Будем тестировать и тренироваться на todo приложении.

Приложение предназначено для ведения списка задач.

1. Получите уникальный токен, по которому приложение сможет вас опознать.

Для этого отправьте POST запрос на https://apichallenges.herokuapp.com/challenger.

В ответе есть header X-Challenger. Значение header сохраните!

По нему и буду зачитываться задания.

При выполнении любого задания, нужно указать хедер X-Challenger и его значение.

2. Чтобы просмотреть список задач, отправьте GET запрос на https://apichallenges.herokuapp.com/challenges и не забудьте указать ваш хедер)))

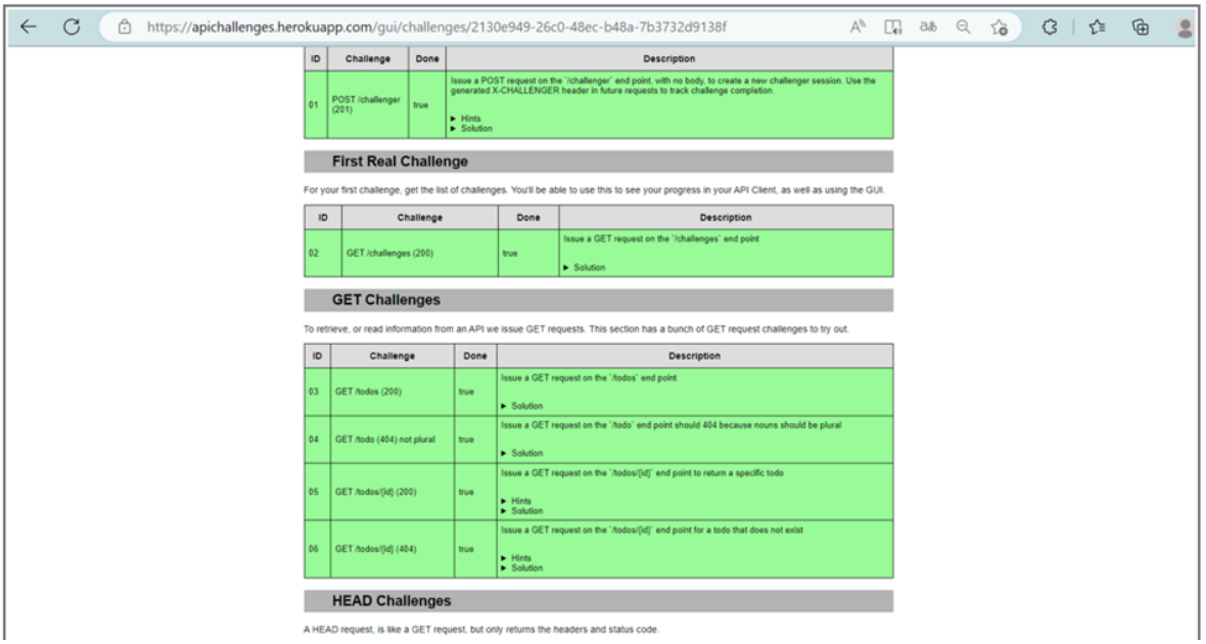
Документация на api, где можно подсмотреть адреса эндпоинтов

https://apichallenges.herokuapp.com/docs

Вот так внешне выглядит само приложение.

https://apichallenges.herokuapp.com/gui/instances?entity=todo

В чат с преподавателем пришлите скриншоты, где у всех задачек стоит статус true



HEAD Challenges

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

ID	Challenge	Done	Description
07	HEAD /todos (200)	true	Issue a HEAD request on the "/todos" end point ▶ Solution

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 assocated GET request, and amending it.

ID	Challenge	Done	Description
08	POST /todos (201)	true	Issue a POST request to successfully create a todo ▶ Solution
09	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
10	POST /todos (400) doneStatus	true	Issue a POST request to create a todo but fail validation on the "doneStatus" field ▶ Solution

Update Challenges with POST

Use a POST request to amend something that already exists. These are 'partial' content updates so you useually 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
11	POST /todos/[id] (200)	true	Issue a POST request to successfully update a todo ▶ Hints ▶ Solution

DELETE Challenges

DELETE Challenges

Use a DELETE request to delete an entity. Since this is an extreme request, normall 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
12	DELETE /todos/{id} (200)	true	Issue a DELETE request to successfully delete a todo ▶ Hints ▶ Solution

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
13	OPTIONS /todos (200)	true	Issue an OPTIONS request on the '/todos' end point. You might want to manually check the 'Allow' header in the response is as expected. ▶ Solution

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
14	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
15	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
16	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