

Многообразие Моск инструментов



Александр

Кузнецов

MTS

HEISENBUG

План:

1. **Зачем использовать моки?**

План:

1. Зачем использовать моки?
2. **Обзор UI инструментов для создания МОКОВ**

План:

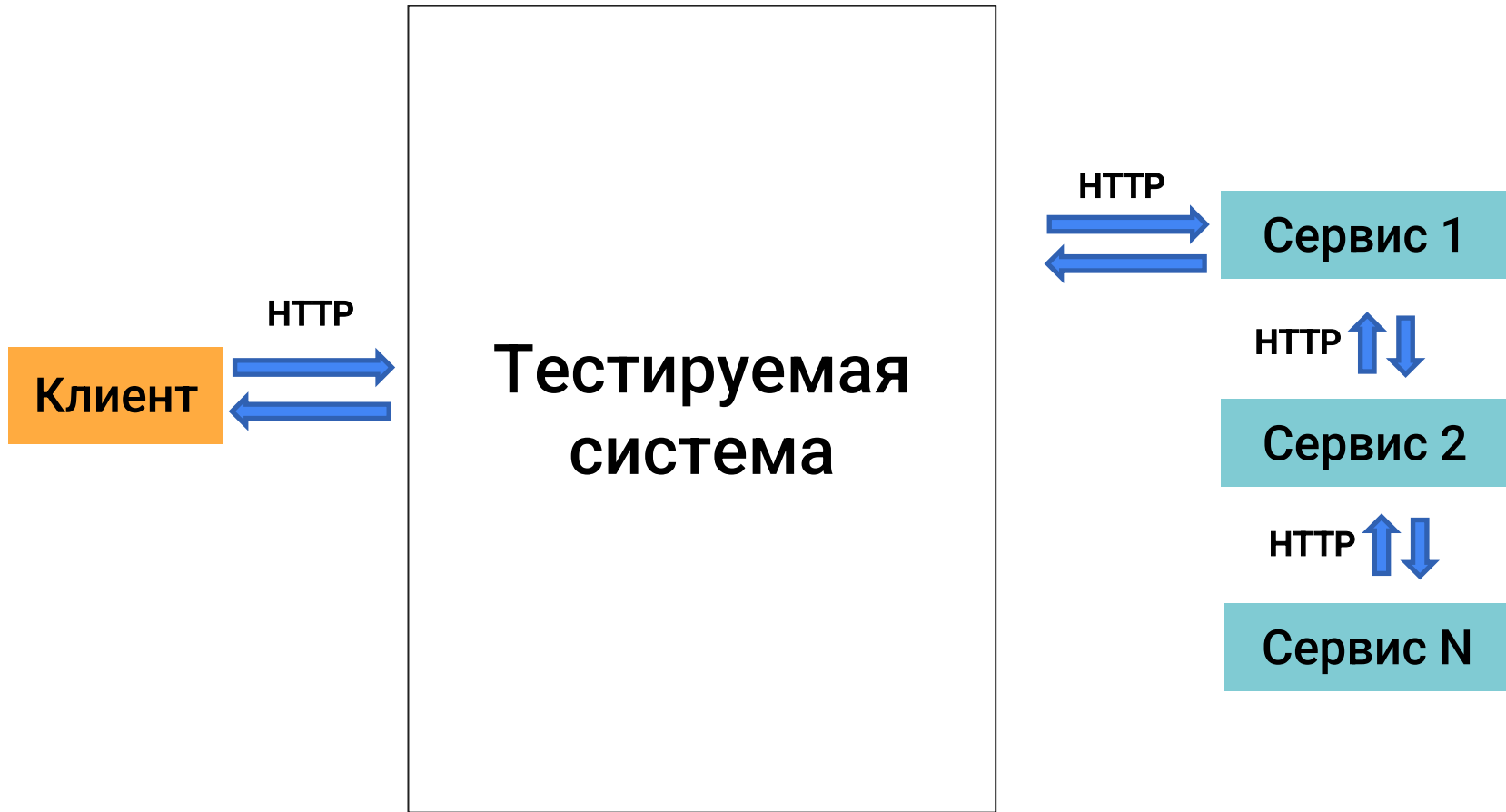
1. Зачем использовать моки?
2. Обзор **UI инструментов** для создания МОКОВ
3. **Обзор инструментов CLI/CODE**

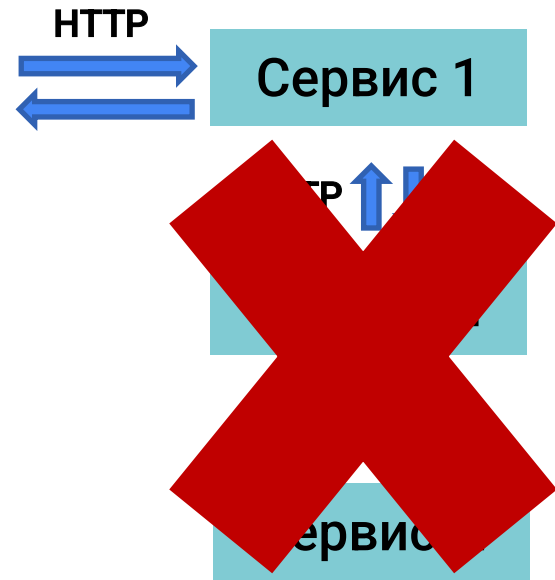
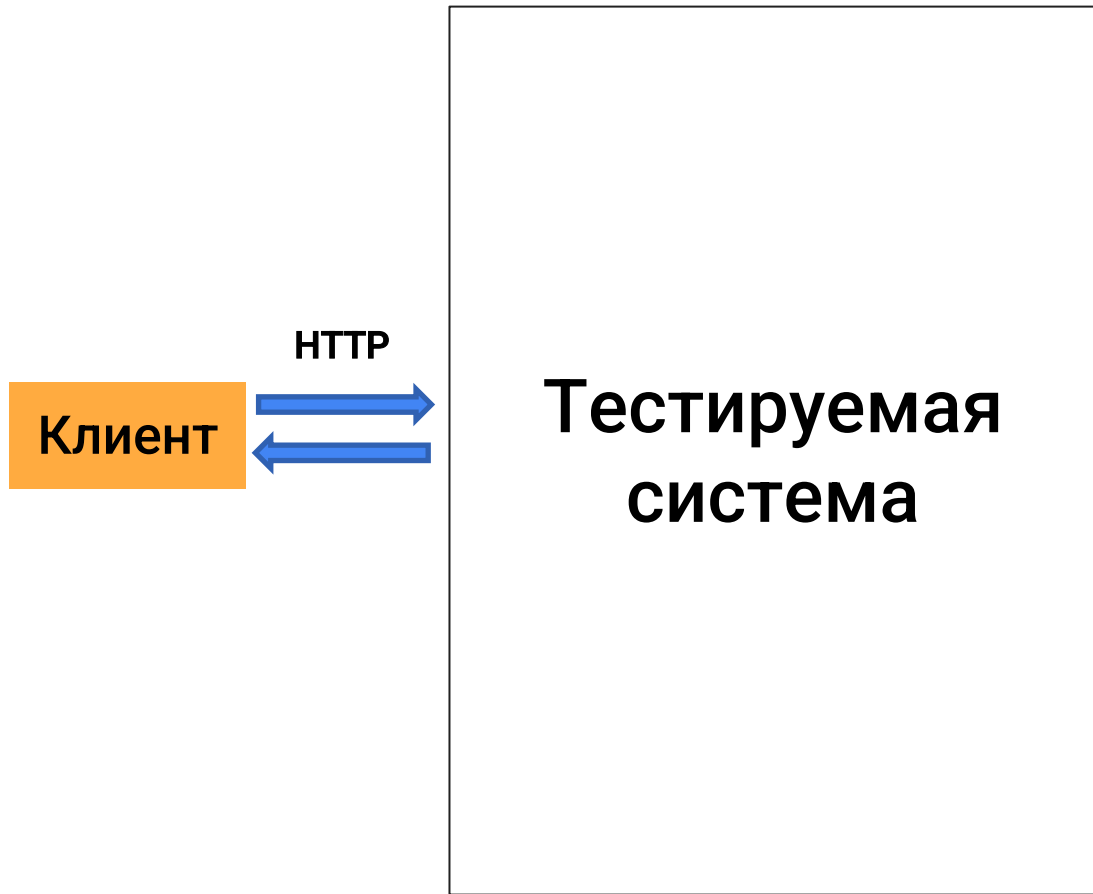
План:

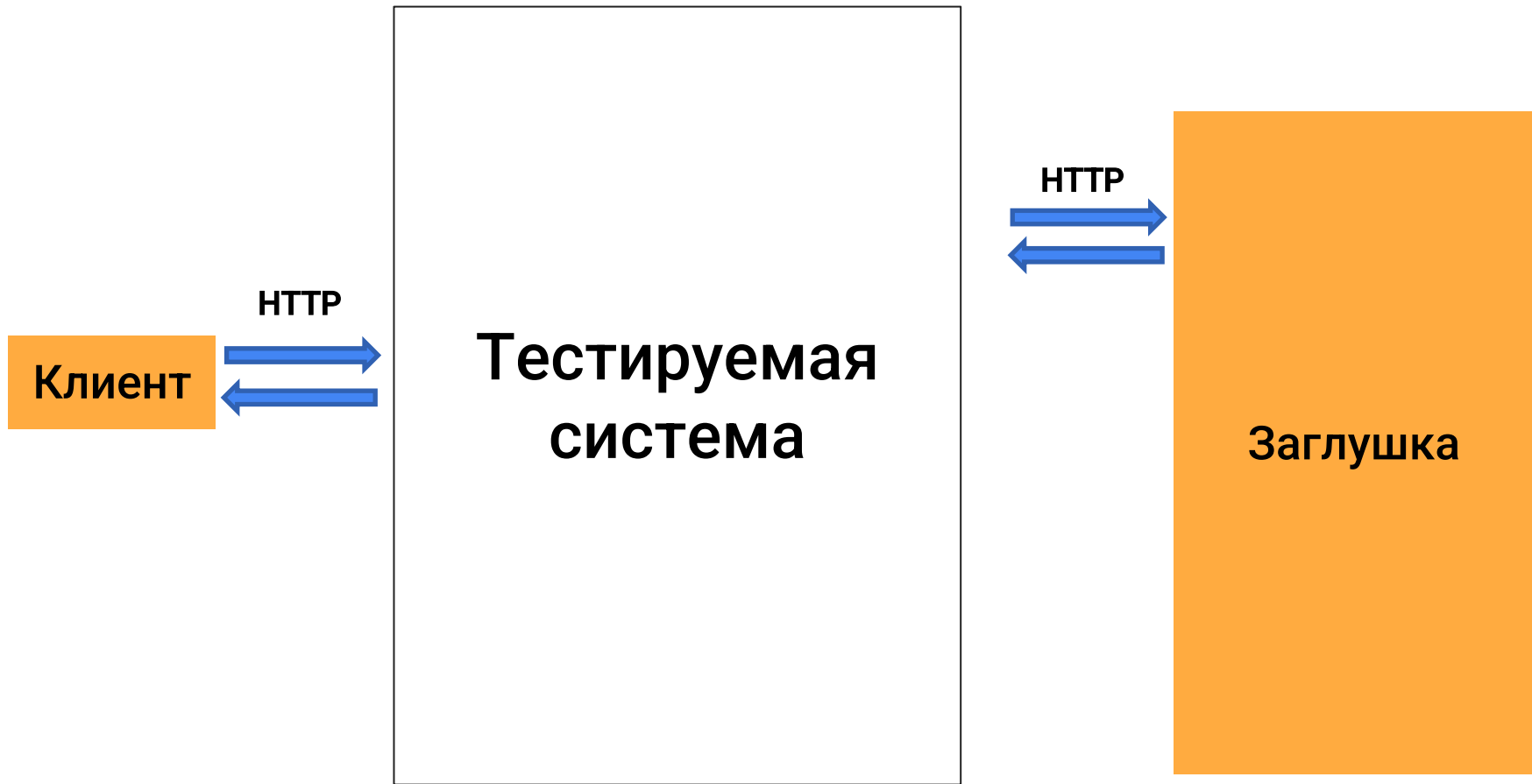
1. Зачем использовать моки?
2. Обзор **UI инструментов** для создания МОКОВ
3. Обзор инструментов **CLI/CODE**
4. **Обзор и сравнение комбинированных**

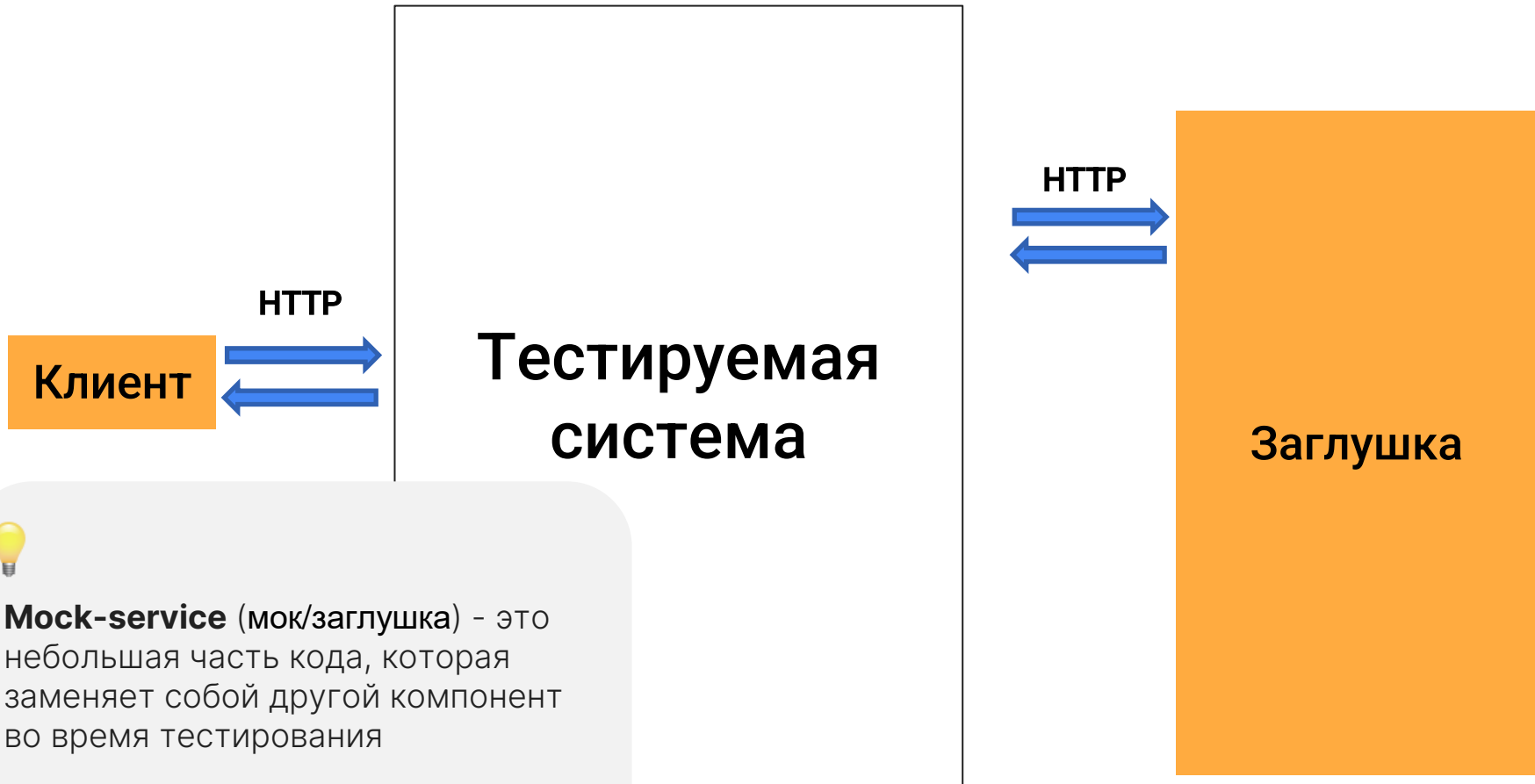
Зачем
использовать
моки?

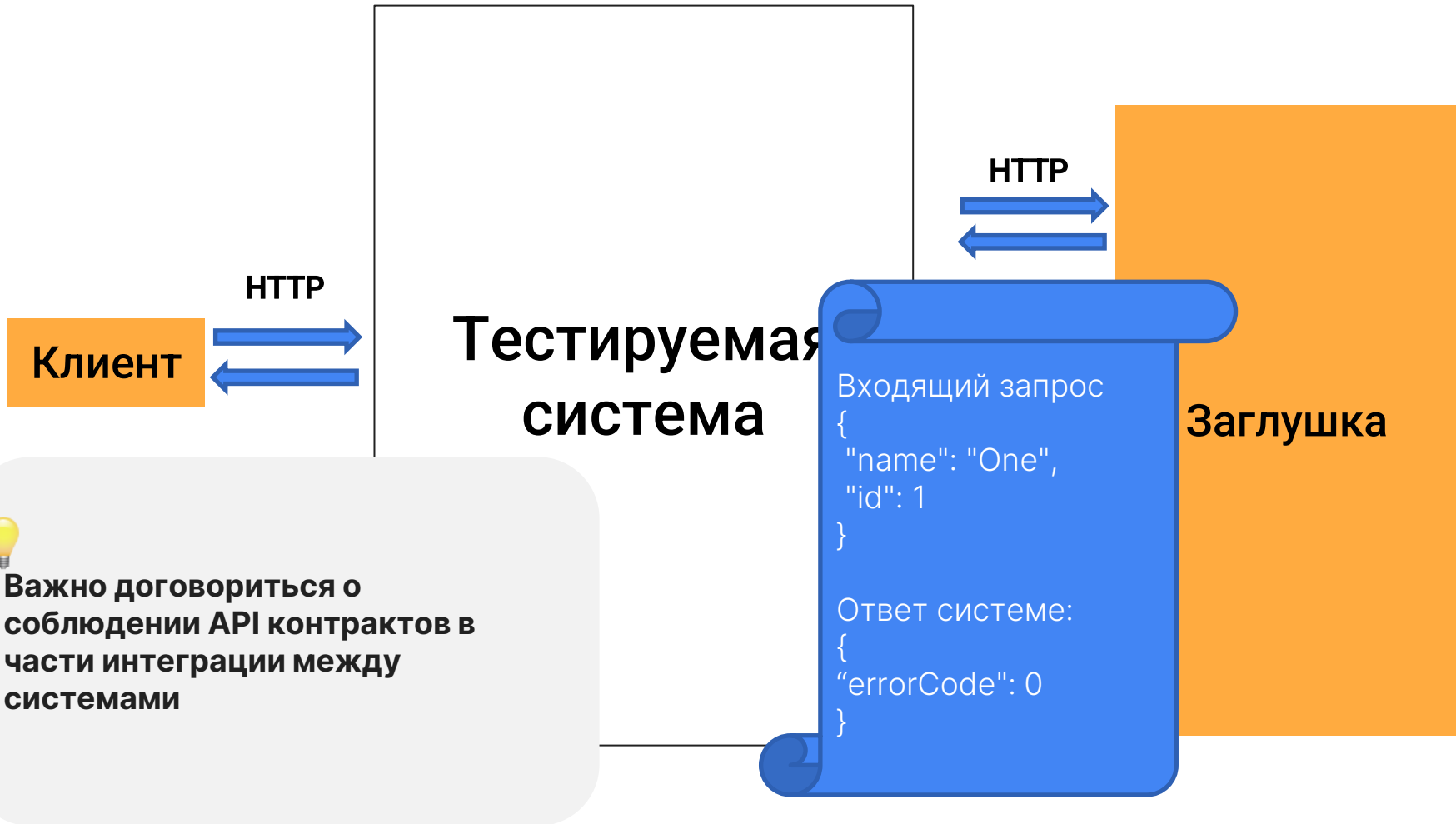




















Сценарии применения

Проверка	Живая интеграция	Заглушка
Позитивный сценарий		
Timeout		
HTTP response status code		
Негативные сценарии		



SoapUI



POSTMAN

WM WireMock Recorder



SMARTBEAR
SwaggerHub



PRISM
Brought to you by **Stoplight**.

CITRUS 

mock  on



MOCKINTOSH

MockServer



mocky

UI инструменты



POSTMAN



mocky



SwaggerHub

CLI/CODE



MOCKINTOSH

CITRUS 



MICROCKS

API MOCKING & TESTING

@MICROCKSIO

Комбинированные



PRISM

Brought to you by **Stoplight**.

MockServer



WireMock Recorder

mock  on



SoapUI



Критерии к инструментам

1. Понятный и простой UI

Критерии к инструментам

1. Понятный и простой UI
2. **Доступность**

Критерии к инструментам

1. Понятный и простой UI
2. Доступность
3. **Возможность локального запуска**

Критерии к инструментам

1. Понятный и простой UI
2. Доступность
3. Возможность локального запуска
- 4. Возможность запуска из под Docker**

Критерии к инструментам

1. Понятный и простой UI
2. Доступность
3. Возможность локального запуска
4. Возможность запуска из под Docker
- 5. Встроенный генератор данных**

Критерии к инструментам

1. Понятный и простой UI
2. Доступность
3. Возможность локального запуска
4. Возможность запуска из под Docker
5. Встроенный генератор данных
- 6. Запись входящих запросов**

Критерии к инструментам

1. Понятный и простой UI
2. Доступность
3. Возможность локального запуска
4. Возможность запуска из под Docker
5. Встроенный генератор данных
6. Запись входящих запросов
7. **Хорошая документация (её наличие)**

Критерии к инструментам

1. Понятный и простой UI
2. Доступность
3. Возможность локального запуска
4. Возможность запуска из под Docker
5. Встроенный генератор данных
6. Запись входящих запросов
7. Хорошая документация (её наличие)
- 8. Разнообразие мокируемых протоколов**

Критерии к инструментам

1. Понятный и простой UI
2. Доступность
3. Возможность локального запуска
4. Возможность запуска из под Docker
5. Встроенный генератор данных
6. Запись входящих запросов
7. Хорошая документация (её наличие)
8. Разнообразие мокируемых протоколов
- 9. Импорт готовых моков**



POSTMAN



mocky

SMARTBEAR

SwaggerHub

Обзор

UI

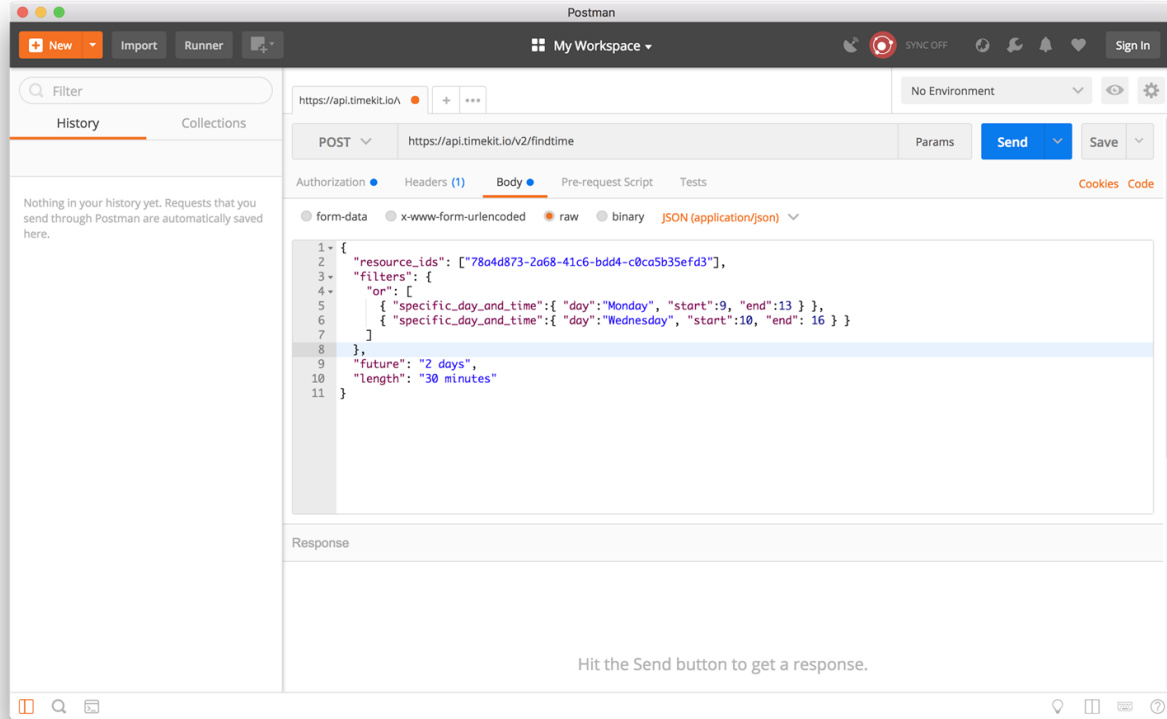
инструментов



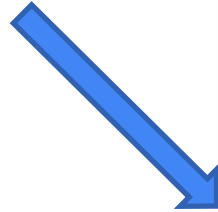
1. Postman



POSTMAN



Открыть вкладку
Mock Server



The screenshot shows the 'Customer APIs' interface. At the top, there are 'New' and 'Import' buttons. Below them is a search bar with a plus icon and a menu icon. A tooltip 'Create mock server' is visible over the plus icon. The left sidebar contains several tabs: 'Collect', 'APIs', 'Environments', 'Mock Servers' (which is highlighted with a blue border and a vertical orange bar), 'Monitors', 'Flows', and 'History'. The main content area features an illustration of an astronaut holding a server rack. Below the illustration, the text reads: 'You don't have any mock servers. Mock servers let you simulate endpoints and their corresponding responses in a collection without actually setting up a back end.' At the bottom of this section is a blue button labeled 'Create Mock Server'.



2. Нажать кнопку
«Создать мок
сервер»



The screenshot shows the 'Customer APIs' management interface. At the top, there are 'New' and 'Import' buttons. Below them is a search bar with a plus icon and a dropdown menu. A blue arrow points to the plus icon, which has a tooltip that says 'Create mock server'. The left sidebar contains navigation options: Collect, APIs, Environments, Mock Servers (highlighted with an orange bar), Monitors, Flows, and History. The main content area features an illustration of a person in a spacesuit holding a tablet, with the text: 'You don't have any mock servers. Mock servers let you simulate endpoints and their corresponding responses in a collection without actually setting up a back end. Create Mock Server'.





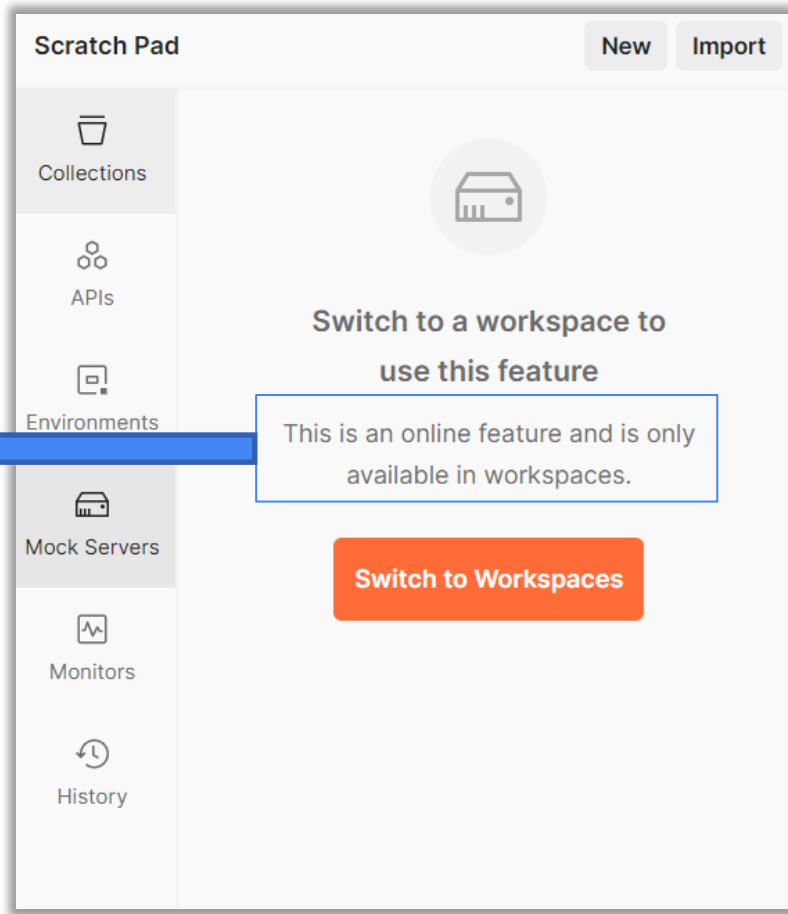
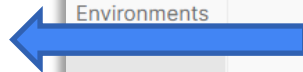
Это онлайн-функция,
доступная только в
Workspaces.



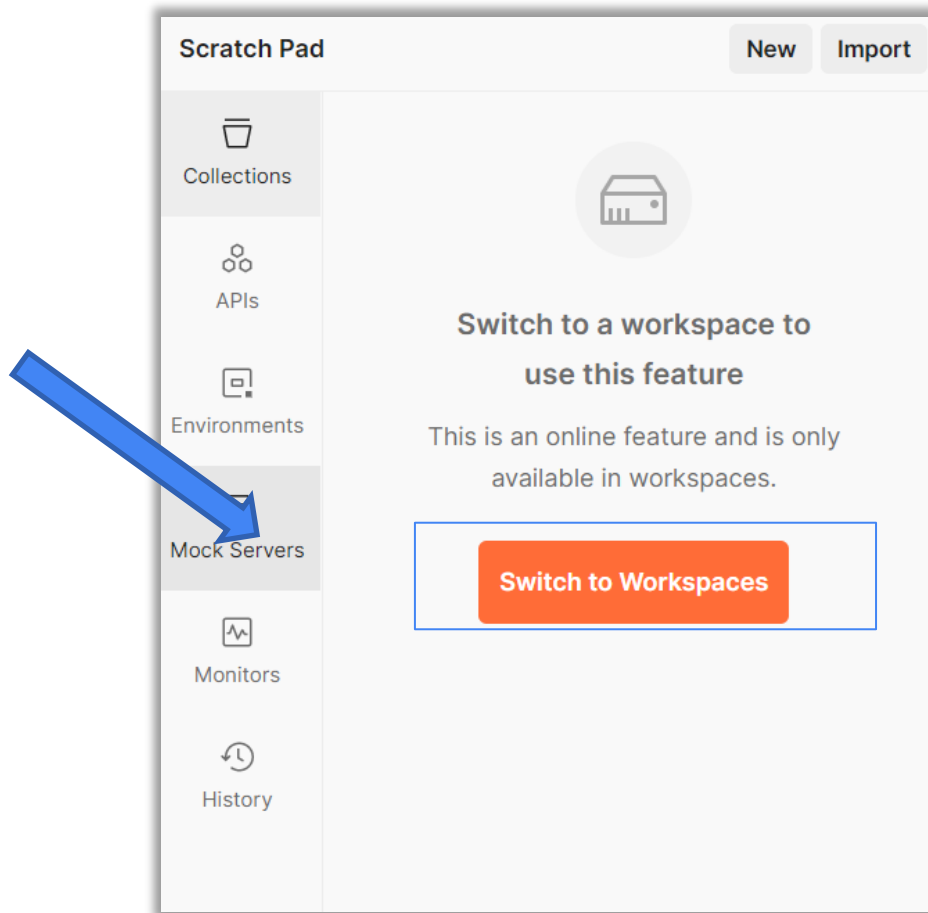
Что такое workspaces?

Рабочие области Postman помогают организовать работу с API и обеспечить совместную работу в организации или по всему миру.

Существует **четыре** различных типа **рабочих пространств** Postman для ваших различных потребностей: **личные, командные, партнерские** и **общедоступные**.



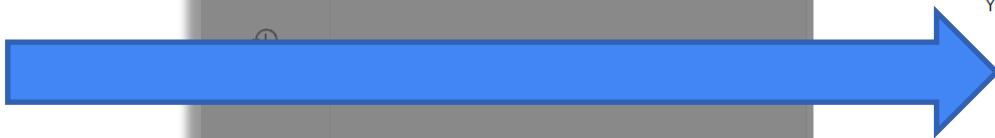
Открываем
Workspaces.





Но
нужен
аккаунт


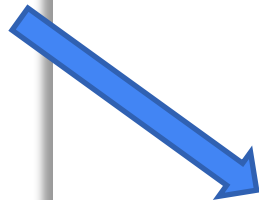
...



The screenshot shows the Postman application interface. On the left is a sidebar with navigation items: Collections, APIs, Environments, Mock Servers, and Monitors. The main area displays a message: "Switch to a workspace to use this feature" with a sub-message: "This is an online feature and is only available in workspaces." Below this is a "Switch to Workspaces" button. A modal dialog titled "CREATE AN ACCOUNT" is open on the right. It features an illustration of an astronaut, the text "Create an account" and "You need an account to continue exploring Postman.", and two buttons: "Create Account" (highlighted in orange) and "Sign In".



Проверьте, стабильно ли ваше интернет-соединение. Если вы используете брандмауэр или прокси-сервер, отключите его или внесите в белый список getpostman.com а затем повторите попытку. Если проблема не устранена, повторите попытку через некоторое время.
(Код ошибки:)



Unable to connect to our servers

Check if your internet connection is stable. If you are using a firewall or a proxy server, disable it or whitelist getpostman.com and then retry. If the problem persists, try again after some time. (Error code:)

[Go To Scratchpad](#)

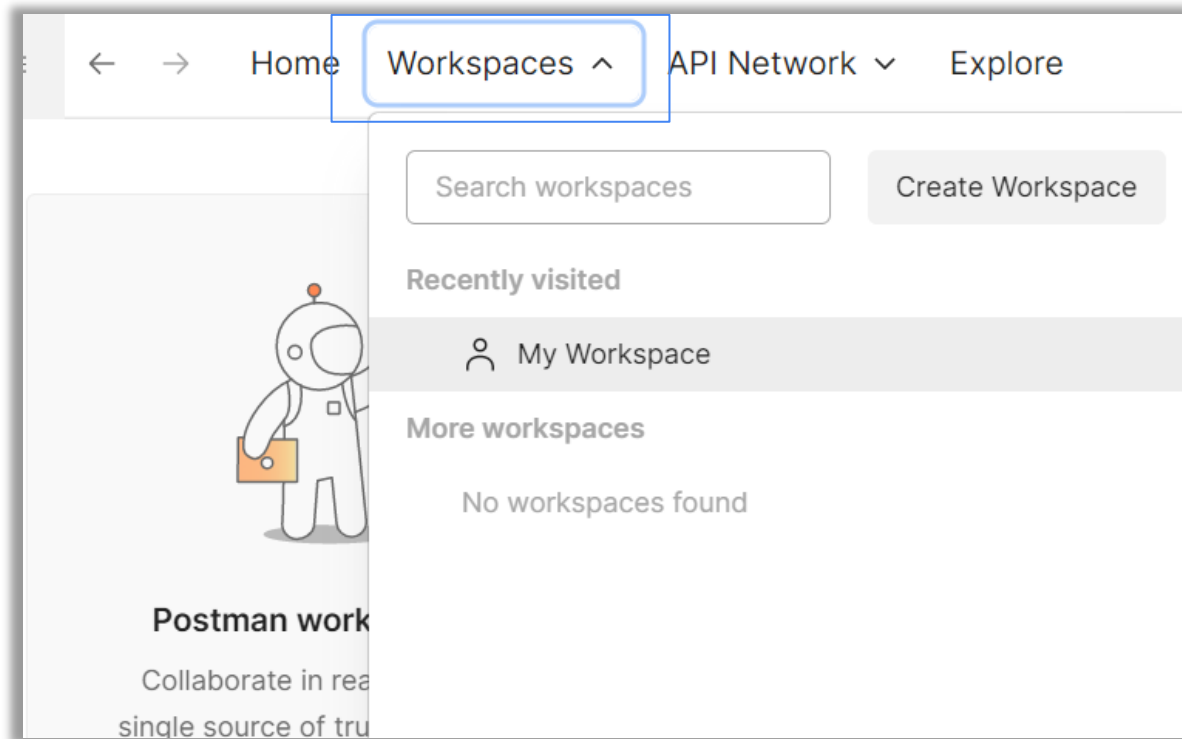
[Try Again](#)



Возможно в вашей организации заблокирована часть ресурсов Postman



Открываем
Workspaces.






← → Home **Workspaces ^** API Network v Explore

Search workspaces Create Workspace

Recently visited

-  My Workspace

More workspaces

No workspaces found

Postman work
Collaborate in real
single source of truth

Открываем Mock Server

My Workspace New Import

Collections

APIs

Environments

Mock Servers

Monitors

Flows

History

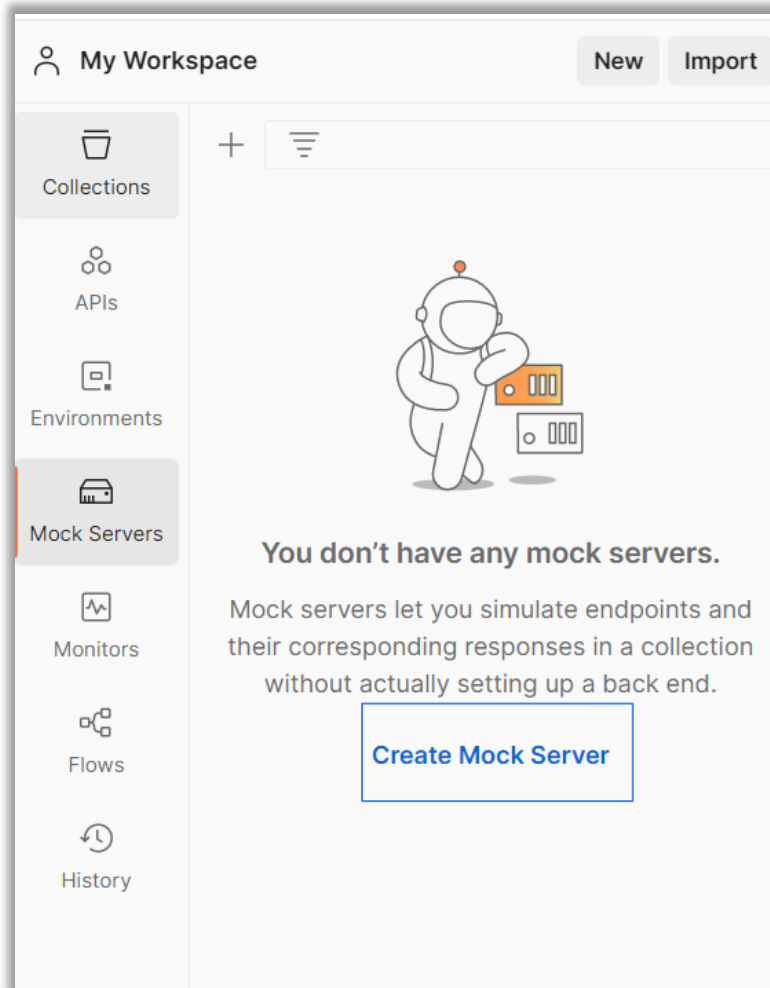
You don't have any mock servers.

Mock servers let you simulate endpoints and their corresponding responses in a collection without actually setting up a back end.

[Create Mock Server](#)



Создаем Mock Server



Указываем параметры Mock Server

Create a mock server

1. Select collection to mock 2. Configuration

Create a new collection Select an existing collection

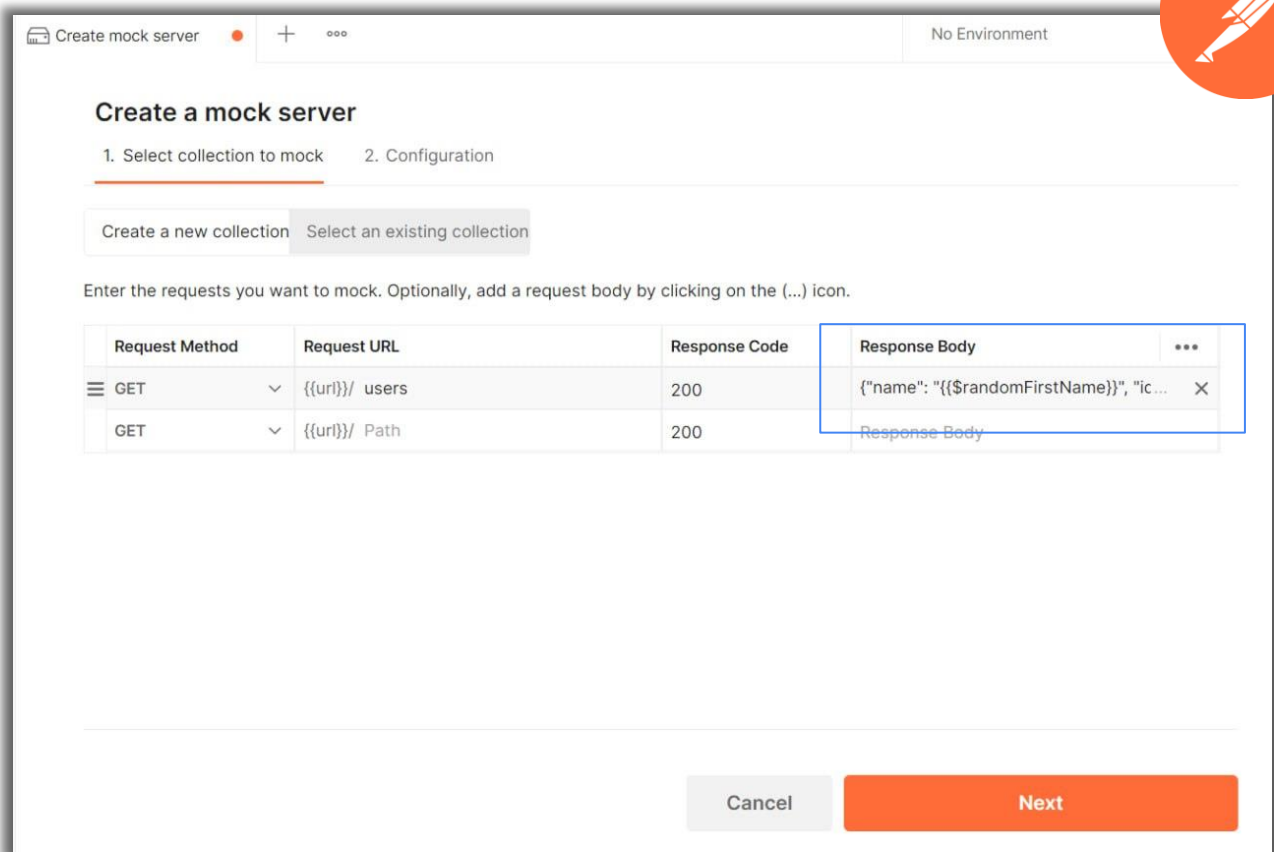
Enter the requests you want to mock. Optionally, add a request body by clicking on the (...) icon.

Request Method	Request URL	Response Code	Response Body	...
GET	⌵ {{url}}/ Path	200	Response Body	

Cancel Next



Указываем тело ответа Mock Server



Create mock server + ... No Environment

Create a mock server

1. Select collection to mock 2. Configuration

Create a new collection Select an existing collection

Enter the requests you want to mock. Optionally, add a request body by clicking on the (...) icon.

Request Method	Request URL	Response Code	Response Body	...
GET	{{url}}/ users	200	{"name": "\${RandomFirstName}", "ic...	×
GET	{{url}}/ Path	200	Response Body	

Cancel Next



Create a mock server

1. Select collection to mock
2. Configuration

Create a new collection Select an existing collection

Enter the requests you want to mock. Optionally, add a request body by clicking on the (...) icon.

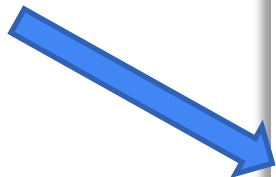
Request Method	Request URL	Response Code	Response Body	...
GET	{{url}}/ users	200	{"name": "{{randomFirstName}}", "ic...	X
GET	{{url}}/ Path	200	Response Body	

Cancel

Next



Имитировать
фиксированную
задержку в
сети



Create a mock server

✓ Select collection to mock 2. Configuration

Mock Server Name

Environment
An environment is a group of variables useful for storing and reusing values.
[Learn more ↗](#)

No Environment ▾

Save the mock server URL as a new environment variable
This will create a new environment containing URL.

Simulate a fixed network delay

Make mock server private
To call a private mock server, you'll need to add an `x-api-key` header to your requests. See how to generate a [Postman API key ↗](#)

Скрыть мок сервер



Postman предоставляет два различных типа мок серверов. Общедоступные мок серверы отлично подходят для проектов с открытым исходным кодом или отдельных проектов, но если вы работаете с внутренними или частными API, вы можете легко обеспечить безопасность своих API, просматривая ответы API с частных мок серверов.

Create a mock server

✓ Select collection to mock 2. Configuration

Mock Server Name

Environment
An environment is a group of variables useful for storing and reusing values.
[Learn more](#)

No Environment

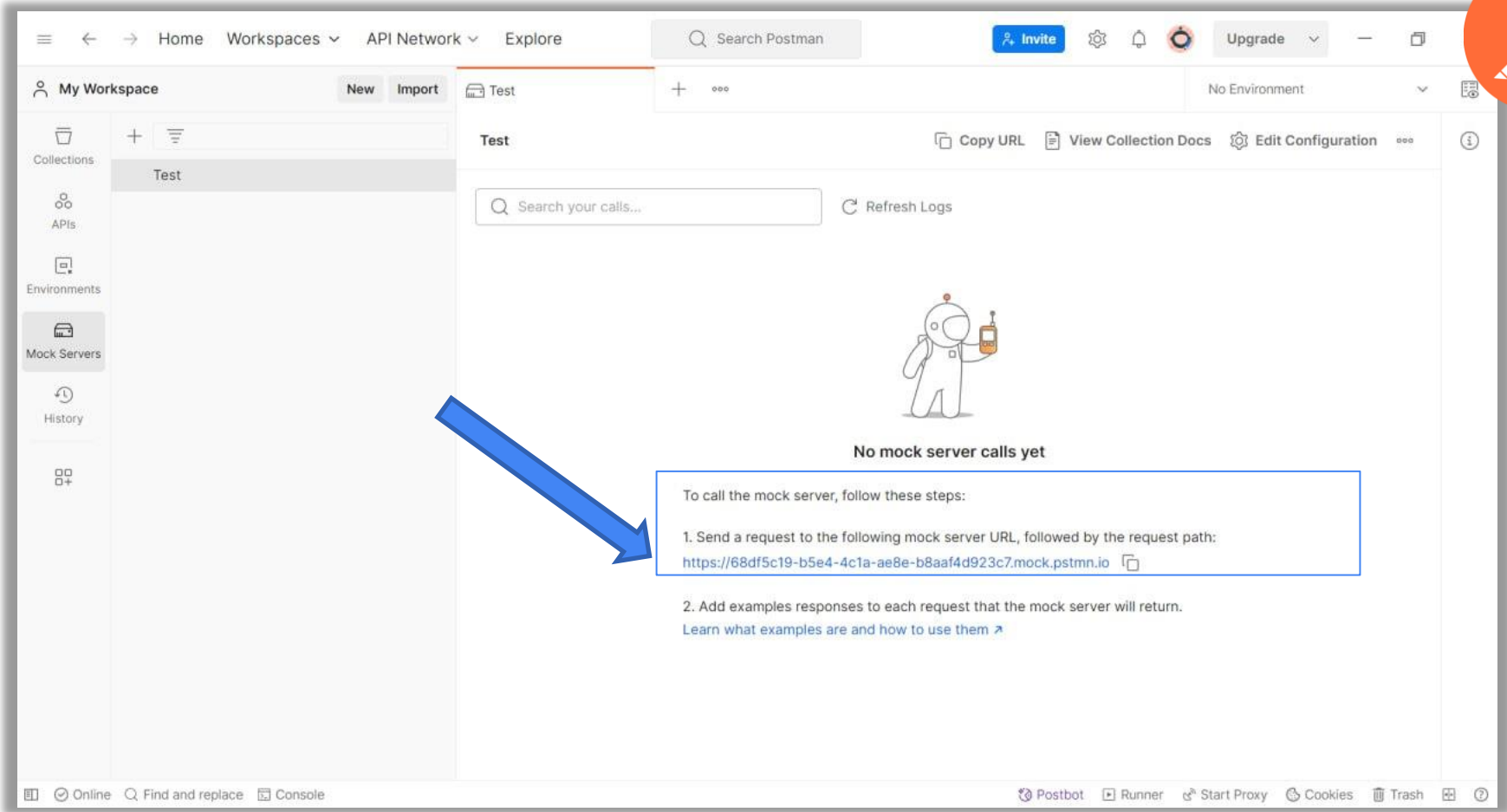
Save the mock server URL as a new environment variable
This will create a new environment containing URL.

Simulate fixed network delay

No delay

Make mock server private
To call a private mock server, you'll need to add an `x-api-key` header to your requests. See how to generate a [Postman API key](#)






Home Workspaces API Network Explore Search Postman Invite Upgrade

My Workspace New Import Test No Environment

Test Copy URL View Collection Docs Edit Configuration

Search your calls... Refresh Logs



No mock server calls yet

To call the mock server, follow these steps:

1. Send a request to the following mock server URL, followed by the request path:
`https://68df5c19-b5e4-4c1a-ae8e-b8aaf4d923c7.mock.pstmn.io`
2. Add examples responses to each request that the mock server will return.
[Learn what examples are and how to use them](#)

Online Find and replace Console Postbot Runner Start Proxy Cookies Trash



GET

https://68df5c19-b5e4-4c1a-ae8e-b8aaf4d923c7.mock.pstmn.io/users

Send



Params Authorization Headers (7) Body Pre-request Script Tests Settings

Cookies

Query Params

Key	Value	Description	⋮ Bulk Edit
Key	Value	Description	

Body Cookies Headers (12) Test Results

200 OK 809 ms 489 B Save as Example

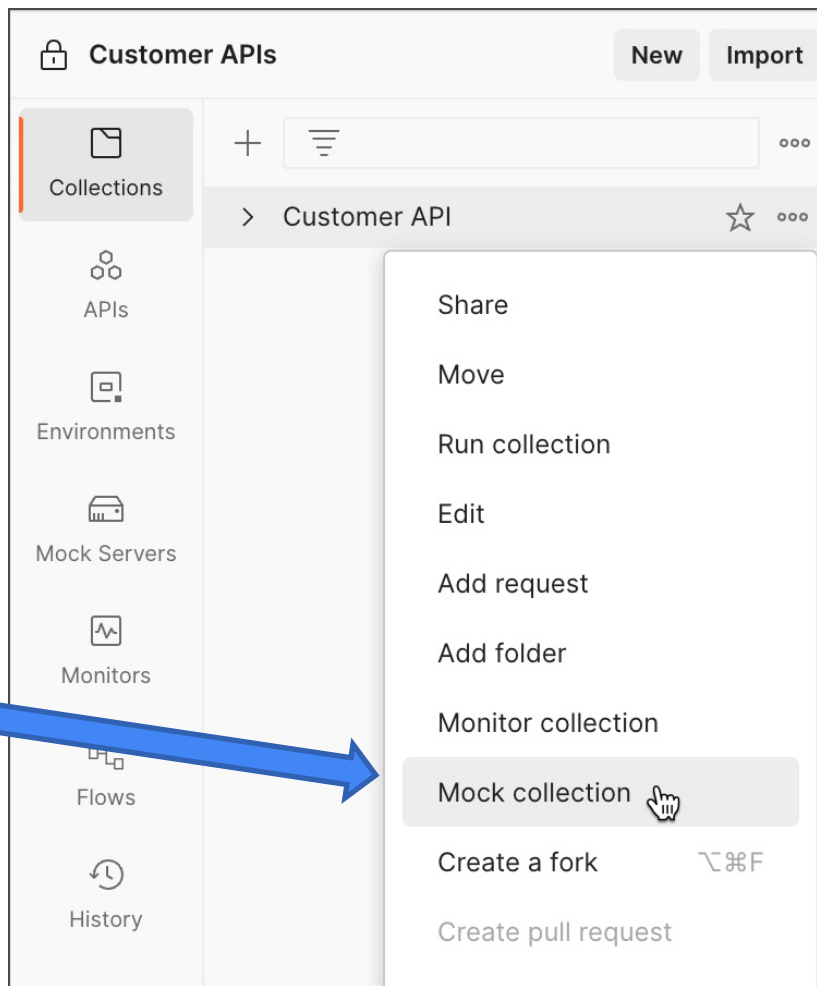
Pretty Raw Preview Visualize

JSON

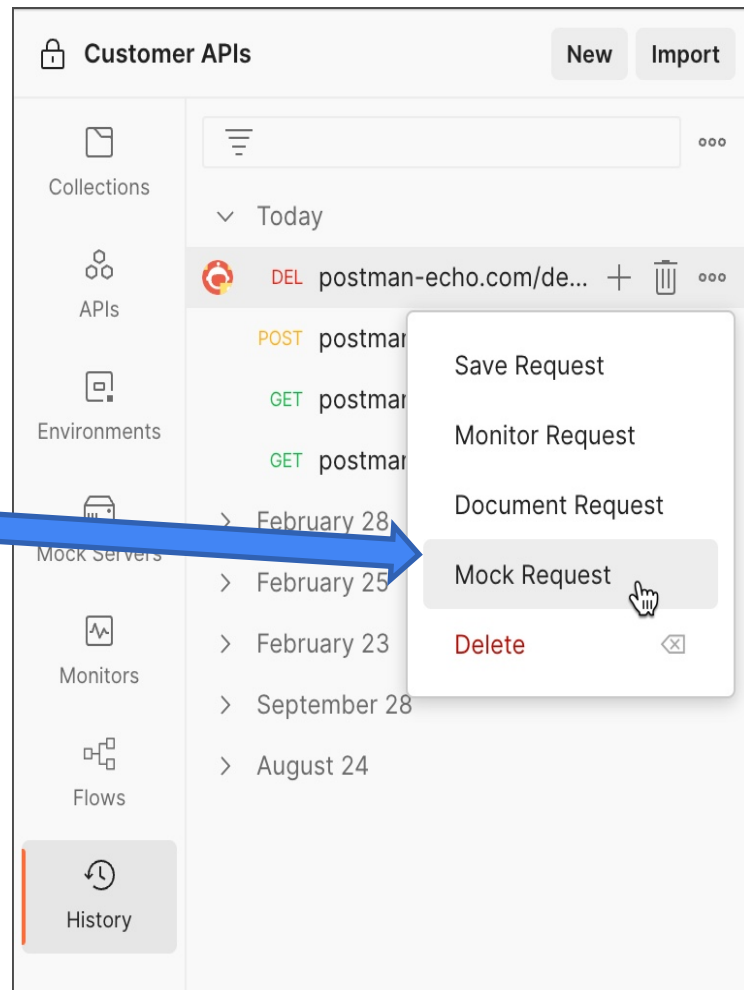


```
1 {
2   "name": "Earline",
3   "id": "53f38dc7-7224-4350-8efd-d535c94dae49",
4   "test": "test",
5   "check": true
6 }
```

Удобное создание МОКОВ ИЗ КОЛЛЕКЦИИ



И из истории запросов





Плюсы Postman

- ✓ **Понятный интерфейс**
- ✓ Бесплатно (но с ограничениями)
- ✓ Возможно скрыть мок (private mock)
- ✓ Удобная работа с моками внутри Postman



Для бесплатной версии
ограничение в 1000 вызовов

	Basic	Professional	Enterprise Essentials	Enterprise Ultimate
	\$12 Per user/month	\$29 Per user/month	\$49 Per user/month	Contact our sales team for pricing, support, and benefits.
	Buy Now	Buy Now	Buy Now	Contact Sales
Mock Servers Calls (monthly) ?	1,000	10,000	10,000	100,000



Плюсы Postman

- ✓ Понятный интерфейс
- ✓ **Бесплатно (но с ограничениями)**
- ✓ Возможно скрыть мок (private mock)
- ✓ Удобная работа с моками внутри Postman



Для бесплатной версии
ограничение в 1000 вызовов

	Basic	Professional	Enterprise Essentials	Enterprise Ultimate
	\$12	\$29	\$49	Contact our sales team for pricing, support, and benefits.
	Per user/month	Per user/month	Per user/month	
	Buy Now	Buy Now	Buy Now	Contact Sales
Mock Servers Calls (monthly) [?]	1,000	10,000	10,000	100,000



Плюсы Postman

- ✓ Понятный интерфейс
- ✓ Бесплатно (но с ограничениями)
- ✓ **Возможно скрыть мок (private mock)**
- ✓ Удобная работа с моками внутри Postman



Для бесплатной версии
ограничение в 1000 вызовов

	Basic	Professional	Enterprise Essentials	Enterprise Ultimate
	\$12	\$29	\$49	Contact our sales team for pricing, support, and benefits.
	Per user/month	Per user/month	Per user/month	
	Buy Now	Buy Now	Buy Now	Contact Sales
Mock Servers Calls (monthly) ?	1,000	10,000	10,000	100,000



Плюсы Postman

- ✓ Понятный интерфейс
- ✓ Бесплатно (но с ограничениями)
- ✓ Возможно скрыть мок (private mock)
- ✓ **Удобная работа с моками внутри Postman**



Для бесплатной версии
ограничение в 1000 вызовов

	Basic	Professional	Enterprise Essentials	Enterprise Ultimate
	\$12	\$29	\$49	Contact our sales team for pricing, support, and benefits.
	Per user/month	Per user/month	Per user/month	
	Buy Now	Buy Now	Buy Now	Contact Sales
Mock Servers Calls (monthly) ?	1,000	10,000	10,000	100,000



Минусы Postman

- × **Требуется аккаунт**
- × Сервер поднят не локально
- × Для бесплатного аккаунта лимит запросов в (1,000 бесплатных вызовов каждый месяц)
- × Нельзя менять готовую коллекцию МОКОВ



Минусы Postman

- × Требуется аккаунт
- × **Сервер поднят не локально**
- × Для бесплатного аккаунта лимит запросов в (1,000 бесплатных вызовов каждый месяц)
- × Нельзя менять готовую коллекцию МОКОВ



Минусы Postman

- × Требуется аккаунт
- × Сервер поднят не локально
- × **Для бесплатного аккаунта лимит запросов в (1,000 бесплатных вызовов каждый месяц)**
- × Нельзя менять готовую коллекцию МОКОВ



Минусы Postman

- × Требуется аккаунт
- × Сервер поднять не локально
- × Для бесплатного аккаунта лимит запросов в (1,000 бесплатных вызовов каждый месяц)
- × **Нельзя менять готовую коллекцию моков**



Итог для Postman

1. Подходит для быстрых экспериментов на моках
2. Удобно генерировать одноразовые моки и работать в рамках Postman
3. Встроенный генератор данных
<https://github.com/faker-js/faker>

<https://learning.postman.com/docs/designing-and-developing-your-api/mocking-data/mock-with-api/>





[SwaggerHub](https://swaggerhub.com)



Swagger
powered by SMARTBEAR

[Why Swagger?](#)
[Tools](#)
[Resources](#)

Search
Sign In

SwaggerHub

[Features](#)
[Pricing](#)
[Enterprise](#)
[Integrations](#)

Create Free Account

NEW
API Exploration Made Easy
→

SETTINGS

The Single Source of Truth for API Development

Accelerating API delivery and quality through standards and collaboration, built on OpenAPI.

Create Free Account

We Value Your Privacy

NewRelic, Inc. ("NewRelic") and our **advertising partners** use cookies and a variety of technologies as described below to collect and store personal data (e.g., your IP address). If you consent, the cookies, device identifiers, or other information can be stored or accessed on your device for the purposes described below. You can click "Allow All" or "Decline All" to click Settings above to customize your choices.

NewRelic and our **advertising partners** process personal data for:

- Store a color access information on a device
- Create a personalized user profile
- Select personalized content
- Personalized ads, ad measurement and audience insights
- Product development

 For some of the purposes above, our **advertising partners**

- Use precise geolocation data

 Even if our partners may or their legitimate interests in order to process personal data.

View our **advertising partners**

- You may to provide or deny consent for specific purposes, review the purposes each partner believes they have a legitimate interest in, and object to such processing.

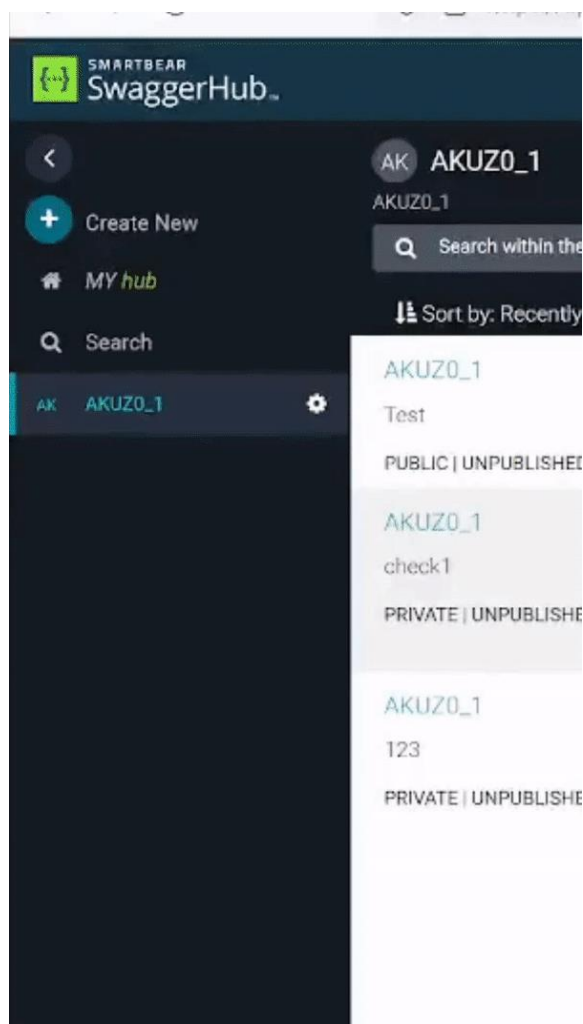
If you **decline** cookies, All you will still be able to view content on this site and you will still receive advertising, but the advertising will not be tailored for you. You may change your setting whenever you see the on this site.

DECLINE ALL

ALLOW ALL

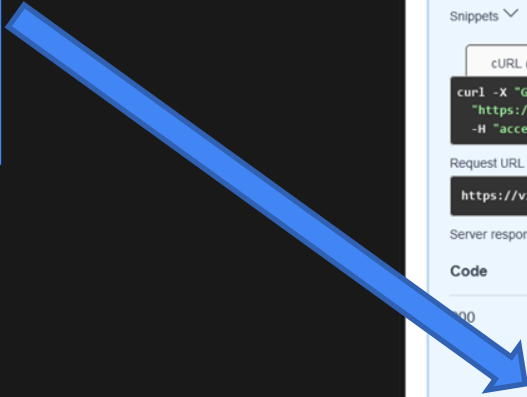
Join thousands of teams who depend on SwaggerHub to get their products to market, faster.

<div style="font-size: 24px; font-weight: bold; margin: 0;">100k+</div> <div style="font-size: 10px; margin-top: 5px;">API Practitioners</div>	<div style="font-size: 24px; font-weight: bold; margin: 0;">40k+</div> <div style="font-size: 10px; margin-top: 5px;">Organizations</div>	<div style="font-size: 24px; font-weight: bold; margin: 0;">200k+</div> <div style="font-size: 10px; margin-top: 5px;">API Projects</div>
--	---	---



```

1 openapi: 3.0.0
2 servers:
3   # Added by API Auto Mocking Plugin
4   - description: SwaggerHub API Auto Mocking
5     url: https://virtserver.swaggerhub.com/AKUZ0_1/Test/1.0.0
6 info:
7   description: This is a simple API
8   version: "1.0.0"
9   title: Simple Inventory API
10
11 paths:
12   /user:
13     get:
14       responses:
15         '200':
16           description: OK
17           content:
18             application/json:
19               schema:
20                 $ref: '#/components/schemas/User'
21         '400':
22           description: bad input parameter
23
24 components:
25   schemas:
26     User:
27       type: object
28       properties:
29         id:
30           type: integer
31           format: int64
32           example: 42
33         name:
34           type: string
35           example: Widget Adapter
36
  
```



Servers

https://virtserver.swaggerhub.com/AKUZ0_1/Test/1.0.0 - SwaggerHub API A... ▼

default ⤴

GET /user 🗑️ ⤴ ⤵

Parameters Cancel

No parameters

Execute
Clear

Responses

Snippets ▼

cURL (bash) cURL (PowerShell) **cURL (CMD)**

```

curl -X "GET" ^
  "https://virtserver.swaggerhub.com/AKUZ0_1/Test/1.0.0/user" ^
  -H "accept: application/json"
        
```

Request URL

https://virtserver.swaggerhub.com/AKUZ0_1/Test/1.0.0/user

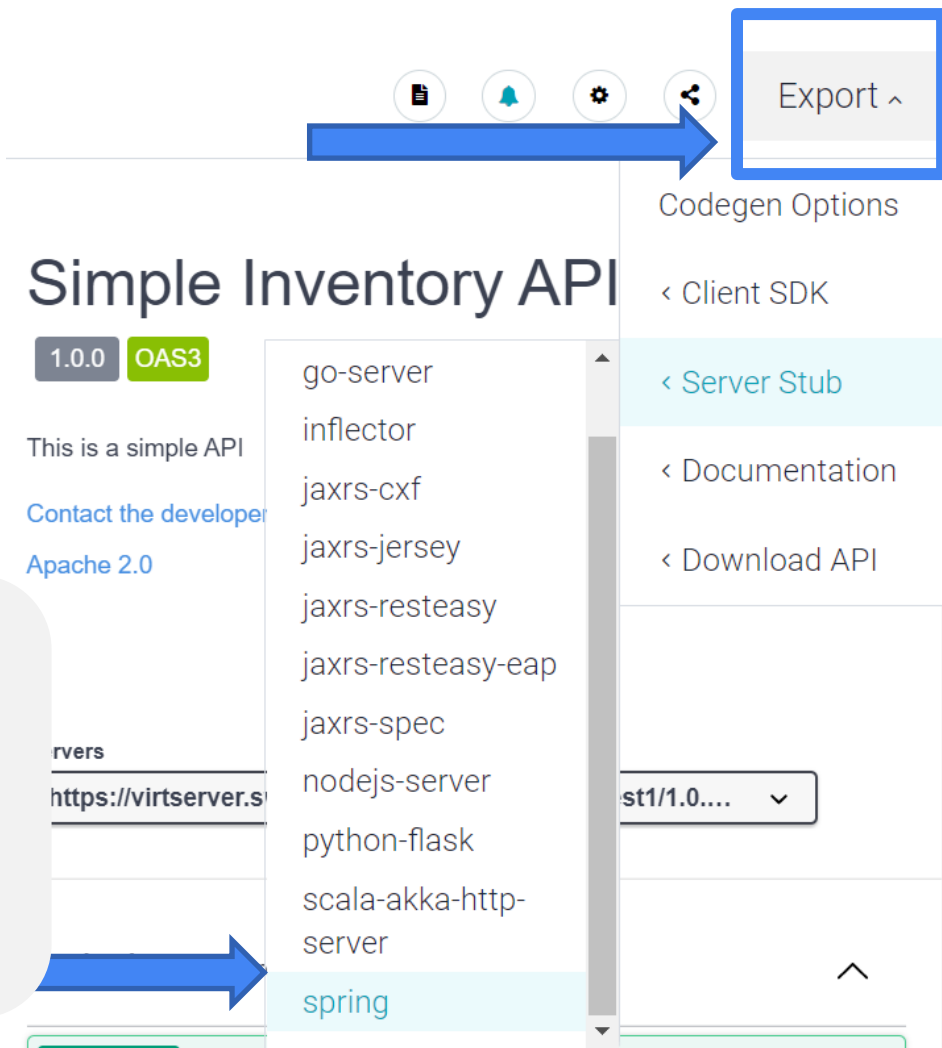
Server response

Code	Details
200	<p>Response body</p> <pre> { "id": 42, "name": "Widget Adapter" } </pre> <p style="text-align: right;">🗑️ Download</p>
	Response headers

Возможно выгрузить шаблон созданного мока на любом



Генерация кода (код ген) - это автоматическое создание программного кода специальным приложением, при котором по заданным условиям полностью или частично формируется исходный код программы.



The screenshot shows the SwaggerHub interface for the 'Simple Inventory API' (version 1.0.0, OAS3). The 'Export' button is highlighted with a blue box and a blue arrow pointing to it. A dropdown menu is open, showing a list of code generation options: go-server, inflector, jaxrs-cxf, jaxrs-jersey, jaxrs-resteasy, jaxrs-resteasy-eap, jaxrs-spec, nodejs-server, python-flask, scala-akka-http-server, and spring. The 'spring' option is highlighted in light blue. A blue arrow points from the text box on the left to the 'spring' option in the dropdown menu.

SwaggerHub

1.0.2.yaml

- Project
 - APIS
 - ace
 - IOT
 - oauth
 - passwords-api
 - petstore
 - reports-api
 - simple_api
 - 1.0.0
 - 1.0.1
 - 1.0.2
 - DOMAINS

```

1  openapi: 3.0.0
2  info:
3    description: This is a simple API
4    version: 1.0.2
5    title: Simple Inventory API
6  contact:
7    email: you@your-company.com
8  license:
9    name: Apache 2.0
10   url: 'http://www.apache.org/licenses/LICENSE-2.0.html'
11 tags:
12 - name: admins
13   description: Secured Admin-only calls
14 - name: developers
15   description: Operations available to regular developers
16 paths:
17   /inventory:
18     get:
19       tags:
20         - developers
21       summary: searches inventory
22       operationId: searchInventory
23       description: |
24         By passing in the appropriate options, you can search for
  
```

Document 1/1 > openapi: > 3.0.0

Event Log

1:15 LF UTF-8 2 spaces Schema: OAS 3: API

Плюсы SwaggerHub

- ✓ **Генерирует ответы JSON, XML и YAML на основе схем и примеров ответов**
- ✓ Бесплатно (но с ограничениями)
- ✓ Возможно скрыть мок (private mock)
- ✓ Возможность выгрузить мок в ГОТОВЫЙ КОД
- ✓ Мок генерируется по документации

Плюсы SwaggerHub

- ✓ Генерирует ответы JSON, XML и YAML на основе схем и примеров ответов
- ✓ **Бесплатно (но с ограничениями)**
- ✓ Возможно скрыть мок (private mock)
- ✓ Возможность выгрузить мок в ГОТОВЫЙ КОД
- ✓ Мок генерируется по документации

Плюсы SwaggerHub

- ✓ Генерирует ответы JSON, XML и YAML на основе схем и примеров ответов
- ✓ Бесплатно (но с ограничениями)
- ✓ **Возможно скрыть мок (private mock)**
- ✓ Возможность выгрузить мок в готовый код
- ✓ Мок генерируется по документации

Плюсы SwaggerHub

- ✓ Генерирует ответы JSON, XML и YAML на основе схем и примеров ответов
- ✓ Бесплатно (но с ограничениями)
- ✓ Возможно скрыть мок (private mock)
- ✓ **Возможность выгрузить мок в готовый код**
- ✓ Мок генерируется по документации

Плюсы SwaggerHub

- ✓ Генерирует ответы JSON, XML и YAML на основе схем и примеров ответов
- ✓ Бесплатно (но с ограничениями)
- ✓ Возможно скрыть мок (private mock)
- ✓ Возможность выгрузить мок в ГОТОВЫЙ КОД
- ✓ **Мок генерируется по документации**

Минусы SwaggerHub

- × **Требуется аккаунт**
- × Сервер поднят не локально
- × Для аккаунта лимит запросов (10 запросов в минуту для каждой версии API)
- × Ограниченные возможности в комбинации моков
- × Не работает для OpenAPI 3.1.0

Минусы SwaggerHub

- × Требуется аккаунт
- × **Сервер поднят не локально**
- × Для аккаунта лимит запросов (10 запросов в минуту для каждой версии API)
- × Ограниченные возможности в комбинации моков
- × Не работает для OpenAPI 3.1.0

Минусы SwaggerHub

- × Требуется аккаунт
- × Сервер поднят не локально
- × **Для аккаунта лимит запросов (10 запросов в минуту для каждой версии API)**
- × Ограниченные возможности в комбинации моков
- × Не работает для OpenAPI 3.1.0

Минусы SwaggerHub

- × Требуется аккаунт
- × Сервер поднят не локально
- × Для лимит запросов (10 запросов в минуту для каждой версии API)
- × **Ограниченные возможности в комбинации моков**
- × Не работает для OpenAPI 3.1.0

Минусы SwaggerHub

- × Требуется аккаунт
- × Сервер поднят не локально
- × Для аккаунта лимит запросов (10 запросов в минуту для каждой версии API)
- × Ограниченные возможности в комбинации моков
- × **Не работает для OpenAPI 3.1.0**

Итог для SwaggerHub

1. Удобно, если проект использует OpenApi
2. Можно использовать, если команда пользуется SwaggerHub.

<https://support.smartbear.com/swaggerhub/docs/integrations/api-auto-mocking.html>





<https://designer.mocky.io/>

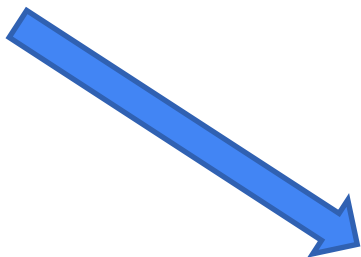


The screenshot shows the Mocky website interface. At the top left is the Mocky logo with a code icon. To the right are buttons for 'MANAGE MY MOCKS' and 'NEW MOCK'. The main heading is 'Mocky' with the tagline 'The world easiest & fastest tool belts to mock your APIs'. Below this is a section for 'API Mocks for Free' with a sub-heading 'Don't wait for the backend to be ready, generate custom API responses with Mocky and start working on your application straightaway'. A code block shows a curl command and its response:

```
curl https://run.mocky.io/v3/5185415ba171ea3a00704eed
< HTTP/1.1 200 OK
< Content-Type: application/json; charset=UTF-8
{ "hello": "world" }
```

 Below the code block is a button 'Start designing your mock >'. A dark banner at the bottom of the main content area contains a cookie policy notice: 'By using our website you agree to our [Cookie Policy](#)'. Below the banner are three buttons: 'No signup', 'Start designing your mock', and 'NEW MOCK'. The footer section is divided into four columns: 'Free & Unlimited' (no ads, no hidden subscriptions), 'Total control' (update/delete mocks anytime), 'Developer Friendly' (compatible with JS, Mobile, CORS, JSONP, GZIP), and 'Open Source' (Apache 2 license, GitHub contributions welcome). At the bottom, there are links for 'ABOUT', 'FAQ', and 'YOU LIKE IT? BUY ME A COFFEE!', a social media link 'Be an awesome Hacker, fork me', and a credit 'Made with love by @julien_lafont'. A sponsorship notice for 'Abstract API' is also present, along with copyright and policy links.

Форма создания HTTP мока



HTTP Status REQUIRED

200 - OK

The HTTP Code of the HTTP response you'll receive.

Response Content Type REQUIRED

application/json

The Content-Type header that will be sent with the response.

Charset REQUIRED

UTF-8

The Charset used to encode/decode your payload.

HTTP Headers OPTIONAL

```
{
  "X-Foo-Bar": "Hello World"
}
```


Customize the HTTP headers sent in the response. Define the headers as a JSON object.

HTTP Response Body OPTIONAL


```
{
  "identity": {
    "id": "b06cd03f-75d0-413a-b94b-35e155444d70",
    "login": "John Doe"
  },
  "permissions": {
    "roles": [
      "moderator"
    ]
  }
}
```

Готовый публичный адрес мока

Your mock is ready!

Mock URL 

```
https://run.mocky.io/v3/8d09ba26-8710-40da-ba8c-457fda7191fe
```

Secret delete link 

```
https://designer.mocky.io/manage/delete/8d09ba26-8710-40da-ba8c-457fda7191fe/sFaGGd0qNjUQ08inC469xkMwYFfwztDPqMQF
```

```
run.mocky.io/v3/6af6b538-2508-402 X
```

← → ↻

JSON Необработанные данные За

Сохранить Копировать Свернуть все F

key: "Hello!"

Плюсы Mocky

- ✓ Бесплатно
- ✓ Не требует регистрации


Минусы Mocky

- × Сервер поднят не локально
- × Простой HTTP мок, без возможности управления поведением/генерации данных

Итог по Mocky

1. Подходит для простых моков
2. Бесплатная альтернатива postman

Итоги по UI инструментам

Инструмент	Стоимость	Поддерживаемые протоколы	Облачное решение	Поддержка OpenApi
Postman	Бесплатная версия с ограничениями	HTTP(S) gRPC GraphQL	Да	Нет
SwaggerHub 	Бесплатная версия с ограничениями	HTTP(S)	Да	Да
Mocky	Бесплатно	HTTPS	Да	Нет

Обзор CLI/CODE ИНСТРУМЕНТОВ




MOCKINTOSH





Castle Mock



Castle Mock 

Home / Project / Application / Resource / 1

Method: 1 Update method Create response Delete method

Type GET
Status Mocked
Response strategy Random
Address http://localhost:8080/castlemock/mock/rest/project/QK1Tv/application/5IzeG3/HisenBug
Forwarded endpoint
Simulate network delay
Network delay 0 ms
Default response

Mock responses

<input type="checkbox"/> Name ↑	Status ↑	HTTP status code ↑
<input type="checkbox"/> eer	Enabled	200

10 1

Update status Duplicate Delete mock response



1. Скачать Java Web Archive файл и развернуть в Tomcat или запустить в Docker
2. Перейти по адресу <http://localhost:8080/castlemock>
3. Для входа в ПУ ввести пароль и логин admin/admin

<https://github.com/castlemock/castlemock/wiki/Use-Castle-Mock#install-and-run-castle-mock>






Castle Mock

Login with your Castle Mock ID

SIGN IN →




Пример панели


Castle Mock 





[Home](#) / Heisenbug


Project: Heisenbug

 Update project

 Create application

 Upload ▾

 Export project

 Delete project

Description

Type

REST

Applications

<input type="checkbox"/>	Name ↑↓	Mocked ↑↓	Disabled ↑↓	Forwarded ↑↓	Recording ↑↓	Record once ↑↓	Echo ↑↓
Click on the 'Upload' to upload a REST API definition							

10 ▾

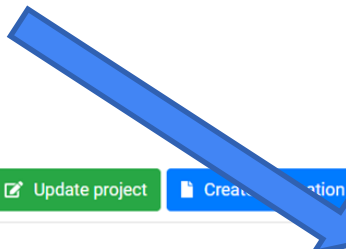
 Update status

 Update endpoint

 Delete application



Функция загрузки и выгрузки шаблона мокируемого контракта



[Home](#) / Heisenbug

Project: Heisenbug

[Update project](#) [Create application](#) [Upload](#) [Export project](#) [Delete project](#)

Description

Type REST

- RAML
- Swagger
- WADL

Applications

Search

<input type="checkbox"/>	Name ↑↓	Mocked ↑↓	Disabled ↑↓	Forwarded ↑↓	Recording ↑↓	Record once ↑↓	Echo ↑↓
Click on the 'Upload' to upload a REST API definition							

10 ▾

[Update status](#) [Update endpoint](#) [Delete application](#)



Home / Heisenbug

Project: Heisenbug

Update project

Create application

Upload

Export project

Delete project

Description

Type

REST

Create application

Name

Create

Applications

<input type="checkbox"/>	Name ↑↓	Mocked ↑↓	Disabled ↑↓	Forwarded ↑↓	Recording ↑↓	Record once ↑↓	Echo ↑↓
<input type="checkbox"/>	HEISENBUG	0	0	0	0	0	0

10

1

Update status

Update endpoint

Delete application



[Home](#) / Heisenbug

Project: Heisenbug

Update project

Create application

Upload ▾

Export project

Delete project

Description

Type

REST

Applications

Search

<input type="checkbox"/>	Name ↑↓	Mocked ↑↓	Disabled ↑↓	Forwarded ↑↓	Recording ↑↓	Record once ↑↓	Echo ↑↓
<input type="checkbox"/>	HEISENBUG	0	0	0	0	0	0

10 ▾

1

Update status

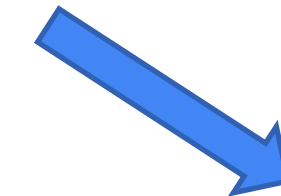
Update endpoint

Delete application



[Home](#) / [Project](#) / HEISENBUG

Application: HEISENBUG



[Update application](#) [Create resource](#) [Delete application](#)

Resources

<input type="checkbox"/>	Name ↑↓	Mocked ↑↓	Disabled ↑↓	Forwarded ↑↓	Recording ↑↓	Record once ↑↓	Echo ↑↓
--------------------------	---------	-----------	-------------	--------------	--------------	----------------	---------

10 ▾

[Update status](#) [Update endpoint](#) [Delete resource](#)



Home / Project / HEISENBUG

Application: HEISENBUG

[Update application](#) [Create resource](#) [Delete application](#)

Resources

Name ↑↓ Mocked ↑↓

10 ▾

[Update status](#) [Update endpoint](#)

Search

Force ↑↓ Echo ↑↓

Create resource

Name

Uri

[Create](#)



[Home](#) / [Project](#) / [Application](#) / Test1

Resource: Test1

[Update resource](#) [Create method](#) [Delete resource](#)

URI /test1
Address http://localhost:8080/castlemock/mock/rest/project/kbYNhK/application/YIY67C/test1

Methods

Search

<input type="checkbox"/>	Name ↑↓	Method type ↑↓	Method Status ↑↓	Forwarded endpoint ↑↓
--------------------------	---------	----------------	------------------	-----------------------

10 ▾

[Update status](#) [Update endpoint](#) [Delete method](#)



Home / Project / Application / Test1

Resource: Test1

URI: /test1

Address: http://

Update resource Create method Delete resource

Methods

Name ↑

10

Update status Update endpoint Delete method

Search

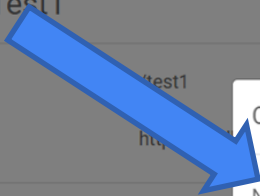
Forwarded endpoint ↑↓

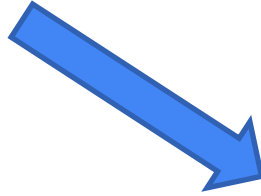
Create method

Name: users

HTTP method: GET

Create





[Home](#) / [Project](#) / [Application](#) / [Resource](#) / users

Method: users

[✎ Update method](#) [📄 Create response](#) [🗑 Delete method](#)

Type	GET
Status	Mocked
Response strategy	Random
Address	http://localhost:8080/castlemock/mock/rest/project/kbYNhK/application/YIY67C/test1
Forwarded endpoint	
Simulate network delay	<input type="checkbox"/>
Network delay	0 ms
Default response	

Mock responses

Гибкая настройка параметров мокируемого эндпоинта

Update method ✕

Name	<input type="text" value="1"/>
Type	<input type="text" value="GET"/>
Status	<input type="text" value="Mocked"/>
Response strategy	<input type="text" value="Random"/>
Forwarded endpoint	<input type="text"/>
Simulate network delay	<input type="checkbox"/>
Network delay	<input type="text" value="0"/>
Default response	<input type="text" value="-- select an option --"/>





[Home](#) / [Project](#) / [Application](#) / [Resource](#) / users

Method: users

Update method Create response Delete method

A large blue arrow points from the top right towards the 'Create response' button. The buttons are arranged horizontally: a green 'Update method' button with a pencil icon, a blue 'Create response' button with a document icon, and a red 'Delete method' button with a trash can icon.

Type	GET
Status	Mocked
Response strategy	Random
Address	http://localhost:8080/castlemock/mock/rest/project/kbYNhK/application/YIY67C/test1
Forwarded endpoint	
Simulate network delay	<input type="checkbox"/>
Network delay	0 ms
Default response	

Mock responses



Home / Project / Application / Resource / users

Method: users

[Update method](#) [Create response](#) [Delete](#)

Type GET
Status Mocked
Response strategy
Address
Forwarded endpoint
Simulate network delay
Network delay
Default response



Create response? ×

Name

Status

Mock responses



[Home](#) / [Project](#) / [Application](#) / [Resource](#) / [Method](#) / [Response1](#)

Mock Response: Response1

[Validate expression](#) [Delete mock response](#)

Name

HTTP Status code

Status

Use Expression



[Body](#) [Headers](#) [XPath](#) [JSON Path](#) [Parameter Queries](#) [Header Queries](#)

Add header

Name

Value

[+ Add header](#)

Name ↑	Value ↑
No headers	

[Update response](#) [Discard changes](#)



Mock Response: Response1

Name

HTTP Status code

Status

Use Expression

Body Headers XPath JSON Path Parameter Queries Header Queries



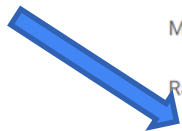
```
{"email": "${RANDOM_EMAIL(domain="castlemock",min=5,max=10,topDomain"com")}"}
```



Method: users

[Update method](#) [Create response](#) [Delete method](#)

Type	GET
Status	Mocked
Response strategy	Random
Address	http://localhost:8080/castlemock/mock/rest/project/kbYNhK/application/YIY67C/test1
Forwarded endpoint	
Simulate network delay	<input type="checkbox"/>
Network delay	0 ms
Default response	




Mock responses



GET ▾ p://localhost:8080/castlemock/mock/rest/project/kbYNhK/application/YIY67C/test1 Send ▾ 200 OK 6.41 ms 41 B 15 Minutes Ago ▾

Body ▾ Auth ▾ Query Headers Docs Preview ▾ Headers⁹ Cookies Timeline

```
1 {  
2   "email": "9chVbuOrcXTxW@castlemock.com"  
3 }
```



Enter a URL and send to get a response

Select a body type from above to send data in the body of a request

[Introduction to Insomnia](#) ↗



Плюсы CastleMock

- ✓ **Бесплатно**
- ✓ ПУ UI
- ✓ Разделение прав
- ✓ Логирование
- ✓ Работает с SOAP



Плюсы CastleMock

- ✓ Бесплатно
- ✓ **ПУ UI**
- ✓ Разделение прав
- ✓ Логирование
- ✓ Работает с SOAP



Плюсы CastleMock

- ✓ Бесплатно
- ✓ ПУ UI
- ✓ **Разделение прав**
- ✓ Логирование
- ✓ Работает с SOAP



Плюсы CastleMock

- ✓ Бесплатно
- ✓ ПУ UI
- ✓ Разделение прав
- ✓ **Логирование**
- ✓ Работает с SOAP



Плюсы CastleMock

- ✓ Бесплатно
- ✓ ПУ UI
- ✓ Разделение прав
- ✓ Логирование
- ✓ **Работает с SOAP**



Минусы CastleMock

- × Требуется сервер
(желательно с докером)



Итог по CastleMock

1. Мощная утилита
2. Бесплатная версия
3. Применяется на проектах для создание моков HTTP, SOAP

Mockintosh



MOCKINTOSH

<https://mockintosh.io/>



MOCKINTOSH



Untitled-1

```
mockintosh --sample-config example.yaml|
```

```
templatingEngine: Jinja2
services:
  - port: 8081
    endpoints:
      - path: "/example/{{ id }}"
        response: "id is {{ id }} and a random integer: {{ random.int(1, 100) }}"
```

<https://handlebarsjs.com/guide/#what-is-handlebars>

<https://jinja.palletsprojects.com/en/2.11.x/>

[HTTP Endpoints](#)[Async Actors](#)[Traffic Log](#)[Statistics](#)[Configuration](#)[Unhandled Requests](#)

Pick any endpoint below and use "Try it out" button to send the request

https://service1.example.com:8000

0.1.9OAS3

Automatically generated Open API Specification.

default



POST /v2/pet

PUT /v2/pet

GET /v2/pet/findByStatus

GET /v2/pet/findByTags

GET /v2/pet/{petId}

POST /v2/pet/{petId}

DELETE /v2/pet/{petId}



MOCKINTOSH

Итог по Mockintosh

1. **Бесплатный**
2. Работает с Apache Kafka, RabbitMQ, Apache ActiveMQ, Redis etc. or cloud services such as Google Cloud Pub/Sub, Amazon SQS
3. Поддерживается (но не активно)
4. Написан на Python



MOCKINTOSH

Итог по Mockintosh

1. Бесплатный
2. **Работает с Apache Kafka, RabbitMQ, Apache ActiveMQ, Redis etc. or cloud services such as Google Cloud Pub/Sub, Amazon SQS**
3. Поддерживается (но не активно)
4. Написан на Python



MOCKINTOSH

Итог по Mockintosh

1. Бесплатный
2. Работает с Apache Kafka, RabbitMQ, Apache ActiveMQ, Redis etc. or cloud services such as Google Cloud Pub/Sub, Amazon SQS
3. **Поддерживается (но не активно)**
4. Написан на Python



MOCKINTOSH

Итог по Mockintosh

1. Бесплатный
2. Работает с Apache Kafka, RabbitMQ, Apache ActiveMQ, Redis etc. or cloud services such as Google Cloud Pub/Sub, Amazon SQS
3. Поддерживается (но не активно)
4. **Написан на Python**

Citrus Simulator

<https://github.com/citrusframework/citrus-simulator>



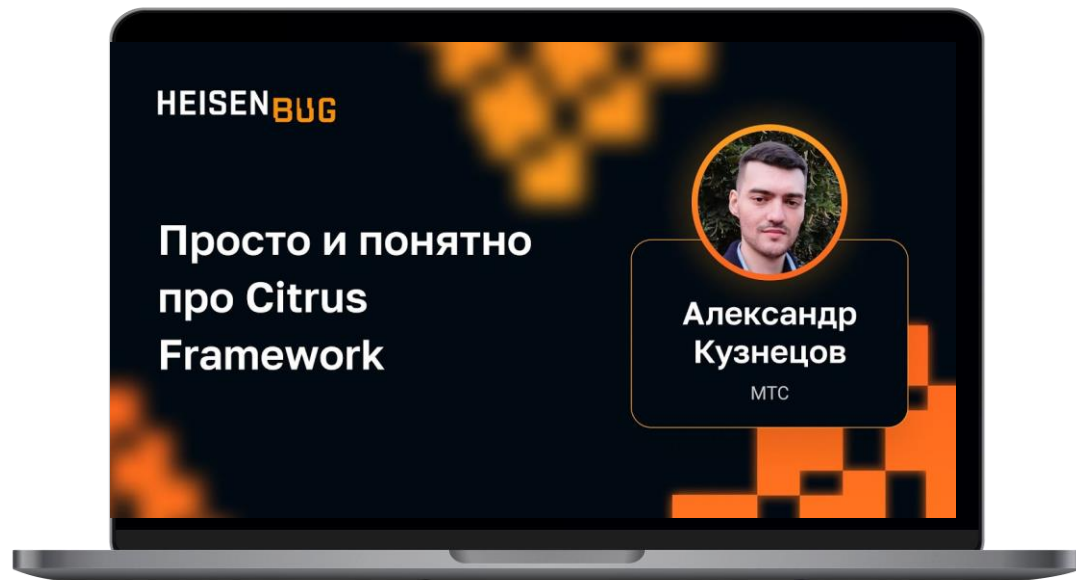
Simulator: REST Petstore Simulator

Name: REST Petstore Simulator

Version: 1.0.0-SNAPSHOT

Status
















```
/** Test Http REST client */
@Autowired
private HttpClient simulatorClient;

/**
 * Sends a hello request to server expecting positive response message.
 */
@CitrusTest
public void testHelloRequest() {
    $(http().client(simulatorClient)
        .send()
        .post("hello")
        .message()
        .contentType(MediaType.APPLICATION_XML_VALUE)
        .body("<HelloHello xmlns=\"http://citrusframework.org/schemas/hello\">"
            + "Hello!" +
            "</HelloHello>"));

    $(http().client(simulatorClient)
        .receive()
        .response(HttpStatus.OK)
        .message()
        .body("<Response xmlns=\"http://citrusframework.org/schemas/hello\">" +
            "Hi!" +
            "</Response>"));
}
```



 sample-combined	Use next snapshot version 2.2.0-SNAPSHOT
 sample-jms-fax	chore(deps): include jdk17 compatible jaxb-runtime
 sample-jms	Use next snapshot version 2.2.0-SNAPSHOT
 sample-mail	Use next snapshot version 2.2.0-SNAPSHOT
 sample-rest	Use next snapshot version 2.2.0-SNAPSHOT
 sample-swagger	chore(deps): bump com.consol.citrus and corresponding deps
 sample-ws-client	Use next snapshot version 2.2.0-SNAPSHOT
 sample-ws	Use next snapshot version 2.2.0-SNAPSHOT
 sample-wsdl	Use next snapshot version 2.2.0-SNAPSHOT



Плюсы Citrus Simulator

- ✓ Бесплатно
- ✓ Реализует любой из интерфейсов: SOAP (WSDL), REST, JMS, File
- ✓ Мощное моделирование реакции (динамические идентификаторы, моделирование ошибок, извлечение значений и т. д.)
- ✓ Простое определение сообщения (Citrus Java DSL)
- ✓ Проверка схемы XSD для входящих запросов
- ✓ Обработка формата сообщения JSON
- ✓ Мощная поддержка заголовков (действие SOAP, фрагменты заголовков SOAP, заголовки Http)
- ✓ Автоматическое создание симулятора из спецификаций Swagger Open API
- ✓ Автоматическое создание симулятора из файлов SOAP WSDL



Минусы Citrus Simulator

- × Есть баги
- × Нет возможности редактировать/создавать заглушки через UI

Microcks.io



microcks.io



Microcks-UI | Meet - Sync Microcks Labs test | localhost:4200/#/services/62a9a90efc6a1a7dab01dfa1

Mettre à jour

MICROCKS | Anonymous Admin

- Dashboard
- APIs | Services
- Importers
- Microcks Hub
- Administration

Services & APIs > API Pastry - 2.0 - 2.0.0

API Pastry - 2.0 - 2.0.0

Created 3 months ago, last update on Aug 26, 2022, 4:12:54 PM

domain=pastry status=GA team=A [Edit Labels](#)

PROPERTIES

Service Type: REST

Contracts: [API Pastry - 2.0-2.0.0.yaml](#) [OPEN_API_SPEC](#)
[View Documentation](#)

Statistics: [Mocks invocations](#)

OPERATIONS

Method	Path	Dispatcher	Sample(s)
GET	/pastry	SCRIPT dispatcher	1 sample(s)
GET	/pastry/{name}	URI_PARTS dispatcher	3 sample(s)
PATCH	/pastry/{name}	SCRIPT dispatcher	2 sample(s)

Tests

Conformance index

A B C D E

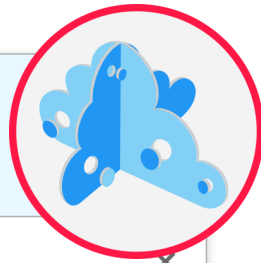
Conformance score

82.5% →

[Learn More](#)

[View Tests Runs](#) [+ New Test](#)





GET /hello

GET with dispatcher

1 sample(s)

DISPATCHING PROPERTIES

Dispatcher: [Learn More](#)
Dispatching rules: -

HTTP Verb: GET
Default delay: 0 ms

MOCKS

dynamic

Request

Mock URL

http://localhost:4200/rest/Hello+Dynamic+API/1.0.0/hello

```
{ "name": "Laurent" }
```

Header name	Values
Accept	application/json
Content-Type	application/json

Response

Response Code and Type 200: application/json

```
{  
  "id": "{{ randomString(64) }}",  
  "date": "{{ now(dd/MM/yyyy) }}",  
  "message": "Hello {{ request.body/name }}!"  
}
```



```
1 docker run -it quay.io/microcks/microcks-cli:latest microcks-cli
  test 'Beer Catalog API:0.9' \
2     http://beer-catalog-impl-beer-catalog-
  dev.apps.144.76.24.92.nip.io/api/ POSTMAN \
3     --microcksURL=http://microcks.apps.144.76.24.92.nip.io/api/ \
4     --keycloakClientId=microcks-serviceaccount \ --
  keycloakClientSecret=7deb71e8-8c80-4376-95ad-00a399ee3ca1 --
  waitFor=8sec \
5     --operationsHeaders='{ "globals": [{"name": "x-api-key",
  "values": "my-values"}], "GET /beer": [{"name": "x-trace-id",
  "values": "xcvbnsdfghjklm"}]}'
```

```
1 microcks-cli test <apiName:apiVersion>
  <testEndpoint> <runner> \
2     --microcksURL=<> --waitFor=5sec \
3     --keycloakClientId=<> --keycloakClientSecret=<>
```




Итог по Microcks

1. **Бесплатный**
2. Работает с OpenAPI specs, AsyncAPI specs, gRPC protobuf, GraphQL schema, Postman collections, SoapUI projects
3. Поддерживается
4. Запуск с Podman, Docker, Kubernetes
5. CLI
6. Плагин Github action, Jenkins
7. Работа через API



Итог по Microcks

1. Бесплатный
2. **Работает с OpenAPI specs, AsyncAPI specs, gRPC protobuf, GraphQL schema, Postman collections, SoapUI projects**
3. Поддерживается
4. Запуск с Podman, Docker, Kubernetes
5. CLI
6. Плагин Github action, Jenkins
7. Работа через API



Итог по Microcks

1. Бесплатный
2. Работает с OpenAPI specs, AsyncAPI specs, gRPC protobuf, GraphQL schema, Postman collections, SoapUI projects
- 3. Поддерживается**
4. Запуск с Podman, Docker, Kubernetes
5. CLI
6. Плагин Github action, Jenkins
7. Работа через API



Итог по Microcks

1. Бесплатный
2. Работает с OpenAPI specs, AsyncAPI specs, gRPC protobuf, GraphQL schema, Postman collections, SoapUI projects
3. Поддерживается
- 4. Запуск с Podman, Docker, Kubernetes**
5. CLI
6. Плагин Github action, Jenkins
7. Работа через API



Итог по Microcks

1. Бесплатный
2. Работает с OpenAPI specs, AsyncAPI specs, gRPC protobuf, GraphQL schema, Postman collections, SoapUI projects
3. Поддерживается
4. Запуск с Podman, Docker, Kubernetes
- 5. CLI**
6. Плагин Github action, Jenkins
7. Работа через API



Итог по Microcks

1. Бесплатный
2. Работает с OpenAPI specs, AsyncAPI specs, gRPC protobuf, GraphQL schema, Postman collections, SoapUI projects
3. Поддерживается
4. Запуск с Podman, Docker, Kubernetes
5. CLI
- 6. Плагин Github action, Jenkins**
7. Работа через API



Итог по Microcks

1. Бесплатный
2. Работает с OpenAPI specs, AsyncAPI specs, gRPC protobuf, GraphQL schema, Postman collections, SoapUI projects
3. Поддерживается
4. Запуск с Podman, Docker, Kubernetes
5. CLI
6. Плагин Github action, Jenkins
- 7. Работа через API**

Итоги по CLI/Code

Инструмент	Стоимость	Поддерживаемые протоколы	Облачное решение
Castle Mock	Продукт с открытым кодом	HTTP(S)/SOAP	Нет
Mockintosh	Продукт с открытым кодом	HTTP(S), Apache Kafka, RabbitMQ, Apache ActiveMQ, Redis	Нет
Citrus Simulator	Продукт с открытым кодом	HTTP(S), SMTP, WS	Нет
Microcks 	Продукт с открытым кодом	HTTP(S), TCP, SMTP	Нет

Обзор комбинированных инструментов





github.com/SmartBear/soapui

The screenshot shows the SoapUI 5.4.0 interface. The main window displays a REST client configuration for a GET request to the endpoint `https://sarojk-dev-ed.my.salesforce.com/services/data/v41.0/subjects/Account/00100000028Tze`. The request is configured with an Authorization header: `Bearer 00d0000000mohAR...`. The response is displayed in JSON format, showing account and shipping address details.

```
1 {
2   "attributes": {
3     "type": "Account",
4     "url": "/services/data/v41.0/subjects/Account/00100000028TzeAAB"
5   },
6   "id": "001100000028TzeAAB",
7   "isDeleted": false,
8   "MasterRecordId": null,
9   "Name": "United Oil & Gas Corp.",
10  "Type": "Customer - Direct",
11  "ParentId": null,
12  "BillingStreet": "1301 Avenue of the Americas \nNew York, NY 10019\nUSA",
13  "BillingCity": "New York",
14  "BillingPostalCode": null,
15  "BillingCountry": null,
16  "BillingLatitude": null,
17  "BillingLongitude": null,
18  "BillingGeocodeAccuracy": null,
19  "BillingAddress": {
20    "city": "New York",
21    "country": null,
22    "geocodeAccuracy": null,
23    "latitude": null,
24    "longitude": null,
25    "postalCode": null,
26    "state": "NY",
27    "street": "1301 Avenue of the Americas \nNew York, NY 10019\nUSA"
28  },
29  "ShippingStreet": "1301 Avenue of the Americas \nNew York, NY 10019\nUSA",
30  "ShippingCity": null,
31  "ShippingState": null,
32  "ShippingPostalCode": null,
33  "ShippingCountry": null,
34  "ShippingLatitude": null,
35  "ShippingLongitude": null,
36  "ShippingGeocodeAccuracy": null,
37  "ShippingAddress": {
38    "city": null,
39    "country": null,
40    "geocodeAccuracy": null,
41    "latitude": null,
42    "longitude": null,
43    "postalCode": null,
44    "state": null,
45    "street": "1301 Avenue of the Americas \nNew York, NY 10019\nUSA"
46  },
47 }
```





Empty SOAP REST Import Save All Forum Trial Preferences Proxy

Endpoint Explorer

Search Forum



Online Help

Navigator



Projects

Inspector

Project 1
REST MockService 1
GET /users
REST Response 1

Project 1

REST MockService 1

Actions

REST Response 1

GET /users

Method: GET, Resource Path: /users

MockResponses

REST Response 1

Description

Enable

Message Log

Dispatch: SEQUENCE

Value

MockAction Properti...

Property	Value
Name	/users
Descri...	

$$\sum_{m=1}^{\infty} q_m(\omega) \int_0^{L_0} \left\{ (1 + i\eta) \frac{d^2}{dx^2} \left[k(x) \frac{d^2 \psi_m(x)}{dx^2} \right] - \omega^2 \psi_m(x) \right.$$

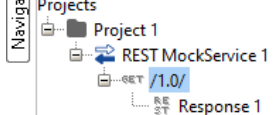
$$\times \left[\rho_l(x) + \frac{\pi}{4} \rho_f b^2(x) \Gamma(\beta(x, \omega), \alpha(x)) \right] \psi_n(x) dx$$

$$= \omega^2 \int_0^{L_0} \left\{ \hat{\theta}_B(\omega)(x + L_0) \left[\rho_l(x) + \frac{\pi}{4} \rho_f b^2(x) \Gamma(\beta(x, \omega), \alpha(x)) \right] \right.$$

$$\left. + \frac{\pi}{4} \rho_f b^2(x) \Delta \left(\beta(x, \omega), \frac{1}{b(x)} \left| \sum_{m=1}^{\infty} q_m(\omega) \psi_m(x) \right. \right. \right.$$

$$\left. \left. + \hat{\theta}_B(\omega)(x + L_0) \right|, \alpha(x) \right)$$

$$\times \left[\sum_{m=1}^{\infty} q_m(\omega) \psi_m(x) + \hat{\theta}_B(\omega)(x + L_0) \right] \psi_n(x) dx.$$



Method: GET Resource Path: /1.0/

MockResponses

Response 1

Dispatch: SCRIPT Default Response: Response 1

SCRIPT is invoked with log, context, requestContext, mockRequest and mockOperation...

```
1 /*
2 // Script dispatcher is used to select a response based on the incoming request.
3 // Here are few examples showing how to match based on path, query param, header and body
4
5 // Match based on path
6 def requestPath = mockRequest.getPath()
7 log.info "Path: " + requestPath
8
9 if( requestPath.contains("json") ) {
10 {
11 // return the name of the response you want to dispatch
12 return "JSON Response"
13 }
14
15
16 // Match based on query parameter
17 def queryString = mockRequest.getRequest().getQueryString()
18 log.info "QueryString: " + queryString
19
20 if( queryString.contains("stockholm") ) {
21 {
22 // return the name of the response you want to dispatch
23 return "Response Stockholm"
24 }
```



SCRIPT Позволяет выстраивать ответ на основе входящих данных.

The screenshot displays a REST client interface. On the left, a tree view shows a project named 'REST Project 1' containing a request for 'http://test.com'. A context menu is open over this request, listing several actions: 'Show Request Editor' (Enter), 'Add to TestCase' (Ctrl+Alt-A), 'Add to REST mock service' (Ctrl+Alt-M), 'Clone Request' (F9), 'Rename' (F2), and 'Delete' (Delete). The 'Add to REST mock service' option is highlighted. The main interface shows 'REST Project 1' with tabs for 'Overview', 'TestSuites', 'WS-Security Configurations', and 'Security'. Below these are summary sections: 'Project Summary', 'Interface Summary' (listing 'http://test.com' and 'http://test.com.wadl'), 'Tests Summary' (listing TestSuites, TestCases, TestSteps, Assertions, and LoadTests, all with a count of 0), and 'SOAP Mock Summary' (listing WsdMockServices, WsdMockOperations, and WsdMockResponses, all with a count of 0). A script editor at the bottom shows a snippet of XML/JSON with comments like '// Script dispatch' and '// Here are fe'.



Создание моков на основе запросов.

Code Blame 48 lines (32 loc) · 1.37 KB


```

1  @echo off
2
3  set SOAPUI_HOME=%~dp0
4  if exist "%SOAPUI_HOME%\jre\bin" goto SET_BUNDLED_JAVA
5
6  if exist "%JAVA_HOME%" goto SET_SYSTEM_JAVA
7
8  echo JAVA_HOME is not set, unexpected results may occur.
9  echo Set JAVA_HOME to the directory of your local JDK to avoid this message.
10 goto SET_SYSTEM_JAVA
11
12 :SET_BUNDLED_JAVA
13 set JAVA=%SOAPUI_HOME%\jre\bin\java
14 goto END_SETTING_JAVA

```

Go to file

- dist_standalone/bin
 - installationcomplete.bat
 - installationcomplete.sh
 - loadtestrunner.bat
 - loadtestrunner.sh
 - mockservicerunner.bat
 - mockservicerunner.sh
 - securitytestrunner.bat
 - securitytestrunner.sh

 Запуск мока из командной строки

```

1 C:\Program Files\SmartBear\soapUI-4.5.2\bin\mockservicerunner.bat "C:\Path\my-soapui-project.xml"

```


Плюсы SoapUI

- ✓ **Бесплатно**
- ✓ Не требует регистрации
- ✓ Многофункциональный
- ✓ Maven Plugin

Плюсы SoapUI

- ✓ Бесплатно
- ✓ **Не требует регистрации**
- ✓ Многофункциональный
- ✓ Maven Plugin

Плюсы SoapUI

- ✓ Бесплатно
- ✓ Не требует регистрации
- ✓ **Многофункциональный**
- ✓ Maven Plugin

Плюсы SoapUI

- ✓ Бесплатно
- ✓ Не требует регистрации
- ✓ Многофункциональный
- ✓ **Maven Plugin**

Минусы SoapUI

- × **Перегруженный UI**
- × В бесплатной версии баги
- × Groovy

Минусы SoapUI

- × Перегруженный UI
- × **В бесплатной версии баги**
- × Groovy

Минусы SoapUI

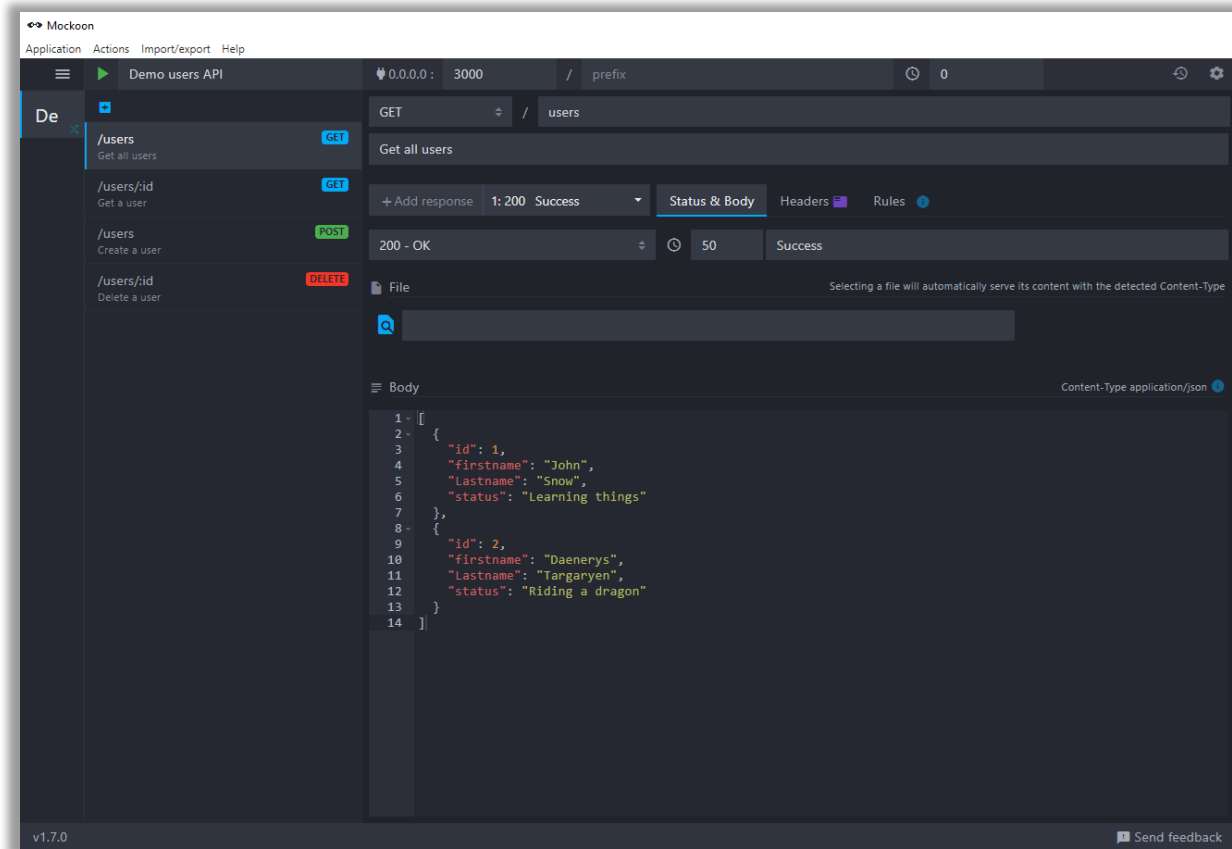
- × Перегруженный UI
- × В бесплатной версии баги
- × **Groovy**

Итог по SoapUI

1. Мощная утилита
2. Бесплатная версия
3. Создание мока для REST/SOAP



mockoon.com





Routes 1 Headers 1 Logs Proxy Settings

Getting started API
0.0.0.0:3000/

Filter GET / Path supports regexes and params: */:var/a(b)?c/[0-9]{1,5}

/ GET Add notes or documentation

Response 1 (200)

Status & Body Headers Rules Settings

200 - OK 0 Add label or notes

File

Body

```
1 {}
```

application/json [Open templating documentation](#)

Mockoon CLI is now available! x



Getting started API
0.0.0.0:3000/

Routes 2 Headers 1 Logs Proxy Settings

Filter POST / tutorials

GET / Add notes or documentation

POST /tutorials

Response 1 (200)

Status & Body Headers Rules Settings

200 - OK 0 Add label or notes

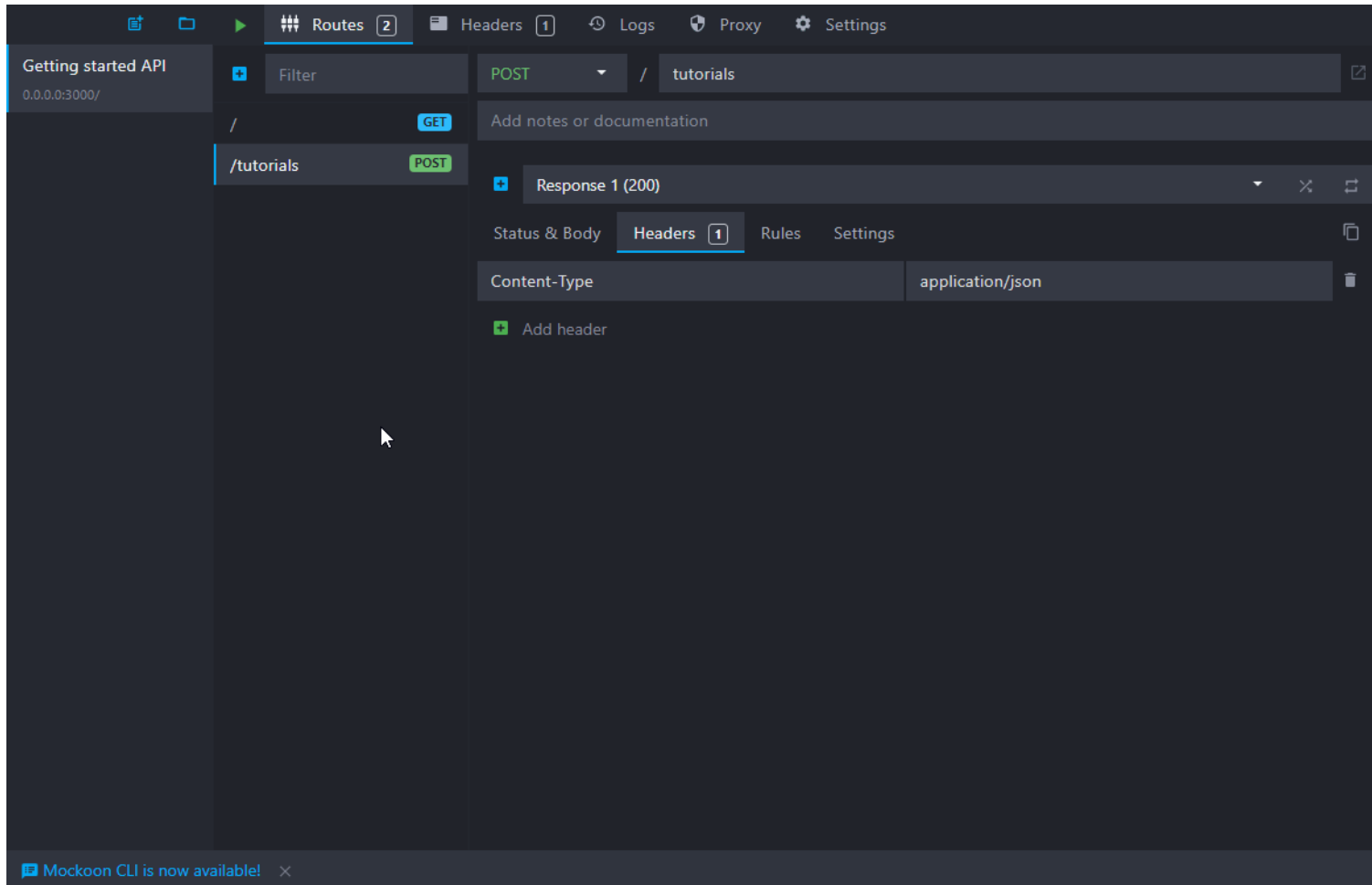
File

Body

```
1 {}
```

application/json [Open templating documentation](#)

Mockoon CLI is now available! x

A screenshot of the Mockoon web interface. The top navigation bar includes icons for a clipboard, a folder, a play button, and tabs for "Routes" (2), "Headers" (1), "Logs", "Proxy", and "Settings". The main area is divided into three panels. The left panel shows a sidebar with "Getting started API" and "0.0.0.0:3000/". The middle panel displays a route for "POST /tutorials" with a "POST" method label. The right panel shows a "Response 1 (200)" with tabs for "Status & Body", "Headers" (1), "Rules", and "Settings". The "Headers" tab is active, showing a single header: "Content-Type" with the value "application/json". There is an "Add header" button below the header list. At the bottom of the interface, a notification banner reads "Mockoon CLI is now available!".

GET http://localhost:3000/template Send 200 OK 13.8 ms 395 B Just Now

Body Auth Query Headers Docs Preview Headers⁵ Cookies Timeline

```

1 { "users": [
2   {{# repeat (queryParams 'total' '5') }}
3   {
4     "userId": "{{ faker 'datatype.number' min=10000 max=100000 }}",
5     "firstname": "{{ faker 'name.firstName' }}",
6   },
7   {{/ repeat }}
8 ]
9 }

```

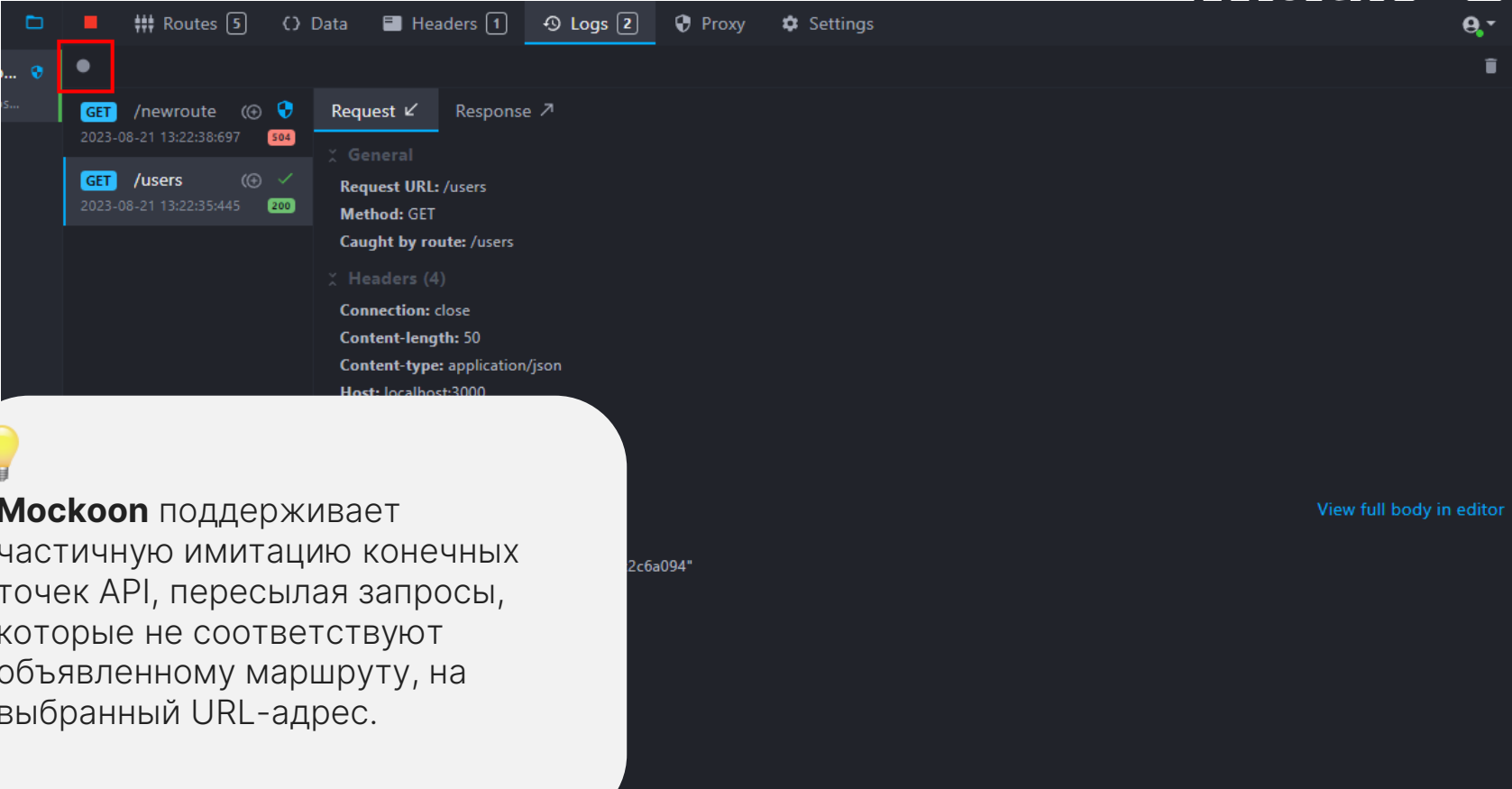
Enter a URL and send to get a response

Select a body type from above to send data in the body of a request

```

1 {
2   "users": [
3     {
4       "userId": "64463",
5       "firstname": "Theo",
6     },
7     {
8       "userId": "91337",
9       "firstname": "Rosario",
10    },
11    {
12      "userId": "18072",
13      "firstname": "Alfred",
14    },
15    {
16      "userId": "63606",
17      "firstname": "Velva",
18    },
19    {
20      "userId": "67623",
21      "firstname": "Emmy",
22    }
23  ]
24 }

```



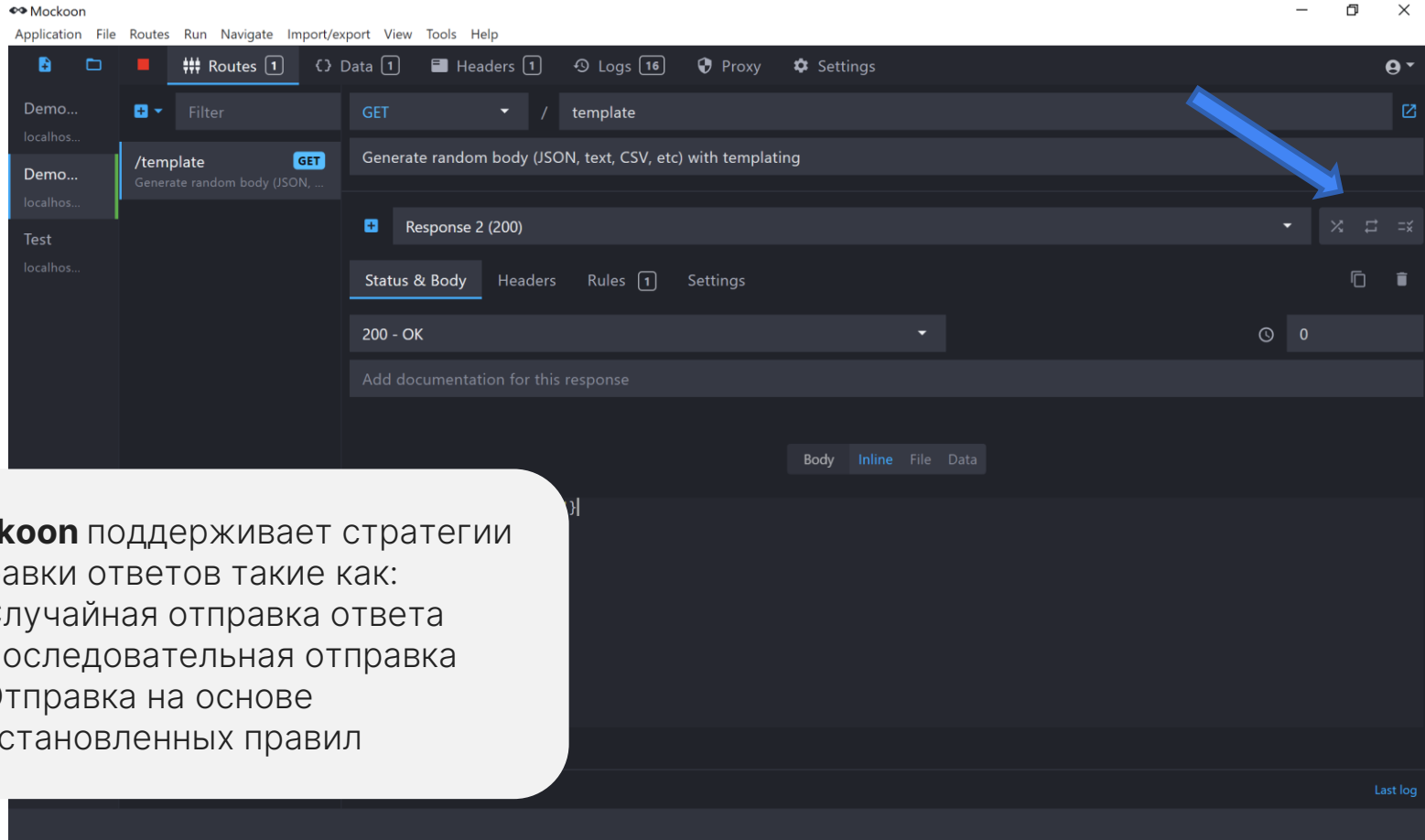
The screenshot shows the Mockoon application interface. At the top, there are tabs for Routes (5), Data, Headers (1), Logs (2), Proxy, and Settings. The Logs tab is active, showing a list of log entries. The first entry is a 504 error for a GET request to /newroute at 2023-08-21 13:22:38:697. The second entry is a 200 response for a GET request to /users at 2023-08-21 13:22:35:445. A red box highlights a small grey circle in the top-left corner of the application window. The details for the 200 response are expanded, showing the following information:

- Request URL: /users
- Method: GET
- Caught by route: /users
- Connection: close
- Content-length: 50
- Content-type: application/json
- Host: localhost:3000

At the bottom right of the interface, there is a link that says "View full body in editor".



Mockoon поддерживает частичную имитацию конечных точек API, пересылая запросы, которые не соответствуют объявленному маршруту, на выбранный URL-адрес.



The screenshot shows the Mockoon application interface. The top menu includes Application, File, Routes, Run, Navigate, Import/export, View, Tools, and Help. The main area displays a route configuration for a GET request to /template. The response is configured as 'Response 2 (200)' with a status of '200 - OK'. A blue arrow points to the 'Response 2 (200)' tab.

Mockoon поддерживает стратегии отправки ответов такие как:

- Случайная отправка ответа
- Последовательная отправка
- Отправка на основе установленных правил

Routes 1 Data 1 Headers 1 Logs 11 Proxy Settings

Filter GET / template

/template GET
Generate random body (JSON, ...)

Response 1 (200) Creates 10 random users, or the amount specified in the 'total' query param

Status & Body Headers Rules 1 Settings

200 - OK 0

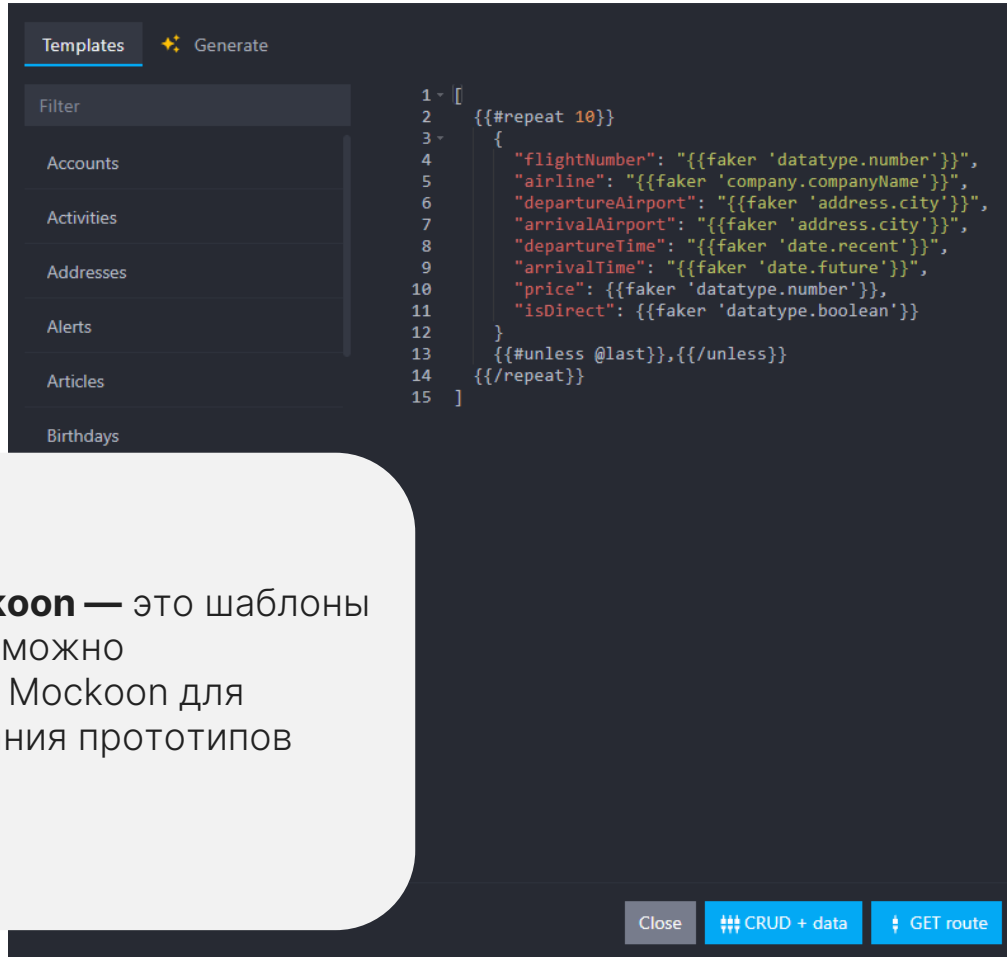
Creates 10 random users, or the amount specified in the 'total' query param

Body Inline File Data

```
1 {
2   "Templating example": "For more information about templating, click the blue 'i' above this editor",
3   "users": [
4     {#{ repeat (queryParams 'total' '10') }}
5     {
6       "userId": "{{ faker 'datatype.number' min=10000 max=100000 }}",
7       "firstname": "{{ faker 'name.firstName' }}",
8       "lastname": "{{ faker 'name.lastName' }}",
9       "friends": [
10        {#{ repeat (faker 'datatype.number' 5) }}
11        {
12          "id": "{{ faker 'datatype.uuid' }}"
13        }
14      ]
15    }
16  ]
17 }
```

Beautify JSON

application/json Last log



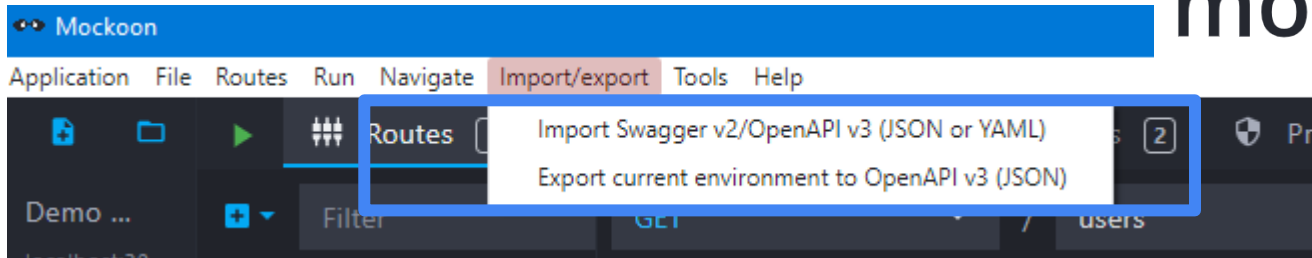
The screenshot shows the Mockoon interface with a 'Templates' sidebar on the left and a code editor on the right. The sidebar lists various API categories like Accounts, Activities, Addresses, Alerts, Articles, and Birthdays. The code editor displays a JSON template for a flight API endpoint, using Mustache syntax for dynamic values and control structures like repeat and unless.

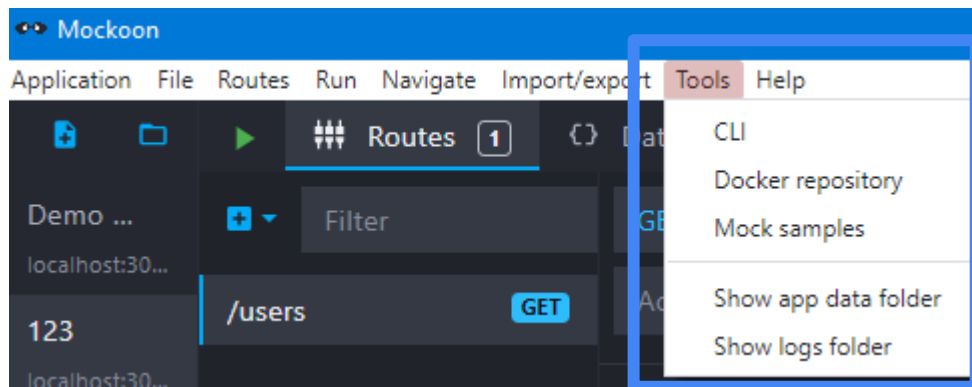
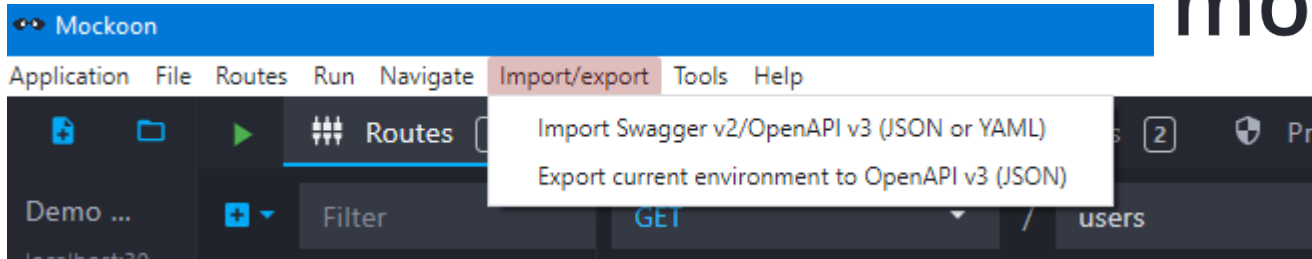
```
1 [
2   {{#repeat 10}}
3   {
4     "flightNumber": "{{faker 'datatype.number'}}",
5     "airline": "{{faker 'company.companyName'}}",
6     "departureAirport": "{{faker 'address.city'}}",
7     "arrivalAirport": "{{faker 'address.city'}}",
8     "departureTime": "{{faker 'date.recent'}}",
9     "arrivalTime": "{{faker 'date.future'}}",
10    "price": {{faker 'datatype.number'}},
11    "isDirect": {{faker 'datatype.boolean'}}
12  }
13  {{#unless @last}},{{/unless}}
14  {{/repeat}}
15 ]
```

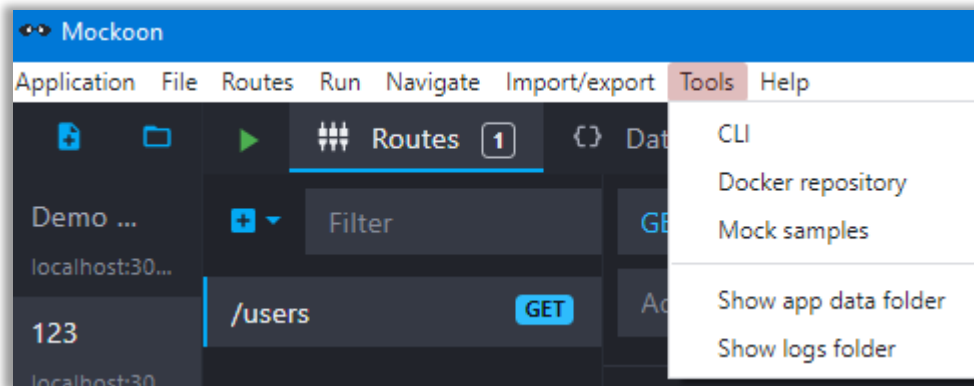
At the bottom of the interface, there are three buttons: 'Close', 'CRUD + data', and 'GET route'.



Шаблоны Mockoon — это шаблоны JSON, которые можно использовать в Mockoon для быстрого создания прототипов фиктивных API.



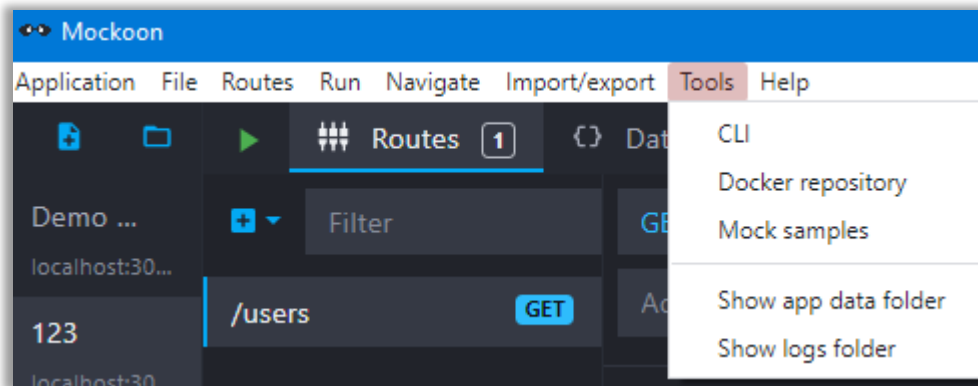




CLI example

```
npm install -g @mockoon/cli
```

```
mockoon-cli start --data ./data-file.json
```



CLI example

```
docker run -d --mount type=bind,source=/data-file.json,target=/data,readonly \
-p 3000:3000 mockoon/cli:latest \
-d data \
-p 3000
```

Плюсы Mockoon

- ✓ **Понятный интерфейс**
- ✓ Бесплатно
- ✓ Работает локально
- ✓ Работает с OpenApi
- ✓ Понятная документация
- ✓ CLI
- ✓ Docker
- ✓ Fakerjs
- ✓ Proxy

Плюсы Mockoon

- ✓ Понятный интерфейс
- ✓ **Бесплатно**
- ✓ Работает локально
- ✓ Работает с OpenApi
- ✓ Понятная документация
- ✓ CLI
- ✓ Docker
- ✓ Fakerjs
- ✓ Proxy

Плюсы Mockoon

- ✓ Понятный интерфейс
- ✓ Бесплатно
- ✓ **Работает локально**
- ✓ Работает с OpenApi
- ✓ Понятная документация
- ✓ CLI
- ✓ Docker
- ✓ Fakerjs
- ✓ Proxy

Плюсы Mockoon

- ✓ Понятный интерфейс
- ✓ Бесплатно
- ✓ Работает локально
- ✓ **Работает с OpenAPI**
- ✓ Понятная документация
- ✓ CLI
- ✓ Docker
- ✓ Faker.js
- ✓ Proxy

Плюсы Mockoon

- ✓ Понятный интерфейс
- ✓ Бесплатно
- ✓ Работает локально
- ✓ Работает с OpenApi
- ✓ **Понятная документация**
- ✓ CLI
- ✓ Docker
- ✓ Fakerjs
- ✓ Proxy

Плюсы Mockoon

- ✓ Понятный интерфейс
- ✓ Бесплатно
- ✓ Работает локально
- ✓ Работает с OpenApi
- ✓ Понятная документация
- ✓ **CLI**
- ✓ Docker
- ✓ Fakerjs
- ✓ Proxy

Плюсы Mockoon

- ✓ Понятный интерфейс
- ✓ Бесплатно
- ✓ Работает локально
- ✓ Работает с OpenApi
- ✓ Понятная документация
- ✓ CLI
- ✓ **Docker**
- ✓ Fakerjs
- ✓ Proxy

Плюсы Mockoon

- ✓ Понятный интерфейс
- ✓ Бесплатно
- ✓ Работает локально
- ✓ Работает с OpenApi
- ✓ Понятная документация
- ✓ CLI
- ✓ Docker
- ✓ **Fakerjs**
- ✓ Proxy

Плюсы Mockoon

- ✓ Понятный интерфейс
- ✓ Бесплатно
- ✓ Работает локально
- ✓ Работает с OpenApi
- ✓ Понятная документация
- ✓ CLI
- ✓ Docker
- ✓ Fakerjs
- ✓ **Proxy**

Минусы Mockoon

- ✗ Встроенный генератор данных не всё может генерировать (используем `fakerjs.dev`)
- ✗ Требуется изучать внутренний синтаксис генератора данных
- ✗ Ограничение в 10 темплейтов
- ✗ Есть платная версия, с более расширенным функционалом

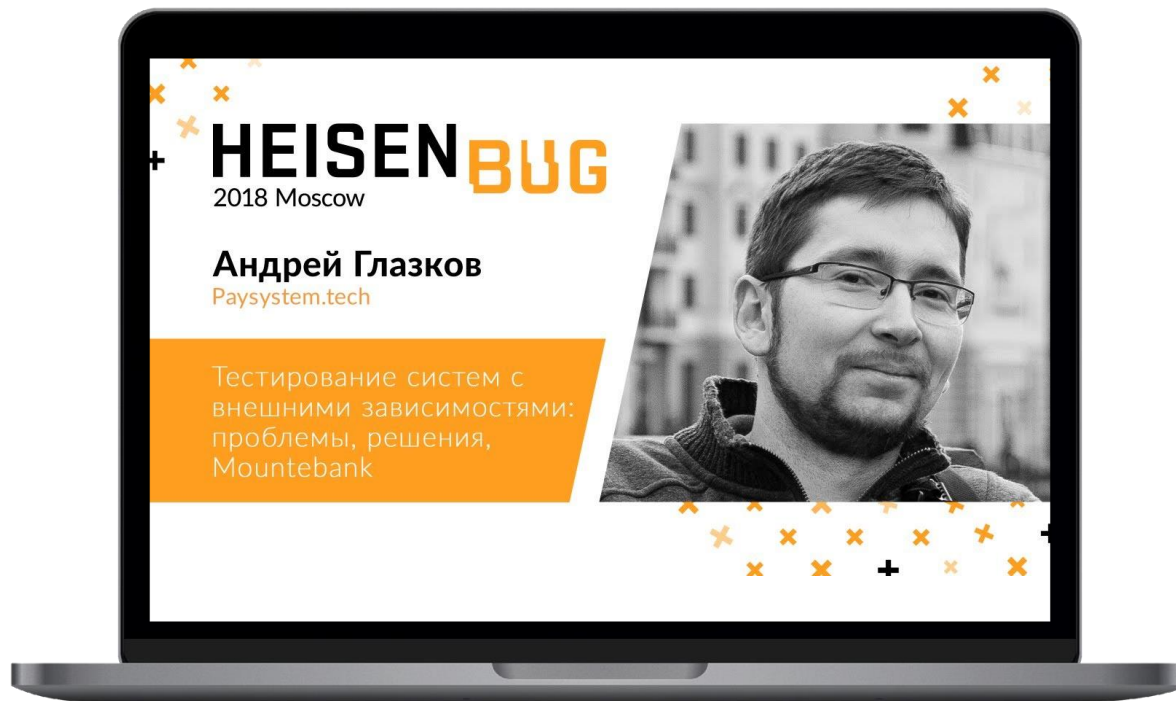


mountebank

www.mbttest.org



The screenshot shows the main page of the mountebank web application. At the top left is a circular profile picture of the pharmacist character. To its right is the title "mountebank - over the wire test doubles" and a search input field. Below the title is a navigation menu with links for "home", "imposters", "logs", "config", "metrics", and a RSS icon. The main content area is divided into two columns. The left column, titled "the apothecary", contains a vertical list of links: "getting started", "mental model", "3rd party plugins", "command line", "security", "faqs", "support", "api:", "overview", "contracts", "mock verification", "stub responses", "proxies", "injection", "behaviors", "faults", and "stub predicates". The right column features a large heading "Welcome, friend" followed by two paragraphs of introductory text. The first paragraph states that mountebank is an open-source tool for providing cross-platform, multi-protocol test doubles over the wire, designed to replace traditional stubs and mocks. The second paragraph highlights it as the most capable open-source service virtualization tool available. Below the text is a section titled "How it works" which explains that the tool uses a legion of "imposters" to act as on-demand test doubles, communicating with the application over HTTP. It details how users can set up stubs, record and replay proxies, and verify mock expectations. The final paragraph notes that imposters are configured at startup via a config file and respond to specific protocols based on the port and socket information provided.



[Андрей Глазков — Тестирование систем с внешними зависимостями: проблемы, решения, Mountebank](https://www.youtube.com/watch?v=Kc4wLfBDYfl)

<https://www.youtube.com/watch?v=Kc4wLfBDYfl>





```
1 npm install -g mountebank
```

```
2 mb start
```



```
1 docker pull bbyars/mountebank:2.8.2
```

```
2 docker run --rm -p 2525:2525 -p 4545:4545 -p 5555:5555
```

```
bbyars/mountebank:2.8.2 start
```



Client Libraries

While mountebank speaks [REST](#), he truly loves your native language and has no desire to make you learn another language. If you have a native library you'd like others to be able to use, please send mountebank a note on the [support page](#) or through a pull request to this page and he will add it to the list below.

Some of these are still in their early stages. Feel free to help the authors out ;>

Language	Name	Author
C#	MbDotNet	Matthew Herman
Clojure	Charlatan	Matthew Daley
Delphi	mountebank-delphi	Jamie Geddes
F#	MbDotNet.FSharp	Matthew Herman
Go	GoBank	Erkan
	mbgo	Senseye
Java	javabank	James Thomas
JavaScript	mountebank-helper	Alex
Perl	Test::Mountebank	Dagfinn Reiersøl
PHP	Juggler	Andrejs Mironovs
	mountebank-api-php	Demyanovsky Ruslan
	mountebank.php	Abraham Valdez

Mountebank Imposter UI [Home](#) [Collection Maintenance](#) [Mountebank JSON](#) [Import/Export](#) [Help](#)

Collection

Imposters

- Item 1 (GET)
- Item 2 (GET)

Documentation

Selected Response [+ Add Response](#) [- Delete Current Response](#)

1


Click To Use Injection All other criteria will be lost

Status

(Click on the section below)

Headers

Body



- [Mountebank-UI by Don Henton](http://donhenton.github.io/mountebank-UI/public_html/index.html#/) - http://donhenton.github.io/mountebank-UI/public_html/index.html#/
- [Disguise by Opus Software](https://github.com/Opus-Software/disguise) - <https://github.com/Opus-Software/disguise>
- [Mountebank UI by The Testing Gospels](https://github.com/testinggospels/mountebank-ui) - <https://github.com/testinggospels/mountebank-ui>



Итог по mountebank

1. Бесплатный
2. Многофункциональный (поддерживает **HTTP, HTTPS, TCP, SMTP**)
3. Множество доработок сторонними разработчиками (**IDAP, WebSockets, GraphQL, SNMP, Telnet, SSH**)

PRISM



stoplight.io/open-source/prism



Пример запуска CLI

```
[→ Stoplight prism mock --help
Start a mock server with the given spec file

USAGE
  $ prism mock SPEC

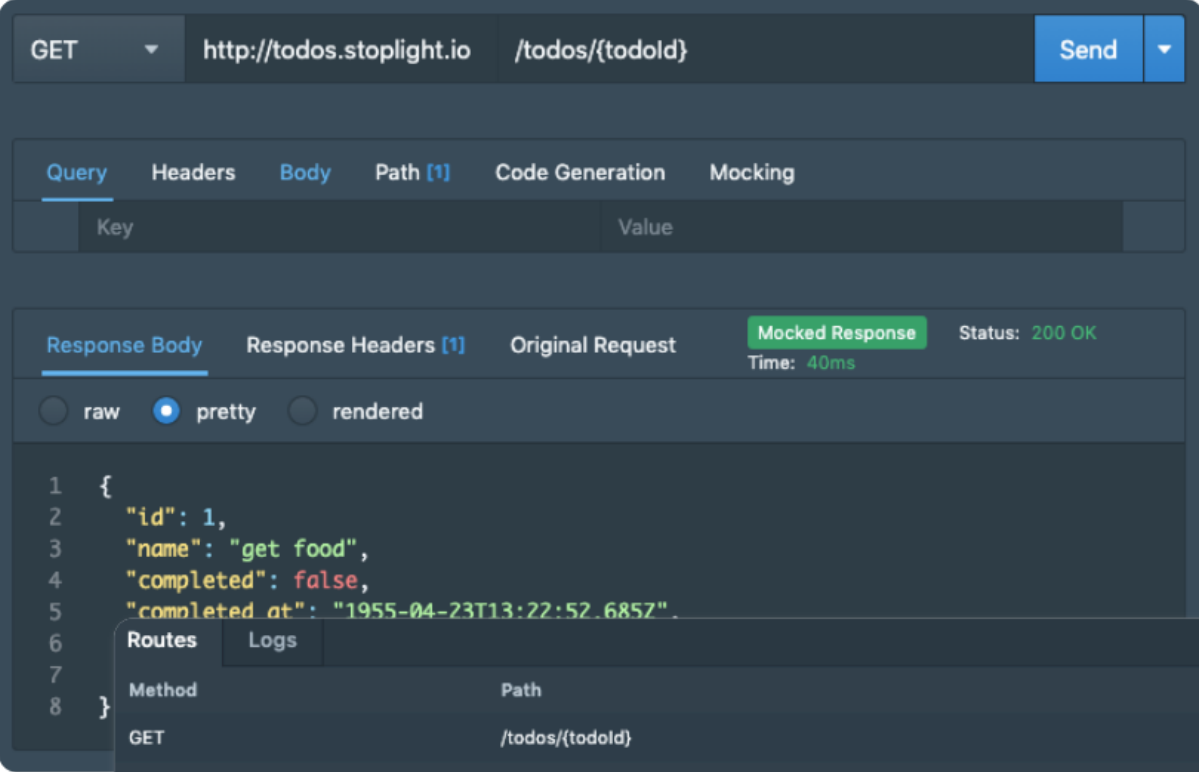
ARGUMENTS
  SPEC  Path to a spec file. Can be both a file or a fetchable resource on the web

OPTIONS
  -d, --dynamic      Dynamically generate examples.
  -h, --host=host  [default: 127.0.0.1] Host that Prism will listen to.
  -m, --multiprocess Fork the http server from the CLI
  -p, --port=port (required) [default: 4010] Port that Prism will run on.
```

Пример запуска CLI

```
→ Stoplight prism mock petstore.yml
[CLI] ... awaiting Starting Prism...
[HTTP SERVER] i info Server listening at http://127.0.0.1:4010
[CLI] • note GET http://127.0.0.1:4010/pets
[CLI] • note POST http://127.0.0.1:4010/pets
[CLI] • note GET http://127.0.0.1:4010/pets/{petId}
[CLI] ► start Prism is listening on http://127.0.0.1:4010
[HTTP SERVER] get /pets i info Request received
[NEGOTIATOR] i info Request contains an accept header: */*
[VALIDATOR] ⚠ warning Request did not pass the validation rules
[NEGOTIATOR] • note Unable to find a 422 response definition
[NEGOTIATOR] • note Unable to find a 400 response definition.
[NEGOTIATOR] ✓ success Created a 422 from a default response
[NEGOTIATOR] • note Unable to find a content with an example defined fo
[NEGOTIATOR] ✓ success The response 422 has a schema. I'll keep going with
[NEGOTIATOR] ✓ success Responding with the requested status code 422
```


Пример UI



The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** `http://todos.stoplight.io /todos/{todoid}`
- Response Status:** 200 OK
- Response Time:** 40ms
- Response Body (pretty):**

```
1 {
2   "id": 1,
3   "name": "get food",
4   "completed": false,
5   "completed_at": "1955-04-23T13:22:52.685Z"
6 }
```

A "Routes" panel is overlaid on the bottom right, displaying the following table:

Method	Path	Mock URL
GET	<code>/todos/{todoid}</code>	<code>http://127.0.0.1:3101/todos/{todoid}</code>
PUT	<code>/todos/{todoid}</code>	<code>http://127.0.0.1:3101/todos/{todoid}</code>
DELETE	<code>/todos/{todoid}</code>	<code>http://127.0.0.1:3101/todos/{todoid}</code>

Итог по Prism

1. Бесплатный CLI
2. Stoplight Studio: Free visual OpenAPI designer that comes integrated with mocking powered by Prism.
3. Понятный интерфейс в командной строке
4. OpenAPI v3.1, OpenAPI v3.0, OpenAPI v2.0 (formerly Swagger) and Postman Collections.
5. Запуск из под Docker
6. Библиотеки Faker (fakerjs.dev) и JSON Schema Faker (json-schema-faker.js.org)

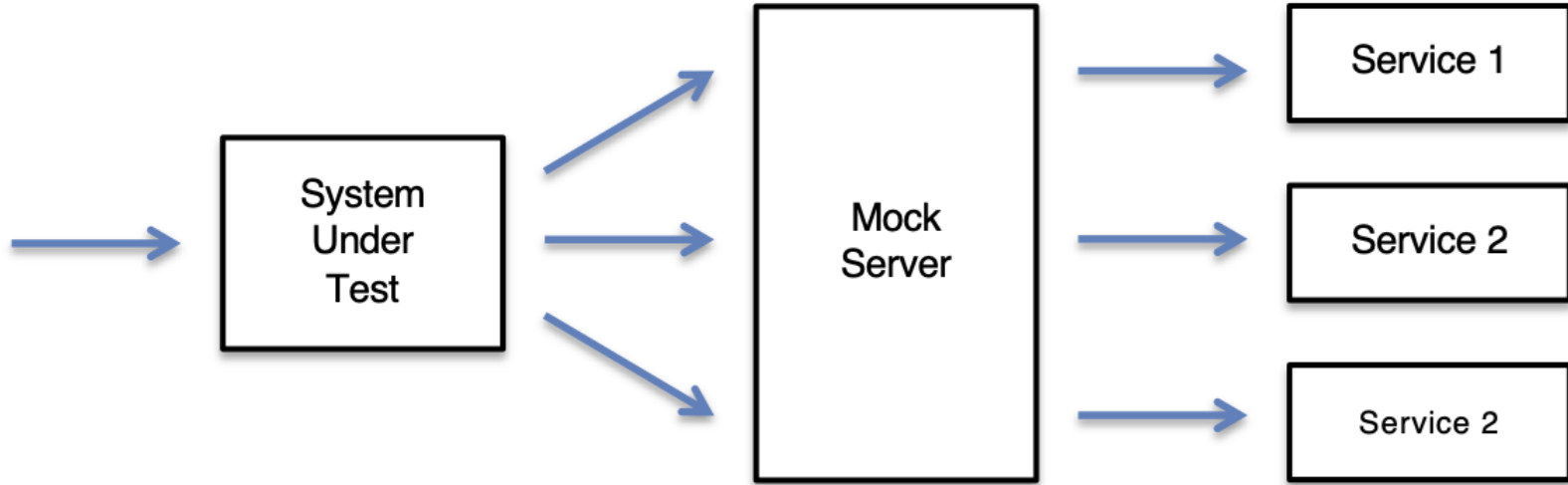
Что такое MockServer

MockServer

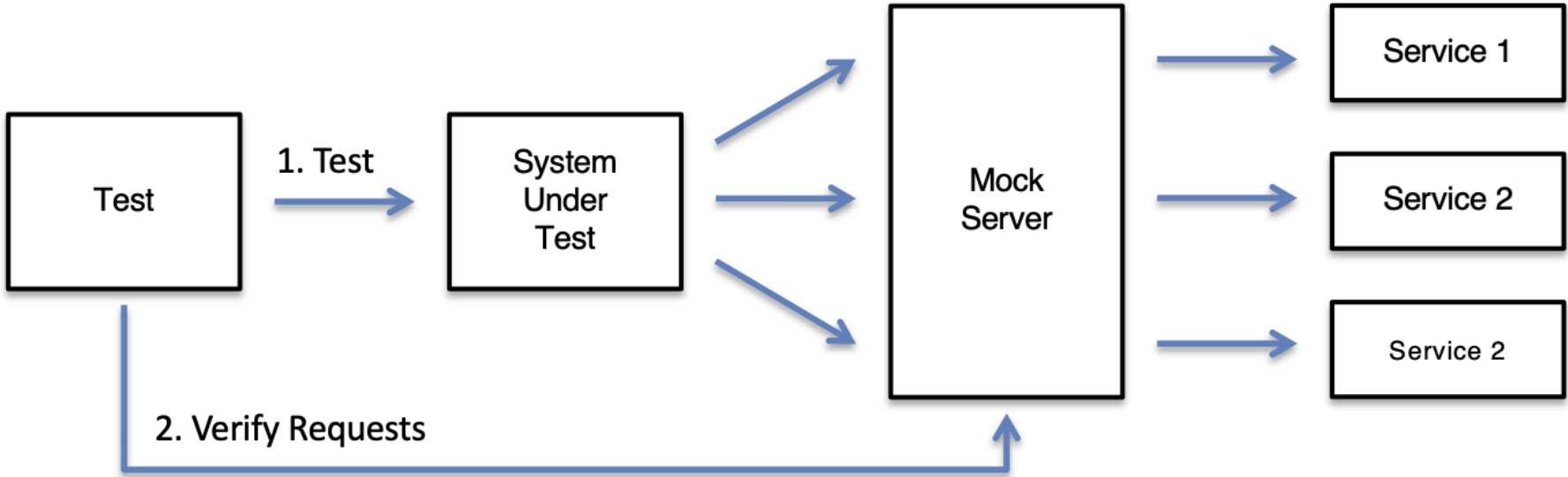
[MockServer \(mock-server.com\)](https://mock-server.com)



MockServer



MockServer





Untitled-1

```
@Before
```

```
public void startMockServer() {  
    mockServer = startClientAndServer(1080);  
}
```



Untitled-1

```
new ClientAndServer(1080)
    .verify(
        request()
            .withPath("/some/path"),
        VerificationTimes.atLeast(2)
    );
```

```
curl -v -X PUT "http://localhost:1080/mockserver/verify" -d '{
  "httpRequest": {
    "path": "/simple"
  },
  "times": {
    "atLeast": 2
  }
}'
```

[Проверка запросов \(mock-server.com\)](https://www.mock-server.com/proxy/verification.html)
<https://www.mock-server.com/proxy/verification.html>

Log Messages (most recent at the top)

```
07-03 13:18:28.729 NO_MATCH_RESPONSE ▶ ...
07-03 13:18:23.280 EXPECTATION_RESPONSE ▶ ...
07-03 13:18:11.027 WARN TSL handshake failure: Client does not trust MockServer Certificate Authority for: ▶ (...) See http://mock-
07-03 13:18:07.067 EXPECTATION_RESPONSE ▼
-----
07-03 13:18:07.067 EXPECTATION_RESPONSE returning response: ▶ (...) for request: ▶ (...) for action: ▶ (...)
07-03 13:18:07.061 EXPECTATION_MATCHED request: ▶ (...) matched expectation: ▶ (...)
07-03 13:18:07.060 RECEIVED_REQUEST received request: ▼ {
  "method": "GET"
  "path": "/simple"
  ▶ "headers": {...}
  ▶ "cookies": {...}
  "keepAlive": true
  "secure": false
}
-----
07-03 13:17:57.128 CREATED_EXPECTATION creating expectation: ▶ (...)
07-03 13:17:44.825 CREATED_EXPECTATION creating expectation: ▼ {
  "id": "cb2f10af-e34a-47fb-b55a-5b92ef6c2c88"
  "priority": 0
  ▼ "httpRequest": {
    ▶ "specUrlOrPayload": {...}
    "operationId": "somePath"
  }
  ▶ "times": {...}
  ▶ "timeToLive": {...}
  ▶ "httpResponse": {...}
}
07-03 13:17:44.807 CREATED_EXPECTATION creating expectation: ▶ (...)
07-03 13:17:44.794 CREATED_EXPECTATION creating expectation: ▶ (...)
07-03 13:17:44.777 CREATED_EXPECTATION creating expectation: ▶ (...)
```

Плюсы MockServer

- ✓ Понятный интерфейс
- ✓ MockServer поддерживает спецификации OpenAPI версии 3 в формате JSON, либо в формате YAML.
- ✓ Выполнения через аннотированный тестовый класс Spring `@MockServerTest`
- ✓ Поддерживается в качестве контейнера Docker в любой среде с поддержкой Docker из командной строки для автономного процесса в тестовой среде
- ✓ Node.js (npm) module
- ✓ Maven Plugin
- ✓ CLI
- ✓ MockServer (testcontainers.com/modules/mockserver/)

Что такое WireMock



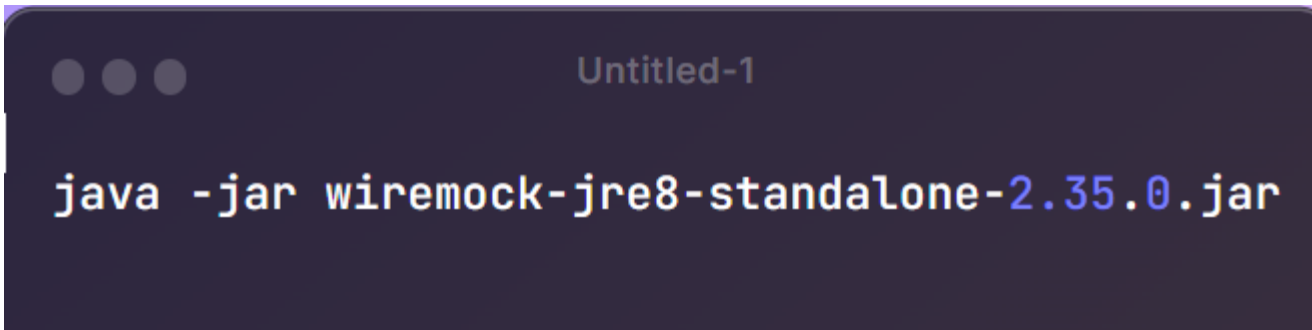
WIREFMOCK

WM WireMock Recorder



<https://wiremock.org/>

```
Untitled-1  
  
docker run -it --rm \  
  -p 8080:8080 \  
  --name wiremock \  
  wiremock/wiremock:2.35.0
```

A terminal window with a dark background and rounded corners. The title bar at the top reads "Untitled-1" and has three small grey circles on the left. The main area contains a single line of white text: "java -jar wiremock-jre8-standalone-2.35.0.jar".

```
java -jar wiremock-jre8-standalone-2.35.0.jar
```

Untitled-1

```
WireMockServer wireMockServer = new WireMockServer(options().port(8089));  
wireMockServer.start();  
wireMockServer.stop();
```

Spring Cloud Contract WireMock

<https://docs.spring.io/spring-cloud-contract/docs/current/reference/html/project-features.html#features-wiremock>

```

@SpringBootTest(webEnvironment = WebEnvironment.RANDOM_PORT)
@AutoConfigureWireMock(port = 0)
public class WiremockForDocsTests {

    // A service that calls out over HTTP
    @Autowired
    private Service service;

    @BeforeEach
    public void setup() {
        this.service.setBase("http://localhost:"
            + this.environment.getProperty("wiremock.server.port"));
    }

    // Using the WireMock APIs in the normal way:
    @Test
    public void contextLoads() throws Exception {
        // Stubbing WireMock
        stubFor(get(urlEqualTo("/resource")).willReturn(aResponse()
            .withHeader("Content-Type", "text/plain").withBody("Hello
World!")));
        // We're asserting if WireMock responded properly
        assertThat(this.service.go()).isEqualTo("Hello World!");
    }
}

```

```

@SpringBootTest(webEnvironment = WebEnvironment.RANDOM_PORT)
@AutoConfigureWireMock(port = 0)
public class WiremockForDocsTests {

    // A service that calls out over HTTP
    @Autowired
    private Service service;


    @BeforeEach
    public void setup() {
        this.service.setBase("http://localhost:"
            + this.environment.getProperty("wiremock.server.port"));
    }

    // Using the WireMock APIs in the normal way:
    @Test
    public void contextLoads() throws Exception {
        // Stubbing WireMock
        stubFor(get(urlEqualTo("/resource")).willReturn(aResponse()
            .withHeader("Content-Type", "text/plain").withBody("Hello
World!")));
        // We're asserting if WireMock responded properly
        assertThat(this.service.go()).isEqualTo("Hello World!");
    }
}

```


Spring Cloud Contract WireMock

<https://docs.spring.io/spring-cloud-contract/docs/current/reference/html/project-features.html#features-wiremock>



```

@SpringBootTest(webEnvironment = WebEnvironment.RANDOM_PORT)
@AutoConfigureWireMock(port = 0)
public class WiremockForDocsTests {

    // A service that calls out over HTTP
    @Autowired
    private Service service;

    @BeforeEach
    public void setup() {
        this.service.setBase("http://localhost:"
            + this.environment.getProperty("wiremock.server.port"));
    }

    // Using the WireMock APIs in the normal way:
    @Test
    public void contextLoads() throws Exception {
        // Stubbing WireMock
        stubFor(get(urlEqualTo("/resource")).willReturn(aResponse()
            .withHeader("Content-Type", "text/plain").withBody("Hello
World!")));
        // We're asserting if WireMock responded properly
        assertThat(this.service.go()).isEqualTo("Hello World!");
    }
}

```

Spring Cloud Contract WireMock

<https://docs.spring.io/spring-cloud-contract/docs/current/reference/html/project-features.html#features-wiremock>



```
@SpringBootTest(webEnvironment = WebEnvironment.RANDOM_PORT)
@AutoConfigureWireMock(port = 0)
public class WiremockForDocsTests {

    // A service that calls out over HTTP
    @Autowired
    private Service service;

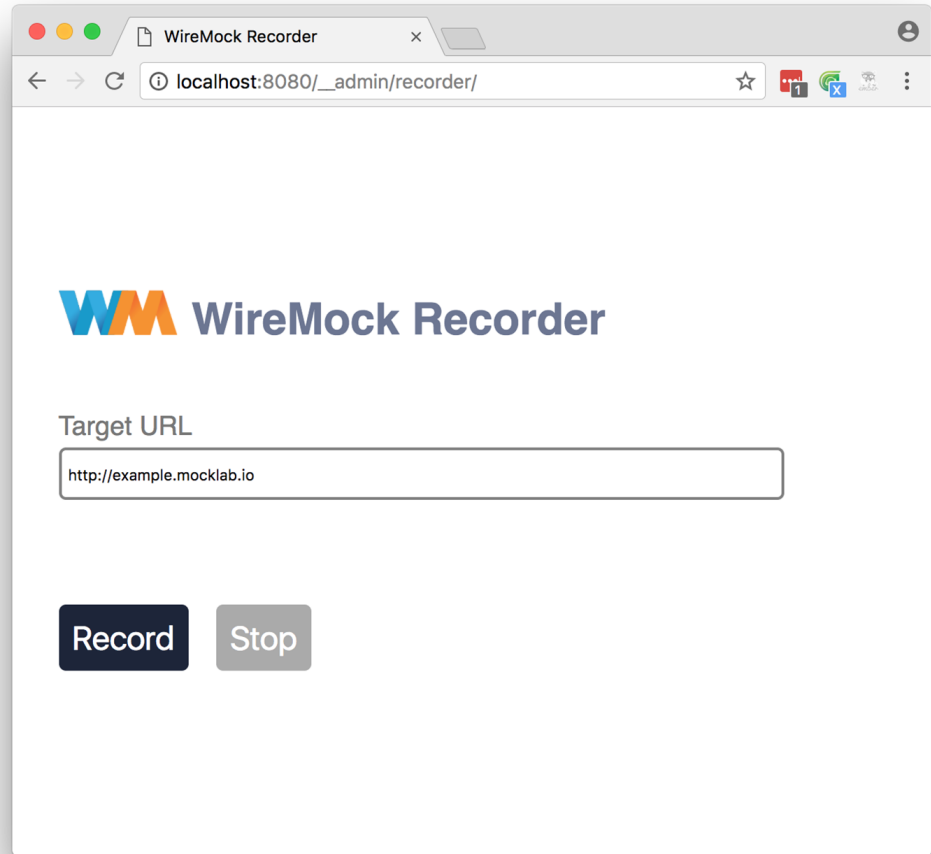
    @BeforeEach
    public void setup() {
        this.service.setBase("http://localhost:"
            + this.environment.getProperty("wiremock.server.port"));
    }

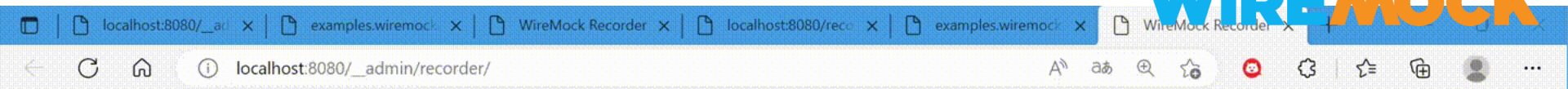
    // Using the WireMock APIs in the normal way:
    @Test
    public void contextLoads() throws Exception {
        // Stubbing WireMock
        stubFor(get(urlEqualTo("/resource")).willReturn(aResponse()
            .withHeader("Content-Type", "text/plain").withBody("Hello
World!")));
        // We're asserting if WireMock responded properly
        assertThat(this.service.go()).isEqualTo("Hello World!");
    }
}
```

Record and Playback

WireMock может создавать сопоставления заглушек на основе полученных запросов.

В сочетании с функцией проксирования это позволяет вам «записывать» сопоставления заглушек при взаимодействии с существующими API.





WireMock Recorder

Target URL

Captured 0 stub mappings.

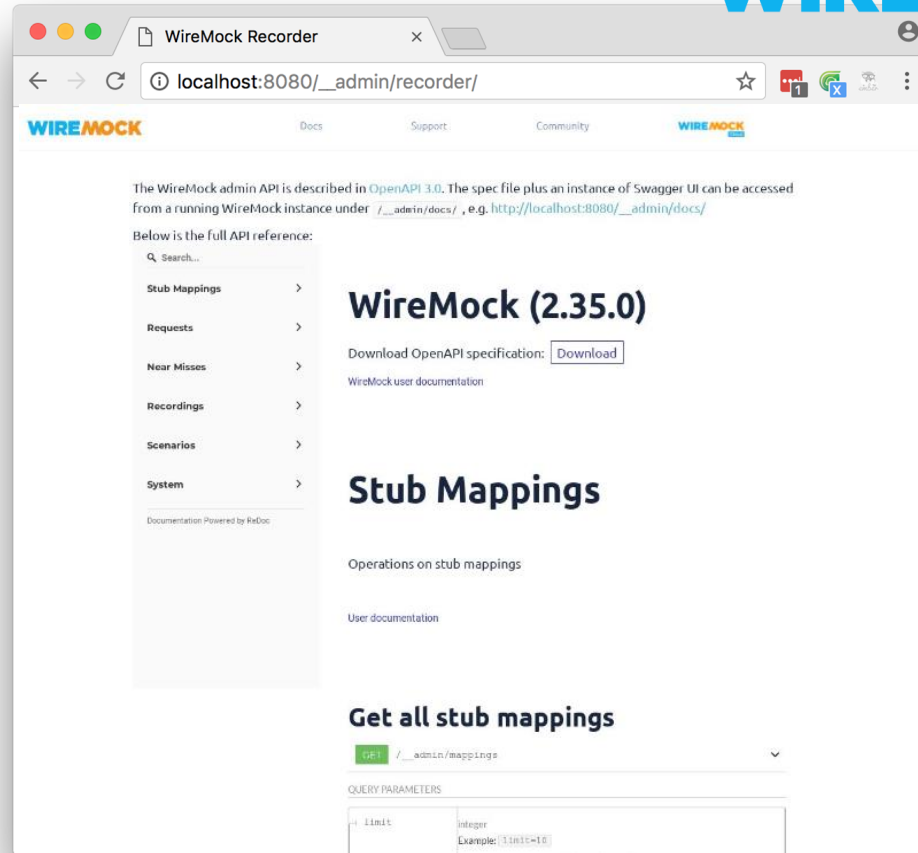
Record

Stop

Admin api

API администратора WireMock описан в OpenAPI 3.0. К файлу спецификации и экземпляру пользовательского интерфейса Swagger можно получить доступ из запущенного экземпляра WireMock в `/__admin/docs/`, например

`http://localhost:8080/__admin/docs`



The screenshot shows the WireMock Recorder admin interface in a browser. The browser address bar displays `localhost:8080/__admin/recorder/`. The page features the WireMock logo and navigation links for Docs, Support, and Community. The main content area includes a search bar and a list of API endpoints: Stub Mappings, Requests, Near Misses, Recordings, Scenarios, and System. The 'Stub Mappings' section is expanded, showing a search bar, a 'Download OpenAPI specification' button, and a 'Get all stub mappings' section with a query parameter 'limit' set to 10.

Плюсы WireMock

- ✓ **Понятный интерфейс**
- ✓ Богатый функционал
- ✓ Поддерживает разные форматы запуска
- ✓ Хорошая документация
- ✓ WireMock (testcontainers.com)
- ✓ Testcontainers - python-wiremock
wiremock.org/docs/solutions/python

Плюсы WireMock

- ✓ Понятный интерфейс
- ✓ **Богатый функционал**
- ✓ Поддерживает разные форматы запуска
- ✓ Хорошая документация
- ✓ WireMock (testcontainers.com)
- ✓ Testcontainers - python-wiremock
wiremock.org/docs/solutions/python

Плюсы WireMock

- ✓ Понятный интерфейс
- ✓ Богатый функционал
- ✓ **Поддерживает разные форматы запуска**
- ✓ Хорошая документация
- ✓ WireMock (testcontainers.com)
- ✓ Testcontainers - python-wiremock
[wiremock.org/docs/solutions/python](https://www.wiremock.org/docs/solutions/python)

Плюсы WireMock

- ✓ Понятный интерфейс
- ✓ Богатый функционал
- ✓ **Поддерживает разные форматы запуска**
- ✓ Хорошая документация
- ✓ WireMock (testcontainers.com)
- ✓ Testcontainers - python-wiremock
wiremock.org/docs/solutions/python

Плюсы WireMock

- ✓ Понятный интерфейс
- ✓ Богатый функционал
- ✓ Поддерживает разные форматы запуска
- ✓ **Хорошая документация**
- ✓ WireMock (testcontainers.com)
- ✓ Testcontainers - python-wiremock
[wiremock.org/docs/solutions/python](https://www.wiremock.org/docs/solutions/python)

Плюсы WireMock

- ✓ Понятный интерфейс
- ✓ Богатый функционал
- ✓ Поддерживает разные форматы запуска
- ✓ Хорошая документация
- ✓ **WireMock (testcontainers.com)**
- ✓ Testcontainers - python-wiremock
wiremock.org/docs/solutions/python

Плюсы WireMock

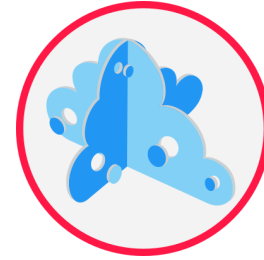
- ✓ Понятный интерфейс
- ✓ Богатый функционал
- ✓ Поддерживает разные форматы запуска
- ✓ Хорошая документация
- ✓ WireMock (testcontainers.com)
- ✓ **Testcontainers - python-wiremock**
wiremock.org/docs/solutions/python

Итоги по комбинированным

Инструмент	Стоимость	Поддерживаемые протоколы	Облачное решение
Mockoon 	Продукт с открытым кодом	HTTP(S)	Нет
Mountebank 	Продукт с открытым кодом	HTTP(S), TCP, SMTP	Нет
Stoplight/Prism	Prism отдельный модуль с открытым кодом	HTTP(S)	Нет
MockServer	Продукт с открытым кодом	HTTP(S), RPC	Нет
WireMock	Продукт с открытым кодом + платный wiremock cloud	HTTP(S), SOAP, gRPC, GraphQL	Есть

Выводы:

Microcks.io

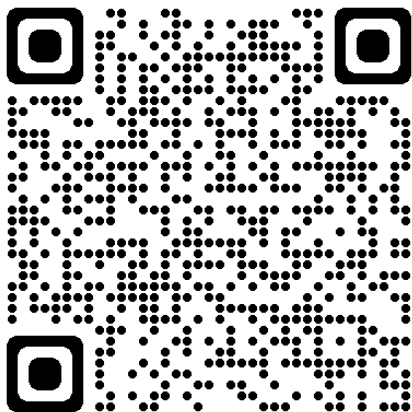


Mountebank



Mockoon mock👁️👁️n

Больше информации [Comparison of API simulation tools - Wikipedia](#)



Контакт: @akuz0