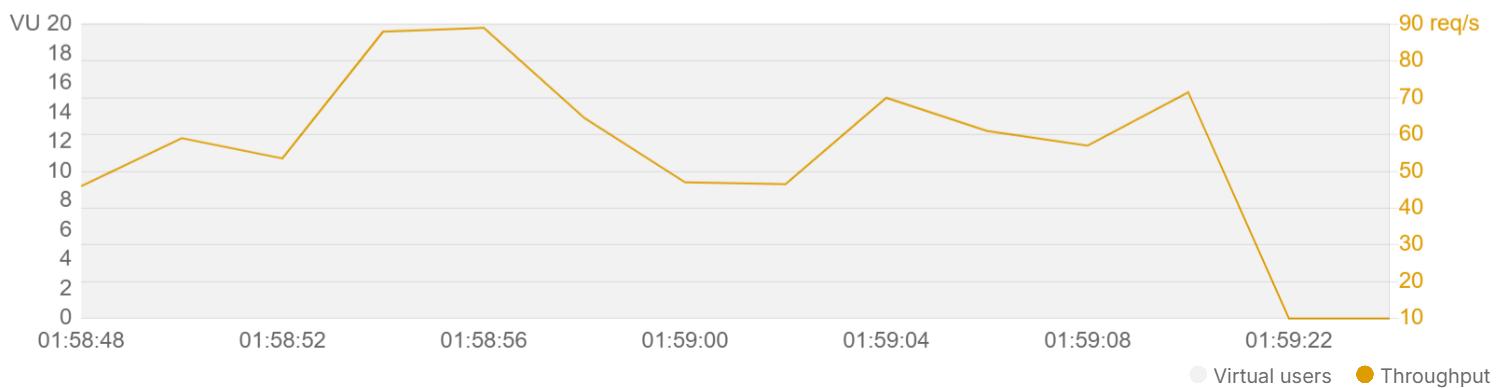
### 1.1 Response time

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)
<code>GET viacep_wrong_uf</code> <code>{url}/ws/SP/{{cidade}}/{{logradouro}}/json/</code>	450	505	506	509	192	509
<code>GET viacep_json_with_space</code> <code>{url}/ws/010 1000/json/</code>	445	460	468	579	413	579
<code>GET viacep_json_alphanumeric</code> <code>{url}/ws/010a1000/json/</code>	416	460	461	684	145	684
<code>GET viacep_3 char_50_hits_xml</code> <code>{url}/ws/{{UF}}/{{cidade}}/dom/xml/</code>	368	466	486	545	293	545
<code>GET viacep_3 char_50_hits_json</code> <code>{url}/ws/{{UF}}/{{cidade}}/dom/json/</code>	342	412	414	552	153	552

## 1.4 Requests with most errors

Top 5 requests with the most errors, along with the most frequently occurring errors for each request.

Request	Total error count	Error 1	Error 2	Other errors
<b>POST</b> viacep_json_post {{url}}/ws/{{cep}}/json/	40	405 Not Allowed (40)	-	0
<b>PATCH</b> viacep_json_path {{url}}/ws/{{cep}}/json/	40	405 Not Allowed (40)	-	0
<b>PUT</b> viacep_json_put {{url}}/ws/{{cep}}/json/	40	405 Not Allowed (40)	-	0
<b>DELETE</b> viacep_json_delete {{url}}/ws/{{cep}}/json/	40	405 Not Allowed (40)	-	0
<b>POST</b> viacep_json_post_callback {{url}}/ws/{{cep}}/json/?callback={{callback}}	40	405 Not Allowed (40)	-	0

## 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 %
<b>GET</b> viacep_json {{url}}/ws/{{cep}}/json/	40	0.60	133	314	518	546	0
<b>GET</b> viacep_callback {{url}}/ws/{{cep}}/json/?callback={{callback}}	40	0.60	136	153	167	185	0
<b>GET</b> viacep_xml {{url}}/ws/{{cep}}/xml/	40	0.60	135	152	170	199	0
<b>GET</b> viacep_partial_name_json {{url}}/ws/{{UF}}/{{cidade}}/{{logradouro}}/json/	40	0.60	142	204	336	429	0
<b>GET</b> viacep_partial_name_callback {{url}}/ws/{{UF}}/{{cidade}}/{{logradouro}}/json/?callback={{callback}}	40	0.60	142	207	277	323	0

<b>GET</b> viacep_partial_name_xml	40	0.60	152	227	324	368	0
{{url}}/ws/{{UF}}/{{cidade}}/{{logradouro}}/xml/							
<b>GET</b> viacep_partial_name_with_space_json	40	0.60	141	180	238	269	0
{{url}}/ws/{{UF}}/{{cidade}}/{{logradouro_com_espaco}}/json/							
<b>GET</b> viacep_partial_name_with_space_callback	40	0.60	145	188	258	275	0
{{url}}/ws/{{UF}}/{{cidade}}/{{logradouro_com_espaco}}/json/?callback={{callback}}							
<b>GET</b> viacep_partial_name_with_space_xml	40	0.60	146	198	290	303	0
{{url}}/ws/{{UF}}/{{cidade}}/{{logradouro_com_espaco}}/xml/							
<b>GET</b> viacep_partial_name_with_plus_json	40	0.60	142	189	248	325	0
{{url}}/ws/{{UF}}/{{cidade}}/{{logradouro_com_mais}}/json/							
on/?callback={{callback}}							
<b>GET</b> viacep_partial_name_with_plus_xml	40	0.60	147	199	277	317	0
{{url}}/ws/{{UF}}/{{cidade}}/{{logradouro_com_mais}}/xml/							
<b>GET</b> viacep_3_char_50_hits_json	40	0.60	153	342	412	552	0
{{url}}/ws/{{UF}}/{{cidade}}/dom/json/							
<b>GET</b> viacep_3_char_50_hits_callback	40	0.60	147	183	223	306	0
{{url}}/ws/{{UF}}/{{cidade}}/dom/json/?callback={{callback}}							
<b>GET</b> viacep_3_char_50_hits_xml	40	0.60	293	368	466	545	0
{{url}}/ws/{{UF}}/{{cidade}}/dom/xml/							
<b>POST</b> viacep_json_post	40	0.60	135	149	167	208	100
{{url}}/ws/{{cep}}/json/							
<b>PATCH</b> viacep_json_path	40	0.60	134	148	162	172	100
{{url}}/ws/{{cep}}/json/							

<b>PUT</b> viacep_json_put	40	0.60	134	153	170	213	100
<code>{url}/ws/{{cep}}/json/</code>							
<b>DELETE</b> viacep_json_delete	40	0.60	132	149	161	214	100
<code>{url}/ws/{{cep}}/json/</code>							
<b>POST</b> viacep_json_post_callback	40	0.60	131	164	170	591	100
<code>{url}/ws/{{cep}}/json/?callback={{callback}}</code>							
<b>PATCH</b> viacep_json_path_callback	40	0.60	134	168	187	554	100
<code>{url}/ws/{{cep}}/json/?callback={{callback}}</code>							
<b>PUT</b> viacep_json_put_callback	40	0.60	135	151	162	211	100
<code>{url}/ws/{{cep}}/json/?callback={{callback}}</code>							
<b>DELETE</b> viacep_json_delete_callback	40	0.60	131	149	166	187	100
<code>{url}/ws/{{cep}}/json/?callback={{callback}}</code>							
<b>POST</b> viacep_json_post_xml	40	0.60	133	158	185	203	100
<code>{url}/ws/{{cep}}/xml/</code>							
<b>PATCH</b> viacep_json_path_xml	40	0.60	138	162	178	544	100
<code>{url}/ws/{{cep}}/json/</code>							
<b>PUT</b> viacep_json_put_xml	40	0.60	133	151	169	186	100
<code>{url}/ws/{{cep}}/json/</code>							
<b>DELETE</b> viacep_json_delete_xml	39	0.59	134	162	206	403	100
<code>{url}/ws/{{cep}}/json/</code>							
<b>POST</b> viacep_json_post_without_cep	34	0.51	133	155	179	191	100
<code>{url}/ws/{{UF}}/{{cidade}}/{{logradouro_com_espaco}}</code> <code>/json/</code>							
<b>PATCH</b> viacep_json_path_without_cep	33	0.50	135	147	159	183	100
<code>{url}/ws/{{UF}}/{{cidade}}/{{logradouro_com_espaco}}</code> <code>/json/</code>							
<b>PUT</b> viacep_json_put_without_cep	33	0.50	135	149	164	188	100
<code>{url}/ws/{{UF}}/{{cidade}}/{{logradouro_com_espaco}}</code> <code>/json/</code>							

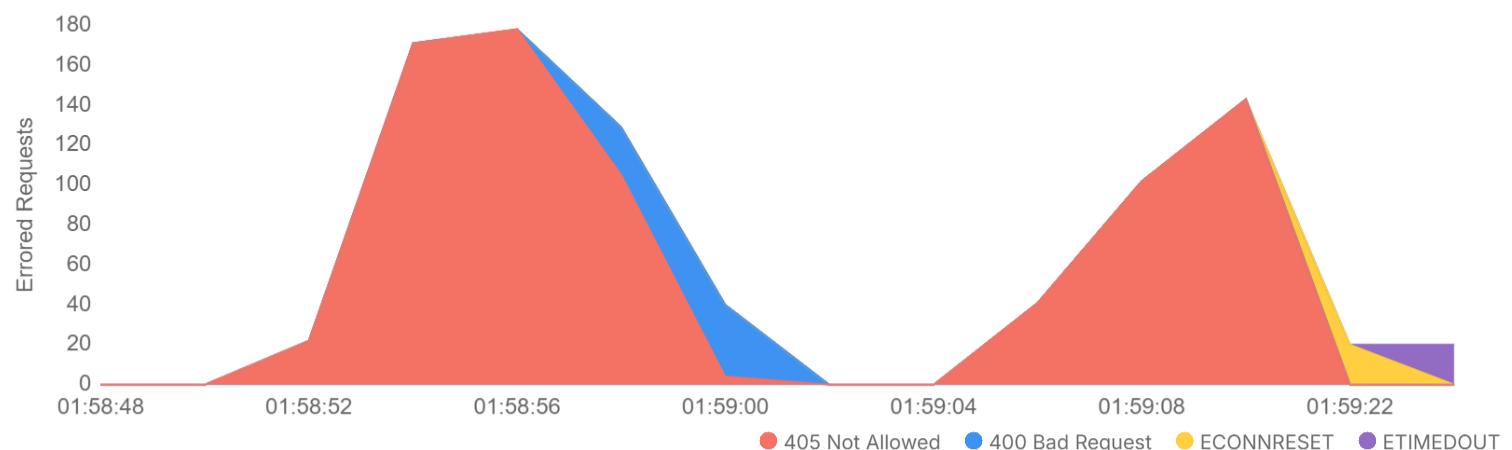
<b>DELETE</b> viacep_json_delete_without_cep	30	0.45	137	147	157	187	100
{{url}}/ws/{{UF}}/{{cidade}}/{{logradouro_com_espaco}}/json/							
<b>POST</b> viacep_json_post_without_cep_callback	28	0.42	133	151	182	202	100
{{url}}/ws/{{UF}}/{{cidade}}/{{logradouro_com_espaco}}/json/?callback={{callback}}							
<b>PATCH</b> viacep_json_path_without_cep_callback	28	0.42	132	149	160	190	100
{{url}}/ws/{{UF}}/{{cidade}}/{{logradouro_com_espaco}}/json/?callback={{callback}}							
<b>PUT</b> viacep_json_put_without_cep_callback	27	0.41	135	146	156	181	100
{{url}}/ws/{{UF}}/{{cidade}}/{{logradouro_com_espaco}}/json/?callback={{callback}}							
<b>DELETE</b> viacep_json_delete_without_cep_callback	26	0.39	138	150	156	201	100
{{url}}/ws/{{UF}}/{{cidade}}/{{logradouro_com_espaco}}/json/?callback={{callback}}							
<b>POST</b> viacep_json_post_without_cep_xml	25	0.38	136	146	154	158	100
{{url}}/ws/{{UF}}/{{cidade}}/{{logradouro_com_espaco}}/xml/							
<b>PATCH</b> viacep_json_path_without_cep_xml	22	0.33	133	143	153	155	100
{{url}}/ws/{{UF}}/{{cidade}}/{{logradouro_com_espaco}}/xml/							
<b>PUT</b> viacep_json_put_without_cep_xml	21	0.32	136	160	160	409	100
{{url}}/ws/{{UF}}/{{cidade}}/{{logradouro_com_espaco}}/xml/							
<b>DELETE</b> viacep_json_delete_without_cep_xml	20	0.30	135	146	157	160	100
{{url}}/ws/{{UF}}/{{cidade}}/{{logradouro_com_espaco}}/xml/							
<b>GET</b> viacep_json_9_digitos	20	0.30	135	200	416	440	100
{{url}}/ws/{{cep}}0/json/							
<b>GET</b> viacep_json_alphanumeric	20	0.30	145	416	460	684	100
{{url}}/ws/010a1000/json/							
<b>GET</b> viacep_json_with_space	20	0.30	413	445	460	579	100
{{url}}/ws/010 1000/json/							

<code>GET</code> viacep_wrong_uf	20	0.30	192	450	505	509	0
{{url}}/ws/SP/{{cidade}}/{{logradouro}}/json/							
<code>GET</code> viacep_wrong_city	20	0.30	165	188	207	213	0
{{url}}/ws/{{UF}}/São Paulo/{{logradouro}}/json/							
<code>GET</code> viacep_wrong_street	20	0.30	158	201	222	507	0
{{url}}/ws/{{UF}}/{{cidade}}/Humberto Borella/json/							
<code>GET</code> viacep_wrong_street Copy	20	0.30	152	177	201	218	0
{{url}}/ws/{{UF}}/{{cidade}}/Humberto Borella/json/							

## 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
405 Not Allowed	766

<b>POST</b> viacep_json_post	40
<b>PATCH</b> viacep_json_path	40
<b>PUT</b> viacep_json_put	40
<b>DELETE</b> viacep_json_delete	40
<b>POST</b> viacep_json_post_callback	40
<b>PATCH</b> viacep_json_path_callback	40
<b>PUT</b> viacep_json_put_callback	40
<b>DELETE</b> viacep_json_delete_callback	40
<b>POST</b> viacep_json_post_xml	40
<b>PATCH</b> viacep_json_path_xml	39
<b>PUT</b> viacep_json_put_xml	34
<b>DELETE</b> viacep_json_delete_xml	33
<b>POST</b> viacep_json_post_without_cep	33
<b>PATCH</b> viacep_json_path_without_cep	30
<b>PUT</b> viacep_json_put_without_cep	28
<b>DELETE</b> viacep_json_delete_without_cep	28
<b>POST</b> viacep_json_post_without_cep_callback	27
<b>PATCH</b> viacep_json_path_without_cep_callback	26
<b>PUT</b> viacep_json_put_without_cep_callback	25
<b>DELETE</b> viacep_json_delete_without_cep_callback	22
<b>POST</b> viacep_json_post_without_cep_xml	21
<b>PATCH</b> viacep_json_path_without_cep_xml	20
<b>PUT</b> viacep_json_put_without_cep_xml	20
<b>DELETE</b> viacep_json_delete_without_cep_xml	20
400 Bad Request	60
<b>GET</b> viacep_json_9_digitos	20
<b>GET</b> viacep_json_alphanumeric	20
<b>GET</b> viacep_json_with_space	20
ECONNRESET	20

PATCH viacep_json_path_xml	1
PUT viacep_json_put_xml	5
DELETE viacep_json_delete_xml	1
PATCH viacep_json_path_without_cep	3
PUT viacep_json_put_without_cep	2
POST viacep_json_post_without_cep_callback	1
PATCH viacep_json_path_without_cep_callback	1
PUT viacep_json_put_without_cep_callback	1
DELETE viacep_json_delete_without_cep_callback	3
POST viacep_json_post_without_cep_xml	1
PATCH viacep_json_path_without_cep_xml	1
ETIMEDOUT	20
PUT viacep_json_put_xml	1
DELETE viacep_json_delete_xml	5
POST viacep_json_post_without_cep	1
PUT viacep_json_put_without_cep	3
DELETE viacep_json_delete_without_cep	2
PATCH viacep_json_path_without_cep_callback	1
PUT viacep_json_put_without_cep_callback	1
DELETE viacep_json_delete_without_cep_callback	1
POST viacep_json_post_without_cep_xml	3
PATCH viacep_json_path_without_cep_xml	1
PUT viacep_json_put_without_cep_xml	1



## Testing API performance on Postman

Postman enables you to simulate user traffic and observe how your API behaves under load. It also helps you identify any issues or bottlenecks that affect performance.

Learn more about [testing API performance](#).