

Упражнение 1

Шаги:

1. В любом браузере (желательно на Chromium) открыть ссылку <https://jsonplaceholder.typicode.com>
2. Открыть панель разработчика на вкладке Console/Консоль
3. Отправить запрос
4. Проанализировать ответ

Запрос:

```
fetch( '<ссылка на ресурс>' )  
  .then(response => response.json())  
  .then(json => console.log(json))
```

<ссылка на ресурс> состоит из протокола передачи данных, полного доменного имени и дополнением к URL:

Часть URL	Ограничение
/posts	100 posts
/comments	500 comments
/albums	100 albums
/photos	5000 photos
/todos	200 todos
/users	10 users

Приблизительный результат:

The screenshot shows a web browser window with the URL jsonplaceholder.typicode.com. The page has a blue header with the text "{JSON} Placeholder" and a description: "Free fake API for testing and prototyping. Powered by JSON Server + LowDB. Tested with XV. As of Oct 2021, serving ~1.7 billion requests each month." The Chrome DevTools Console is open, showing a successful fetch request to `https://jsonplaceholder.typicode.com/users/1`. The response is a JSON object representing a user:

```
{  
  "id": 1,  
  "name": "Leanne Graham",  
  "username": "Bret",  
  "email": "Sincere@april.biz",  
  "address": {  
    "city": "Gwenborough",  
    "geo": {  
      "lat": "-37.3159",  
      "lng": "81.1496"   
    },  
    "street": "Kulas Light",  
    "suite": "Apt. 556",  
    "zipcode": "92998-3874"   
  },  
  "phone": "1-770-736-8031 x56442",  
  "website": "hildegard.org"  
}
```

Упражнение 2

Шаги:

1. В любом браузере (желательно на Chromium) открыть ссылку <https://reqres.in>
2. Открыть панель разработчика на вкладке Console/Консоль
3. Отправить запрос
4. Проанализировать ответ

Запрос:

```
fetch('<ссылка на ресурс>', {  
  method: '<метод запроса>',  
  body: JSON.stringify(<тело запроса>)  
})  
  
  .then(response => response.json())  
  .then(json => console.log(json))
```

<ссылка на ресурс> состоит из протокола передачи данных, полного доменного имени и дополнением к URL, описанные на странице ресурса

<метод запроса> имеет значения - GET, POST, PUT, PATCH, DELETE в зависимости от дополнения к URL, описанные на странице ресурса

<тело запроса> описаны на странице ресурса

Приблизительный результат:

The image shows a screenshot of the reqres.in API interface and a browser console. On the left, there is a list of API endpoints with their corresponding HTTP methods: GET LIST USERS, GET SINGLE USER, GET SINGLE USER NOT FOUND, GET LIST <RESOURCE>, GET SINGLE <RESOURCE>, GET SINGLE <RESOURCE> NOT FOUND, and POST CREATE. In the center, there is a 'Request' section showing the endpoint /api/users and a 'Response' section showing a 201 status code and a JSON object: { "name": "morpheus", "job": "leader", "id": "819", "createdAt": "2022-05-01T16:55:45.3" }. On the right, there is a screenshot of a browser console showing a successful POST request and response, with the response being a JSON object: { "id": "856", "createdAt": "2022-05-01T16:55:54.121Z" }.