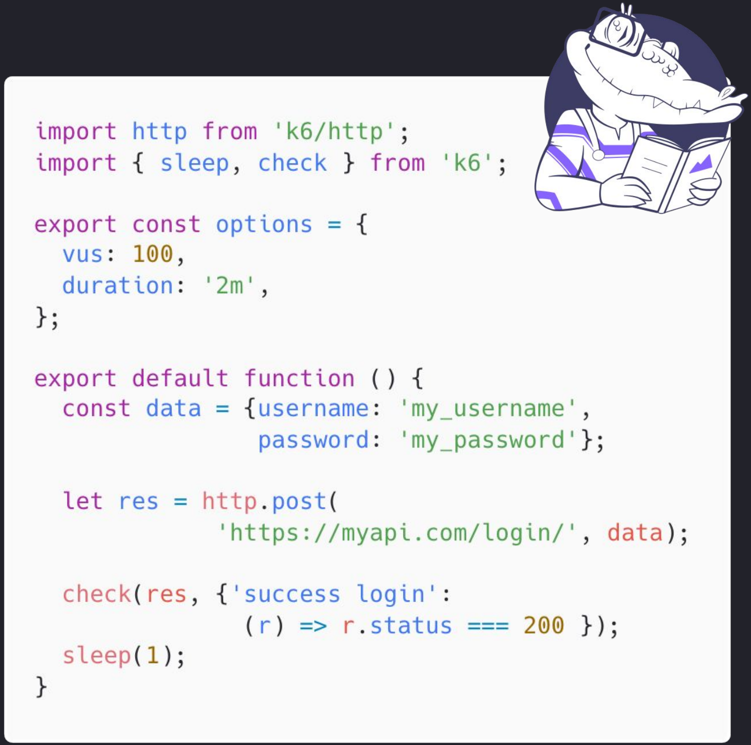
Grafana Labs

Performance testing with OpenAPI Specification and k6



```
import http from 'k6/http';
import { sleep, check } from 'k6';

export const options = {
  vus: 100,
  duration: '2m',
};

export default function () {
  const data = {username: 'my_username',
                password: 'my_password'};

  let res = http.post(
           'https://myapi.com/login/', data);

  check(res, {'success login':
                (r) => r.status === 200 });
  sleep(1);
}
```

Ayush Goyal
Senior Software Engineer
Grafana Labs

Agenda

- Challenges of traditional API performance testing
- How openapi-to-k6 tool transforms this process
- Demo
- Best practices and real-world implementation strategies

Typical workflow

- Backend Development
- OpenAPI specification for documentation
- Write basic boilerplate code for load testing to send the requests
- Run the load test

Common Challenges

- Manual script creation is time-consuming and error-prone
- Keeping test scripts synchronized with API changes is challenging
- Limited IDE support makes writing tests more difficult
- Inconsistencies between documentation and actual implementation

Role of OpenAPI Specification

- They serve as a single source of truth for API documentation
- They provide a machine-readable format for automation
- They're widely adopted in the industry
- They contain all the information needed for generating tests

Introducing openapi-to-k6

What is openapi-to-k6?

- A CLI tool to generate TypeScript client from OpenAPI specifications
- Just call a function from the client to send requests from inside the K6 scripts
- Optionally, generate a sample k6 script to get started instantly.

Benefits

Automated Test Generation

TypeScript Integration

Created by Mohamed Mb
from Noun Project

Seamless k6 Integration

Automated Test Generation

- Eliminates manual script writing
- Ensures test accuracy through specification-based generation
- Reduces maintenance overhead

TypeScript Integration

- Provides excellent IDE support with autocompletion
- Catches errors during development
- Makes tests more maintainable

Seamless k6 Integration

- Works with existing k6 knowledge and tooling
- Supports all k6 features and metrics
- Enables custom test scenarios and load patterns

Demo Time

Best Practices

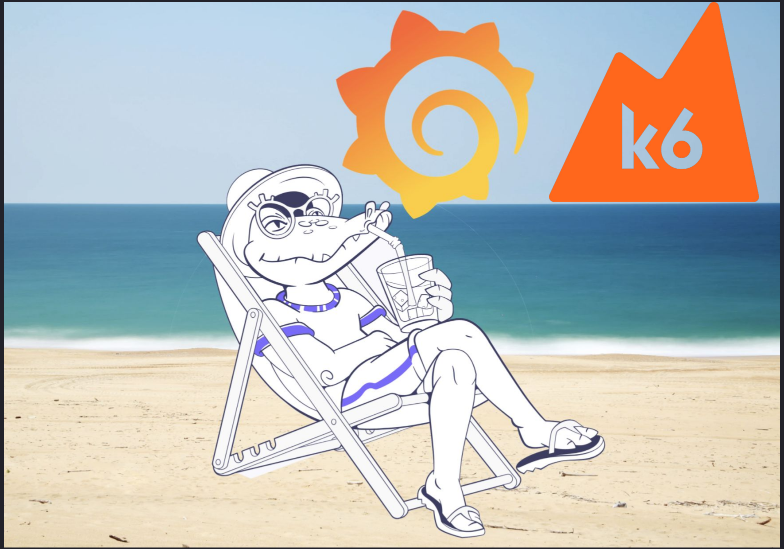
- Do not make changes to the generated client directly

- Create a new class and inherit the generated client to make any changes

- Keep reusability and modularity in mind when writing tests

Grafana Labs

Thank you



Questions?

- https://www.npmjs.com/package/@grafana/openapi-to-k6
- https://github.com/grafana/openapi-to-k6