Practices used

Sessions are stored in MongoDB, and the corresponding user information is cached in Redis with the same lifetime as the access token. The session can be refreshed with a refresh token, which is also stored in MongoDB. If the access token is not refreshed within an hour, the session is automatically deleted.

To cache the browser information, I chose Redis and stored the user's payload there. Redis was chosen because of its high speed of data access and efficiency in cases where you need to provide quick access to frequently used information.

I stored the main information in PostgreSQL. PostgreSQL was chosen because of its powerful relational database capabilities, reliability, and support for complex queries and transactions, which is important for storing critical data and ensuring data integrity.

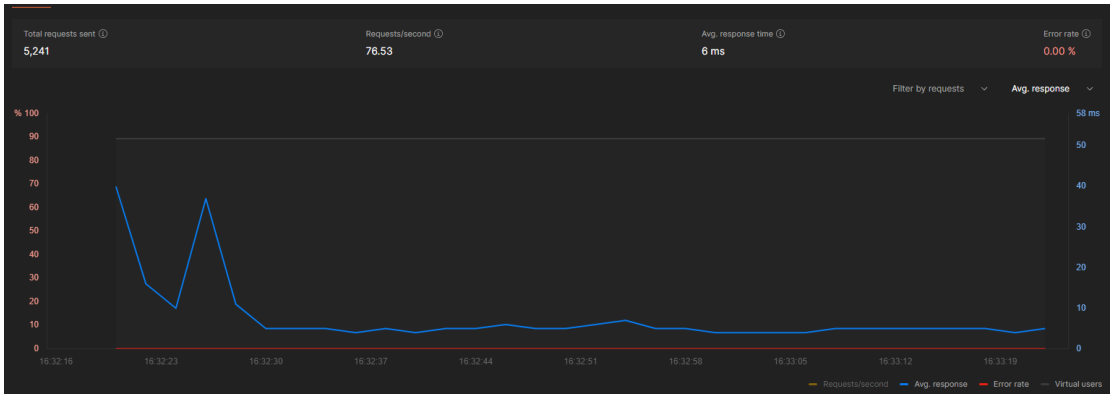
I set up clustering for the server so that the program runs in parallel on all threads. This allows for efficient use of server resources and ensures high availability and performance of the application.

I also added validation of all data before it is processed and global error handlers to ensure correct processing of incoming data and reduce the risk of unforeseen errors.

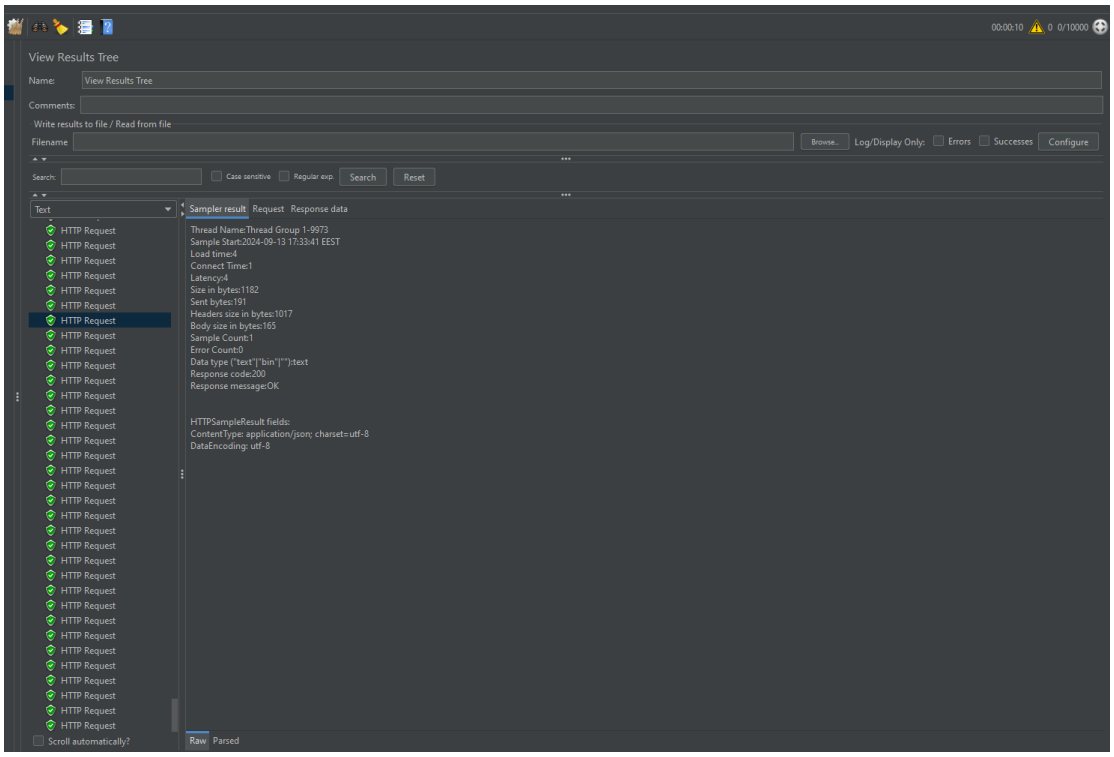
In addition, all users' passwords are stored in an encrypted form to ensure the security and confidentiality of user data.

I have also used various strategies to protect against attacks to ensure the security of the application and protect it from possible threats.

Postman result



JMeter result



Server architecture