

API Challenge link - <https://apichallenges.herokuapp.com/gui/challenges>

API documentation link - <https://apichallenges.herokuapp.com/docs>

The solved challenge has a green color.

Challenge solutions:

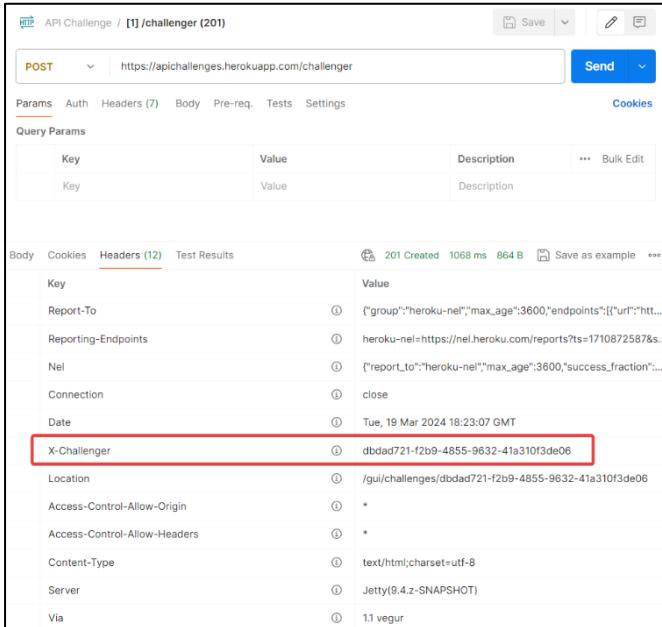
1. Create a new challenger session

Getting Started

If you want to track your challenge progress, in multi-user mode then you need to solve the challenges in this section to generate a unique ID that we can associate your progress with.

ID	Challenge	Done	Description
01	POST / challenger (201)	false	<p>Issue a POST request on the '/challenger' end point, with no body, to create a new challenger session. Use the generated X-CHALLENGER header in future requests to track challenge completion.</p> <p>► Hints ► Solution</p>

Solution:



The screenshot shows the API Challenge tool interface. A POST request is being made to the URL <https://apichallenges.herokuapp.com/challenger>. The 'Headers' tab is selected, showing 12 headers. One of the headers, 'X-Challenger', is highlighted with a red box. The response status is 201 Created, and the response body contains JSON data related to a challenge session.

01	POST /challenger (201)	true	<p>Issue a POST request on the '/challenger' end point, with no body, to create a new challenger session. Use the generated X-CHALLENGER header in future requests to track challenge completion.</p> <p>► Hints ► Solution</p>
----	------------------------	------	---

2. Retrieve a list of challenges

First Real Challenge

For your first challenge, get the list of challenges. You'll be able to use this to see your progress in your API Client, as well as using the GUI.

ID	Challenge	Done	Description
02	GET /challenges (200)	false	Issue a GET request on the `/challenges` end point ► Solution

Solution:

The screenshot shows a Postman request for `https://apichallenges.herokuapp.com/challenges`. The request method is `GET`. The response status is `200 OK` with a response time of 839 ms and a size of 12.11 KB. The JSON response body contains a single challenge object:

```
1 {
2   "challenges": [
3     {
4       "id": 1,
5       "name": "POST /challenger (201)",
6       "description": "Issue a POST request on the `/challenger` end point, with no body, to create a new challenger session. Use the generated X-CHALLENGER header in future requests to track the session."}]
```

02	GET /challenges (200)	true	Issue a GET request on the `/challenges` end point ► Solution
----	-----------------------	------	--

3. Retrieve a list of 'todos'

GET Challenges

To retrieve, or read information from an API we issue GET requests. This section has a bunch of GET request challenges to try out.

ID	Challenge	Done	Description
03	GET /todos (200)	false	Issue a GET request on the `/todos` end point ► Solution

Solution:

The screenshot shows a Postman request for `https://apichallenges.herokuapp.com/todos`. The request method is `GET`. The response status is `200 OK` with a response time of 801 ms and a size of 147 KB. The JSON response body contains two todo items:

```
1 [
2   {
3     "id": 2,
4     "title": "file paperwork",
5     "doneStatus": false,
6     "description": ""
7   },
8   {
9     "id": 10,
10    "title": "install webcam",
11    "doneStatus": false,
12    "description": ""
13  }
]
```

03	GET /todos (200)	true	Issue a GET request on the `/todos` end point ► Solution
----	------------------	------	---

4. Retrieve the 404 status-code

04	GET /todo (404) not plural	false	Issue a GET request on the `/todo` end point should 404 because nouns should be plural ► Solution
----	----------------------------	-------	--

Solution:

The screenshot shows the Postman interface with a failed GET request to `https://apichallenges.herokuapp.com/todo`. The response status is `404 Not Found`. The request headers include `Postman-Token`, `Host`, `User-Agent`, `Accept`, `Accept-Encoding`, `Connection`, and `Content-Type`.

04	GET /todo (404) not plural	true	Issue a GET request on the `/todo` end point should 404 because nouns should be plural ► Solution
----	----------------------------	------	--

5. Retrieve specific 'todo' with status code 200

05	GET /todos/{id} (200)	false	Issue a GET request on the `/todos/{id}` end point to return a specific todo ► Hints ► Solution
----	-----------------------	-------	---

Solution:

The screenshot shows the Postman interface with a successful GET request to `https://apichallenges.herokuapp.com/todos/1`. The response status is `200 OK`. The request headers include `X-CHALLENGER`, `Postman-Token`, `Host`, `User-Agent`, `Accept`, and `Accept-Encoding`. The JSON response body is:

```
1 {  
2   "todos": [  
3     {  
4       "id": 1,  
5       "title": "scan paperwork",  
6       "doneStatus": false,  
7       "description": ""  
8     }  
9   ]  
10 }
```

05	GET /todos/{id} (200)	true	Issue a GET request on the `/todos/{id}` end point to return a specific todo ► Hints ► Solution
----	-----------------------	------	---

6. Retrieve 404 status code when {todo id} doesn't exist

06	GET /todos/{id} (404)	false	Issue a GET request on the `/todos/{id}` end point for a todo that does not exist ► Hints ► Solution
----	-----------------------	-------	--

Solution:

The screenshot shows a Postman request configuration for a GET request to `https://apichallenges.herokuapp.com/todos/10000`. The response status is 404 Not Found, and the body contains an error message: `{"errorMessages": ["Could not find an instance with todos/10000"]}`.

06	GET /todos/{id} (404)	true	Issue a GET request on the `/todos/{id}` end point for a todo that does not exist ► Hints ► Solution
----	-----------------------	------	--

7. Get a filtered list of todos

07	GET /todos (200) ?filter	false	Issue a GET request on the `/todos` end point with a query filter to get only todos which are 'done'. There must exist both 'done' and 'not done' todos, to pass this challenge. ► Hints ► Solution
----	--------------------------	-------	---

Solution:

The screenshot shows a Postman request configuration for a GET request to `https://apichallenges.herokuapp.com/todos?doneStatus=true`. The response status is 200 OK, and the body is an empty array: `[{"todos": []}]`.

07	GET /todos (200) ?filter	true	Issue a GET request on the `/todos` end point with a query filter to get only todos which are 'done'. There must exist both 'done' and 'not done' todos, to pass this challenge. ► Hints ► Solution
----	--------------------------	------	---

8. Create a HEAD request for todos

HEAD Challenges

A HEAD request, is like a GET request, but only returns the headers and status code.

ID	Challenge	Done	Description
08	HEAD /todos (200)	false	Issue a HEAD request on the '/todos' end point ► Solution

Solution:

The screenshot shows the Postman interface for a HEAD request to the '/todos' endpoint. The 'Headers' tab is selected, showing various standard headers like X-CHALLENGER, Postman-Token, Host, User-Agent, Accept, Accept-Encoding, and Connection. The 'Body' tab shows a 200 OK response with a size of 782 B. The 'Pretty' tab displays the JSON response, which is empty.

08	HEAD /todos (200)	true	Issue a HEAD request on the '/todos' end point ► Solution
----	-------------------	------	--

9. Create a new todo record

Creation Challenges with POST

A POST request can be used to create and update data, these challenges are to 'create' data. As a Hint, if you are not sure what the message body should be, try copying in the response from the associated GET request, and amending it.

ID	Challenge	Done	Description
09	POST /todos (201)	false	Issue a POST request to successfully create a todo ► Solution

Solution:

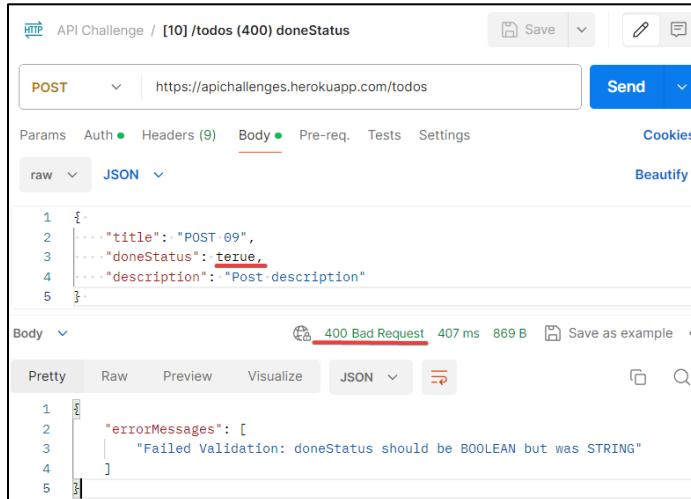
The screenshot shows the Postman interface for a POST request to the '/todos' endpoint. The 'Headers' tab is selected, showing X-CHALLENGER, Postman-Token, Content-Type (application/json), Content-Length, Host, and User-Agent. The 'Body' tab shows a 201 Created response with a size of 875 B. The 'Pretty' tab displays the JSON response, which includes an 'id' field (53), 'title' ('POST 09'), 'doneStatus' (true), and 'description' ('Post description').

09	POST /todos (201)	true	Issue a POST request to successfully create a todo ► Solution
----	-------------------	------	--

10. Fail validation on the 'doneStatus' field

10	POST /todos (400) doneStatus	false	Issue a POST request to create a todo but fail validation on the 'doneStatus' field ► Solution
----	---------------------------------	-------	---

Solution:



The screenshot shows a Postman interface with the following details:

- Method: POST
- URL: https://apichallenges.herokuapp.com/todos
- Body (JSON):

```
1 {  
2   ... "title": "POST_09",  
3   ... "doneStatus": terue,  
4   ... "description": "Post description"  
5 }
```

- Response status: 400 Bad Request
- ResponseBody (Pretty):

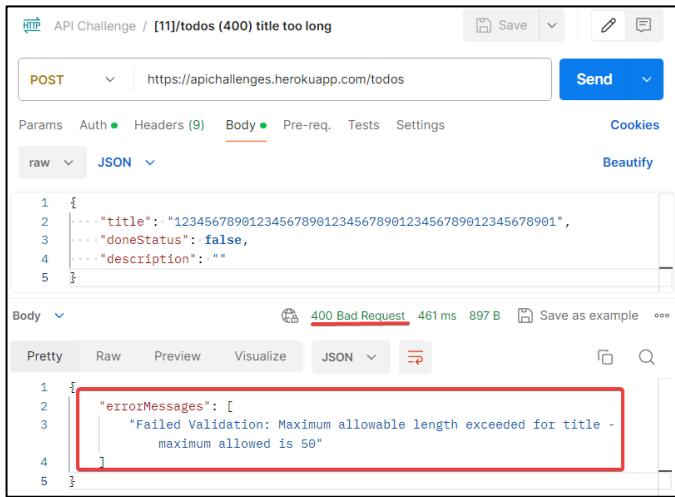
```
1 {  
2   "errorMessages": [  
3     "Failed Validation: doneStatus should be BOOLEAN but was STRING"  
4   ]  
5 }
```

10	POST /todos (400) doneStatus	true	Issue a POST request to create a todo but fail validation on the 'doneStatus' field ► Solution
----	---------------------------------	------	---

11. Create a todo record with fail validation on the 'title' field:

11	POST /todos (400) title too long	false	Issue a POST request to create a todo but fail length validation on the 'title' field because your title exceeds maximum allowable characters.
----	-------------------------------------	-------	--

Solution:



The screenshot shows a Postman interface with the following details:

- Method: POST
- URL: https://apichallenges.herokuapp.com/todos
- Body (JSON):

```
1 {  
2   ... "title": "123456789012345678901234567890123456789012345678901",  
3   ... "doneStatus": false,  
4   ... "description": "  
5 }
```

- Response status: 400 Bad Request
- ResponseBody (Pretty):

```
1 {  
2   "errorMessages": [  
3     "Failed Validation: Maximum allowable length exceeded for title -  
4       maximum allowed is 50"  
5   ]  
6 }
```

11	POST /todos (400) title too long	true	Issue a POST request to create a todo but fail length validation on the 'title' field because your title exceeds maximum allowable characters.
----	----------------------------------	------	--

12. Create a todo record with fail validation on the 'title' field:

12	POST /todos (400) description too long	false	Issue a POST request to create a todo but fail length validation on the `description` because your description exceeds maximum allowable characters.
----	---	-------	--

Solution:

The screenshot shows a Postman interface with a red box highlighting the error message in the response body:

```

1 {
2     "errorMessages": [
3         "Failed Validation: Maximum allowable length exceeded for description
4             - maximum allowed is 200"
5     ]
}

```

12	POST /todos (400) description too long	true	Issue a POST request to create a todo but fail length validation on the `description` because your description exceeds maximum allowable characters.
----	---	------	--

13. Create a todo record with max length of title and description fields

13	POST /todos (201) max out content	false	Issue a POST request to create a todo with maximum length title and description fields. ► Hints
----	--------------------------------------	-------	--

Solution:

The screenshot shows a Postman interface with a red box highlighting the error message in the response body:

```

1 {
2     "id": 13,
3     "title": "12345678901234567890123456789012345678901234567890",
4     "doneStatus": false,
5     "description": "text text text text text text text text text text
6         text text
7         text text
8         text text
9         text."
}

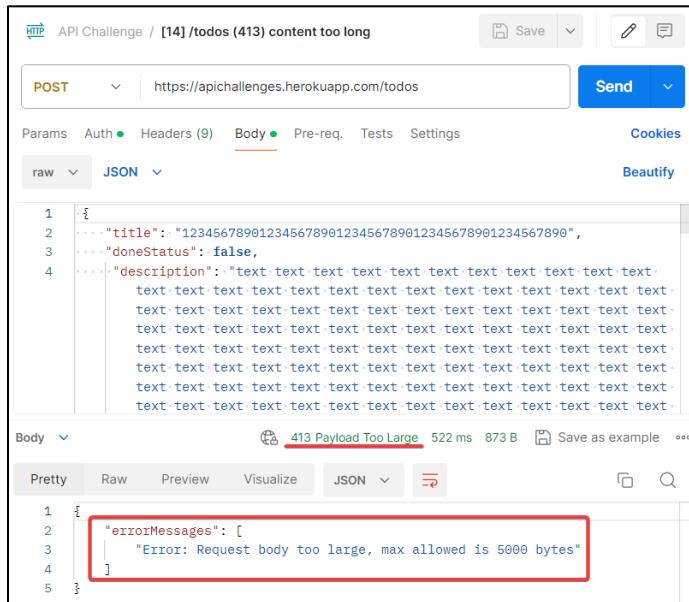
```

13	POST /todos (201) max out content	true	Issue a POST request to create a todo with maximum length title and description fields. ► Hints
----	-----------------------------------	------	--

14. Create todo with long content

14	POST /todos (413) content too long	false	Issue a POST request to create a todo but fail payload length validation on the `description` because your whole payload exceeds maximum allowable 5000 characters. ► Hints
----	------------------------------------	-------	--

Solution:



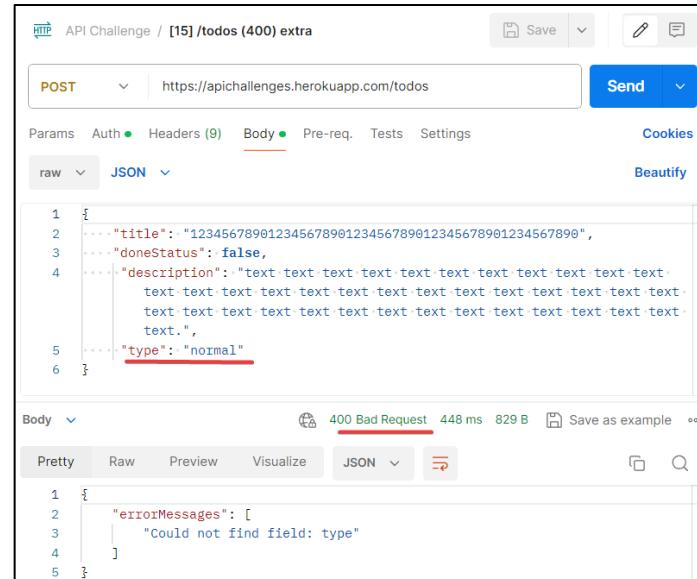
The screenshot shows the Postman interface with a POST request to `https://apichallenges.herokuapp.com/todos`. The JSON body contains a long string of text for the 'description' field. The response status is `413 Payload Too Large`, and the error message is `"Error: Request body too large, max allowed is 5000 bytes"`.

14	POST /todos (413) content too long	true	Issue a POST request to create a todo but fail payload length validation on the `description` because your whole payload exceeds maximum allowable 5000 characters. ► Hints
----	------------------------------------	------	--

15. Create a todo record using an unrecognized field in the payload

15	POST /todos (400) extra	false	Issue a POST request to create a todo but fail validation because your payload contains an unrecognised field. ► Hints
----	-------------------------	-------	---

Solution:



The screenshot shows the Postman interface with a POST request to `https://apichallenges.herokuapp.com/todos`. The JSON body contains a long string of text for the 'description' field and an additional field `"type": "normal"`. The response status is `400 Bad Request`, and the error message is `"Could not find field: type"`.

15	POST /todos (400) extra	true	Issue a POST request to create a todo but fail validation because your payload contains an unrecognised field. ► Hints
----	-------------------------	------	---

16. Use a PUT request for unsuccessfully creating a todo record

Creation Challenges with PUT

A PUT request can often be used to create and update data. The todo application we are using has automatically generated IDs, so you cannot use PUT to create. As a Hint, if you are not sure what the message body should be, try copying in the response from the associated GET request, and amending it.

ID	Challenge	Done	Description
16	PUT /todos/{id} (400)	false	Issue a PUT request to unsuccessfully create a todo

Solution:

The screenshot shows the Postman interface with a PUT request to `https://apichallenges.herokuapp.com/todos/156`. The request body is a JSON object with various fields. The response status is `400 Bad Request` with a duration of `453 ms` and a size of `856 B`. The response body is a JSON object with an `errorMessages` array containing the message `'Cannot create todo with PUT due to Auto fields id'`.

ID	Challenge	Done	Description
16	PUT /todos/{id} (400)	true	Issue a PUT request to unsuccessfully create a todo

17. Update todo record

Update Challenges with POST

Use a POST request to amend something that already exists. These are 'partial' content updates so you usually don't need to have all details of the entity in the request, e.g. you could just update a title, or a description, or a status

ID	Challenge	Done	Description
17	POST /todos/{id} (200)	false	Issue a POST request to successfully update a todo ► Hints ► Solution

Solution:

The screenshot shows the Postman interface with a POST request to `https://apichallenges.herokuapp.com/todos/1`. The request body is a JSON object with fields `title`, `doneStatus`, and `description`. The response status is `200 OK` with a duration of `1325 ms` and a size of `863 B`. The response body is a JSON object with the updated todo record.

17	POST /todos/{id} (200)	true	Issue a POST request to successfully update a todo ► Hints ► Solution
----	------------------------	------	---

18. Update a todo record which doesn't exist

18	POST /todos/{id} (404)	false	Issue a POST request for a todo which does not exist. Expect to receive a 404 response. ► Hints
----	------------------------	-------	--

Solution:

18	POST /todos/{id} (404)	true	Issue a POST request for a todo which does not exist. Expect to receive a 404 response. ► Hints
----	------------------------	------	--

19. Update an existing todo record with a complete payload

Update Challenges with PUT

A PUT request can be used to amend data. REST Put requests are idempotent, they provide the same result each time.

ID	Challenge	Done	Description
19	PUT /todos/{id} full (200)	false	Issue a PUT request to update an existing todo with a complete payload i.e. title, description and donestatus.

Solution:

19	PUT /todos/{id} full (200)	true	Issue a PUT request to update an existing todo with a complete payload i.e. title, description and donestatus.
----	----------------------------	------	--

20. Update mandatory items for an existing todo record

20	PUT /todos/{id} partial (200)	false	Issue a PUT request to update an existing todo with just mandatory items in payload i.e. title.
----	-------------------------------	-------	---

Solution:

The screenshot shows the Postman interface with a PUT request to `https://apichallenges.herokuapp.com/todos/1`. The body contains a JSON object with only the `"title"` field set to `"file paperworking (UPDATED!)"`. The response status is 200 OK, and the response body is a full JSON object including `"id": 1`, `"title": "file paperworking (UPDATED!)"`, `"doneStatus": false`, and `"description": ""`.

20	PUT /todos/{id} partial (200)	true	Issue a PUT request to update an existing todo with just mandatory items in payload i.e. title.
----	-------------------------------	------	---

21. Fail to update an existing todo record

21	PUT /todos/{id} no title (400)	false	Issue a PUT request to fail to update an existing todo because title is missing in payload. ► Hints
----	--------------------------------	-------	--

Solution:

The screenshot shows the Postman interface with a PUT request to `https://apichallenges.herokuapp.com/todos/5`. The body is a JSON object with `"doneStatus": false` and `"description": ""`. The response status is 400 Bad Request, and the response body is a JSON object with an `"errorMessages"` array containing the message `"title : field is mandatory"`.

21	PUT /todos/{id} no title (400)	true	Issue a PUT request to fail to update an existing todo because title is missing in payload. ► Hints
----	--------------------------------	------	--

22. Fail to update an existing todo

22	PUT /todos/{id} no amend id (400)	false	Issue a PUT request to fail to update an existing todo because id different in payload. ► Hints
----	-----------------------------------	-------	--

Solution:

The screenshot shows the Postman interface with a PUT request to `https://apichallenges.herokuapp.com/todos/5`. The request body is JSON with an ID of 22. The response is a 400 Bad Request with the message "Can not amend id from 5 to 22".

22	PUT /todos/{id} no amend id (400)	true	Issue a PUT request to fail to update an existing todo because id different in payload. ► Hints
----	-----------------------------------	------	--

23. Delete the todo record

DELETE Challenges

Use a DELETE request to delete an entity. Since this is an extreme request, normally you have to be logged in or authenticated, but we wanted to make life easier for you so we cover authentication later. Anyone can delete To Do items without authentication in this system.

ID	Challenge	Done	Description
23	DELETE /todos/{id} (200)	false	Issue a DELETE request to successfully delete a todo ► Hints ► Solution

Solution:

The screenshot shows the Postman interface with a DELETE request to `https://apichallenges.herokuapp.com/todos/10`. The request includes auto-generated headers: X-CHALLENGER, Postman-Token, Host, User-Agent, Accept, and Accept-Encoding. The response is a 200 OK.

23	DELETE /todos/{id} (200)	true	Issue a DELETE request to successfully delete a todo ► Hints ► Solution
----	--------------------------	------	---

24. Get an 'Allow' header in the OPTION request

OPTIONS Challenges

Use an OPTIONS verb and check the 'Allow' header, this will show you what verbs are allowed to be used on an endpoint. When you test APIs it is worth checking to see if all the verbs listed are allowed or not.

ID	Challenge	Done	Description
24	OPTIONS / todos (200)	false	<p>Issue an OPTIONS request on the `/todos` end point. You might want to manually check the 'Allow' header in the response is as expected.</p> <p>► Solution</p>

Solution:

The screenshot shows the Postman interface for challenge 24. The URL is https://apichallenges.herokuapp.com/todos (200). The Headers tab is active, displaying several pre-defined headers. In the Response section, the Allow header is highlighted with a red box, showing its value as OPTIONS, GET, HEAD, POST.

24	OPTIONS / todos (200)	true	<p>Issue an OPTIONS request on the `/todos` end point. You might want to manually check the 'Allow' header in the response is as expected.</p> <p>► Solution</p>
----	-----------------------	------	--

25. Receive a result in XML format

Accept Challenges

The 'Accept' header, tells the server what format you want the response to be in. By changing the 'Accept' header you can specify JSON or XML.

ID	Challenge	Done	Description
25	GET /todos (200) XML	false	Issue a GET request on the '/todos' end point with an 'Accept' header of 'application/xml' to receive results in XML format ► Solution

Solution:

The screenshot shows a Postman request configuration for challenge 25. The URL is https://apichallenges.herokuapp.com/todos. The 'Headers' tab is selected, showing the following headers:

- User-Agent: PostmanRuntime/7.37.0
- Accept: application/xml (highlighted with a red box)
- Accept-Encoding: gzip, deflate, br
- Connection: keep-alive

The response section shows a 200 OK status with a response time of 479 ms and a size of 1.69 KB. The response body is displayed in XML format, showing a single todo item with id 2 and title "file paperwork".

25	GET /todos (200) XML	true	Issue a GET request on the '/todos' end point with an 'Accept' header of 'application/xml' to receive results in XML format ► Solution
----	----------------------	------	---

26. Receive a result in JSON format

26	GET /todos (200) JSON	false	Issue a GET request on the '/todos' end point with an 'Accept' header of 'application/json' to receive results in JSON format ► Solution
----	-----------------------	-------	---

Solution:

The screenshot shows a Postman request configuration for challenge 26. The URL is https://apichallenges.herokuapp.com/todos. The 'Headers' tab is selected, showing the following headers:

- Host: <calculated when request is...>
- User-Agent: PostmanRuntime/7.37.0
- Accept: application/json (highlighted with a red box)
- Accept-Encoding: gzip, deflate, br
- Connection: keep-alive

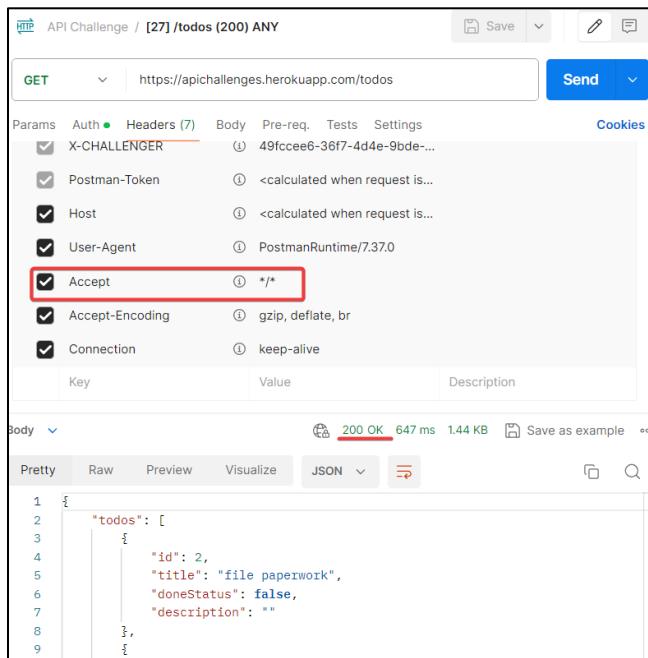
The response section shows a 200 OK status with a response time of 691 ms and a size of 1.44 KB. The response body is displayed in JSON format, showing a single todo item with id 2 and title "file paperwork".

26	GET /todos (200) JSON	true	Issue a GET request on the '/todos' end point with an 'Accept' header of 'application/json' to receive results in JSON format ► Solution
----	-----------------------	------	---

27. Receive a result in default JSON format

27	GET /todos (200) ANY	false	Issue a GET request on the `/todos` end point with an `Accept` header of `/*` to receive results in default JSON format ► Solution
----	-------------------------	-------	---

Solution:



The screenshot shows a Postman request configuration for step 27. The URL is `https://apichallenges.herokuapp.com/todos`. In the Headers section, the `Accept` header is set to `/*`. The response body is displayed in JSON format, showing a single todo item with id 2, title "file paperwork", done status false, and no description.

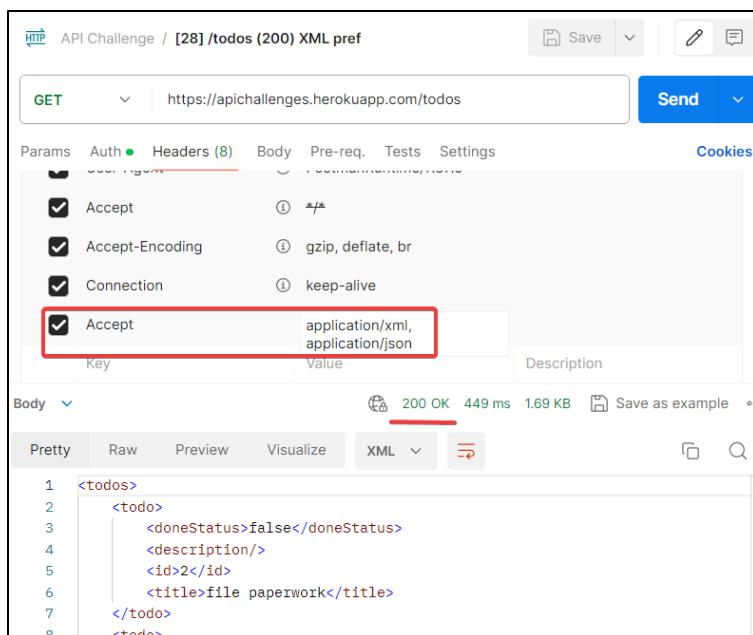
```
1 {
2     "todos": [
3         {
4             "id": 2,
5             "title": "file paperwork",
6             "doneStatus": false,
7             "description": ""
8         }
9     ]
}
```

27	GET /todos (200) ANY	true	Issue a GET request on the `/todos` end point with an `Accept` header of `/*` to receive results in default JSON format ► Solution
----	-------------------------	------	---

28. Receive a result in the XML format

28	GET /todos (200) XML pref	false	Issue a GET request on the `/todos` end point with an `Accept` header of `application/xml, application/json` to receive results in the preferred XML format ► Solution
----	------------------------------	-------	---

Solution:



The screenshot shows a Postman request configuration for step 28. The URL is `https://apichallenges.herokuapp.com/todos`. In the Headers section, the `Accept` header is set to `application/xml, application/json`. The response body is displayed in XML format, representing the same todo item as the JSON response.

```
1 <todos>
2     <todo>
3         <doneStatus>false</doneStatus>
4         <description/>
5         <id>2</id>
6         <title>file paperwork</title>
7     </todo>
8     <todo>
```

28	GET /todos (200) XML pref	true	Issue a GET request on the `/todos` end point with an `Accept` header of `application/xml, application/json` to receive results in the preferred XML format ► Solution
----	------------------------------	------	---

29. Receive results in default JSON format

29	GET /todos (200) no accept	false	Issue a GET request on the `/todos` end point with no `Accept` header present in the message to receive results in default JSON format ► Solution
----	-------------------------------	-------	--

Solution:

The screenshot shows the Postman interface with a GET request to `https://apichallenges.herokuapp.com/todos`. In the Headers tab, the `Accept` header is explicitly unselected (), while other headers like `X-CHALLENGER`, `Postman-Token`, `Host`, and `User-Agent` are selected. The response status is 200 OK, and the body is displayed in JSON format:

```

1 {
2   "todos": [
3     {
4       "id": 2,
5       "title": "file paperwork",
6       "doneStatus": false,
7       "description": ""
8     }
9   ]

```

29	GET /todos (200) no accept	true	Issue a GET request on the `/todos` end point with no `Accept` header present in the message to receive results in default JSON format ► Solution
----	-------------------------------	------	--

30. Create a GET request with an ‘Accept’ header ‘application/gzip’

30	GET /todos (406)	false	Issue a GET request on the `/todos` end point with an `Accept` header `application/gzip` to receive 406 'NOT ACCEPTABLE' status code ► Solution
----	---------------------	-------	--

Solution:

The screenshot shows the Postman interface with a GET request to `https://apichallenges.herokuapp.com/todos`. In the Headers tab, the `Accept` header is set to `application/gzip` (), while other headers like `Postman-Token`, `Host`, `User-Agent`, `Accept-Encoding`, and `Connection` are selected. The response status is 406 Not Acceptable, and the body is displayed in JSON format:

```

1 {
2   "errorMessages": [
3     "Unrecognised Accept Type"
4   ]
5 }

```

30	GET /todos (406)	true	Issue a GET request on the `/todos` end point with an `Accept` header `application/gzip` to receive 406 'NOT ACCEPTABLE' status code ► Solution
----	---------------------	------	--

31. Create a todo record using Content-Type ‘application/xml’

Content-Type Challenges			
ID	Challenge	Done	Description
31	POST /todos XML	false	<p>Issue a POST request on the `/todos` end point to create a todo using Content-Type `application/xml`, and Accepting only XML ie. Accept header of `application/xml`</p> <p>► Solution</p>

Solution:

The screenshot shows a Postman request configuration for challenge 31. The method is POST, the URL is <https://apichallenges.herokuapp.com/todos>, and the headers section is selected. It contains the following entries:

- Connection: keep-alive
- Accept: application/xml (highlighted with a red box)
- Content-Type: application/xml (highlighted with a red box)

The body section is also selected, showing a pretty-printed XML response:

```

1 <todo>
2   <doneStatus>true</doneStatus>
3   <description>Lorem Ipsum</description>
4   <id>11</id>
5   <title>file paperwork</title>
6 </todo>

```

31	POST /todos XML	true	<p>Issue a POST request on the `/todos` end point to create a todo using Content-Type `application/xml`, and Accepting only XML ie. Accept header of `application/xml`</p> <p>► Solution</p>
----	-----------------	------	--

32. Create POST request using Content-Type ‘application/json’

32	POST /todos JSON	false	<p>Issue a POST request on the `/todos` end point to create a todo using Content-Type `application/json`, and Accepting only JSON ie. Accept header of `application/json`</p> <p>► Solution</p>
----	------------------	-------	---

Solution:

The screenshot shows a Postman request configuration for challenge 32. The method is POST, the URL is <https://apichallenges.herokuapp.com/todos>, and the headers section is selected. It contains the following entries:

- Content-Type: application/json (highlighted with a red box)
- Content-Length: <calculated when request is...
- Host: <calculated when request is...
- User-Agent: PostmanRuntime/7.37.0
- Accept: */*
- Accept-Encoding: gzip, deflate, br
- Connection: keep-alive
- Accept: application/json (highlighted with a red box)

The body section is also selected, showing a pretty-printed JSON response:

```

1 {
2   "id": 12,
3   "title": "Lorem Ipsum",
4   "doneStatus": true,
5   "description": "Lorem Ipsum"
6 }

```

32	POST /todos JSON	true	<p>Issue a POST request on the `/todos` end point to create a todo using Content-Type `application/json`, and Accepting only JSON ie. Accept header of `application/json`</p> <p>► Solution</p>
----	------------------	------	---

33. Create a request with an unsupported content type

33	POST /todos (415)	false	Issue a POST request on the `/todos` end point with an unsupported content type to generate a 415 status code ► Solution
----	----------------------	-------	---

Solution:

The screenshot shows the Postman interface with a POST request to the URL `https://apichallenges.herokuapp.com/todos`. In the Headers tab, the `Content-Type` header is explicitly set to `unsupported`, which is highlighted with a red box. The response status is `415 Unsupported Media Type`, and the response body contains the JSON object `{"errorMessages": ["Unsupported Content Type - unsupported"]}`.

33	POST /todos (415)	true	Issue a POST request on the `/todos` end point with an unsupported content type to generate a 415 status code ► Solution
----	----------------------	------	---

34. Return the progress data payload for current challenger GUID

Fancy a Break? Restore your session

Your challenge progress can be saved, and as long as you remember your challenger ID you can restore it. Leaving a challenger idle in the system for more than 10 minutes will remove the challenger from memory. Challenger status and the todos database can be saved to, and restored from, the browser localStorage.

ID	Challenge	Done	Description
34	GET /challenger/guid (existing X-CHALLENGER)	false	Issue a GET request on the `/challenger/{guid}` end point, with an existing challenger GUID. This will return the progress data payload that can be used to later restore your progress to this status. ► Hints

Solution:

The screenshot shows the Postman interface with a GET request to the URL `https://apichallenges.herokuapp.com/challenger/49fccee6-36f7-4d4e-9bde-75b4c2790f59`. In the Headers tab, the `X-CHALLENGER` header is set to the value `49fccee6-36f7-4d4e-9bde-75b4c2790f59`. The response status is `200 OK`, and the response body is a JSON object containing various challenge status flags such as `xAuthToken`, `xChallenger`, `secretNote`, and `challengeStatus`.

34	GET /challenger/guid (existing X-CHALLENGER)	true	Issue a GET request on the `/challenger/{guid}` end point, with an existing challenger GUID. This will return the progress data payload that can be used to later restore your progress to this status. ► Hints
----	---	------	--

35. Restore the challenger's progress

35	PUT /challenger/guid RESTORE	false	Issue a PUT request on the `/challenger/{guid}` end point, with an existing challenger GUID to restore that challenger's progress into memory. ▶ Hints ▶ Solution
----	---------------------------------	-------	---

Solution:

HTTP API Challenge / [35] /challenger/guid RESTORE

PUT https://apichallenges.herokuapp.com/challenger/49fccee6-36f7-4d4e-9bde-75b4c2790f59

Send

Params Auth Headers (9) Body Pre-req. Tests Settings Cookies Beautify

raw JSON

```
1 {
2   ...
3   "xAuthToken": "cdadba06-cb86-4ba5-a42f-5b68f4211e39",
4   "xChallenger": "49fccee6-36f7-4d4e-9bde-75b4c2790f59",
5   "secretNote": "",
6   "challengeStatus": {
7     ...
8     "POST_TODOS_INVALID_EXTRA_FIELD": false,
9     "GET_HEARTBEAT_204": false
10    ...
11  }
12}
```

Body 200 OK 476 ms 2.66 KB Save as example

Pretty Raw Preview Visualize JSON

```
1 {
2   "xAuthToken": "cdadba06-cb86-4ba5-a42f-5b68f4211e39",
3   "xChallenger": "49fccee6-36f7-4d4e-9bde-75b4c2790f59",
4   "secretNote": "",
5   "challengeStatus": {
6     ...
7     "GET_RESTORABLE_TODOS": false,
8     "GET_ACCEPT_ANY_DEFAULT_JSON": false,
9     "CREATE_SECRET_TOKEN_201": false,
10    "PUT_RESTORABLE_CHALLENGER_PROGRESS_STATUS": false,
11    "OVERWRITE_PATCH_HEARTBEAT_500": false,
12    "POST_TODOS_TOO_LONG_PAYLOAD_SIZE": false,
13  }
14}
```

35	PUT /challenger/guid RESTORE	true	Issue a PUT request on the `/challenger/{guid}` end point, with an existing challenger GUID to restore that challenger's progress into memory. ▶ Hints ▶ Solution
----	---------------------------------	------	---

36. Restore challenger's progress, when challenger GUID is not currently in memory

36	PUT /challenger/guid CREATE	false	Issue a PUT request on the `/challenger/{guid}` end point, with a challenger GUID not currently in memory to restore that challenger's progress into memory. ▶ Hints ▶ Solution
----	--------------------------------	-------	---

Solution:

HTTP API Challenge / [36] /challenger/guid CREATE

PUT https://apichallenges.herokuapp.com/challenger/49fccee6-36f7-4d4e-9bde-75b4c2790f59

Send

Params Auth Headers (9) Body Pre-req. Tests Settings Cookies Beautify

raw JSON

```
1 {
2   ...
3   "xAuthToken": "2f3766c1-0b84-4264-bddc-90f63dea8ecd",
4   "xChallenger": "49fccee6-36f7-4d4e-9bde-75b4c2790f59",
5   "secretNote": "",
6   "challengeStatus": {
7     ...
8     "POST_TODOS_INVALID_EXTRA_FIELD": false,
9     "GET_HEARTBEAT_204": false,
10    ...
11    "POST_SECRET_NOTE_403": false,
12    ...
13    "GET_ACCEPT_XML_PREFERRED": false,
14    ...
15    "POST_MAX_OUT_TITLE_DESCRIPTION_LENGTH": false,
16    ...
17    "POST_TODOS_TOO_LONG_PAYLOAD_SIZE": false,
18    ...
19    "GET_ACCEPT_ANY_DEFAULT_JSON": true
20  }
21}
```

Body 200 OK 488 ms 2.66 KB Save as example

Pretty Raw Preview Visualize JSON

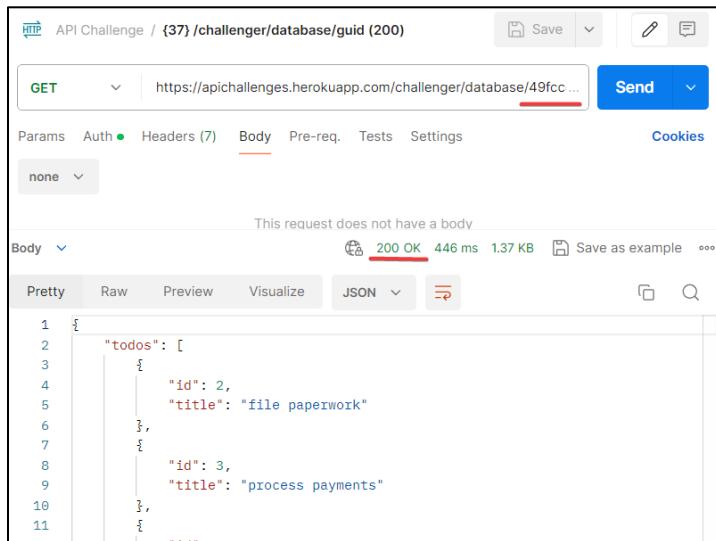
```
1 {
2   "xAuthToken": "2f3766c1-0b84-4264-bddc-90f63dea8ecd",
3   "xChallenger": "49fccee6-36f7-4d4e-9bde-75b4c2790f59",
4   "secretNote": "",
5   "challengeStatus": {
6     ...
7     "GET_RESTORABLE_TODOS": false,
8     "GET_ACCEPT_ANY_DEFAULT_JSON": true,
9   }
10}
```

36	PUT /challenger/guid CREATE	true	Issue a PUT request on the `/challenger/{guid}` end point, with a challenger GUID not currently in memory to restore that challenger's progress into memory. ▶ Hints ▶ Solution
----	--------------------------------	------	---

37. Retrieve the current todos database for the user.

37	GET /challenger/database/guid (200)	false	Issue a GET request on the `/challenger/database/{guid}` end point, to retrieve the current todos database for the user. You can use this to restore state later. ► Hints
----	-------------------------------------	-------	--

Solution:



The screenshot shows a Postman interface with a GET request to `https://apichallenges.herokuapp.com/challenger/database/49fcc...`. The response status is 200 OK, 446 ms, 1.37 KB. The response body is a JSON object with two todos:

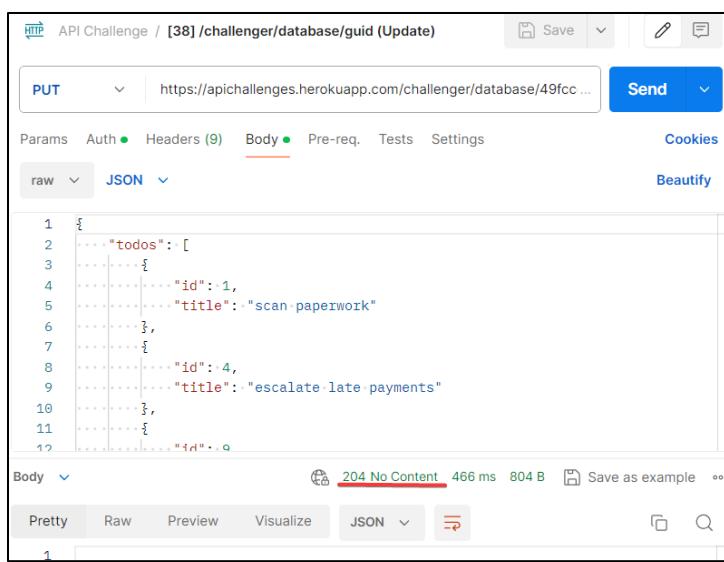
```
1 {  
2   "todos": [  
3     {  
4       "id": 2,  
5       "title": "file paperwork"  
6     },  
7     {  
8       "id": 3,  
9       "title": "process payments"  
10    },  
11  ]  
12 }
```

37	GET /challenger/database/guid (200)	true	Issue a GET request on the `/challenger/database/{guid}` end point, to retrieve the current todos database for the user. You can use this to restore state later. ► Hints
----	-------------------------------------	------	--

38. Restore the todo database in memory

38	PUT /challenger/database/guid (Update)	false	Issue a PUT request on the `/challenger/database/{guid}` end point, with a payload to restore the Todos database in memory. ► Hints ► Solution
----	--	-------	--

Solution:



The screenshot shows a Postman interface with a PUT request to `https://apichallenges.herokuapp.com/challenger/database/49fcc...`. The response status is 204 No Content, 466 ms, 804 B. The response body is empty.

38	PUT /challenger/database/guid (Update)	true	Issue a PUT request on the `/challenger/database/{guid}` end point, with a payload to restore the Todos database in memory. ► Hints ► Solution
----	--	------	--

39. Create a todo using Content-Type ‘application/xml’, but Accept ‘application/json’

Mix Accept and Content-Type Challenges

We can mix the ‘Accept’ and ‘Content-Type’ headers so that we can send JSON but receive XML. These challenges encourage you to explore some combinations.

ID	Challenge	Done	Description
39	POST /todos XML to JSON	false	<p>Issue a POST request on the `/todos` end point to create a todo using Content-Type `application/xml` but Accept `application/json`</p> <p>► Solution</p>

Solution:

The screenshot shows the Postman interface. The URL is https://apichallenges.herokuapp.com/todos. The method is set to POST. In the Headers section, the Content-Type is set to application/xml and the Accept is set to application/json. The Body section is set to XML, containing the XML code for a new todo item. The response shows a 201 Created status with a JSON object containing the created todo's id, title, doneStatus, and description.

39	POST /todos XML to JSON	true	<p>Issue a POST request on the `/todos` end point to create a todo using Content-Type `application/xml` but Accept `application/json`</p> <p>► Solution</p>
----	----------------------------	------	---

40. Create a todo using Content-Type ‘application/json’, but Accept ‘application/xml’

40	POST /todos JSON to XML	false	<p>Issue a POST request on the `/todos` end point to create a todo using Content-Type `application/json` but Accept `application/xml`</p> <p>► Solution</p>
----	----------------------------	-------	---

Solution:

The screenshot shows the Postman interface. The URL is https://apichallenges.herokuapp.com/todos. The method is set to POST. In the Headers section, the Content-Type is set to application/json and the Accept is set to application/xml. The Body section is set to JSON, containing the JSON object for a new todo item. The response shows a 201 Created status with an XML representation of the created todo's id, title, doneStatus, and description.

40	POST /todos JSON to XML	true	<p>Issue a POST request on the `/todos` end point to create a todo using Content-Type `application/json` but Accept `application/xml`</p> <p>► Solution</p>
----	----------------------------	------	---

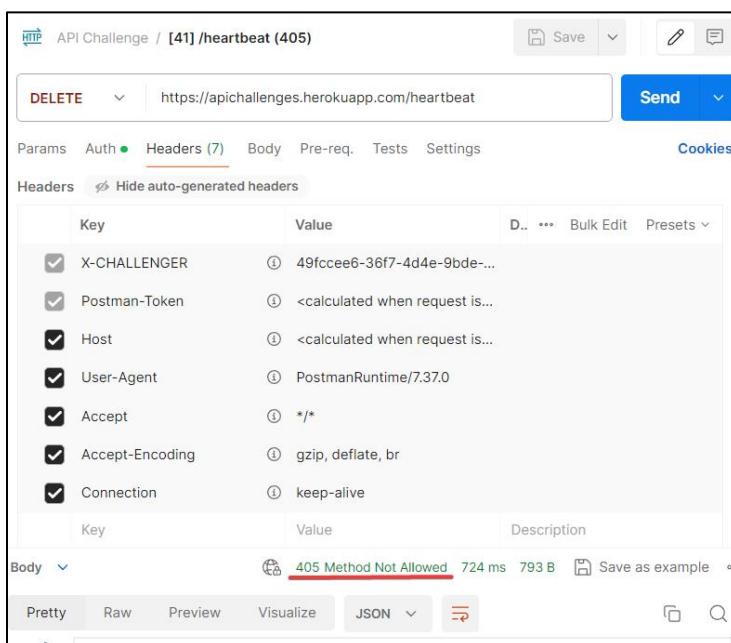
41. Receive 405 status code in DELETE request

Status Code Challenges

Status-codes are essential to understand, so we created some challenges that help you trigger more status codes. Remember to review httpstatuses.com to learn what the status codes mean.

ID	Challenge	Done	Description
41	DELETE / heartbeat (405)	false	Issue a DELETE request on the `/heartbeat` end point and receive 405 (Method Not Allowed) ▶ Solution

Solution:



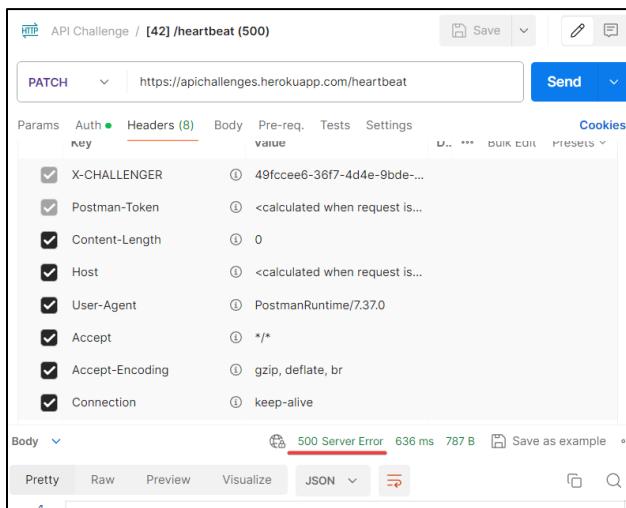
The screenshot shows a Postman request configuration for challenge 41. The method is set to DELETE, and the URL is https://apichallenges.herokuapp.com/heartbeat. The Headers tab shows several auto-generated headers like X-Challenger, Postman-Token, Host, User-Agent, Accept, Accept-Encoding, and Connection. The Body tab is empty. The response status is 405 Method Not Allowed, with a response time of 724 ms and a body size of 793 B.

41	DELETE /heartbeat (405)	true	Issue a DELETE request on the `/heartbeat` end point and receive 405 (Method Not Allowed) ▶ Solution
----	-------------------------	------	---

42. Receive 500 status code in the PATCH request

42	PATCH /heartbeat (500)	false	Issue a PATCH request on the `/heartbeat` end point and receive 500 (internal server error) ▶ Solution
----	------------------------	-------	---

Solution:



The screenshot shows a Postman request configuration for challenge 42. The method is set to PATCH, and the URL is https://apichallenges.herokuapp.com/heartbeat. The Headers tab shows several auto-generated headers like X-Challenger, Postman-Token, Content-Length, Host, User-Agent, Accept, Accept-Encoding, and Connection. The Body tab is empty. The response status is 500 Server Error, with a response time of 636 ms and a body size of 787 B.

42	PATCH /heartbeat (500)	true	Issue a PATCH request on the `/heartbeat` end point and receive 500 (internal server error) ▶ Solution
----	------------------------	------	---

43. Receive 501 status code in TRACE request

43	TRACE /heartbeat (501)	false	Issue a TRACE request on the `/heartbeat` end point and receive 501 (Not Implemented) ► Solution
----	---------------------------	-------	---

Solution:

The screenshot shows a Postman collection named "API Challenge / [43] /heartbeat (501)". A TRACE request is made to the URL `https://apichallenges.herokuapp.com/heartbeat`. The response status is `501 Not Implemented`, with a duration of `568 ms` and a body size of `794 B`. The Headers tab shows the following configuration:

Key	Value	Description
X-CHALLENGER	49fccee6-36f7-4d4e-9bde-...	
Postman-Token	<calculated when request is...	
Host	<calculated when request is...	
User-Agent	PostmanRuntime/7.37.0	
Accept	*/*	
Accept-Encoding	gzip, deflate, br	
Connection	keep-alive	

43	TRACE /heartbeat (501)	true	Issue a TRACE request on the `/heartbeat` end point and receive 501 (Not Implemented) ► Solution
----	------------------------	------	---

44. Receive 204 status code when the server is running

44	GET /heartbeat (204)	false	Issue a GET request on the `/heartbeat` end point and receive 204 when server is running ► Solution
----	-------------------------	-------	--

Solution:

The screenshot shows a Postman collection named "API Challenge / [44] /heartbeat (204)". A GET request is made to the URL `https://apichallenges.herokuapp.com/heartbeat`. The response status is `204 No Content`, with a duration of `664 ms` and a body size of `809 B`. The Headers tab shows the following configuration:

Key	Value	Description
X-CHALLENGER	49fccee6-36f7-4d4e-9bde-...	
Postman-Token	<calculated when request is...	
Host	<calculated when request is...	
User-Agent	PostmanRuntime/7.37.0	
Accept	*/*	
Accept-Encoding	gzip, deflate, br	
Connection	keep-alive	

44	GET /heartbeat (204)	true	Issue a GET request on the `/heartbeat` end point and receive 204 when server is running ► Solution
----	-------------------------	------	--

45. Create a POST request and receive 405 status code

HTTP Method Override Challenges

Some HTTP Clients can not send all verbs e.g. PATCH, DELETE, PUT. Use an X-HTTP-Method-Override header to simulate these with a POST request

ID	Challenge	Done	Description
45	POST /heartbeat as DELETE (405)	false	<p>Issue a POST request on the '/heartbeat' end point and receive 405 when you override the Method Verb to a DELETE</p> <p>► Hints ► Solution</p>

Solution:

The screenshot shows the Postman API Challenge interface. A POST request is being sent to the URL <https://apichallenges.herokuapp.com/heartbeat>. In the Headers tab, the 'X-HTTP-Method-Override' header is explicitly set to 'DELETE'. The response status is 405 Method Not Allowed, indicating that the server does not support the requested method.

45	POST /heartbeat as DELETE (405)	true	Issue a POST request on the '/heartbeat' end point and receive 405 when you override the Method Verb to a DELETE ► Hints ► Solution
----	---------------------------------	------	---

46. Receive 500 status code when override the Method Verb to a PATCH

46	POST /heartbeat as PATCH (500)	false	<p>Issue a POST request on the '/heartbeat' end point and receive 500 when you override the Method Verb to a PATCH</p> <p>► Hints ► Solution</p>
----	--------------------------------	-------	--

Solution:

The screenshot shows the Postman API Challenge interface. A POST request is being sent to the URL <https://apichallenges.herokuapp.com/heartbeat>. In the Headers tab, the 'X-HTTP-Method-Override' header is explicitly set to 'PATCH'. The response status is 500 Server Error, indicating an internal server error.

46	POST /heartbeat as PATCH (500)	true	Issue a POST request on the '/heartbeat' end point and receive 500 when you override the Method Verb to a PATCH ► Hints ► Solution
----	--------------------------------	------	--

47. Receive 500 status code when override the Method Verb to a TRACE

47	POST /heartbeat as Trace (501)	false	Issue a POST request on the `/heartbeat` end point and receive 501 (Not Implemented) when you override the Method Verb to a TRACE ► Hints ► Solution
----	--------------------------------	-------	--

Solution:

The screenshot shows a Postman API challenge for task 47. The request method is POST, and the URL is https://apichallenges.herokuapp.com/heartbeat. In the Headers tab, the X-HTTP-Method-Override header is explicitly set to TRACE. The response status is 501 Not Implemented, which is highlighted in red.

47	POST /heartbeat as Trace (501)	true	Issue a POST request on the `/heartbeat` end point and receive 501 (Not Implemented) when you override the Method Verb to a TRACE ► Hints ► Solution
----	--------------------------------	------	--

48. Receive 401 status code when Basic auth username/password is not admin/password

Authentication Challenges

Authentication is telling the system who you are. In multi-user mode you are already doing that with the X-CHALLENGER header, but we have added an extra level of security on the /secret section. So first Authenticate with Basic Authentication to find out the token to use for authorisation for later challenges.

ID	Challenge	Done	Description
48	POST /secret/token (401)	false	Issue a POST request on the `/secret/token` end point and receive 401 when Basic auth username/password is not admin/password ► Hints ► Solution

Solution:

The screenshot shows a Postman API challenge for task 48. The request method is POST, and the URL is https://apichallenges.herokuapp.com/secret/token. In the Auth dropdown, Basic Auth is selected. The response status is 401 Unauthorized, which is highlighted in red.

48	POST /secret/token (401)	true	Issue a POST request on the `/secret/token` end point and receive 401 when Basic auth username/password is not admin/password ► Hints ► Solution
----	--------------------------	------	--

49. Receive 201 status code when Basic auth has admin/password values for username/password fields

49	POST /secret/token (201)	false	Issue a POST request on the `/secret/token` end point and receive 201 when Basic auth username/password is admin/password ► Hints ► Solution
----	--------------------------	-------	--

Solution:

The screenshot shows the Postman interface with a successful POST request to `https://apichallenges.herokuapp.com/secret/token`. The response status is `201 Created`, and the response body is `{} 449 ms 844 B`.

49	POST /secret/token (201)	true	Issue a POST request on the `/secret/token` end point and receive 201 when Basic auth username/password is admin/password ► Hints ► Solution
----	--------------------------	------	--

50. Receive 403 status code when X-AUTH-TOKEN doesn't match a valid token

Authorization Challenges

Once the system knows who you are, authorization is if you have the correct level of access. In these challenges the authorization is granted using a custom API header X-AUTH-TOKEN or using a Bearer Authorization header.

ID	Challenge	Done	Description
50	GET /secret/note (403)	false	Issue a GET request on the `/secret/note` end point and receive 403 when X-AUTH-TOKEN does not match a valid token ► Hints ► Solution

Solution:

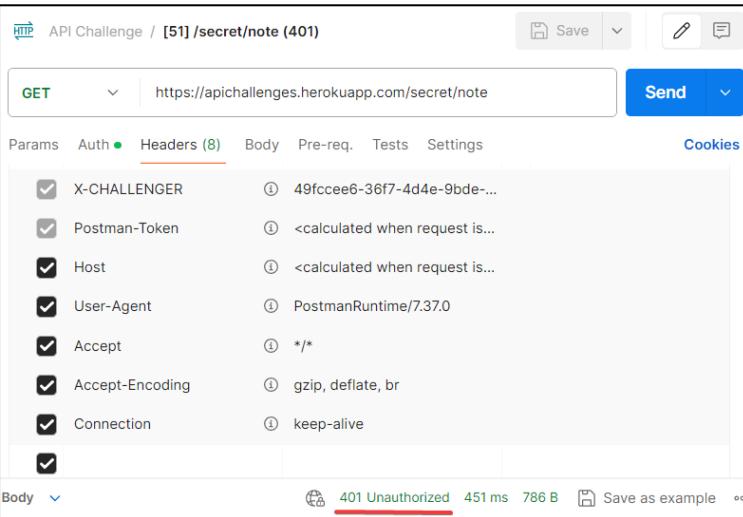
The screenshot shows the Postman interface with a failed GET request to `https://apichallenges.herokuapp.com/secret/note`. The response status is `403 Forbidden`, and the response body is `{} 458 ms 783 B`. The X-AUTH-TOKEN header is set to `key`.

50	GET /secret/note (403)	true	Issue a GET request on the `/secret/note` end point and receive 403 when X-AUTH-TOKEN does not match a valid token ► Hints ► Solution
----	------------------------	------	---

51. Receive 401 when no X-AUTH-TOKEN header present

51	GET /secret/note (401)	false	Issue a GET request on the `/secret/note` end point and receive 401 when no X-AUTH-TOKEN header present ► Hints ► Solution
----	------------------------	-------	--

Solution:



The screenshot shows the Postman interface with a failed GET request to `https://apichallenges.herokuapp.com/secret/note`. The Headers tab is selected, displaying the following headers:

- X-CHALLENGER: 49fccee6-36f7-4d4e-9bde-...
- Postman-Token: <calculated when request is...
- Host: <calculated when request is...
- User-Agent: PostmanRuntime/7.37.0
- Accept: */*
- Accept-Encoding: gzip, deflate, br
- Connection: keep-alive

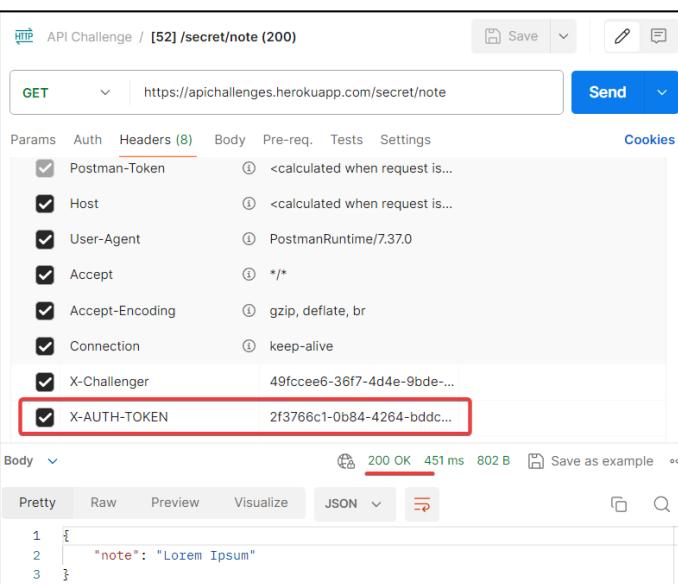
The status bar at the bottom indicates a 401 Unauthorized response.

51	GET /secret/note (401)	true	Issue a GET request on the `/secret/note` end point and receive 401 when no X-AUTH-TOKEN header present ► Hints ► Solution
----	------------------------	------	--

52. Receive 200 status code when valid X-AUTH-TOKEN used – response body should contain the note

52	GET /secret/note (200)	false	Issue a GET request on the `/secret/note` end point receive 200 when valid X-AUTH-TOKEN used - response body should contain the note ► Hints ► Solution
----	------------------------	-------	---

Solution:



The screenshot shows the Postman interface with a successful GET request to `https://apichallenges.herokuapp.com/secret/note`. The Headers tab is selected, displaying the following headers:

- Postman-Token: <calculated when request is...
- Host: <calculated when request is...
- User-Agent: PostmanRuntime/7.37.0
- Accept: */*
- Accept-Encoding: gzip, deflate, br
- Connection: keep-alive
- X-Challenger: 49fccee6-36f7-4d4e-9bde-...
- X-AUTH-TOKEN: 2f3766c1-0b84-4264-bddc...

The status bar at the bottom indicates a 200 OK response. The response body is shown in JSON format:

```
1 {  
2   "note": "Lorem Ipsum"  
3 }
```

52	GET /secret/note (200)	true	Issue a GET request on the `/secret/note` end point receive 200 when valid X-AUTH-TOKEN used - response body should contain the note ► Hints ► Solution
----	------------------------	------	---

53. Receive 200 status code when valid X-AUTH-TOKEN used.

53	POST /secret/note (200)	false	<p>Issue a POST request on the `/secret/note` end point with a note payload e.g. {"note":"my note"} and receive 200 when valid X-AUTH-TOKEN used. Note is maximum length 100 chars and will be truncated when stored.</p> <p>► Hints ► Solution</p>
----	-------------------------	-------	---

Solution:

The screenshot shows the Postman interface with a successful POST request to the endpoint [53] /secret/note (200). The response status is 200 OK, and the response body is {"note": "Lorem Ipsum Loire Amore"}. The request body was {"note": "Lorem Ipsum Loire Amore"}

53	POST /secret/note (200)	true	<p>Issue a POST request on the `/secret/note` end point with a note payload e.g. {"note":"my note"} and receive 200 when valid X-AUTH-TOKEN used. Note is maximum length 100 chars and will be truncated when stored.</p> <p>► Hints ► Solution</p>
----	-------------------------	------	---

54. Receive 401 when no X-AUTH-TOKEN present

54	POST /secret/note (401)	false	<p>Issue a POST request on the `/secret/note` end point with a note payload {"note":"my note"} and receive 401 when no X-AUTH-TOKEN present</p> <p>► Hints ► Solution</p>
----	-------------------------	-------	---

Solution:

The screenshot shows the Postman interface with an unauthorized POST request to the endpoint [54] /secret/note (401). The response status is 401 Unauthorized, and the response body is {"error": "Unauthorized"}. The request body was {"note": "Lorem Ipsum Loire Amore"}

54	POST /secret/note (401)	true	<p>Issue a POST request on the `/secret/note` end point with a note payload {"note":"my note"} and receive 401 when no X-AUTH-TOKEN present</p> <p>► Hints ► Solution</p>
----	-------------------------	------	---

55. Receive 403 status code when X-AUTH-TOKEN does not match a valid token

55	POST /secret/note (403)	false	Issue a POST request on the `/secret/note` end point with a note payload {"note":"my note"} and receive 403 when X-AUTH-TOKEN does not match a valid token ► Hints ► Solution
----	-------------------------------	-------	---

Solution:

HTTP API Challenge / [55] /secret/note (403)

POST https://apichallenges.herokuapp.com/secret/note Send

Params Auth Headers (10) Body Pre-req. Tests Settings Cookies

Key	Value
X-CHALLENGER	49fccee6-36f7-4d4e-9bde-...
Postman-Token	<calculated when request is...
Content-Type	application/json
Content-Length	<calculated when request is...
Host	<calculated when request is...
User-Agent	PostmanRuntime/7.37.0
Accept	*/*
Accept-Encoding	gzip, deflate, br
Connection	keep-alive
X-AUTH-TOKEN	KEY

Body

403 Forbidden 451 ms 787 B Save as example

Pretty Raw Preview Visualize JSON

55	POST /secret/note (403)	true	Issue a POST request on the `/secret/note` end point with a note payload {"note":"my note"} and receive 403 when X-AUTH-TOKEN does not match a valid token ► Hints ► Solution
----	-------------------------------	------	---

56. Receive 200 when using the X-AUTH-TOKEN value as an Authorization Bearer token

56	GET /secret/note (Bearer)	false	Issue a GET request on the `/secret/note` end point receive 200 when using the X-AUTH-TOKEN value as an Authorization Bearer token - response body should contain the note ► Hints ► Solution
----	---------------------------	-------	---

Solution:

HTTP API Challenge / [56] /secret/note (Bearer)

GET https://apichallenges.herokuapp.com/secret/note Send

Params Auth Headers (8) Body Pre-req. Tests Settings Cookies

Key	Value
Authorization	Bearer 2f3766c1-0b84-426...
Postman-Token	<calculated when request is...
Host	<calculated when request is...
User-Agent	PostmanRuntime/7.37.0
Accept	*/*
Accept-Encoding	gzip, deflate, br
Connection	keep-alive
X-Challenger	49fccee6-36f7-4d4e-9bde-...

Body

200 OK 446 ms 817 B Save as example

Pretty Raw Preview Visualize JSON

```

1 {
2   "note": "Lorem Ipsum Lore Amore"
3 }

```

56	GET /secret/note (Bearer)	true	Issue a GET request on the `/secret/note` end point receive 200 when using the X-AUTH-TOKEN value as an Authorization Bearer token - response body should contain the note ► Hints ► Solution
----	---------------------------------	------	---

57. Receive 200 status code when a valid X-AUTH-TOKEN value used as an Authorization Bearer token.

57	POST /secret/note (Bearer)	false	<p>Issue a POST request on the `/secret/note` end point with a note payload e.g. {"note":"my note"} and receive 200 when valid X-AUTH-TOKEN value used as an Authorization Bearer token. Status code 200 received. Note is maximum length 100 chars and will be truncated when stored.</p> <p>► Hints ► Solution</p>
----	----------------------------	-------	--

Solution:

The screenshot shows a Postman request configuration for a POST to the endpoint `https://apichallenges.herokuapp.com/secret/note`. The Headers tab is selected, showing the following configuration:

- Authorization**: Bearer 2f3766c1-0b84-426...
- Content-Type**: application/json
- Content-Length**: <calculated when request is...
- Host**: <calculated when request is...
- User-Agent**: PostmanRuntime/7.37.0
- Accept**: */*
- Accept-Encoding**: gzip, deflate, br
- Connection**: keep-alive
- X-Challenger**: 49fccee6-36f7-4d4e-9bde-...

The Body tab shows a JSON response with the following content:

```

1 {
2   "note": "Lorem Ipsum"
3 }

```

The response status is 200 OK with a duration of 444 ms and a size of 802 B.

57	POST /secret/note (Bearer)	true	<p>Issue a POST request on the `/secret/note` end point with a note payload e.g. {"note":"my note"} and receive 200 when valid X-AUTH-TOKEN value used as an Authorization Bearer token. Status code 200 received. Note is maximum length 100 chars and will be truncated when stored.</p> <p>► Hints ► Solution</p>
----	----------------------------	------	--

58. Delete the last todo record in the system so that there are no more todos in the system

Miscellaneous Challenges

We left these challenges to the end because they seemed fun, but... different.

ID	Challenge	Done	Description
58	DELETE /todos/{id} (200) all	false	Issue a DELETE request to successfully delete the last todo in system so that there are no more todos in the system ► Hints

Solution:

A screenshot of the Postman application interface. The URL in the address bar is `https://apichallenges.herokuapp.com/todos/10`. The method dropdown shows "DELETE". The "Send" button is highlighted. Below the URL, the "Headers" tab is selected, showing several default headers like X-Challenger, Postman-Token, Host, User-Agent, Accept, Accept-Encoding, and Connection. The "Body" tab is also visible. At the bottom, the response status is shown as "200 OK" with a green success icon.

58	DELETE /todos/{id} (200) all	true	Issue a DELETE request to successfully delete the last todo in system so that there are no more todos in the system ► Hints
----	------------------------------	------	--

59. Create as many POST requests as possible for todos

59	POST /todos (201) all	false	Issue as many POST requests as it takes to add the maximum number of TODOS allowed for a user. The maximum number should be listed in the documentation.
----	-----------------------	-------	--

Solution:

A screenshot of the Postman application interface. The URL in the address bar is `https://apichallenges.herokuapp.com/todos`. The method dropdown shows "POST". The "Send" button is highlighted. Below the URL, the "Headers" tab is selected, showing several default headers. The "Body" tab is selected and set to "JSON". The JSON payload is a single object with properties: title ("POST_21"), doneStatus (false), and description ("Post-description_21"). At the bottom, the response status is shown as "400 Bad Request" with a red error icon. The response body is a JSON object with a single key "errorMessages": ["ERROR: Cannot add instance, maximum limit of 20 reached"].

59	POST /todos (201) all	true	Issue as many POST requests as it takes to add the maximum number of TODOS allowed for a user. The maximum number should be listed in the documentation.
----	-----------------------	------	--