



44
3

Scenarios

2025-06-12 01:54:56 p. m.

[1.1:28] [Validate new booking creation](#)

[2.1:41] [Validate mandatory fields in the request](#)

[2.2:42] [Validate mandatory fields in the request](#)

[2.3:43] [Validate mandatory fields in the request](#)

[2.4:44] [Validate mandatory fields in the request](#)

[3.1:55] [Validate errors for invalid request content](#)

[3.2:55] [Validate errors for invalid request content](#)

[3.3:55] [Validate errors for invalid request content](#)

[3.4:55] [Validate errors for invalid request content](#)

[3.5:55] [Validate errors for invalid request content](#)

[3.6:55] [Validate errors for invalid request content](#)

[3.7:55] [Validate errors for invalid request content](#)

[3.8:55] [Validate errors for invalid request content](#)

[3.9:55] [Validate errors for invalid request content](#)

[3.10:55] [Validate errors for invalid request content](#)

[3.11:55] [Validate errors for invalid request content](#)

[3.12:55] [Validate errors for invalid request content](#)

Summary Tags Feature: src/test/java/bookingOperations/createBooking.feature Booking Operations	
<div><div>bookingOperations</div><div>newBooking</div><div>Operations</div></div>	ms: 472
Scenario: [1.1:28] Validate new booking creation	
<div>>> Background:</div> <div><div>5</div><div>* def dataRequest = read('classpath:data/bookingOperations/requestCreateNewBooking.json')</div><div>43</div></div> <div><div>6</div><div>* def structureError400 = read('classpath:data/structureError.json')</div><div>3</div></div> <div><div>7</div><div>* def structureResponse = read('classpath:data/bookingOperations/structureResponse200.json')</div><div>2</div></div> <div><div>8</div><div>* def dateValidator = read('classpath:helpers/dateValidator.js')</div><div>10</div></div> <div><div>9</div><div>Given url apiUrl</div><div>1</div></div> <div><div>10</div><div>Given path 'booking'</div><div>1</div></div> <div><div>11</div><div>Given headers { accept: 'application/json', Content-Type: 'application/json' }</div><div>1</div></div> <div><div>16</div><div>* print dataRequest</div><div>5</div></div> <div>13:54:54.849 [print] { "date": "2024-06-10", "destination": "MEX", "origin": "JFK", "userId": 1 }</div> <div><div>17</div><div>And request dataRequest</div><div>1</div></div> <div><div>18</div><div>When method POST</div><div>371</div></div> <div>13:54:55.159 request: 1 > POST http://127.0.0.1:8900/booking 1 > accept: application/json 1 > Content-Type: application/json; charset=UTF-8 1 > Content-Length: 67 1 > Host: 127.0.0.1:8900 1 > Connection: Keep-Alive 1 > User-Agent: Apache-HttpClient/4.5.14 (Java/22.0.1) 1 > Accept-Encoding: gzip,deflate { "date": "2024-06-10", "destination": "MEX", "origin": "JFK", "userId": 1 }</div> <div>13:54:55.220 response time in milliseconds: 59 1 < 201 1 < X-Powered-By: Express 1 < content-type: application/json 1 < Date: Thu, 12 Jun 2025 18:54:55 GMT 1 < Connection: keep-alive 1 < Keep-Alive: timeout=5 1 < Transfer-Encoding: chunked { "date": "2024-06-10", "destination": "MEX", "id": 3, "origin": "JFK", "userId": 1 }</div> <div><div>19</div><div>Then status 201</div><div>3</div></div> <div><div>20</div><div>* print response</div><div>1</div></div> <div>13:54:55.226 [print] { "date": "2024-06-10", "destination": "MEX", "id": 3, "origin": "JFK", "userId": 1 }</div> <div><div>21</div><div>And match response == structureResponse</div><div>27</div></div> <div><div>22</div><div>And match response.date == date</div><div>1</div></div> <div><div>23</div><div>And match response.destination == destination</div><div>1</div></div> <div><div>24</div><div>And match response.origin == origin</div><div>0</div></div> <div><div>25</div><div>And match response.userId == dataRequest.userId</div><div>1</div></div>	
<div><div>bookingOperations</div><div>MandatoryFieldsReqValNB</div><div>Operations</div></div>	ms: 24
Scenario: [2.1:41] Validate mandatory fields in the request	
<div>>> Background:</div>	



44

3

Scenarios

2025-06-12 01:54:56 p. m.

[\[1.1:28\] Validate new booking creation](#)[\[2.1:41\] Validate mandatory fields in the request](#)[\[2.2:42\] Validate mandatory fields in the request](#)[\[2.3:43\] Validate mandatory fields in the request](#)[\[2.4:44\] Validate mandatory fields in the request](#)[\[3.1:55\] Validate errors for invalid request content](#)[\[3.2:55\] Validate errors for invalid request content](#)[\[3.3:55\] Validate errors for invalid request content](#)[\[3.4:55\] Validate errors for invalid request content](#)[\[3.5:55\] Validate errors for invalid request content](#)[\[3.6:55\] Validate errors for invalid request content](#)[\[3.7:55\] Validate errors for invalid request content](#)[\[3.8:55\] Validate errors for invalid request content](#)[\[3.9:55\] Validate errors for invalid request content](#)[\[3.10:55\] Validate errors for invalid request content](#)[\[3.11:55\] Validate errors for invalid request content](#)[\[3.12:55\] Validate errors for invalid request content](#)

32 * remove dataRequest.date	1
33 * print dataRequest	1
13:54:55.276 [print] { "destination": "MEX", "origin": "JFK", "userId": 1 }	
34 And request dataRequest	0
35 When method POST	10
13:54:55.277 request: 1 > POST http://127.0.0.1:8900/booking 1 > accept: application/json 1 > Content-Type: application/json; charset=UTF-8 1 > Content-Length: 47 1 > Host: 127.0.0.1:8900 1 > Connection: Keep-Alive 1 > User-Agent: Apache-HttpClient/4.5.14 (Java/22.0.1) 1 > Accept-Encoding: gzip,deflate { "destination": "MEX", "origin": "JFK", "userId": 1 }	
13:54:55.285 response time in milliseconds: 8 1 < 400 1 < X-Powered-By: Express 1 < Content-Type: application/json; charset=utf-8 1 < Content-Length: 222 1 < ETag: W/"de-Bx8Bjdbiu0pdreKV0JSU3Bk8X9k" 1 < Date: Thu, 12 Jun 2025 18:54:55 GMT 1 < Connection: keep-alive 1 < Keep-Alive: timeout=5 { "errors": [{ "location": { "in": "request", "name": "body", "docPath": "/paths/~1booking/post/requestBody/content/application~1json/schema", "path": "" }, "message": "must have required property 'date'" }, { "message": "Validation errors" }] }	
36 Then status 400	0
37 * print response	1
13:54:55.287 [print] { "errors": [{ "location": { "in": "request", "name": "body", "docPath": "/paths/~1booking/post/requestBody/content/application~1json/schema", "path": "" }, "message": "must have required property 'date'" }, { "message": "Validation errors" }], "message": "Validation errors" }	
38 And match response == structureError400	4

bookingOperations MandatoryFieldsReqValNB Operations	ms: 12
Scenario: [2.2:42] Validate mandatory fields in the request	
>> Background:	
32 * remove dataRequest.destination	0
33 * print dataRequest	0
13:54:55.301 [print] { "date": "2024-06-10", "origin": "JFK", "userId": 1 }	
34 And request dataRequest	0
35 When method POST	7
13:54:55.302 request: 1 > POST http://127.0.0.1:8900/booking 1 > accept: application/json 1 > Content-Type: application/json; charset=UTF-8 1 > Content-Length: 47	



44
3

Scenarios

2025-06-12 01:54:56 p. m.

[\[1.1:28\] Validate new booking creation](#)

[\[2.1:41\] Validate mandatory fields in the request](#)

[\[2.2:42\] Validate mandatory fields in the request](#)

[\[2.3:43\] Validate mandatory fields in the request](#)

[\[2.4:44\] Validate mandatory fields in the request](#)

[\[3.1:55\] Validate errors for invalid request content](#)

[\[3.2:55\] Validate errors for invalid request content](#)

[\[3.3:55\] Validate errors for invalid request content](#)

[\[3.4:55\] Validate errors for invalid request content](#)

[\[3.5:55\] Validate errors for invalid request content](#)

[\[3.6:55\] Validate errors for invalid request content](#)

[\[3.7:55\] Validate errors for invalid request content](#)

[\[3.8:55\] Validate errors for invalid request content](#)

[\[3.9:55\] Validate errors for invalid request content](#)

[\[3.10:55\] Validate errors for invalid request content](#)

[\[3.11:55\] Validate errors for invalid request content](#)

[\[3.12:55\] Validate errors for invalid request content](#)

1 > Host: 127.0.0.1:8900	
1 > Connection: Keep-Alive	
1 > User-Agent: Apache-HttpClient/4.5.14 (Java/22.0.1)	
1 > Accept-Encoding: gzip,deflate	
{"date":"2024-06-10","origin":"JFK","userId":1}	
13:54:55.307 response time in milliseconds: 5	
1 < 400	
1 < X-Powered-By: Express	
1 < Content-Type: application/json; charset=utf-8	
1 < Content-Length: 229	
1 < ETag: W/"e5-5wSq4vFK03d2mNhYUt8fGs/QxYc"	
1 < Date: Thu, 12 Jun 2025 18:54:55 GMT	
1 < Connection: keep-alive	
1 < Keep-Alive: timeout=5	
{"errors":[{"location":{"in":"request","name":"body","docPath":"/paths/~1booking/post/requestBody/content/application~1json/schema","path":""},"message":"must have required property 'destination'"]},"message":"Validation errors"}	
36 Then status 400	0
37 * print response	1
13:54:55.309 [print] {	
"errors": [
{	
"location": {	
"in": "request",	
"name": "body",	
"docPath": "/paths/~1booking/post/requestBody/content/application~1json/schema",	
"path": ""	
},	
"message": "must have required property 'destination'"	
}	
],	
"message": "Validation errors"	
}	
38 And match response == structureError400	0

bookingOperations	MandatoryFieldsReqValNB	Operations	ms: 13
Scenario: [2.3:43] Validate mandatory fields in the request			
>> Background:			
32 * remove dataRequest.origin			0
33 * print dataRequest			0
13:54:55.321 [print] {			
"date": "2024-06-10",			
"destination": "MEX",			
"userId": 1			
}			
34 And request dataRequest			0
35 When method POST			7
13:54:55.322 request:			
1 > POST http://127.0.0.1:8900/booking			
1 > accept: application/json			
1 > Content-Type: application/json; charset=UTF-8			
1 > Content-Length: 52			
1 > Host: 127.0.0.1:8900			
1 > Connection: Keep-Alive			
1 > User-Agent: Apache-HttpClient/4.5.14 (Java/22.0.1)			
1 > Accept-Encoding: gzip,deflate			
{"date":"2024-06-10","destination":"MEX","userId":1}			
13:54:55.327 response time in milliseconds: 5			
1 < 400			
1 < X-Powered-By: Express			
1 < Content-Type: application/json; charset=utf-8			
1 < Content-Length: 224			
1 < ETag: W/"e0-Rw9X7VHTEUg8Umle9sjjBKUoXrM"			
1 < Date: Thu, 12 Jun 2025 18:54:55 GMT			
1 < Connection: keep-alive			
1 < Keep-Alive: timeout=5			
{"errors":[{"location":{"in":"request","name":"body","docPath":"/paths/~1booking/post/requestBody/			

44
3

Scenarios

2025-06-12 01:54:56 p. m.

[\[1.1:28\] Validate new booking creation](#)[\[2.1:41\] Validate mandatory fields in the request](#)[\[2.2:42\] Validate mandatory fields in the request](#)[\[2.3:43\] Validate mandatory fields in the request](#)[\[2.4:44\] Validate mandatory fields in the request](#)[\[3.1:55\] Validate errors for invalid request content](#)[\[3.2:55\] Validate errors for invalid request content](#)[\[3.3:55\] Validate errors for invalid request content](#)[\[3.4:55\] Validate errors for invalid request content](#)[\[3.5:55\] Validate errors for invalid request content](#)[\[3.6:55\] Validate errors for invalid request content](#)[\[3.7:55\] Validate errors for invalid request content](#)[\[3.8:55\] Validate errors for invalid request content](#)[\[3.9:55\] Validate errors for invalid request content](#)[\[3.10:55\] Validate errors for invalid request content](#)[\[3.11:55\] Validate errors for invalid request content](#)[\[3.12:55\] Validate errors for invalid request content](#)

```
content/application~1json/schema","path":""},"message":"must have required property 'origin'"]},"message":"Validation errors"}
```

36 Then status 400

[37](#) * print response

```
13:54:55.328 [print] {
  "errors": [
    {
      "location": {
        "in": "request",
        "name": "body",
        "docPath": "/paths/~1booking/post/requestBody/content/application~1json/schema",
        "path": ""
      },
      "message": "must have required property 'origin'"
    }
  ],
  "message": "Validation errors"
}
```

38 And match response == structureError400

bookingOperations

MandatoryFieldsReqValNB

Operations

ms: 13

Scenario: [2.4:44] **Validate mandatory fields in the request**

>> Background:

32 * remove dataRequest.userId

[33](#) * print dataRequest

```
13:54:55.338 [print] {
  "date": "2024-06-10",
  "destination": "MEX",
  "origin": "JFK"
}
```

34 And request dataRequest

[35](#) When method POST

```
13:54:55.339 request:
1 > POST http://127.0.0.1:8900/booking
1 > accept: application/json
1 > Content-Type: application/json; charset=UTF-8
1 > Content-Length: 56
1 > Host: 127.0.0.1:8900
1 > Connection: Keep-Alive
1 > User-Agent: Apache-HttpClient/4.5.14 (Java/22.0.1)
1 > Accept-Encoding: gzip,deflate
{"date":"2024-06-10","destination":"MEX","origin":"JFK"}
```

13:54:55.345 response time in milliseconds: 5

```
1 < 400
1 < X-Powered-By: Express
1 < Content-Type: application/json; charset=utf-8
1 < Content-Length: 224
1 < ETag: W/"e0-XiTcppVhbGEvrPnt75nYKFwIuYo"
1 < Date: Thu, 12 Jun 2025 18:54:55 GMT
1 < Connection: keep-alive
1 < Keep-Alive: timeout=5
{"errors":[{"location":{"in":"request","name":"body","docPath":"/paths/~1booking/post/requestBody/content/application~1json/schema","path":""},"message":"must have required property 'userId'"]},"message":"Validation errors"}
```

36 Then status 400

[37](#) * print response

38 And match response == structureError400

reqContValNB

bookingOperations

Operations

ms: 17

Scenario: [3.1:55] **Validate errors for invalid request content**

>> Background:

[5](#) * def dataRequest = read('classpath:data/bookingOperations/requestCreateNewBooking.json')

6 * def structureError400 = read('classpath:data/structureError.json')



44
3

Scenarios

2025-06-12 01:54:56 p. m.

[\[1.1:28\] Validate new booking creation](#)

[\[2.1:41\] Validate mandatory fields in the request](#)

[\[2.2:42\] Validate mandatory fields in the request](#)

[\[2.3:43\] Validate mandatory fields in the request](#)

[\[2.4:44\] Validate mandatory fields in the request](#)

[\[3.1:55\] Validate errors for invalid request content](#)

[\[3.2:55\] Validate errors for invalid request content](#)

[\[3.3:55\] Validate errors for invalid request content](#)

[\[3.4:55\] Validate errors for invalid request content](#)

[\[3.5:55\] Validate errors for invalid request content](#)

[\[3.6:55\] Validate errors for invalid request content](#)

[\[3.7:55\] Validate errors for invalid request content](#)

[\[3.8:55\] Validate errors for invalid request content](#)

[\[3.9:55\] Validate errors for invalid request content](#)

[\[3.10:55\] Validate errors for invalid request content](#)

[\[3.11:55\] Validate errors for invalid request content](#)

[\[3.12:55\] Validate errors for invalid request content](#)

7	* def structureResponse = read('classpath:data/bookingOperations/structureResponse200.json')	1
8	* def dateValidator = read('classpath:helpers/dateValidator.js')	1
9	Given url apiUrl	0
10	Given path 'booking'	0
11	Given headers { accept: 'application/json', Content-Type: 'application/json' }	0
48	* dataRequest.userId = 1	5
49	And request dataRequest	0
50	When method POST	8
13:54:55.385	request:	
1	> POST http://127.0.0.1:8900/booking	
1	> accept: application/json	
1	> Content-Type: application/json; charset=UTF-8	
1	> Content-Length: 57	
1	> Host: 127.0.0.1:8900	
1	> Connection: Keep-Alive	
1	> User-Agent: Apache-HttpClient/4.5.14 (Java/22.0.1)	
1	> Accept-Encoding: gzip,deflate	
	{"date":"","destination":"BOG","origin":"MAD","userId":1}	
13:54:55.390	response time in milliseconds: 5	
1	< 400	
1	< X-Powered-By: Express	
1	< Content-Type: application/json; charset=utf-8	
1	< Content-Length: 219	
1	< ETag: W/"db-VVy2zHmDTwFS0Lx4GtJxeG0Wn7Q"	
1	< Date: Thu, 12 Jun 2025 18:54:55 GMT	
1	< Connection: keep-alive	
1	< Keep-Alive: timeout=5	
	{"errors":[{"location":{"in":"request","name":"body","docPath":"/paths/~1booking/post/requestBody/content/application~1json/schema","path":"/date"},"message":"must match format \\\"date\\\""}],"message":"Validation errors"}	
51	Then status 400	0
52	* print response	1
13:54:55.392	[print] {	
	"errors": [
	{	
	"location": {	
	"in": "request",	
	"name": "body",	
	"docPath": "/paths/~1booking/post/requestBody/content/application~1json/schema",	
	"path": "/date"	
	},	
	"message": "must match format \\\"date\\\""	
	},	
	"message": "Validation errors"	
	}	
53	And match response == structureError400	0

<div>reqContValNB bookingOperations Operations</div>		ms: 14
Scenario: [3.2:55] Validate errors for invalid request content		
>> Background:		
5	* def dataRequest = read('classpath:data/bookingOperations/requestCreateNewBooking.json')	1
6	* def structureError400 = read('classpath:data/structureError.json')	1
7	* def structureResponse = read('classpath:data/bookingOperations/structureResponse200.json')	0
8	* def dateValidator = read('classpath:helpers/dateValidator.js')	1
9	Given url apiUrl	0
10	Given path 'booking'	0
11	Given headers { accept: 'application/json', Content-Type: 'application/json' }	0
48	* dataRequest.userId = 1	0
49	And request dataRequest	0
50	When method POST	10



44
3

Scenarios

2025-06-12 01:54:56 p. m.

[1.1:28] [Validate new booking creation](#)

[2.1:41] [Validate mandatory fields in the request](#)

[2.2:42] [Validate mandatory fields in the request](#)

[2.3:43] [Validate mandatory fields in the request](#)

[2.4:44] [Validate mandatory fields in the request](#)

[3.1:55] [Validate errors for invalid request content](#)

[3.2:55] [Validate errors for invalid request content](#)

[3.3:55] [Validate errors for invalid request content](#)

[3.4:55] [Validate errors for invalid request content](#)

[3.5:55] [Validate errors for invalid request content](#)

[3.6:55] [Validate errors for invalid request content](#)

[3.7:55] [Validate errors for invalid request content](#)

[3.8:55] [Validate errors for invalid request content](#)

[3.9:55] [Validate errors for invalid request content](#)

[3.10:55] [Validate errors for invalid request content](#)

[3.11:55] [Validate errors for invalid request content](#)

[3.12:55] [Validate errors for invalid request content](#)

13:54:55.403 request:	
1 > POST http://127.0.0.1:8900/booking	
1 > accept: application/json	
1 > Content-Type: application/json; charset=UTF-8	
1 > Content-Length: 61	
1 > Host: 127.0.0.1:8900	
1 > Connection: Keep-Alive	
1 > User-Agent: Apache-HttpClient/4.5.14 (Java/22.0.1)	
1 > Accept-Encoding: gzip,deflate	
{ "date": "2015", "destination": "BOG", "origin": "MAD", "userId": 1 }	
13:54:55.411 response time in milliseconds: 7	
1 < 400	
1 < X-Powered-By: Express	
1 < Content-Type: application/json; charset=utf-8	
1 < Content-Length: 219	
1 < ETag: W/"db-VVy2zHmDTwFS0Lx4GtJxeG0Wn7Q"	
1 < Date: Thu, 12 Jun 2025 18:54:55 GMT	
1 < Connection: keep-alive	
1 < Keep-Alive: timeout=5	
{ "errors": [{ "location": { "in": "request", "name": "body", "docPath": "/paths/~1booking/post/requestBody/content/application~1json/schema", "path": "/date" }, "message": "must match format \\"date\\""}], "message": "Validation errors" }	
51 Then status 400	0
52 * print response	1
13:54:55.413 [print] {	
"errors": [
{	
"location": {	
"in": "request",	
"name": "body",	
"docPath": "/paths/~1booking/post/requestBody/content/application~1json/schema",	
"path": "/date"	
},	
"message": "must match format \\"date\\""	
},	
"message": "Validation errors"	
]	
}	
53 And match response == structureError400	0

<div>reqContValNB</div>	<div>bookingOperations</div>	<div>Operations</div>	ms: 17
Scenario: [3.3:55] Validate errors for invalid request content			
>> Background:			
5 * def dataRequest = read('classpath:data/bookingOperations/requestCreateNewBooking.json')			2
6 * def structureError400 = read('classpath:data/structureError.json')			1
7 * def structureResponse = read('classpath:data/bookingOperations/structureResponse200.json')			1
8 * def dateValidator = read('classpath:helpers/dateValidator.js')			1
9 Given url apiUri			0
10 Given path 'booking'			0
11 Given headers { accept: 'application/json', Content-Type: 'application/json' }			0
48 * dataRequest.userId = 1			0
49 And request dataRequest			0
50 When method POST			10
13:54:55.426 request:			
1 > POST http://127.0.0.1:8900/booking			
1 > accept: application/json			
1 > Content-Type: application/json; charset=UTF-8			
1 > Content-Length: 61			
1 > Host: 127.0.0.1:8900			
1 > Connection: Keep-Alive			
1 > User-Agent: Apache-HttpClient/4.5.14 (Java/22.0.1)			
1 > Accept-Encoding: gzip,deflate			
{ "date": "ASDF", "destination": "BOG", "origin": "MAD", "userId": 1 }			
13:54:55.434 response time in milliseconds: 8			



44
3

Scenarios

2025-06-12 01:54:56 p. m.

[1.1:28] [Validate new booking creation](#)

[2.1:41] [Validate mandatory fields in the request](#)

[2.2:42] [Validate mandatory fields in the request](#)

[2.3:43] [Validate mandatory fields in the request](#)

[2.4:44] [Validate mandatory fields in the request](#)

[3.1:55] [Validate errors for invalid request content](#)

[3.2:55] [Validate errors for invalid request content](#)

[3.3:55] [Validate errors for invalid request content](#)

[3.4:55] [Validate errors for invalid request content](#)

[3.5:55] [Validate errors for invalid request content](#)

[3.6:55] [Validate errors for invalid request content](#)

[3.7:55] [Validate errors for invalid request content](#)

[3.8:55] [Validate errors for invalid request content](#)

[3.9:55] [Validate errors for invalid request content](#)

[3.10:55] [Validate errors for invalid request content](#)

[3.11:55] [Validate errors for invalid request content](#)

[3.12:55] [Validate errors for invalid request content](#)

1 < 400	
1 < X-Powered-By: Express	
1 < Content-Type: application/json; charset=utf-8	
1 < Content-Length: 219	
1 < ETag: W/"db-VVy2zHmDTwFS0Lx4GtJxeG0Wn7Q"	
1 < Date: Thu, 12 Jun 2025 18:54:55 GMT	
1 < Connection: keep-alive	
1 < Keep-Alive: timeout=5	
{ "errors": [{ "location": { "in": "request", "name": "body", "docPath": "/paths/~1booking/post/requestBody/content/application~1json/schema", "path": "/date" }, "message": "must match format \\\"date\\\""}], "message": "Validation errors" }	
51 Then status 400	0
52 * print response	1
13:54:55.436 [print] { "errors": [{ "location": { "in": "request", "name": "body", "docPath": "/paths/~1booking/post/requestBody/content/application~1json/schema", "path": "/date" }, "message": "must match format \\\"date\\\"" },], "message": "Validation errors" }	
53 And match response == structureError400	1

<div>reqContValNB bookingOperations Operations</div> <div>Scenario: [3.4:55] Validate errors for invalid request content</div>		ms: 12
>> Background:		
5 * def dataRequest = read('classpath:data/bookingOperations/requestCreateNewBooking.json')		1
13:54:55.442 karate.env system property was: null		
6 * def structureError400 = read('classpath:data/structureError.json')		1
7 * def structureResponse = read('classpath:data/bookingOperations/structureResponse200.json')		1
8 * def dateValidator = read('classpath:helpers/dateValidator.js')		1
9 Given url apiUrl		0
10 Given path 'booking'		0
11 Given headers { accept: 'application/json', Content-Type: 'application/json' }		0
48 * dataRequest.userId = 1		0
49 And request dataRequest		0
50 When method POST		7
13:54:55.447 request:		
1 > POST http://127.0.0.1:8900/booking		
1 > accept: application/json		
1 > Content-Type: application/json; charset=UTF-8		
1 > Content-Length: 59		
1 > Host: 127.0.0.1:8900		
1 > Connection: Keep-Alive		
1 > User-Agent: Apache-HttpClient/4.5.14 (Java/22.0.1)		
1 > Accept-Encoding: gzip, deflate		
{ "date": "!@", "destination": "BOG", "origin": "MAD", "userId": 1 }		
13:54:55.453 response time in milliseconds: 6		
1 < 400		
1 < X-Powered-By: Express		
1 < Content-Type: application/json; charset=utf-8		
1 < Content-Length: 219		
1 < ETag: W/"db-VVy2zHmDTwFS0Lx4GtJxeG0Wn7Q"		
1 < Date: Thu, 12 Jun 2025 18:54:55 GMT		
1 < Connection: keep-alive		
1 < Keep-Alive: timeout=5		
{ "errors": [{ "location": { "in": "request", "name": "body", "docPath": "/paths/~1booking/post/requestBody/content/application~1json/schema", "path": "/date" }, "message": "must match format \\\"date\\\""}], "message": "Validation errors" }		



44

3

Scenarios

2025-06-12 01:54:56 p. m.

[\[1.1:28\] Validate new booking creation](#)[\[2.1:41\] Validate mandatory fields in the request](#)[\[2.2:42\] Validate mandatory fields in the request](#)[\[2.3:43\] Validate mandatory fields in the request](#)[\[2.4:44\] Validate mandatory fields in the request](#)[\[3.1:55\] Validate errors for invalid request content](#)[\[3.2:55\] Validate errors for invalid request content](#)[\[3.3:55\] Validate errors for invalid request content](#)[\[3.4:55\] Validate errors for invalid request content](#)[\[3.5:55\] Validate errors for invalid request content](#)[\[3.6:55\] Validate errors for invalid request content](#)[\[3.7:55\] Validate errors for invalid request content](#)[\[3.8:55\] Validate errors for invalid request content](#)[\[3.9:55\] Validate errors for invalid request content](#)[\[3.10:55\] Validate errors for invalid request content](#)[\[3.11:55\] Validate errors for invalid request content](#)[\[3.12:55\] Validate errors for invalid request content](#)

51 Then status 400	0
52 * print response	1
13:54:55.455 [print] { "errors": [{ "location": { "in": "request", "name": "body", "docPath": "/paths/~1booking/post/requestBody/content/application~1json/schema", "path": "/date" }, "message": "must match format \"date\"" },], "message": "Validation errors" }	
53 And match response == structureError400	0

reqContValNB bookingOperations Operations	ms: 12
Scenario: [3.5:55] Validate errors for invalid request content	
>> Background:	
5 * def dataRequest = read('classpath:data/bookingOperations/requestCreateNewBooking.json')	1
6 * def structureError400 = read('classpath:data/structureError.json')	1
7 * def structureResponse = read('classpath:data/bookingOperations/structureResponse200.json')	1
8 * def dateValidator = read('classpath:helpers/dateValidator.js')	1
9 Given url apiUri	0
10 Given path 'booking'	0
11 Given headers { accept: 'application/json', Content-Type: 'application/json' }	0
48 * dataRequest.userId = 1	0
49 And request dataRequest	0
50 When method POST	8
13:54:55.465 request: 1 > POST http://127.0.0.1:8900/booking 1 > accept: application/json 1 > Content-Type: application/json; charset=UTF-8 1 > Content-Length: 67 1 > Host: 127.0.0.1:8900 1 > Connection: Keep-Alive 1 > User-Agent: Apache-HttpClient/4.5.14 (Java/22.0.1) 1 > Accept-Encoding: gzip,deflate { "date": "2028-30-10", "destination": "BOG", "origin": "MAD", "userId": 1 }	
13:54:55.470 response time in milliseconds: 5 1 < 400 1 < X-Powered-By: Express 1 < Content-Type: application/json; charset=utf-8 1 < Content-Length: 219 1 < ETag: W/"db-VVy2zHmDTwFS0Lx4GtJxeG0Wn7Q" 1 < Date: Thu, 12 Jun 2025 18:54:55 GMT 1 < Connection: keep-alive 1 < Keep-Alive: timeout=5 { "errors": [{ "location": { "in": "request", "name": "body", "docPath": "/paths/~1booking/post/requestBody/content/application~1json/schema", "path": "/date" }, "message": "must match format \"date\"" },], "message": "Validation errors" }	
51 Then status 400	0
52 * print response	1
53 And match response == structureError400	0

reqContValNB bookingOperations Operations	ms: 12
Scenario: [3.6:55] Validate errors for invalid request content	
>> Background:	
5 * def dataRequest = read('classpath:data/bookingOperations/	1



44
3

Scenarios

2025-06-12 01:54:56 p. m.

[\[1.1:28\] Validate new booking creation](#)

[\[2.1:41\] Validate mandatory fields in the request](#)

[\[2.2:42\] Validate mandatory fields in the request](#)

[\[2.3:43\] Validate mandatory fields in the request](#)

[\[2.4:44\] Validate mandatory fields in the request](#)

[\[3.1:55\] Validate errors for invalid request content](#)

[\[3.2:55\] Validate errors for invalid request content](#)

[\[3.3:55\] Validate errors for invalid request content](#)

[\[3.4:55\] Validate errors for invalid request content](#)

[\[3.5:55\] Validate errors for invalid request content](#)

[\[3.6:55\] Validate errors for invalid request content](#)

[\[3.7:55\] Validate errors for invalid request content](#)

[\[3.8:55\] Validate errors for invalid request content](#)

[\[3.9:55\] Validate errors for invalid request content](#)

[\[3.10:55\] Validate errors for invalid request content](#)

[\[3.11:55\] Validate errors for invalid request content](#)

[\[3.12:55\] Validate errors for invalid request content](#)

requestCreateNewBooking.json'	
6 * def structureError400 = read('classpath:data/structureError.json')	0
7 * def structureResponse = read('classpath:data/bookingOperations/structureResponse200.json')	1
8 * def dateValidator = read('classpath:helpers/dateValidator.js')	0
9 Given url apiUrl	0
10 Given path 'booking'	0
11 Given headers { accept: 'application/json', Content-Type: 'application/json' }	0
48 * dataRequest.userId = 1	0
49 And request dataRequest	0
50 When method POST	8
13:54:55.482 request:	
1 > POST http://127.0.0.1:8900/booking	
1 > accept: application/json	
1 > Content-Type: application/json; charset=UTF-8	
1 > Content-Length: 67	
1 > Host: 127.0.0.1:8900	
1 > Connection: Keep-Alive	
1 > User-Agent: Apache-HttpClient/4.5.14 (Java/22.0.1)	
1 > Accept-Encoding: gzip,deflate	
{ "date": "2028-02-50", "destination": "BOG", "origin": "MAD", "userId": 1 }	
13:54:55.488 response time in milliseconds: 6	
1 < 400	
1 < X-Powered-By: Express	
1 < Content-Type: application/json; charset=utf-8	
1 < Content-Length: 219	
1 < ETag: W/"db-VVy2zHmDTwFS0Lx4GtJxeG0Wn7Q"	
1 < Date: Thu, 12 Jun 2025 18:54:55 GMT	
1 < Connection: keep-alive	
1 < Keep-Alive: timeout=5	
{ "errors": [{ "location": { "in": "request", "name": "body", "docPath": "/paths/~1booking/post/requestBody/content/application~1json/schema", "path": "/date" }, "message": "must match format \\\"date\\\""}], "message": "Validation errors" }	
51 Then status 400	0
52 * print response	1
53 And match response == structureError400	0

reqContValNB bookingOperations Operations	ms: 11
Scenario: [3.7:55] Validate errors for invalid request content	
>> Background:	
5 * def dataRequest = read('classpath:data/bookingOperations/requestCreateNewBooking.json')	1
6 * def structureError400 = read('classpath:data/structureError.json')	1
7 * def structureResponse = read('classpath:data/bookingOperations/structureResponse200.json')	1
8 * def dateValidator = read('classpath:helpers/dateValidator.js')	0
9 Given url apiUrl	0
10 Given path 'booking'	0
11 Given headers { accept: 'application/json', Content-Type: 'application/json' }	0
48 * dataRequest.userId = 1	0
49 And request dataRequest	0
50 When method POST	7
13:54:55.499 request:	
1 > POST http://127.0.0.1:8900/booking	
1 > accept: application/json	
1 > Content-Type: application/json; charset=UTF-8	
1 > Content-Length: 68	
1 > Host: 127.0.0.1:8900	
1 > Connection: Keep-Alive	
1 > User-Agent: Apache-HttpClient/4.5.14 (Java/22.0.1)	
1 > Accept-Encoding: gzip,deflate	
{ "date": "2024-06-10", "destination": "BOGO", "origin": "MAD", "userId": 1 }	
13:54:55.504 response time in milliseconds: 5	



44
3

Scenarios

2025-06-12 01:54:56 p. m.

[\[1.1:28\] Validate new booking creation](#)

[\[2.1:41\] Validate mandatory fields in the request](#)

[\[2.2:42\] Validate mandatory fields in the request](#)

[\[2.3:43\] Validate mandatory fields in the request](#)

[\[2.4:44\] Validate mandatory fields in the request](#)

[\[3.1:55\] Validate errors for invalid request content](#)

[\[3.2:55\] Validate errors for invalid request content](#)

[\[3.3:55\] Validate errors for invalid request content](#)

[\[3.4:55\] Validate errors for invalid request content](#)

[\[3.5:55\] Validate errors for invalid request content](#)

[\[3.6:55\] Validate errors for invalid request content](#)

[\[3.7:55\] Validate errors for invalid request content](#)

[\[3.8:55\] Validate errors for invalid request content](#)

[\[3.9:55\] Validate errors for invalid request content](#)

[\[3.10:55\] Validate errors for invalid request content](#)

[\[3.11:55\] Validate errors for invalid request content](#)

[\[3.12:55\] Validate errors for invalid request content](#)

```
1 < 400
1 < X-Powered-By: Express
1 < Content-Type: application/json; charset=utf-8
1 < Content-Length: 233
1 < ETag: W/"e9-udqyy/B30CLkylZod5/y/UsB0k4"
1 < Date: Thu, 12 Jun 2025 18:54:55 GMT
1 < Connection: keep-alive
1 < Keep-Alive: timeout=5
{"errors":[{"location":{"in":"request","name":"body","docPath":"/paths/~1booking/post/requestBody/content/application~1json/schema","path":"/destination"},"message":"must match pattern \"^[A-Z]{3}$\""}],"message":"Validation errors"}

51 Then status 400
52 * print response
13:54:55.505 [print] {
  "errors": [
    {
      "location": {
        "in": "request",
        "name": "body",
        "docPath": "/paths/~1booking/post/requestBody/content/application~1json/schema",
        "path": "/destination"
      },
      "message": "must match pattern \"^[A-Z]{3}$\""
    }
  ],
  "message": "Validation errors"
}

53 And match response == structureError400
```

0
1

0

reqContValNB	bookingOperations	Operations	ms: 12
Scenario: [3.8:55] Validate errors for invalid request content			
>> Background:			
5	* def dataRequest = read('classpath:data/bookingOperations/requestCreateNewBooking.json')		1
6	* def structureError400 = read('classpath:data/structureError.json')		1
7	* def structureResponse = read('classpath:data/bookingOperations/structureResponse200.json')		1
8	* def dateValidator = read('classpath:helpers/dateValidator.js')		0
9	Given url apiUrl		0
10	Given path 'booking'		0
11	Given headers { accept: 'application/json', Content-Type: 'application/json' }		0
48	* dataRequest.userId = 1		0
49	And request dataRequest		0
50	When method POST		8
13:54:55.514 request:			
1 > POST http://127.0.0.1:8900/booking			
1 > accept: application/json			
1 > Content-Type: application/json; charset=UTF-8			
1 > Content-Length: 66			
1 > Host: 127.0.0.1:8900			
1 > Connection: Keep-Alive			
1 > User-Agent: Apache-HttpClient/4.5.14 (Java/22.0.1)			
1 > Accept-Encoding: gzip,deflate			
{ "date": "2024-06-10", "destination": "BO", "origin": "MAD", "userId": 1 }			
13:54:55.520 response time in milliseconds: 5			
1 < 400			
1 < X-Powered-By: Express			
1 < Content-Type: application/json; charset=utf-8			
1 < Content-Length: 233			
1 < ETag: W/"e9-udqyy/B30CLkylZod5/y/UsB0k4"			
1 < Date: Thu, 12 Jun 2025 18:54:55 GMT			
1 < Connection: keep-alive			
1 < Keep-Alive: timeout=5			
{ "errors": [{"location": {"in": "request", "name": "body", "docPath": "/paths/~1booking/post/requestBody/content/application~1json/schema", "path": "/destination"}, "message": "must match pattern \"^[A-Z]{3}\$\""}], "message": "Validation errors" }			



44

3

Scenarios

2025-06-12 01:54:56 p. m.

[\[1.1:28\] Validate new booking creation](#)[\[2.1:41\] Validate mandatory fields in the request](#)[\[2.2:42\] Validate mandatory fields in the request](#)[\[2.3:43\] Validate mandatory fields in the request](#)[\[2.4:44\] Validate mandatory fields in the request](#)[\[3.1:55\] Validate errors for invalid request content](#)[\[3.2:55\] Validate errors for invalid request content](#)[\[3.3:55\] Validate errors for invalid request content](#)[\[3.4:55\] Validate errors for invalid request content](#)[\[3.5:55\] Validate errors for invalid request content](#)[\[3.6:55\] Validate errors for invalid request content](#)[\[3.7:55\] Validate errors for invalid request content](#)[\[3.8:55\] Validate errors for invalid request content](#)[\[3.9:55\] Validate errors for invalid request content](#)[\[3.10:55\] Validate errors for invalid request content](#)[\[3.11:55\] Validate errors for invalid request content](#)[\[3.12:55\] Validate errors for invalid request content](#)

51 Then status 400	0
52 * print response	1
13:54:55.522 [print] { "errors": [{ "location": { "in": "request", "name": "body", "docPath": "/paths/~1booking/post/requestBody/content/application~1json/schema", "path": "/destination" }, "message": "must match pattern \"^[A-Z]{3}\$\"" }], "message": "Validation errors" }	
53 And match response == structureError400	0

reqContValNB bookingOperations Operations	ms: 11
Scenario: [3.9:55] Validate errors for invalid request content	
>> Background:	
48 * dataRequest.userId = 1	0
49 And request dataRequest	0
50 When method POST	7
13:54:55.531 request: 1 > POST http://127.0.0.1:8900/booking 1 > accept: application/json 1 > Content-Type: application/json; charset=UTF-8 1 > Content-Length: 72 1 > Host: 127.0.0.1:8900 1 > Connection: Keep-Alive 1 > User-Agent: Apache-HttpClient/4.5.14 (Java/22.0.1) 1 > Accept-Encoding: gzip,deflate { "date": "2024-06-10", "destination": "Alicante", "origin": "MAD", "userId": 1 }	
13:54:55.536 response time in milliseconds: 5 1 < 400 1 < X-Powered-By: Express 1 < Content-Type: application/json; charset=utf-8 1 < Content-Length: 233 1 < ETag: W/"e9-udqyy/B30CLkylZod5/y/UsB0k4" 1 < Date: Thu, 12 Jun 2025 18:54:55 GMT 1 < Connection: keep-alive 1 < Keep-Alive: timeout=5 { "errors": [{ "location": { "in": "request", "name": "body", "docPath": "/paths/~1booking/post/requestBody/content/application~1json/schema", "path": "/destination" }, "message": "must match pattern \"^[A-Z]{3}\$\" \"\" }], "message": "Validation errors" }	
51 Then status 400	0
52 * print response	1
13:54:55.537 [print] { "errors": [{ "location": { "in": "request", "name": "body", "docPath": "/paths/~1booking/post/requestBody/content/application~1json/schema", "path": "/destination" }, "message": "must match pattern \"^[A-Z]{3}\$\"" }], "message": "Validation errors" }	
53 And match response == structureError400	0

reqContValNB bookingOperations Operations	ms: 11
Scenario: [3.10:55] Validate errors for invalid request content	



44

3

Scenarios

2025-06-12 01:54:56 p. m.

[\[1.1:28\] Validate new booking creation](#)[\[2.1:41\] Validate mandatory fields in the request](#)[\[2.2:42\] Validate mandatory fields in the request](#)[\[2.3:43\] Validate mandatory fields in the request](#)[\[2.4:44\] Validate mandatory fields in the request](#)[\[3.1:55\] Validate errors for invalid request content](#)[\[3.2:55\] Validate errors for invalid request content](#)[\[3.3:55\] Validate errors for invalid request content](#)[\[3.4:55\] Validate errors for invalid request content](#)[\[3.5:55\] Validate errors for invalid request content](#)[\[3.6:55\] Validate errors for invalid request content](#)[\[3.7:55\] Validate errors for invalid request content](#)[\[3.8:55\] Validate errors for invalid request content](#)[\[3.9:55\] Validate errors for invalid request content](#)[\[3.10:55\] Validate errors for invalid request content](#)[\[3.11:55\] Validate errors for invalid request content](#)[\[3.12:55\] Validate errors for invalid request content](#)

>> Background:

```
5 * def dataRequest = read('classpath:data/bookingOperations/
  requestCreateNewBooking.json')
6 * def structureError400 = read('classpath:data/structureError.json')
7 * def structureResponse = read('classpath:data/bookingOperations/
  structureResponse200.json')
8 * def dateValidator = read('classpath:helpers/dateValidator.js')
9 Given url apiUri
10 Given path 'booking'
11 Given headers { accept: 'application/json', Content-Type: 'application/json' }
48 * dataRequest.userId = 1
49 And request dataRequest
50 When method POST
```

```
13:54:55.547 request:
1 > POST http://127.0.0.1:8900/booking
1 > accept: application/json
1 > Content-Type: application/json; charset=UTF-8
1 > Content-Length: 72
1 > Host: 127.0.0.1:8900
1 > Connection: Keep-Alive
1 > User-Agent: Apache-HttpClient/4.5.14 (Java/22.0.1)
1 > Accept-Encoding: gzip,deflate
{"date":"2024-06-10","destination":"ALICANTE","origin":"MAD","userId":1}

13:54:55.552 response time in milliseconds: 5
1 < 400
1 < X-Powered-By: Express
1 < Content-Type: application/json; charset=utf-8
1 < Content-Length: 233
1 < ETag: W/"e9-udqyy/B30CLkylZod5/y/UsB0k4"
1 < Date: Thu, 12 Jun 2025 18:54:55 GMT
1 < Connection: keep-alive
1 < Keep-Alive: timeout=5
{"errors":[{"location":{"in":"request","name":"body","docPath":"/paths/~1booking/post/requestBody/content/application~1json/schema","path":"/destination"},"message":"must match pattern \"^[A-Z]{3}$\""}],\"message\":\"Validation errors\"}

51 Then status 400
52 * print response
13:54:55.553 [print] {
  "errors": [
    {
      "location": {
        "in": "request",
        "name": "body",
        "docPath": "/paths/~1booking/post/requestBody/content/application~1json/schema",
        "path": "/destination"
      },
      "message": "must match pattern \"^[A-Z]{3}$\""
    }
  ],
  "message": "Validation errors"
}

53 And match response == structureError400
```

reqContValNB bookingOperations Operations

ms: 11

Scenario: [3.11:55] Validate errors for invalid request content

>> Background:

```
48 * dataRequest.userId = 1
49 And request dataRequest
50 When method POST
```

```
13:54:55.562 request:
1 > POST http://127.0.0.1:8900/booking
1 > accept: application/json
1 > Content-Type: application/json; charset=UTF-8
1 > Content-Length: 67
1 > Host: 127.0.0.1:8900
```




44
3

Scenarios

2025-06-12 01:54:56 p. m.

[1.1:28] [Validate new booking creation](#)

[2.1:41] [Validate mandatory fields in the request](#)

[2.2:42] [Validate mandatory fields in the request](#)

[2.3:43] [Validate mandatory fields in the request](#)

[2.4:44] [Validate mandatory fields in the request](#)

[3.1:55] [Validate errors for invalid request content](#)

[3.2:55] [Validate errors for invalid request content](#)

[3.3:55] [Validate errors for invalid request content](#)

[3.4:55] [Validate errors for invalid request content](#)

[3.5:55] [Validate errors for invalid request content](#)

[3.6:55] [Validate errors for invalid request content](#)

[3.7:55] [Validate errors for invalid request content](#)

[3.8:55] [Validate errors for invalid request content](#)

[3.9:55] [Validate errors for invalid request content](#)

[3.10:55] [Validate errors for invalid request content](#)

[3.11:55] [Validate errors for invalid request content](#)

[3.12:55] [Validate errors for invalid request content](#)

1 > Connection: Keep-Alive	
1 > User-Agent: Apache-HttpClient/4.5.14 (Java/22.0.1)	
1 > Accept-Encoding: gzip,deflate	
{ "date": "2024-06-10", "destination": "@@@", "origin": "MAD", "userId": 1 }	
13:54:55.567 response time in milliseconds: 5	
1 < 400	
1 < X-Powered-By: Express	
1 < Content-Type: application/json; charset=utf-8	
1 < Content-Length: 233	
1 < ETag: W/"e9-udqyy/B30CLkylZod5/y/UsB0k4"	
1 < Date: Thu, 12 Jun 2025 18:54:55 GMT	
1 < Connection: keep-alive	
1 < Keep-Alive: timeout=5	
{ "errors": [{ "location": { "in": "request", "name": "body", "docPath": "/paths/~1booking/post/requestBody/content/application~1json/schema", "path": "/destination" }, "message": "must match pattern \"^[A-Z]{3}\$\" \"\" }], "message": "Validation errors" }	
51 Then status 400	0
52 * print response	1
13:54:55.568 [print] {	
"errors": [
{	
"location": {	
"in": "request",	
"name": "body",	
"docPath": "/paths/~1booking/post/requestBody/content/application~1json/schema",	
"path": "/destination"	
},	
"message": "must match pattern \"^[A-Z]{3}\$\" \"\"	
}	
],	
"message": "Validation errors"	
}	
53 And match response == structureError400	0

reqContValNB	bookingOperations	Operations	ms: 14
Scenario: [3.12:55] Validate errors for invalid request content			
>> Background:			
48 * dataRequest.userId = 1			0
49 And request dataRequest			0
50 When method POST			9
13:54:55.577 request:			
1 > POST http://127.0.0.1:8900/booking			
1 > accept: application/json			
1 > Content-Type: application/json; charset=UTF-8			
1 > Content-Length: 67			
1 > Host: 127.0.0.1:8900			
1 > Connection: Keep-Alive			
1 > User-Agent: Apache-HttpClient/4.5.14 (Java/22.0.1)			
1 > Accept-Encoding: gzip,deflate			
{ "date": "2024-06-10", "destination": "123", "origin": "MAD", "userId": 1 }			
13:54:55.585 response time in milliseconds: 7			
1 < 400			
1 < X-Powered-By: Express			
1 < Content-Type: application/json; charset=utf-8			
1 < Content-Length: 233			
1 < ETag: W/"e9-udqyy/B30CLkylZod5/y/UsB0k4"			
1 < Date: Thu, 12 Jun 2025 18:54:55 GMT			
1 < Connection: keep-alive			
1 < Keep-Alive: timeout=5			
{ "errors": [{ "location": { "in": "request", "name": "body", "docPath": "/paths/~1booking/post/requestBody/content/application~1json/schema", "path": "/destination" }, "message": "must match pattern \"^[A-Z]{3}\$\" \"\" }], "message": "Validation errors" }			
51 Then status 400			0
52 * print response			1
13:54:55.586 [print] {			
"errors": [
{			



2025-06-12 01:54:56 p. m.

[3.12:55] Validate errors for invalid request content

```
53 And match response == structureError400
```

0

ms: 13

0

0

8

0

1

```
53 And match response == structureError400
```

0

ms: 11

0

0

7

```
13:54:55.613 request:
```



44
3

Scenarios

2025-06-12 01:54:56 p. m.

[1.1:28] [Validate new booking creation](#)

[2.1:41] [Validate mandatory fields in the request](#)

[2.2:42] [Validate mandatory fields in the request](#)

[2.3:43] [Validate mandatory fields in the request](#)

[2.4:44] [Validate mandatory fields in the request](#)

[3.1:55] [Validate errors for invalid request content](#)

[3.2:55] [Validate errors for invalid request content](#)

[3.3:55] [Validate errors for invalid request content](#)

[3.4:55] [Validate errors for invalid request content](#)

[3.5:55] [Validate errors for invalid request content](#)

[3.6:55] [Validate errors for invalid request content](#)

[3.7:55] [Validate errors for invalid request content](#)

[3.8:55] [Validate errors for invalid request content](#)

[3.9:55] [Validate errors for invalid request content](#)

[3.10:55] [Validate errors for invalid request content](#)

[3.11:55] [Validate errors for invalid request content](#)

[3.12:55] [Validate errors for invalid request content](#)

1 > POST http://127.0.0.1:8900/booking	
1 > accept: application/json	
1 > Content-Type: application/json; charset=UTF-8	
1 > Content-Length: 72	
1 > Host: 127.0.0.1:8900	
1 > Connection: Keep-Alive	
1 > User-Agent: Apache-HttpClient/4.5.14 (Java/22.0.1)	
1 > Accept-Encoding: gzip,deflate	
{ "date": "2024-06-10", "destination": "BOG", "origin": "Alicante", "userId": 1 }	
13:54:55.618 response time in milliseconds: 4	
1 < 400	
1 < X-Powered-By: Express	
1 < Content-Type: application/json; charset=utf-8	
1 < Content-Length: 228	
1 < ETag: W/"e4-XseQcs19QbsgA1YDs+6sPoiX38M"	
1 < Date: Thu, 12 Jun 2025 18:54:55 GMT	
1 < Connection: keep-alive	
1 < Keep-Alive: timeout=5	
{ "errors": [{ "location": { "in": "request", "name": "body", "docPath": "/paths/~1booking/post/requestBody/content/application~1json/schema", "path": "/origin" }, "message": "must match pattern \"^[A-Z]{3}\$\" \"\" }], "message": "Validation errors" }	
51 Then status 400	0
52 * print response	1
13:54:55.619 [print] {	
"errors": [
{	
"location": {	
"in": "request",	
"name": "body",	
"docPath": "/paths/~1booking/post/requestBody/content/application~1json/schema",	
"path": "/origin"	
},	
"message": "must match pattern \"^[A-Z]{3}\$\" \"\"	
},	
"message": "Validation errors"	
]	
}	
53 And match response == structureError400	0

reqContValNB	bookingOperations	Operations	ms: 13
Scenario: [3.15:55] Validate errors for invalid request content			
>> Background:			
48 * dataRequest.userId = 1			0
49 And request dataRequest			0
50 When method POST			8
13:54:55.630 request:			
1 > POST http://127.0.0.1:8900/booking			
1 > accept: application/json			
1 > Content-Type: application/json; charset=UTF-8			
1 > Content-Length: 72			
1 > Host: 127.0.0.1:8900			
1 > Connection: Keep-Alive			
1 > User-Agent: Apache-HttpClient/4.5.14 (Java/22.0.1)			
1 > Accept-Encoding: gzip,deflate			
{ "date": "2024-06-10", "destination": "BOG", "origin": "ALICANTE", "userId": 1 }			
13:54:55.636 response time in milliseconds: 5			
1 < 400			
1 < X-Powered-By: Express			
1 < Content-Type: application/json; charset=utf-8			
1 < Content-Length: 228			
1 < ETag: W/"e4-XseQcs19QbsgA1YDs+6sPoiX38M"			
1 < Date: Thu, 12 Jun 2025 18:54:55 GMT			
1 < Connection: keep-alive			
1 < Keep-Alive: timeout=5			
{ "errors": [{ "location": { "in": "request", "name": "body", "docPath": "/paths/~1booking/post/requestBody/content/application~1json/schema", "path": "/origin" }, "message": "must match pattern \"^[A-Z]{3}\$\" \"\" }], "message": "Validation errors" }			



44

3

Scenarios

2025-06-12 01:54:56 p. m.

[\[1.1:28\] Validate new booking creation](#)[\[2.1:41\] Validate mandatory fields in the request](#)[\[2.2:42\] Validate mandatory fields in the request](#)[\[2.3:43\] Validate mandatory fields in the request](#)[\[2.4:44\] Validate mandatory fields in the request](#)[\[3.1:55\] Validate errors for invalid request content](#)[\[3.2:55\] Validate errors for invalid request content](#)[\[3.3:55\] Validate errors for invalid request content](#)[\[3.4:55\] Validate errors for invalid request content](#)[\[3.5:55\] Validate errors for invalid request content](#)[\[3.6:55\] Validate errors for invalid request content](#)[\[3.7:55\] Validate errors for invalid request content](#)[\[3.8:55\] Validate errors for invalid request content](#)[\[3.9:55\] Validate errors for invalid request content](#)[\[3.10:55\] Validate errors for invalid request content](#)[\[3.11:55\] Validate errors for invalid request content](#)[\[3.12:55\] Validate errors for invalid request content](#)

51 Then status 400	0
52 * print response	1
13:54:55.638 [print] { "errors": [{ "location": { "in": "request", "name": "body", "docPath": "/paths/~1booking/post/requestBody/content/application~1json/schema", "path": "/origin" }, "message": "must match pattern \"^[A-Z]{3}\$\"" }, { "location": { "in": "request", "name": "body", "docPath": "/paths/~1booking/post/requestBody/content/application~1json/schema", "path": "/origin" }, "message": "must match pattern \"^[A-Z]{3}\$\"" }], "message": "Validation errors" }	
53 And match response == structureError400	0

reqContValNB bookingOperations Operations	ms: 11
Scenario: [3.16:55] Validate errors for invalid request content	
>> Background:	
5 * def dataRequest = read('classpath:data/bookingOperations/requestCreateNewBooking.json')	1
6 * def structureError400 = read('classpath:data/structureError.json')	0
7 * def structureResponse = read('classpath:data/bookingOperations/structureResponse200.json')	0
8 * def dateValidator = read('classpath:helpers/dateValidator.js')	1
9 Given url apiUrl	0
10 Given path 'booking'	0
11 Given headers { accept: 'application/json', Content-Type: 'application/json' }	0
48 * dataRequest.userId = 1	0
49 And request dataRequest	0
50 When method POST	7
13:54:55.647 request: 1 > POST http://127.0.0.1:8900/booking 1 > accept: application/json 1 > Content-Type: application/json; charset=UTF-8 1 > Content-Length: 66 1 > Host: 127.0.0.1:8900 1 > Connection: Keep-Alive 1 > User-Agent: Apache-HttpClient/4.5.14 (Java/22.0.1) 1 > Accept-Encoding: gzip,deflate { "date": "2024-06-10", "destination": "BOG", "origin": "AS", "userId": 1 }	
13:54:55.652 response time in milliseconds: 5 1 < 400 1 < X-Powered-By: Express 1 < Content-Type: application/json; charset=utf-8 1 < Content-Length: 228 1 < ETag: W/"e4-XseQcsl9QbSgA1YDs+6sPoiX38M" 1 < Date: Thu, 12 Jun 2025 18:54:55 GMT 1 < Connection: keep-alive 1 < Keep-Alive: timeout=5 { "errors": [{ "location": { "in": "request", "name": "body", "docPath": "/paths/~1booking/post/requestBody/content/application~1json/schema", "path": "/origin" }, "message": "must match pattern \"^[A-Z]{3}\$\""}, { "location": { "in": "request", "name": "body", "docPath": "/paths/~1booking/post/requestBody/content/application~1json/schema", "path": "/origin" }, "message": "must match pattern \"^[A-Z]{3}\$\""}], "message": "Validation errors" }	
51 Then status 400	0
52 * print response	1
13:54:55.653 [print] { "errors": [{ "location": { "in": "request", "name": "body", "docPath": "/paths/~1booking/post/requestBody/content/application~1json/schema", "path": "/origin" }, "message": "must match pattern \"^[A-Z]{3}\$\"" }, { "location": { "in": "request", "name": "body", "docPath": "/paths/~1booking/post/requestBody/content/application~1json/schema", "path": "/origin" }, "message": "must match pattern \"^[A-Z]{3}\$\"" }], "message": "Validation errors" }	



44

3

Scenarios

2025-06-12 01:54:56 p. m.

[\[1.1:28\] Validate new booking creation](#)[\[2.1:41\] Validate mandatory fields in the request](#)[\[2.2:42\] Validate mandatory fields in the request](#)[\[2.3:43\] Validate mandatory fields in the request](#)[\[2.4:44\] Validate mandatory fields in the request](#)[\[3.1:55\] Validate errors for invalid request content](#)[\[3.2:55\] Validate errors for invalid request content](#)[\[3.3:55\] Validate errors for invalid request content](#)[\[3.4:55\] Validate errors for invalid request content](#)[\[3.5:55\] Validate errors for invalid request content](#)[\[3.6:55\] Validate errors for invalid request content](#)[\[3.7:55\] Validate errors for invalid request content](#)[\[3.8:55\] Validate errors for invalid request content](#)[\[3.9:55\] Validate errors for invalid request content](#)[\[3.10:55\] Validate errors for invalid request content](#)[\[3.11:55\] Validate errors for invalid request content](#)[\[3.12:55\] Validate errors for invalid request content](#)

>> Background:

```
5 * def dataRequest = read('classpath:data/bookingOperations/
  requestCreateNewBooking.json')
6 * def structureError400 = read('classpath:data/structureError.json')
7 * def structureResponse = read('classpath:data/bookingOperations/
  structureResponse200.json')
8 * def dateValidator = read('classpath:helpers/dateValidator.js')
9 Given url apiUri
10 Given path 'booking'
11 Given headers { accept: 'application/json', Content-Type: 'application/json' }
48 * dataRequest.userId = 1
49 And request dataRequest
50 When method POST
```

```
13:54:55.678 request:
1 > POST http://127.0.0.1:8900/booking
1 > accept: application/json
1 > Content-Type: application/json; charset=UTF-8
1 > Content-Length: 67
1 > Host: 127.0.0.1:8900
1 > Connection: Keep-Alive
1 > User-Agent: Apache-HttpClient/4.5.14 (Java/22.0.1)
1 > Accept-Encoding: gzip,deflate
{"date":"2024-06-10","destination":"BOG","origin":"@@@","userId":1}

13:54:55.682 response time in milliseconds: 3
1 < 400
1 < X-Powered-By: Express
1 < Content-Type: application/json; charset=utf-8
1 < Content-Length: 228
1 < ETag: W/"e4-XseQcsl9QbsgA1YDs+6sPoiX38M"
1 < Date: Thu, 12 Jun 2025 18:54:55 GMT
1 < Connection: keep-alive
1 < Keep-Alive: timeout=5
{"errors":[{"location":{"in":"request","name":"body","docPath":"/paths/~1booking/post/requestBody/content/application~1json/schema","path":"/origin"},"message":"must match pattern \"^[A-Z]{3}$\""}],\"message\":\"Validation errors\"}
```

51 Then status 400

52 * print response

```
13:54:55.683 [print] {
  "errors": [
    {
      "location": {
        "in": "request",
        "name": "body",
        "docPath": "/paths/~1booking/post/requestBody/content/application~1json/schema",
        "path": "/origin"
      },
      "message": "must match pattern \"^[A-Z]{3}$\""
    }
  ],
  "message": "Validation errors"
}
```

53 And match response == structureError400

reqContValNB bookingOperations Operations

ms: 12

Scenario: [3.19:55] Validate errors for invalid request content

>> Background:

```
5 * def dataRequest = read('classpath:data/bookingOperations/
  requestCreateNewBooking.json')
6 * def structureError400 = read('classpath:data/structureError.json')
7 * def structureResponse = read('classpath:data/bookingOperations/
  structureResponse200.json')
8 * def dateValidator = read('classpath:helpers/dateValidator.js')
9 Given url apiUri
10 Given path 'booking'
```



44
3

Scenarios

2025-06-12 01:54:56 p. m.

[1.1:28] [Validate new booking creation](#)

[2.1:41] [Validate mandatory fields in the request](#)

[2.2:42] [Validate mandatory fields in the request](#)

[2.3:43] [Validate mandatory fields in the request](#)

[2.4:44] [Validate mandatory fields in the request](#)

[3.1:55] [Validate errors for invalid request content](#)

[3.2:55] [Validate errors for invalid request content](#)

[3.3:55] [Validate errors for invalid request content](#)

[3.4:55] [Validate errors for invalid request content](#)

[3.5:55] [Validate errors for invalid request content](#)

[3.6:55] [Validate errors for invalid request content](#)

[3.7:55] [Validate errors for invalid request content](#)

[3.8:55] [Validate errors for invalid request content](#)

[3.9:55] [Validate errors for invalid request content](#)

[3.10:55] [Validate errors for invalid request content](#)

[3.11:55] [Validate errors for invalid request content](#)

[3.12:55] [Validate errors for invalid request content](#)

11	Given headers { accept: 'application/json', Content-Type: 'application/json' }	0
48	* dataRequest.userId = 1	0
49	And request dataRequest	0
50	When method POST	7
13:54:55.694 request:		
1	> POST http://127.0.0.1:8900/booking	
1	> accept: application/json	
1	> Content-Type: application/json; charset=UTF-8	
1	> Content-Length: 67	
1	> Host: 127.0.0.1:8900	
1	> Connection: Keep-Alive	
1	> User-Agent: Apache-HttpClient/4.5.14 (Java/22.0.1)	
1	> Accept-Encoding: gzip,deflate	
	{ "date": "2024-06-10", "destination": "BOG", "origin": "123", "userId": 1 }	
13:54:55.699 response time in milliseconds: 5		
1	< 400	
1	< X-Powered-By: Express	
1	< Content-Type: application/json; charset=utf-8	
1	< Content-Length: 228	
1	< ETag: W/"e4-XseQcsl9QbsgA1YDs+6sPoiX38M"	
1	< Date: Thu, 12 Jun 2025 18:54:55 GMT	
1	< Connection: keep-alive	
1	< Keep-Alive: timeout=5	
	{ "errors": [{ "location": { "in": "request", "name": "body", "docPath": "/paths/~1booking/post/requestBody/content/application~1json/schema", "path": "/origin" }, "message": "must match pattern \"^[A-Z]{3}\$\" \"\" }], "message": "Validation errors" }	
51	Then status 400	0
52	* print response	1
13:54:55.701 [print] {		
	"errors": [
	{	
	"location": {	
	"in": "request",	
	"name": "body",	
	"docPath": "/paths/~1booking/post/requestBody/content/application~1json/schema",	
	"path": "/origin"	
	},	
	"message": "must match pattern \"^[A-Z]{3}\$\" \"\""	
	}	
],	
	"message": "Validation errors"	
	}	
53	And match response == structureError400	0

<div><div>reqContValNB</div><div>bookingOperations</div><div>Operations</div></div>		ms: 14
Scenario: [3.20:55] Validate errors for invalid request content		
>> Background:		
5	* def dataRequest = read('classpath:data/bookingOperations/requestCreateNewBooking.json')	1
6	* def structureError400 = read('classpath:data/structureError.json')	1
7	* def structureResponse = read('classpath:data/bookingOperations/structureResponse200.json')	1
8	* def dateValidator = read('classpath:helpers/dateValidator.js')	1
9	Given url apiUri	0
10	Given path 'booking'	0
11	Given headers { accept: 'application/json', Content-Type: 'application/json' }	0
48	* dataRequest.userId = 1	0
49	And request dataRequest	0
50	When method POST	8
13:54:55.712 request:		
1	> POST http://127.0.0.1:8900/booking	
1	> accept: application/json	
1	> Content-Type: application/json; charset=UTF-8	
1	> Content-Length: 64	
1	> Host: 127.0.0.1:8900	



44
3

Scenarios

2025-06-12 01:54:56 p. m.

[1.1:28] [Validate new booking creation](#)

[2.1:41] [Validate mandatory fields in the request](#)

[2.2:42] [Validate mandatory fields in the request](#)

[2.3:43] [Validate mandatory fields in the request](#)

[2.4:44] [Validate mandatory fields in the request](#)

[3.1:55] [Validate errors for invalid request content](#)

[3.2:55] [Validate errors for invalid request content](#)

[3.3:55] [Validate errors for invalid request content](#)

[3.4:55] [Validate errors for invalid request content](#)

[3.5:55] [Validate errors for invalid request content](#)

[3.6:55] [Validate errors for invalid request content](#)

[3.7:55] [Validate errors for invalid request content](#)

[3.8:55] [Validate errors for invalid request content](#)

[3.9:55] [Validate errors for invalid request content](#)

[3.10:55] [Validate errors for invalid request content](#)

[3.11:55] [Validate errors for invalid request content](#)

[3.12:55] [Validate errors for invalid request content](#)

1 > Connection: Keep-Alive	
1 > User-Agent: Apache-HttpClient/4.5.14 (Java/22.0.1)	
1 > Accept-Encoding: gzip,deflate	
{ "date": "2024-06-10", "destination": "BOG", "origin": "", "userId": 1 }	
13:54:55.718 response time in milliseconds: 6	
1 < 400	
1 < X-Powered-By: Express	
1 < Content-Type: application/json; charset=utf-8	
1 < Content-Length: 228	
1 < ETag: W/"e4-XseQcs19QbsgA1YDs+6sPoiX38M"	
1 < Date: Thu, 12 Jun 2025 18:54:55 GMT	
1 < Connection: keep-alive	
1 < Keep-Alive: timeout=5	
{ "errors": [{ "location": { "in": "request", "name": "body", "docPath": "/paths/~1booking/post/requestBody/content/application~1json/schema", "path": "/origin", "message": "must match pattern \"^[A-Z]{3}\$\" \"\" }], "message": "Validation errors" }	
51 Then status 400	0
52 * print response	1
13:54:55.720 [print] {	
"errors": [
{	
"location": {	
"in": "request",	
"name": "body",	
"docPath": "/paths/~1booking/post/requestBody/content/application~1json/schema",	
"path": "/origin"	
},	
"message": "must match pattern \"^[A-Z]{3}\$\" \"\""	
}	
],	
"message": "Validation errors"	
}	
53 And match response == structureError400	1

reqContValNB	bookingOperations	Operations	ms: 19
Scenario: [3.21:55] Validate errors for invalid request content			
>> Background:			
5 * def dataRequest = read('classpath:data/bookingOperations/requestCreateNewBooking.json')			1
6 * def structureError400 = read('classpath:data/structureError.json')			1
7 * def structureResponse = read('classpath:data/bookingOperations/structureResponse200.json')			1
8 * def dateValidator = read('classpath:helpers/dateValidator.js')			1
9 Given url apiUri			0
10 Given path 'booking'			0
11 Given headers { accept: 'application/json', Content-Type: 'application/json' }			0
48 * dataRequest.userId = 1000			1
49 And request dataRequest			0
50 When method POST			9
13:54:55.733 request:			
1 > POST http://127.0.0.1:8900/booking			
1 > accept: application/json			
1 > Content-Type: application/json; charset=UTF-8			
1 > Content-Length: 70			
1 > Host: 127.0.0.1:8900			
1 > Connection: Keep-Alive			
1 > User-Agent: Apache-HttpClient/4.5.14 (Java/22.0.1)			
1 > Accept-Encoding: gzip,deflate			
{ "date": "2024-06-10", "destination": "BOG", "origin": "MAD", "userId": 1000 }			
13:54:55.740 response time in milliseconds: 7			
1 < 404			
1 < X-Powered-By: Express			
1 < Content-Type: application/json; charset=utf-8			
1 < Content-Length: 48			
1 < ETag: W/"30-T4jfxjWlIoewPgbWzt5b4BlzM7s"			
1 < Date: Thu, 12 Jun 2025 18:54:55 GMT			

44
3

Scenarios

2025-06-12 01:54:56 p. m.

[\[1.1:28\] Validate new booking creation](#)[\[2.1:41\] Validate mandatory fields in the request](#)[\[2.2:42\] Validate mandatory fields in the request](#)[\[2.3:43\] Validate mandatory fields in the request](#)[\[2.4:44\] Validate mandatory fields in the request](#)[\[3.1:55\] Validate errors for invalid request content](#)[\[3.2:55\] Validate errors for invalid request content](#)[\[3.3:55\] Validate errors for invalid request content](#)[\[3.4:55\] Validate errors for invalid request content](#)[\[3.5:55\] Validate errors for invalid request content](#)[\[3.6:55\] Validate errors for invalid request content](#)[\[3.7:55\] Validate errors for invalid request content](#)[\[3.8:55\] Validate errors for invalid request content](#)[\[3.9:55\] Validate errors for invalid request content](#)[\[3.10:55\] Validate errors for invalid request content](#)[\[3.11:55\] Validate errors for invalid request content](#)[\[3.12:55\] Validate errors for invalid request content](#)

```
1 < Connection: keep-alive
1 < Keep-Alive: timeout=5
{"errors":null,"message":"No user with id 1000"}
```

51 Then status 404

[52 * print response](#)

```
13:54:55.742 [print] {
  "errors": null,
  "message": "No user with id 1000"
}
```

[53 And match response == structureError400](#)

```
13:54:55.746 src/test/java/bookingOperations/createBooking.feature:53
And match response == structureError400
match failed: EQUALS
$ | not equal | match failed for name: 'errors' (MAP:MAP)
{"errors":null,"message":"No user with id 1000"}
{"errors":[{"message":"#string","location":
{"in":"#string","name":"#string","docPath":"#string","path":"#string"}]],"message":"#string"}

$.errors | data types don't match (NULL:LIST)
null
[{"message":"#string","location":
{"in":"#string","name":"#string","docPath":"#string","path":"#string"}]]

src/test/java/bookingOperations/createBooking.feature:53
```

[bookingOperations](#) [Operations](#) [getAllBookingOK](#)

ms: 56

Scenario: [4.1:67] **Validate getting all bookings with and without parameters**

>> Background:

[5 * def dataRequest = read\('classpath:data/bookingOperations/requestCreateNewBooking.json'\)](#)

6 * def structureError400 = read('classpath:data/structureError.json')

7 * def structureResponse = read('classpath:data/bookingOperations/structureResponse200.json')

8 * def dateValidator = read('classpath:helpers/dateValidator.js')

9 Given url apiUrl

10 Given path 'booking'

11 Given headers { accept: 'application/json', Content-Type: 'application/json' }

59 * def query = { user: 1, date: 2022-02-01}

60 And params query

[61 When method GET](#)

```
13:54:55.799 request:
1 > GET http://127.0.0.1:8900/booking?user=1&date=2022-02-01
1 > accept: application/json
1 > Content-Type: application/json
1 > Host: 127.0.0.1:8900
1 > Connection: Keep-Alive
1 > User-Agent: Apache-HttpClient/4.5.14 (Java/22.0.1)
1 > Accept-Encoding: gzip,deflate
```

13:54:55.808 response time in milliseconds: 9

1 < 200

1 < X-Powered-By: Express

1 < content-type: application/json

1 < Date: Thu, 12 Jun 2025 18:54:55 GMT

1 < Connection: keep-alive

1 < Keep-Alive: timeout=5

1 < Transfer-Encoding: chunked

```
[{"date":"2022-02-01","destination":"MAD","id":1,"origin":"WRO","userId":1}]
```

62 Then status 200

[63 * print response](#)

```
13:54:55.809 [print] [
{
  "date": "2022-02-01",
  "destination": "MAD",
  "id": 1,
```

44
3

Scenarios

2025-06-12 01:54:56 p. m.

[\[1.1:28\] Validate new booking creation](#)[\[2.1:41\] Validate mandatory fields in the request](#)[\[2.2:42\] Validate mandatory fields in the request](#)[\[2.3:43\] Validate mandatory fields in the request](#)[\[2.4:44\] Validate mandatory fields in the request](#)[\[3.1:55\] Validate errors for invalid request content](#)[\[3.2:55\] Validate errors for invalid request content](#)[\[3.3:55\] Validate errors for invalid request content](#)[\[3.4:55\] Validate errors for invalid request content](#)[\[3.5:55\] Validate errors for invalid request content](#)[\[3.6:55\] Validate errors for invalid request content](#)[\[3.7:55\] Validate errors for invalid request content](#)[\[3.8:55\] Validate errors for invalid request content](#)[\[3.9:55\] Validate errors for invalid request content](#)[\[3.10:55\] Validate errors for invalid request content](#)[\[3.11:55\] Validate errors for invalid request content](#)[\[3.12:55\] Validate errors for invalid request content](#)

```
"origin": "WRO",
"userId": 1
}
]
```

64 And match each response == structureResponse

5

bookingOperations **Operations** **getAllBookingOK**

ms: 19

Scenario: [4.2:68] **Validate getting all bookings with and without parameters**

>> Background:

5 * def dataRequest = read('classpath:data/bookingOperations/requestCreateNewBooking.json')

3

6 * def structureError400 = read('classpath:data/structureError.json')

1

7 * def structureResponse = read('classpath:data/bookingOperations/structureResponse200.json')

0

8 * def dateValidator = read('classpath:helpers/dateValidator.js')

0

9 Given url apiUrl

0

10 Given path 'booking'

0

11 Given headers { accept: 'application/json', Content-Type: 'application/json' }

0

59 * def query = { user: null, date: null}

0

60 And params query

0

61 When method GET

6

13:54:55.825 request:

1 > GET http://127.0.0.1:8900/booking

1 > accept: application/json

1 > Content-Type: application/json

1 > Host: 127.0.0.1:8900

1 > Connection: Keep-Alive

1 > User-Agent: Apache-HttpClient/4.5.14 (Java/22.0.1)

1 > Accept-Encoding: gzip,deflate

13:54:55.830 response time in milliseconds: 5

1 < 200

1 < X-Powered-By: Express

1 < content-type: application/json

1 < Date: Thu, 12 Jun 2025 18:54:55 GMT

1 < Connection: keep-alive

1 < Keep-Alive: timeout=5

1 < Transfer-Encoding: chunked

```
[{"date": "2022-02-01", "destination": "MAD", "id": 1, "origin": "WRO", "userId": 1},
{"date": "2022-03-12", "destination": "WRO", "id": 2, "origin": "DUB", "userId": 1},
{"date": "2024-06-10", "destination": "MEX", "id": 3, "origin": "JFK", "userId": 1}]
```

62 Then status 200

0

63 * print response

0

13:54:55.831 [print] [

```
{
  "date": "2022-02-01",
  "destination": "MAD",
  "id": 1,
  "origin": "WRO",
  "userId": 1
},
```

```
{
  "date": "2022-03-12",
  "destination": "WRO",
  "id": 2,
  "origin": "DUB",
  "userId": 1
},
```

```
{
  "date": "2024-06-10",
  "destination": "MEX",
  "id": 3,
  "origin": "JFK",
  "userId": 1
}
```

]

44
3

Scenarios

2025-06-12 01:54:56 p. m.

[\[1.1:28\] Validate new booking creation](#)[\[2.1:41\] Validate mandatory fields in the request](#)[\[2.2:42\] Validate mandatory fields in the request](#)[\[2.3:43\] Validate mandatory fields in the request](#)[\[2.4:44\] Validate mandatory fields in the request](#)[\[3.1:55\] Validate errors for invalid request content](#)[\[3.2:55\] Validate errors for invalid request content](#)[\[3.3:55\] Validate errors for invalid request content](#)[\[3.4:55\] Validate errors for invalid request content](#)[\[3.5:55\] Validate errors for invalid request content](#)[\[3.6:55\] Validate errors for invalid request content](#)[\[3.7:55\] Validate errors for invalid request content](#)[\[3.8:55\] Validate errors for invalid request content](#)[\[3.9:55\] Validate errors for invalid request content](#)[\[3.10:55\] Validate errors for invalid request content](#)[\[3.11:55\] Validate errors for invalid request content](#)[\[3.12:55\] Validate errors for invalid request content](#)

64 And match each response == structureResponse

7

bookingOperations **Operations** **getAllBookingOK**

ms: 15

Scenario: [4.3:69] **Validate getting all bookings with and without parameters**

>> Background:

5 * def dataRequest = read('classpath:data/bookingOperations/requestCreateNewBooking.json')

3

6 * def structureError400 = read('classpath:data/structureError.json')

1

7 * def structureResponse = read('classpath:data/bookingOperations/structureResponse200.json')

0

8 * def dateValidator = read('classpath:helpers/dateValidator.js')

0

9 Given url apiUrl

0

10 Given path 'booking'

0

11 Given headers { accept: 'application/json', Content-Type: 'application/json' }

0

59 * def query = { user: null, date: 2022-02-01 }

0

60 And params query

0

61 When method GET

7

13:54:55.847 request:

1 > GET http://127.0.0.1:8900/booking?date=2022-02-01

1 > accept: application/json

1 > Content-Type: application/json

1 > Host: 127.0.0.1:8900

1 > Connection: Keep-Alive

1 > User-Agent: Apache-HttpClient/4.5.14 (Java/22.0.1)

1 > Accept-Encoding: gzip,deflate

13:54:55.853 response time in milliseconds: 6

1 < 200

1 < X-Powered-By: Express

1 < content-type: application/json

1 < Date: Thu, 12 Jun 2025 18:54:55 GMT

1 < Connection: keep-alive

1 < Keep-Alive: timeout=5

1 < Transfer-Encoding: chunked

[{"date": "2022-02-01", "destination": "MAD", "id": 1, "origin": "WRO", "userId": 1}]

62 Then status 200

0

63 * print response

0

13:54:55.854 [print] [

```
{
  "date": "2022-02-01",
  "destination": "MAD",
  "id": 1,
  "origin": "WRO",
  "userId": 1
}
```

]

64 And match each response == structureResponse

2

bookingOperations **Operations** **getAllBookingOK**

ms: 17

Scenario: [4.4:70] **Validate getting all bookings with and without parameters**

>> Background:

5 * def dataRequest = read('classpath:data/bookingOperations/requestCreateNewBooking.json')

2

13:54:55.860 karate.env system property was: null

6 * def structureError400 = read('classpath:data/structureError.json')

1

7 * def structureResponse = read('classpath:data/bookingOperations/structureResponse200.json')

0

8 * def dateValidator = read('classpath:helpers/dateValidator.js')

0

9 Given url apiUrl

0

10 Given path 'booking'

0

11 Given headers { accept: 'application/json', Content-Type: 'application/json' }

0

59 * def query = { user: 1, date: null }

0



44

3

Scenarios

2025-06-12 01:54:56 p. m.

[\[1.1:28\] Validate new booking creation](#)[\[2.1:41\] Validate mandatory fields in the request](#)[\[2.2:42\] Validate mandatory fields in the request](#)[\[2.3:43\] Validate mandatory fields in the request](#)[\[2.4:44\] Validate mandatory fields in the request](#)[\[3.1:55\] Validate errors for invalid request content](#)[\[3.2:55\] Validate errors for invalid request content](#)[\[3.3:55\] Validate errors for invalid request content](#)[\[3.4:55\] Validate errors for invalid request content](#)[\[3.5:55\] Validate errors for invalid request content](#)[\[3.6:55\] Validate errors for invalid request content](#)[\[3.7:55\] Validate errors for invalid request content](#)[\[3.8:55\] Validate errors for invalid request content](#)[\[3.9:55\] Validate errors for invalid request content](#)[\[3.10:55\] Validate errors for invalid request content](#)[\[3.11:55\] Validate errors for invalid request content](#)[\[3.12:55\] Validate errors for invalid request content](#)

60 And params query

0

[61 When method GET](#)

6

13:54:55.865 request:

```
1 > GET http://127.0.0.1:8900/booking?user=1
1 > accept: application/json
1 > Content-Type: application/json
1 > Host: 127.0.0.1:8900
1 > Connection: Keep-Alive
1 > User-Agent: Apache-HttpClient/4.5.14 (Java/22.0.1)
1 > Accept-Encoding: gzip,deflate
```

13:54:55.871 response time in milliseconds: 5

```
1 < 200
1 < X-Powered-By: Express
1 < content-type: application/json
1 < Date: Thu, 12 Jun 2025 18:54:55 GMT
1 < Connection: keep-alive
1 < Keep-Alive: timeout=5
1 < Transfer-Encoding: chunked
[{"date": "2022-02-01", "destination": "MAD", "id": 1, "origin": "WRO", "userId": 1},
{"date": "2022-03-12", "destination": "WRO", "id": 2, "origin": "DUB", "userId": 1},
{"date": "2024-06-10", "destination": "MEX", "id": 3, "origin": "JFK", "userId": 1}]
```

62 Then status 200

0

[63 * print response](#)

0

13:54:55.871 [print] [

```
{
  "date": "2022-02-01",
  "destination": "MAD",
  "id": 1,
  "origin": "WRO",
  "userId": 1
},
{
  "date": "2022-03-12",
  "destination": "WRO",
  "id": 2,
  "origin": "DUB",
  "userId": 1
},
{
  "date": "2024-06-10",
  "destination": "MEX",
  "id": 3,
  "origin": "JFK",
  "userId": 1
}
]
```

64 And match each response == structureResponse

5

[bookingOperations](#) [getAllBookingParamKO](#) [Operations](#)

ms: 11

Scenario: [5.1:81] **Validate errors for invalid parameters**

>> Background:

[5 * def dataRequest = read\('classpath:data/bookingOperations/requestCreateNewBooking.json'\)](#)

2

6 * def structureError400 = read('classpath:data/structureError.json')

1

7 * def structureResponse = read('classpath:data/bookingOperations/structureResponse200.json')

0

8 * def dateValidator = read('classpath:helpers/dateValidator.js')

1

9 Given url apiUri

0

10 Given path 'booking'

0

11 Given headers { accept: 'application/json', Content-Type: 'application/json' }

0

74 * def query = { user: , date: 2022-02-01 }

0

75 And params query

0

[76 When method GET](#)

5

13:54:55.891 request:

```
1 > GET http://127.0.0.1:8900/booking?user=&date=2022-02-01
```


44
3

Scenarios

2025-06-12 01:54:56 p. m.

[\[1.1:28\] Validate new booking creation](#)[\[2.1:41\] Validate mandatory fields in the request](#)[\[2.2:42\] Validate mandatory fields in the request](#)[\[2.3:43\] Validate mandatory fields in the request](#)[\[2.4:44\] Validate mandatory fields in the request](#)[\[3.1:55\] Validate errors for invalid request content](#)[\[3.2:55\] Validate errors for invalid request content](#)[\[3.3:55\] Validate errors for invalid request content](#)[\[3.4:55\] Validate errors for invalid request content](#)[\[3.5:55\] Validate errors for invalid request content](#)[\[3.6:55\] Validate errors for invalid request content](#)[\[3.7:55\] Validate errors for invalid request content](#)[\[3.8:55\] Validate errors for invalid request content](#)[\[3.9:55\] Validate errors for invalid request content](#)[\[3.10:55\] Validate errors for invalid request content](#)[\[3.11:55\] Validate errors for invalid request content](#)[\[3.12:55\] Validate errors for invalid request content](#)

```
1 > accept: application/json
1 > Content-Type: application/json
1 > Host: 127.0.0.1:8900
1 > Connection: Keep-Alive
1 > User-Agent: Apache-HttpClient/4.5.14 (Java/22.0.1)
1 > Accept-Encoding: gzip,deflate
```

13:54:55.894 response time in milliseconds: 3

1 < 400

1 < X-Powered-By: Express

1 < Content-Type: application/json; charset=utf-8

1 < Content-Length: 175

1 < ETag: W/"af-zrNqIM5ATZqyInyBJA8i+0TglNY"

1 < Date: Thu, 12 Jun 2025 18:54:55 GMT

1 < Connection: keep-alive

1 < Keep-Alive: timeout=5

```
{"errors":[{"location":{"in":"query","name":"user","docPath":"/paths/~1booking/get/parameters/0/schema","path":""},"message":"must be integer"}],"message":"Validation errors"}
```

77 Then status 400

[78 * print response](#)

13:54:55.895 [print] {

"errors": [

{

"location": {

"in": "query",

"name": "user",

"docPath": "/paths/~1booking/get/parameters/0/schema",

"path": ""

},

"message": "must be integer"

}

],

"message": "Validation errors"

}

79 And match response == structureError400

0

1

0

bookingOperations **getAllBookingParamKO** **Operations**

ms: 10

Scenario: [5.2:81] **Validate errors for invalid parameters**[>> Background:](#)

74 * def query = { user: asdf, date: 2022-02-01}

75 And params query

[76 When method GET](#)

13:54:55.904 request:

1 > GET http://127.0.0.1:8900/booking?user=asdf&date=2022-02-01

1 > accept: application/json

1 > Content-Type: application/json

1 > Host: 127.0.0.1:8900

1 > Connection: Keep-Alive

1 > User-Agent: Apache-HttpClient/4.5.14 (Java/22.0.1)

1 > Accept-Encoding: gzip,deflate

13:54:55.908 response time in milliseconds: 3

1 < 400

1 < X-Powered-By: Express

1 < Content-Type: application/json; charset=utf-8

1 < Content-Length: 175

1 < ETag: W/"af-zrNqIM5ATZqyInyBJA8i+0TglNY"

1 < Date: Thu, 12 Jun 2025 18:54:55 GMT

1 < Connection: keep-alive

1 < Keep-Alive: timeout=5

```
{"errors":[{"location":{"in":"query","name":"user","docPath":"/paths/~1booking/get/parameters/0/schema","path":""},"message":"must be integer"}],"message":"Validation errors"}
```

77 Then status 400

[78 * print response](#)

13:54:55.908 [print] {

"errors": [

{

0

0

44
3

Scenarios

2025-06-12 01:54:56 p. m.

[\[1.1:28\] Validate new booking creation](#)[\[2.1:41\] Validate mandatory fields in the request](#)[\[2.2:42\] Validate mandatory fields in the request](#)[\[2.3:43\] Validate mandatory fields in the request](#)[\[2.4:44\] Validate mandatory fields in the request](#)[\[3.1:55\] Validate errors for invalid request content](#)[\[3.2:55\] Validate errors for invalid request content](#)[\[3.3:55\] Validate errors for invalid request content](#)[\[3.4:55\] Validate errors for invalid request content](#)[\[3.5:55\] Validate errors for invalid request content](#)[\[3.6:55\] Validate errors for invalid request content](#)[\[3.7:55\] Validate errors for invalid request content](#)[\[3.8:55\] Validate errors for invalid request content](#)[\[3.9:55\] Validate errors for invalid request content](#)[\[3.10:55\] Validate errors for invalid request content](#)[\[3.11:55\] Validate errors for invalid request content](#)[\[3.12:55\] Validate errors for invalid request content](#)

```
    "location": {
      "in": "query",
      "name": "user",
      "docPath": "/paths/~1booking/get/parameters/0/schema",
      "path": ""
    },
    "message": "must be integer"
  },
],
"message": "Validation errors"
}
```

79 And match response == structureError400

0

bookingOperations **getAllBookingParamKO** **Operations**

ms: 9

Scenario: [5.3:81] **Validate errors for invalid parameters**

>> Background:

5 * def dataRequest = read('classpath:data/bookingOperations/requestCreateNewBooking.json')

1

6 * def structureError400 = read('classpath:data/structureError.json')

0

7 * def structureResponse = read('classpath:data/bookingOperations/structureResponse200.json')

0

8 * def dateValidator = read('classpath:helpers/dateValidator.js')

0

9 Given url apiUrl

0

10 Given path 'booking'

0

11 Given headers { accept: 'application/json', Content-Type: 'application/json' }

0

74 * def query = { user: @@@, date: 2022-02-01}

0

75 And params query

0

76 When method GET

5

13:54:55.916 request:

1 > GET http://127.0.0.1:8900/booking?user=%40%40%40&date=2022-02-01

1 > accept: application/json

1 > Content-Type: application/json

1 > Host: 127.0.0.1:8900

1 > Connection: Keep-Alive

1 > User-Agent: Apache-HttpClient/4.5.14 (Java/22.0.1)

1 > Accept-Encoding: gzip,deflate

13:54:55.920 response time in milliseconds: 3

1 < 400

1 < X-Powered-By: Express

1 < Content-Type: application/json; charset=utf-8

1 < Content-Length: 175

1 < ETag: W/"af-zrNqIM5ATZqyInyBJA8i+0TglNY"

1 < Date: Thu, 12 Jun 2025 18:54:55 GMT

1 < Connection: keep-alive

1 < Keep-Alive: timeout=5

{ "errors": [{ "location": { "in": "query", "name": "user", "docPath": "/paths/~1booking/get/parameters/0/schema", "path": "" }, "message": "must be integer" }, { "message": "Validation errors" }] }

77 Then status 400

0

78 * print response

0

13:54:55.920 [print] {

"errors": [

{

"location": {

"in": "query",

"name": "user",

"docPath": "/paths/~1booking/get/parameters/0/schema",

"path": ""

},

"message": "must be integer"

}

],

"message": "Validation errors"

}

79 And match response == structureError400

0

44
3

Scenarios

2025-06-12 01:54:56 p. m.

[\[1.1:28\] Validate new booking creation](#)[\[2.1:41\] Validate mandatory fields in the request](#)[\[2.2:42\] Validate mandatory fields in the request](#)[\[2.3:43\] Validate mandatory fields in the request](#)[\[2.4:44\] Validate mandatory fields in the request](#)[\[3.1:55\] Validate errors for invalid request content](#)[\[3.2:55\] Validate errors for invalid request content](#)[\[3.3:55\] Validate errors for invalid request content](#)[\[3.4:55\] Validate errors for invalid request content](#)[\[3.5:55\] Validate errors for invalid request content](#)[\[3.6:55\] Validate errors for invalid request content](#)[\[3.7:55\] Validate errors for invalid request content](#)[\[3.8:55\] Validate errors for invalid request content](#)[\[3.9:55\] Validate errors for invalid request content](#)[\[3.10:55\] Validate errors for invalid request content](#)[\[3.11:55\] Validate errors for invalid request content](#)[\[3.12:55\] Validate errors for invalid request content](#)

bookingOperations	getAllBookingParamKO	Operations	ms: 9
Scenario: [5.4:81] Validate errors for invalid parameters			
>> Background:			
5	* def dataRequest = read('classpath:data/bookingOperations/requestCreateNewBooking.json')		1
13:54:55.924 karate.env system property was: null			
6	* def structureError400 = read('classpath:data/structureError.json')		0
7	* def structureResponse = read('classpath:data/bookingOperations/structureResponse200.json')		0
8	* def dateValidator = read('classpath:helpers/dateValidator.js')		0
9	Given url apiUrl		0
10	Given path 'booking'		0
11	Given headers { accept: 'application/json', Content-Type: 'application/json' }		0
74	* def query = { user: 1000, date: 2022-02-01}		0
75	And params query		0
76	When method GET		5
13:54:55.927 request:			
1 > GET http://127.0.0.1:8900/booking?user=1000&date=2022-02-01			
1 > accept: application/json			
1 > Content-Type: application/json			
1 > Host: 127.0.0.1:8900			
1 > Connection: Keep-Alive			
1 > User-Agent: Apache-HttpClient/4.5.14 (Java/22.0.1)			
1 > Accept-Encoding: gzip,deflate			
13:54:55.932 response time in milliseconds: 4			
1 < 404			
1 < X-Powered-By: Express			
1 < Content-Type: application/json; charset=utf-8			
1 < Content-Length: 48			
1 < ETag: W/"30-T4jfxjWlIoewPgbWzt5b4BlzM7s"			
1 < Date: Thu, 12 Jun 2025 18:54:55 GMT			
1 < Connection: keep-alive			
1 < Keep-Alive: timeout=5			
{ "errors": null, "message": "No user with id 1000" }			
77	Then status 404		0
78	* print response		0
13:54:55.932 [print] {			
"errors": null,			
"message": "No user with id 1000"			
}			
79	And match response == structureError400		0

bookingOperations	getAllBookingParamKO	Operations	ms: 8
Scenario: [5.5:81] Validate errors for invalid parameters			
>> Background:			
74	* def query = { user: 1, date: 2022}		0
75	And params query		0
76	When method GET		5
13:54:55.940 request:			
1 > GET http://127.0.0.1:8900/booking?user=1&date=2022			
1 > accept: application/json			
1 > Content-Type: application/json			
1 > Host: 127.0.0.1:8900			
1 > Connection: Keep-Alive			
1 > User-Agent: Apache-HttpClient/4.5.14 (Java/22.0.1)			
1 > Accept-Encoding: gzip,deflate			
13:54:55.944 response time in milliseconds: 4			
1 < 400			
1 < X-Powered-By: Express			
1 < Content-Type: application/json; charset=utf-8			
1 < Content-Length: 186			



44
3

Scenarios

2025-06-12 01:54:56 p. m.

[1.1:28] [Validate new booking creation](#)

[2.1:41] [Validate mandatory fields in the request](#)

[2.2:42] [Validate mandatory fields in the request](#)

[2.3:43] [Validate mandatory fields in the request](#)

[2.4:44] [Validate mandatory fields in the request](#)

[3.1:55] [Validate errors for invalid request content](#)

[3.2:55] [Validate errors for invalid request content](#)

[3.3:55] [Validate errors for invalid request content](#)

[3.4:55] [Validate errors for invalid request content](#)

[3.5:55] [Validate errors for invalid request content](#)

[3.6:55] [Validate errors for invalid request content](#)

[3.7:55] [Validate errors for invalid request content](#)

[3.8:55] [Validate errors for invalid request content](#)

[3.9:55] [Validate errors for invalid request content](#)

[3.10:55] [Validate errors for invalid request content](#)

[3.11:55] [Validate errors for invalid request content](#)

[3.12:55] [Validate errors for invalid request content](#)

1 < ETag: W/"ba-s+aAibrvRmrIjvu9lofYSGWhxXI"	
1 < Date: Thu, 12 Jun 2025 18:54:55 GMT	
1 < Connection: keep-alive	
1 < Keep-Alive: timeout=5	
{ "errors": [{ "location": { "in": "query", "name": "date", "docPath": "/paths/~1booking/get/parameters/1/schema", "path": "" }, "message": "must match format \"date\"" }], "message": "Validation errors" }	
77 Then status 400	0
78 * print response	0
13:54:55.944 [print] {	
"errors": [
{	
"location": {	
"in": "query",	
"name": "date",	
"docPath": "/paths/~1booking/get/parameters/1/schema",	
"path": ""	
},	
"message": "must match format \"date\""	
}	
],	
"message": "Validation errors"	
}	
79 And match response == structureError400	0

<div>bookingOperationsgetAllBookingParamKOOperations</div>	ms: 8
Scenario: [5.6:81] Validate errors for invalid parameters	
>> Background:	
74 * def query = { user: 1, date: 2022-30-01 }	0
75 And params query	0
76 When method GET	5
13:54:55.950 request:	
1 > GET http://127.0.0.1:8900/booking?user=1&date=2022-30-01	
1 > accept: application/json	
1 > Content-Type: application/json	
1 > Host: 127.0.0.1:8900	
1 > Connection: Keep-Alive	
1 > User-Agent: Apache-HttpClient/4.5.14 (Java/22.0.1)	
1 > Accept-Encoding: gzip,deflate	
13:54:55.954 response time in milliseconds: 4	
1 < 400	
1 < X-Powered-By: Express	
1 < Content-Type: application/json; charset=utf-8	
1 < Content-Length: 186	
1 < ETag: W/"ba-s+aAibrvRmrIjvu9lofYSGWhxXI"	
1 < Date: Thu, 12 Jun 2025 18:54:55 GMT	
1 < Connection: keep-alive	
1 < Keep-Alive: timeout=5	
{ "errors": [{ "location": { "in": "query", "name": "date", "docPath": "/paths/~1booking/get/parameters/1/schema", "path": "" }, "message": "must match format \"date\"" }], "message": "Validation errors" }	
77 Then status 400	0
78 * print response	0
13:54:55.955 [print] {	
"errors": [
{	
"location": {	
"in": "query",	
"name": "date",	
"docPath": "/paths/~1booking/get/parameters/1/schema",	
"path": ""	
},	
"message": "must match format \"date\""	
}	
],	
"message": "Validation errors"	
}	

44
3

Scenarios

2025-06-12 01:54:56 p. m.

[\[1.1:28\] Validate new booking creation](#)[\[2.1:41\] Validate mandatory fields in the request](#)[\[2.2:42\] Validate mandatory fields in the request](#)[\[2.3:43\] Validate mandatory fields in the request](#)[\[2.4:44\] Validate mandatory fields in the request](#)[\[3.1:55\] Validate errors for invalid request content](#)[\[3.2:55\] Validate errors for invalid request content](#)[\[3.3:55\] Validate errors for invalid request content](#)[\[3.4:55\] Validate errors for invalid request content](#)[\[3.5:55\] Validate errors for invalid request content](#)[\[3.6:55\] Validate errors for invalid request content](#)[\[3.7:55\] Validate errors for invalid request content](#)[\[3.8:55\] Validate errors for invalid request content](#)[\[3.9:55\] Validate errors for invalid request content](#)[\[3.10:55\] Validate errors for invalid request content](#)[\[3.11:55\] Validate errors for invalid request content](#)[\[3.12:55\] Validate errors for invalid request content](#)

79 And match response == structureError400

0

bookingOperations**getAllBookingParamKO****Operations**

ms: 9

Scenario: [5.7:81] **Validate errors for invalid parameters**

>> Background:

74 * def query = { user: 1, date: 2022-04-50}

0

75 And params query

0

76 When method GET

5

13:54:55.963 request:

1 > GET http://127.0.0.1:8900/booking?user=1&date=2022-04-50

1 > accept: application/json

1 > Content-Type: application/json

1 > Host: 127.0.0.1:8900

1 > Connection: Keep-Alive

1 > User-Agent: Apache-HttpClient/4.5.14 (Java/22.0.1)

1 > Accept-Encoding: gzip,deflate

13:54:55.967 response time in milliseconds: 3

1 < 400

1 < X-Powered-By: Express

1 < Content-Type: application/json; charset=utf-8

1 < Content-Length: 186

1 < ETag: W/"ba-s+aAibrvRmrIjvu9lofYSGWhxXI"

1 < Date: Thu, 12 Jun 2025 18:54:55 GMT

1 < Connection: keep-alive

1 < Keep-Alive: timeout=5

```
{"errors":[{"location":{"in":"query","name":"date","docPath":"/paths/~1booking/get/parameters/1/schema","path":""},"message":"must match format \\"date\\""}],"message":"Validation errors"}
```

77 Then status 400

0

78 * print response

0

13:54:55.967 [print] {

"errors": [

{

"location": {

"in": "query",

"name": "date",

"docPath": "/paths/~1booking/get/parameters/1/schema",

"path": ""

},

"message": "must match format \\"date\\""

}

],

"message": "Validation errors"

}

79 And match response == structureError400

0

bookingOperations**getAllBookingParamKO****Operations**

ms: 9

Scenario: [5.8:81] **Validate errors for invalid parameters**

>> Background:

74 * def query = { user: 1, date: 20-04-2020}

0

75 And params query

0

76 When method GET

5

13:54:55.975 request:

1 > GET http://127.0.0.1:8900/booking?user=1&date=20-04-2020

1 > accept: application/json

1 > Content-Type: application/json

1 > Host: 127.0.0.1:8900

1 > Connection: Keep-Alive

1 > User-Agent: Apache-HttpClient/4.5.14 (Java/22.0.1)

1 > Accept-Encoding: gzip,deflate

13:54:55.979 response time in milliseconds: 4

1 < 400

1 < X-Powered-By: Express

1 < Content-Type: application/json; charset=utf-8

1 < Content-Length: 186



44
3

Scenarios

2025-06-12 01:54:56 p. m.

[\[1.1:28\] Validate new booking creation](#)

[\[2.1:41\] Validate mandatory fields in the request](#)

[\[2.2:42\] Validate mandatory fields in the request](#)

[\[2.3:43\] Validate mandatory fields in the request](#)

[\[2.4:44\] Validate mandatory fields in the request](#)

[\[3.1:55\] Validate errors for invalid request content](#)

[\[3.2:55\] Validate errors for invalid request content](#)

[\[3.3:55\] Validate errors for invalid request content](#)

[\[3.4:55\] Validate errors for invalid request content](#)

[\[3.5:55\] Validate errors for invalid request content](#)

[\[3.6:55\] Validate errors for invalid request content](#)

[\[3.7:55\] Validate errors for invalid request content](#)

[\[3.8:55\] Validate errors for invalid request content](#)

[\[3.9:55\] Validate errors for invalid request content](#)

[\[3.10:55\] Validate errors for invalid request content](#)

[\[3.11:55\] Validate errors for invalid request content](#)

[\[3.12:55\] Validate errors for invalid request content](#)

```
1 < ETag: W/"ba-s+aAibrvRmrIjvu9lofYSGWhxXI"
1 < Date: Thu, 12 Jun 2025 18:54:55 GMT
1 < Connection: keep-alive
1 < Keep-Alive: timeout=5
{"errors":[{"location":{"in":"query","name":"date","docPath":"/paths/~1booking/get/parameters/1/schema","path":""},"message":"must match format \"date\\\""}],"message":"Validation errors"}
```

77 Then status 400

[78 * print response](#)

```
13:54:55.980 [print] {
  "errors": [
    {
      "location": {
        "in": "query",
        "name": "date",
        "docPath": "/paths/~1booking/get/parameters/1/schema",
        "path": ""
      },
      "message": "must match format \"date\\\"\"
    }
  ],
  "message": "Validation errors"
}
```

79 And match response == structureError400

bookingOperations	getAllBookingParamKO	Operations	ms: 9
Scenario: [5.9:81] Validate errors for invalid parameters			

[>> Background:](#)

[5 * def dataRequest = read\('classpath:data/bookingOperations/requestCreateNewBooking.json'\)](#)

6 * def structureError400 = read('classpath:data/structureError.json')

7 * def structureResponse = read('classpath:data/bookingOperations/structureResponse200.json')

8 * def dateValidator = read('classpath:helpers/dateValidator.js')

9 Given url apiUrl

10 Given path 'booking'

11 Given headers { accept: 'application/json', Content-Type: 'application/json' }

74 * def query = { user: 1, date: 2022/02/01}

75 And params query

[76 When method GET](#)

```
13:54:55.986 request:
1 > GET http://127.0.0.1:8900/booking?user=1&date=2022%2F02%2F01
1 > accept: application/json
1 > Content-Type: application/json
1 > Host: 127.0.0.1:8900
1 > Connection: Keep-Alive
1 > User-Agent: Apache-HttpClient/4.5.14 (Java/22.0.1)
1 > Accept-Encoding: gzip,deflate
```

13:54:55.990 response time in milliseconds: 4

1 < 400

1 < X-Powered-By: Express

1 < Content-Type: application/json; charset=utf-8

1 < Content-Length: 186

1 < ETag: W/"ba-s+aAibrvRmrIjvu9lofYSGWhxXI"

1 < Date: Thu, 12 Jun 2025 18:54:55 GMT

1 < Connection: keep-alive

1 < Keep-Alive: timeout=5

```
{"errors":[{"location":{"in":"query","name":"date","docPath":"/paths/~1booking/get/parameters/1/schema","path":""},"message":"must match format \"date\\\""}],"message":"Validation errors"}
```

77 Then status 400

[78 * print response](#)

```
13:54:55.991 [print] {
  "errors": [
    {
      "location": {
```

44
3

Scenarios

2025-06-12 01:54:56 p. m.

[\[1.1:28\] Validate new booking creation](#)[\[2.1:41\] Validate mandatory fields in the request](#)[\[2.2:42\] Validate mandatory fields in the request](#)[\[2.3:43\] Validate mandatory fields in the request](#)[\[2.4:44\] Validate mandatory fields in the request](#)[\[3.1:55\] Validate errors for invalid request content](#)[\[3.2:55\] Validate errors for invalid request content](#)[\[3.3:55\] Validate errors for invalid request content](#)[\[3.4:55\] Validate errors for invalid request content](#)[\[3.5:55\] Validate errors for invalid request content](#)[\[3.6:55\] Validate errors for invalid request content](#)[\[3.7:55\] Validate errors for invalid request content](#)[\[3.8:55\] Validate errors for invalid request content](#)[\[3.9:55\] Validate errors for invalid request content](#)[\[3.10:55\] Validate errors for invalid request content](#)[\[3.11:55\] Validate errors for invalid request content](#)[\[3.12:55\] Validate errors for invalid request content](#)

```
      "in": "query",
      "name": "date",
      "docPath": "/paths/~1booking/get/parameters/1/schema",
      "path": ""
    },
    "message": "must match format \"date\""
  }
],
"message": "Validation errors"
}
```

79 And match response == structureError400

0

bookingOperations **getAllBookingParamKO** **Operations**

ms: 10

Scenario: [5.10:81] **Validate errors for invalid parameters**

>> Background:

```
5 * def dataRequest = read('classpath:data/bookingOperations/
  requestCreateNewBooking.json')
6 * def structureError400 = read('classpath:data/structureError.json')
7 * def structureResponse = read('classpath:data/bookingOperations/
  structureResponse200.json')
8 * def dateValidator = read('classpath:helpers/dateValidator.js')
9 Given url apiUrl
10 Given path 'booking'
11 Given headers { accept: 'application/json', Content-Type: 'application/json' }
74 * def query = { user: 1, date: ASDC-AS-ER}
75 And params query
76 When method GET
```

13:54:55.999 request:

```
1 > GET http://127.0.0.1:8900/booking?user=1&date=ASDC-AS-ER
1 > accept: application/json
1 > Content-Type: application/json
1 > Host: 127.0.0.1:8900
1 > Connection: Keep-Alive
1 > User-Agent: Apache-HttpClient/4.5.14 (Java/22.0.1)
1 > Accept-Encoding: gzip,deflate
```

13:54:56.003 response time in milliseconds: 4

```
1 < 400
1 < X-Powered-By: Express
1 < Content-Type: application/json; charset=utf-8
1 < Content-Length: 186
1 < ETag: W/"ba-s+aAibrvRmrIjvu9lofYSGWhxXI"
1 < Date: Thu, 12 Jun 2025 18:54:56 GMT
1 < Connection: keep-alive
1 < Keep-Alive: timeout=5
{"errors":[{"location":{"in":"query","name":"date","docPath":"/paths/~1booking/get/parameters/1/
schema","path":""},"message":"must match format \"date\""}],"message":"Validation errors"}
```

77 Then status 400

78 * print response

13:54:56.004 [print] {

```
"errors": [
  {
    "location": {
      "in": "query",
      "name": "date",
      "docPath": "/paths/~1booking/get/parameters/1/schema",
      "path": ""
    },
    "message": "must match format \"date\""
  }
],
"message": "Validation errors"
}
```

79 And match response == structureError400

0

44
3

Scenarios

2025-06-12 01:54:56 p. m.

[\[1.1:28\] Validate new booking creation](#)[\[2.1:41\] Validate mandatory fields in the request](#)[\[2.2:42\] Validate mandatory fields in the request](#)[\[2.3:43\] Validate mandatory fields in the request](#)[\[2.4:44\] Validate mandatory fields in the request](#)[\[3.1:55\] Validate errors for invalid request content](#)[\[3.2:55\] Validate errors for invalid request content](#)[\[3.3:55\] Validate errors for invalid request content](#)[\[3.4:55\] Validate errors for invalid request content](#)[\[3.5:55\] Validate errors for invalid request content](#)[\[3.6:55\] Validate errors for invalid request content](#)[\[3.7:55\] Validate errors for invalid request content](#)[\[3.8:55\] Validate errors for invalid request content](#)[\[3.9:55\] Validate errors for invalid request content](#)[\[3.10:55\] Validate errors for invalid request content](#)[\[3.11:55\] Validate errors for invalid request content](#)[\[3.12:55\] Validate errors for invalid request content](#)

bookingOperations	getAllBookingParamKO	Operations	ms: 12
Scenario: [5.11:81] Validate errors for invalid parameters			
>> Background:			
5	* def dataRequest = read('classpath:data/bookingOperations/requestCreateNewBooking.json')		1
6	* def structureError400 = read('classpath:data/structureError.json')		0
7	* def structureResponse = read('classpath:data/bookingOperations/structureResponse200.json')		0
8	* def dateValidator = read('classpath:helpers/dateValidator.js')		0
9	Given url apiUrl		0
10	Given path 'booking'		0
11	Given headers { accept: 'application/json', Content-Type: 'application/json' }		0
74	* def query = { user: 1, date: aaa}		0
75	And params query		0
76	When method GET		8
13:54:56.013 request:			
1 > GET http://127.0.0.1:8900/booking?user=1&date=aaa			
1 > accept: application/json			
1 > Content-Type: application/json			
1 > Host: 127.0.0.1:8900			
1 > Connection: Keep-Alive			
1 > User-Agent: Apache-HttpClient/4.5.14 (Java/22.0.1)			
1 > Accept-Encoding: gzip,deflate			
13:54:56.019 response time in milliseconds: 5			
1 < 400			
1 < X-Powered-By: Express			
1 < Content-Type: application/json; charset=utf-8			
1 < Content-Length: 186			
1 < ETag: W/"ba-s+aAibrvRmrIjvu9lofYSGWhxXI"			
1 < Date: Thu, 12 Jun 2025 18:54:56 GMT			
1 < Connection: keep-alive			
1 < Keep-Alive: timeout=5			
{ "errors": [{ "location": { "in": "query", "name": "date", "docPath": "/paths/~1booking/get/parameters/1/schema", "path": "" }, "message": "must match format \"date\"" }, { "message": "Validation errors" }] }			
77	Then status 400		0
78	* print response		0
13:54:56.020 [print] {			
"errors": [
{			
"location": {			
"in": "query",			
"name": "date",			
"docPath": "/paths/~1booking/get/parameters/1/schema",			
"path": ""			
},			
"message": "must match format \"date\""			
},			
],			
"message": "Validation errors"			
}			
79	And match response == structureError400		0

bookingOperations	getAllBookingParamKO	Operations	ms: 11
Scenario: [5.12:81] Validate errors for invalid parameters			
>> Background:			
5	* def dataRequest = read('classpath:data/bookingOperations/requestCreateNewBooking.json')		2
13:54:56.023 karate.env system property was: null			
6	* def structureError400 = read('classpath:data/structureError.json')		1
7	* def structureResponse = read('classpath:data/bookingOperations/structureResponse200.json')		0
8	* def dateValidator = read('classpath:helpers/dateValidator.js')		0

44
3

Scenarios

2025-06-12 01:54:56 p. m.

[\[1.1:28\] Validate new booking creation](#)[\[2.1:41\] Validate mandatory fields in the request](#)[\[2.2:42\] Validate mandatory fields in the request](#)[\[2.3:43\] Validate mandatory fields in the request](#)[\[2.4:44\] Validate mandatory fields in the request](#)[\[3.1:55\] Validate errors for invalid request content](#)[\[3.2:55\] Validate errors for invalid request content](#)[\[3.3:55\] Validate errors for invalid request content](#)[\[3.4:55\] Validate errors for invalid request content](#)[\[3.5:55\] Validate errors for invalid request content](#)[\[3.6:55\] Validate errors for invalid request content](#)[\[3.7:55\] Validate errors for invalid request content](#)[\[3.8:55\] Validate errors for invalid request content](#)[\[3.9:55\] Validate errors for invalid request content](#)[\[3.10:55\] Validate errors for invalid request content](#)[\[3.11:55\] Validate errors for invalid request content](#)[\[3.12:55\] Validate errors for invalid request content](#)

9	Given url apiUrl	0
10	Given path 'booking'	0
11	Given headers { accept: 'application/json', Content-Type: 'application/json' }	0
74	* def query = { user: 1, date: @#!@}	0
75	And params query	0
76	When method GET	6
13:54:56.029 request:		
1	> GET http://127.0.0.1:8900/booking?user=1&date=%40%23%21%40	
1	> accept: application/json	
1	> Content-Type: application/json	
1	> Host: 127.0.0.1:8900	
1	> Connection: Keep-Alive	
1	> User-Agent: Apache-HttpClient/4.5.14 (Java/22.0.1)	
1	> Accept-Encoding: gzip,deflate	
13:54:56.033 response time in milliseconds: 4		
1	< 400	
1	< X-Powered-By: Express	
1	< Content-Type: application/json; charset=utf-8	
1	< Content-Length: 186	
1	< ETag: W/"ba-s+aAibrvRmrIjvu9lofYSGWhxXI"	
1	< Date: Thu, 12 Jun 2025 18:54:56 GMT	
1	< Connection: keep-alive	
1	< Keep-Alive: timeout=5	
	{ "errors": [{ "location": { "in": "query", "name": "date", "docPath": "/paths/~1booking/get/parameters/1/schema", "path": "" }, "message": "must match format \"date\"" }], "message": "Validation errors" }	
77	Then status 400	0
78	* print response	0
13:54:56.035 [print] {		
	"errors": [
	{	
	"location": {	
	"in": "query",	
	"name": "date",	
	"docPath": "/paths/~1booking/get/parameters/1/schema",	
	"path": ""	
	},	
	"message": "must match format \"date\""	
	}	
],	
	"message": "Validation errors"	
	}	
79	And match response == structureError400	1

bookingOperations getAllBookingByIdOK Operations		ms: 22
Scenario: [6.1:92] Validate getting bookings by id		
>> Background:		
5	* def dataRequest = read('classpath:data/bookingOperations/requestCreateNewBooking.json')	2
6	* def structureError400 = read('classpath:data/structureError.json')	1
7	* def structureResponse = read('classpath:data/bookingOperations/structureResponse200.json')	0
8	* def dateValidator = read('classpath:helpers/dateValidator.js')	0
9	Given url apiUrl	0
10	Given path 'booking'	0
11	Given headers { accept: 'application/json', Content-Type: 'application/json' }	0
85	Given path id	0
86	When method GET	7
13:54:56.044 request:		
1	> GET http://127.0.0.1:8900/booking/1	
1	> accept: application/json	
1	> Content-Type: application/json	
1	> Host: 127.0.0.1:8900	
1	> Connection: Keep-Alive	
1	> User-Agent: Apache-HttpClient/4.5.14 (Java/22.0.1)	

44
3

Scenarios

2025-06-12 01:54:56 p. m.

[\[1.1:28\] Validate new booking creation](#)[\[2.1:41\] Validate mandatory fields in the request](#)[\[2.2:42\] Validate mandatory fields in the request](#)[\[2.3:43\] Validate mandatory fields in the request](#)[\[2.4:44\] Validate mandatory fields in the request](#)[\[3.1:55\] Validate errors for invalid request content](#)[\[3.2:55\] Validate errors for invalid request content](#)[\[3.3:55\] Validate errors for invalid request content](#)[\[3.4:55\] Validate errors for invalid request content](#)[\[3.5:55\] Validate errors for invalid request content](#)[\[3.6:55\] Validate errors for invalid request content](#)[\[3.7:55\] Validate errors for invalid request content](#)[\[3.8:55\] Validate errors for invalid request content](#)[\[3.9:55\] Validate errors for invalid request content](#)[\[3.10:55\] Validate errors for invalid request content](#)[\[3.11:55\] Validate errors for invalid request content](#)[\[3.12:55\] Validate errors for invalid request content](#)

```
1 > Accept-Encoding: gzip,deflate

13:54:56.050 response time in milliseconds: 5
1 < 200
1 < X-Powered-By: Express
1 < content-type: application/json
1 < Date: Thu, 12 Jun 2025 18:54:56 GMT
1 < Connection: keep-alive
1 < Keep-Alive: timeout=5
1 < Transfer-Encoding: chunked
{"date":"2022-02-01","destination":"MAD","id":1,"origin":"WRO","userId":1}
```

87 Then status 200

[88 * print response](#)

```
13:54:56.058 [print] {
  "date": "2022-02-01",
  "destination": "MAD",
  "id": 1,
  "origin": "WRO",
  "userId": 1
}
```

89 And match response == structureResponse

0

8

2

bookingOperations

Operations

getAllBookingByIdKO

ms: 12

Scenario: [7.1:103] **Validate error for invalid id content**

>> Background:

[5 * def dataRequest = read\('classpath:data/bookingOperations/requestCreateNewBooking.json'\)](#)

6 * def structureError400 = read('classpath:data/structureError.json')

7 * def structureResponse = read('classpath:data/bookingOperations/structureResponse200.json')

8 * def dateValidator = read('classpath:helpers/dateValidator.js')

9 Given url apiUrl

10 Given path 'booking'

11 Given headers { accept: 'application/json', Content-Type: 'application/json' }

96 Given path id

[97 When method GET](#)

13:54:56.070 request:

```
1 > GET http://127.0.0.1:8900/booking/
1 > accept: application/json
1 > Content-Type: application/json
1 > Host: 127.0.0.1:8900
1 > Connection: Keep-Alive
1 > User-Agent: Apache-HttpClient/4.5.14 (Java/22.0.1)
1 > Accept-Encoding: gzip,deflate
```

13:54:56.075 response time in milliseconds: 5

```
1 < 400
1 < X-Powered-By: Express
1 < Content-Type: application/json; charset=utf-8
1 < Content-Length: 178
1 < ETag: W/"b2-ZZWffUMmWZRq5SAc+/7z1/93vxc"
1 < Date: Thu, 12 Jun 2025 18:54:56 GMT
1 < Connection: keep-alive
1 < Keep-Alive: timeout=5
{"errors":[{"location":{"in":"path","name":"id","docPath":"/paths/~1booking~1{id}/get/parameters/0/schema","path":""},"message":"must be integer"}],"message":"Validation errors"}
```

98 Then status 400

[99 * print response](#)

```
13:54:56.076 [print] {
  "errors": [
    {
      "location": {
        "in": "path",
        "name": "id",
```

0

0

44
3

Scenarios

2025-06-12 01:54:56 p. m.

[\[1.1:28\] Validate new booking creation](#)[\[2.1:41\] Validate mandatory fields in the request](#)[\[2.2:42\] Validate mandatory fields in the request](#)[\[2.3:43\] Validate mandatory fields in the request](#)[\[2.4:44\] Validate mandatory fields in the request](#)[\[3.1:55\] Validate errors for invalid request content](#)[\[3.2:55\] Validate errors for invalid request content](#)[\[3.3:55\] Validate errors for invalid request content](#)[\[3.4:55\] Validate errors for invalid request content](#)[\[3.5:55\] Validate errors for invalid request content](#)[\[3.6:55\] Validate errors for invalid request content](#)[\[3.7:55\] Validate errors for invalid request content](#)[\[3.8:55\] Validate errors for invalid request content](#)[\[3.9:55\] Validate errors for invalid request content](#)[\[3.10:55\] Validate errors for invalid request content](#)[\[3.11:55\] Validate errors for invalid request content](#)[\[3.12:55\] Validate errors for invalid request content](#)

```
    "docPath": "/paths/~1booking~1{id}/get/parameters/0/schema",
    "path": ""
  },
  "message": "must be integer"
}
],
"message": "Validation errors"
}
```

100 And match response == structureError400

0

bookingOperations **Operations** **getAllBookingByIdKO**

ms: 10

Scenario: [7.2:104] **Validate error for invalid id content**

>> Background:

5 * def dataRequest = read('classpath:data/bookingOperations/requestCreateNewBooking.json')

2

6 * def structureError400 = read('classpath:data/structureError.json')

1

7 * def structureResponse = read('classpath:data/bookingOperations/structureResponse200.json')

0

8 * def dateValidator = read('classpath:helpers/dateValidator.js')

0

9 Given url apiUrl

0

10 Given path 'booking'

0

11 Given headers { accept: 'application/json', Content-Type: 'application/json' }

0

96 Given path id

0

97 When method GET

6

13:54:56.084 request:

1 > GET http://127.0.0.1:8900/booking/aaa

1 > accept: application/json

1 > Content-Type: application/json

1 > Host: 127.0.0.1:8900

1 > Connection: Keep-Alive

1 > User-Agent: Apache-HttpClient/4.5.14 (Java/22.0.1)

1 > Accept-Encoding: gzip,deflate

13:54:56.088 response time in milliseconds: 4

1 < 400

1 < X-Powered-By: Express

1 < Content-Type: application/json; charset=utf-8

1 < Content-Length: 178

1 < ETag: W/"b2-ZZWffUMmWZRq5SAc+/7z1/93vxc"

1 < Date: Thu, 12 Jun 2025 18:54:56 GMT

1 < Connection: keep-alive

1 < Keep-Alive: timeout=5

```
{
  "errors": [
    {
      "location": {
        "in": "path",
        "name": "id",
        "docPath": "/paths/~1booking~1{id}/get/parameters/0/schema",
        "path": ""
      },
      "message": "must be integer"
    }
  ],
  "message": "Validation errors"
}
```

98 Then status 400

0

99 * print response

0

13:54:56.089 [print] {

"errors": [

{

"location": {

"in": "path",

"name": "id",

"docPath": "/paths/~1booking~1{id}/get/parameters/0/schema",

"path": ""

},

"message": "must be integer"

}

],

"message": "Validation errors"

}

100 And match response == structureError400

0

bookingOperations **Operations** **getAllBookingByIdKO**

ms: 11

Scenario: [7.3:105] **Validate error for invalid id content**

>> Background:



44

3

Scenarios

2025-06-12 01:54:56 p. m.

[1.1:28] Validate new booking creation

[2.1:41] Validate mandatory fields in the request

[2.2:42] Validate mandatory fields in the request

[2.3:43] Validate mandatory fields in the request

[2.4:44] Validate mandatory fields in the request

[3.1:55] Validate errors for invalid request content

[3.2:55] Validate errors for invalid request content

[3.3:55] Validate errors for invalid request content

[3.4:55] Validate errors for invalid request content

[3.5:55] Validate errors for invalid request content

[3.6:55] Validate errors for invalid request content

[3.7:55] Validate errors for invalid request content

[3.8:55] Validate errors for invalid request content

[3.9:55] Validate errors for invalid request content

[3.10:55] Validate errors for invalid request content

[3.11:55] Validate errors for invalid request content

[3.12:55] Validate errors for invalid request content

96 Given path id

0

[97 When method GET](#)

6

13:54:56.098 request:

```
1 > GET http://127.0.0.1:8900/booking/1.5
1 > accept: application/json
1 > Content-Type: application/json
1 > Host: 127.0.0.1:8900
1 > Connection: Keep-Alive
1 > User-Agent: Apache-HttpClient/4.5.14 (Java/22.0.1)
1 > Accept-Encoding: gzip,deflate
```

13:54:56.103 response time in milliseconds: 4

```
1 < 400
1 < X-Powered-By: Express
1 < Content-Type: application/json; charset=utf-8
1 < Content-Length: 178
1 < ETag: W/"b2-ZZWffUMmWZRq5SAc+/7z1/93vxc"
1 < Date: Thu, 12 Jun 2025 18:54:56 GMT
1 < Connection: keep-alive
1 < Keep-Alive: timeout=5
{"errors":[{"location":{"in":"path","name":"id","docPath":"/paths/~1booking~1{id}/get/parameters/0/schema","path":""},"message":"must be integer"}],"message":"Validation errors"}
```

98 Then status 400

0

[99 * print response](#)

0

```
13:54:56.103 [print] {
  "errors": [
    {
      "location": {
        "in": "path",
        "name": "id",
        "docPath": "/paths/~1booking~1{id}/get/parameters/0/schema",
        "path": ""
      },
      "message": "must be integer"
    }
  ],
  "message": "Validation errors"
}
```

100 And match response == structureError400

0

[bookingOperations](#) [Operations](#) [getAllBookingByIdKO](#)

ms: 10

Scenario: [7.4:106] **Validate error for invalid id content**

[>> Background:](#)

[5 * def dataRequest = read\('classpath:data/bookingOperations/requestCreateNewBooking.json'\)](#)

2

13:54:56.106 karate.env system property was: null

6 * def structureError400 = read('classpath:data/structureError.json')

1

7 * def structureResponse = read('classpath:data/bookingOperations/structureResponse200.json')

0

8 * def dateValidator = read('classpath:helpers/dateValidator.js')

0

9 Given url apiUrl

0

10 Given path 'booking'

0

11 Given headers { accept: 'application/json', Content-Type: 'application/json' }

0

96 Given path id

0

[97 When method GET](#)

6

13:54:56.111 request:

```
1 > GET http://127.0.0.1:8900/booking/1000
1 > accept: application/json
1 > Content-Type: application/json
1 > Host: 127.0.0.1:8900
1 > Connection: Keep-Alive
1 > User-Agent: Apache-HttpClient/4.5.14 (Java/22.0.1)
1 > Accept-Encoding: gzip,deflate
```

13:54:56.115 response time in milliseconds: 4



44
3

Scenarios

2025-06-12 01:54:56 p. m.

[1.1:28] Validate new booking creation

[2.1:41] Validate mandatory fields in the request

[2.2:42] Validate mandatory fields in the request

[2.3:43] Validate mandatory fields in the request

[2.4:44] Validate mandatory fields in the request

[3.1:55] Validate errors for invalid request content

[3.2:55] Validate errors for invalid request content

[3.3:55] Validate errors for invalid request content

[3.4:55] Validate errors for invalid request content

[3.5:55] Validate errors for invalid request content

[3.6:55] Validate errors for invalid request content

[3.7:55] Validate errors for invalid request content

[3.8:55] Validate errors for invalid request content

[3.9:55] Validate errors for invalid request content

[3.10:55] Validate errors for invalid request content

[3.11:55] Validate errors for invalid request content

[3.12:55] Validate errors for invalid request content

1 < 404	
1 < X-Powered-By: Express	
1 < Content-Type: application/json; charset=utf-8	
1 < Content-Length: 51	
1 < ETag: W/"33-bprRhTJQi203tPpZqBsyDAFIu50"	
1 < Date: Thu, 12 Jun 2025 18:54:56 GMT	
1 < Connection: keep-alive	
1 < Keep-Alive: timeout=5	
{ "errors": null, "message": "No booking with id 1000" }	
98 Then status 404	0
99 * print response	0
13:54:56.116 [print] {	
"errors": null,	
"message": "No booking with id 1000"	
}	
100 And match response == structureError400	0
13:54:56.117 src/test/java/bookingOperations/createBooking.feature:100	
And match response == structureError400	
match failed: EQUALS	
\$ not equal match failed for name: 'errors' (MAP:MAP)	
{ "errors": null, "message": "No booking with id 1000" }	
{ "errors": [{ "message": "#string", "location":	
{ "in": "#string", "name": "#string", "docPath": "#string", "path": "#string" } }], "message": "#string" }	
\$.errors data types don't match (NULL:LIST)	
null	
[{ "message": "#string", "location":	
{ "in": "#string", "name": "#string", "docPath": "#string", "path": "#string" } }]	
src/test/java/bookingOperations/createBooking.feature:100	