

# Lesson 13 - React Router и Redux.

## Авторизация и Guards

### Redux Routing

Используя Redux и React Router мы получаем конфликт интересов, два источника правды. Состояние приложения хранится в `store`, а информация о маршруте в контексте предоставляемом React Router, это как будто еще один `store` только для раутинга.

Это не очень удобно по нескольким причинам:

- Данные получаются из нескольких источников, что запутывает код.
- Иногда приходится дублировать состояние URL в Redux-хранилище и постоянно обновлять его.

Поэтому желательно чтобы все данные лежали в одном месте и автоматически обновлялись в Redux-хранилище. Таким образом мы сможем получать доступ к этим данным не только из компонентов, но к примеру и из прослоек, экшенов и т. п.

### react-router-redux

`history` - синглтон. Пакет оборачивает `history` и добавляет к ней сохранение данных о `location` и перемещений в приложении.

[Репозиторий react-router-redux](#)

### JWT

# JWT

JSON WEB TOKEN



**HEADER**  
ALGORITHM  
& TOKEN TYPE

```
{  
  "alg": "HS256",  
  "typ": "JWT"  
}
```

+

**PAYLOAD**  
DATA

```
{  
  "sub": "1234567890",  
  "name": "John Doe",  
  "admin": true  
}
```

+

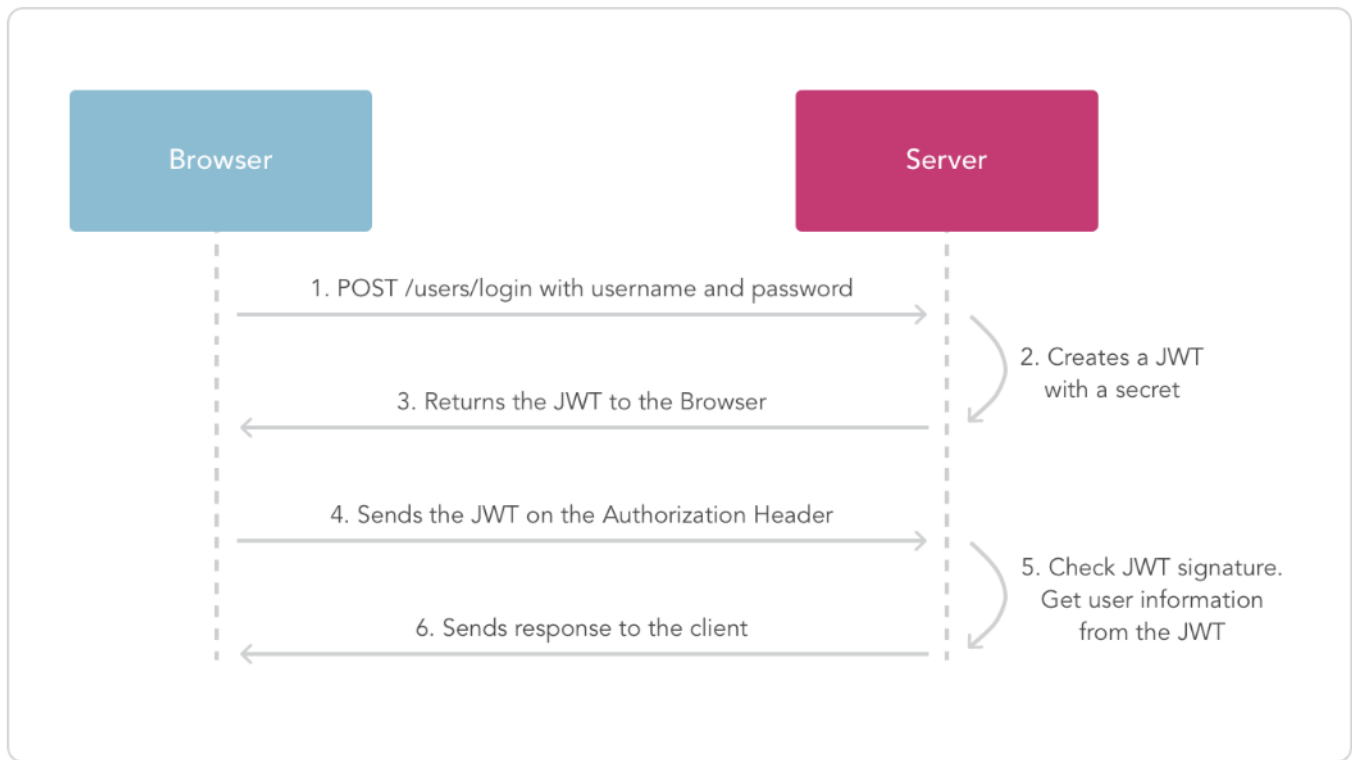
**SIGNATURE**  
VERIFICATION

```
HMACSHA256(  
  base64UrlEncode(header) + "." +  
  base64UrlEncode(payload),secretKey)
```

NORDICAPIS.COM

- [jwt.io](https://jwt.io)
- [JWT \(JSON Web Tokens\) Are Better Than Session Cookies](#)
- [5 Steps to Authenticating Node.js with JWT](#)

**Механизм авторизации пользователя**



- [Protected routes and authentication with React Router v4](#)

## Деплой на GH

[Хаки под GitHub Pages - Building for Relative Paths](#)

## Материалы

- [Build your own React Router v4](#)
- [An Introduction to the Redux-First Routing Model](#)
- [Репозиторий redux-little-router](#)
- [Репозиторий redux-router](#)