



**«Московский государственный технический университет
имени Н.Э. Баумана»
(национальный исследовательский университет)
(МГТУ им. Н.Э. Баумана)**

ФАКУЛЬТЕТ _____ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ_____

КАФЕДРА _____КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)_____

О т ч е т

по лабораторной работе №11

Название лабораторной работы: Аутентификация пользователей с
помощью jwt-токена

Дисциплина: Языки интернет-программирования

Студент гр. ИУ6-33Б

(Подпись, дата)

Цыганчук П. В.
(И.О. Фамилия)

Преподаватель

(Подпись, дата)

В.Д. Шульман
(И.О. Фамилия)

Москва, 2024

Введение

Цель

Получение первичных знаний в области авторизации и аутентификации в контексте веб-приложений

Задание

Реализовать сервис Auth (регистрация пользователя и авторизация с выдачей jwt-токена)

Добавить в сервисы count, hello и query валидацию jwt-токена (если токен не валиден или отсутствует — возвращаем код 401)

Ход работы

Для того, чтобы сделать аутентификацию, был создан отдельный микросервис *auth*, в котором при запросе “/register” создается новый пользователь с именем и паролем, а при “/login” возвращается JWT-токен, который действителен час после создания.

Также в этом микросервисе происходит проверка JWT-токена по запросу “/restricted”.

Другие микросервисы, чтобы проверить токен, обращаются к “/restricted” микросервиса *auth* и если он верен, то уже продолжают делать то, что они делали раньше.

Ниже представлена часть листинга *auth*, а именно создание и проверка JWTтокена.

```
func (j *JWTProvider) GenerateToken(username string) (string, error) {
    claims := JWTClaims{
        Username: username,
        RegisteredClaims: jwt.RegisteredClaims{
            ExpiresAt:      jwt.NewNumericDate(time.Now().Add(1 *
time.Hour)),
        },
    }
    token := jwt.NewWithClaims(jwt.SigningMethodHS256, claims)
    return token.SignedString([]byte(j.secretKey))
}
func (j *JWTProvider) ValidateToken(tokenString string) (*JWTClaims, error) {
    token, err := jwt.ParseWithClaims(tokenString, &JWTClaims{},
func(token *jwt.Token) (interface{}, error) {
    return
[]byte(j.secretKey), nil
})
}
```

```
    })
    if err != nil || !token.Valid {
return nil, err
    }
    claims, ok := token.Claims.(*JWTClaims)
if !ok {
        return nil, fmt.Errorf("invalid claims")
    }
    return claims, nil
}
```

Вывод

Изучен наиболее популярный способ аутентификации и авторизации пользователей в веб-приложениях — jwt-токен.