

Passport JS library of programming language Node Passport JS

Vasilev P. (Russian Federation)

JS языка программирования библиотеки Node JS

Васильев П. А. (Российская Федерация)

*Васильев Петр Алексеевич / Vasilev Petr Alexeevich - студент,
кафедра информационных технологий,
институт математики и информатики
Северо-Восточный федеральный университет, г. Якутск*

Аннотация: в статье предлагаются сведения о библиотеке Passport JS. О ее достоинствах и недостатках в использовании на web-приложениях. Основные методы и использование библиотеки в реальных условиях.

Abstract: in article there is information about library Passport JS. His strengths and weaknesses in using on web-applications. Main things and using of library in real life.

Ключевые слова: Node JS, Passport JS, Web-программирование, JavaScript, авторизация, аутентификация.

Keywords: Node JS, Passport JS, Web-developing, JavaScript, authorization, authentication.

Каждый разработчик web-приложения рано или поздно сталкивается с такой проблемой, как авторизация и аутентификация пользователя. Сегодня любое web-приложение не обходится без него и имеет ряд функций, которые доступны только для авторизованных пользователей. Что такое авторизация? Авторизация [1] – это предоставление определённому лицу или группе лиц прав на выполнение определённых действий; а также процесс проверки данных прав при попытке выполнения этих действий. До недавнего времени программисты разрабатывали свои системы авторизаций. Но в последнее время появились множество готовых решений. Одним из самых популярных библиотек для языка программирования Node JS сейчас является библиотека Passport JS.

Passport JS – это готовое и надежное решение для реализации авторизации и аутентификации в web-приложении, построенном на Node JS. Документация Passport JS описывает его как «простое, компактное связующее приложение аутентификации для Node» и это верно. Самым главным, что предоставляет нам эта библиотека, является не только обычная авторизация с помощью логина и пароля, но и авторизация с помощью социальных сетей, таких как Facebook, Google, Twitter, Vkontakte и т. д. Библиотека имеет около 140 механизмов аутентификаций, которые можно включить для своего приложения.

Библиотека использует функции, которые называются стратегиями. Для каждого механизма описывается отдельная стратегия и вызывается отдельно в роутере. Например, локальная стратегия через имя пользователя и пароль:

```
passport.use (new LocalStrategy({
  usernameField: 'email',
  passwordField: 'password'
}, function(username, password,done){
  // описывается стратегия...
}));
```

Эта стратегия используется в роутере, например /auth/login:

```
app.post (/auth/login',
  passport.authenticate('local'), function(req, res) { //успешное выполнение });
```

Для каждой стратегии требуется установить отдельный модуль, например, passport – google, passport – facebook, passport – vkontakte и т. д. Каждый сервис использует такую технологию, как OAuth2. OAuth2 [1] – это открытый протокол авторизации, который позволяет предоставить третьей стороне ограниченный доступ к защищённым ресурсам пользователя без необходимости передавать ей (третьей стороне) логин и пароль. То есть разработчик может внедрить в своё приложение эту технологию. Тогда пользователи смогут авторизоваться с помощью других сервисов без необходимости пароля и имени пользователя.

Passport JS прекрасно справляется с OAuth2 и имеет готовое решение. Для каждого сервиса разработчик должен создать приложение внутри сервиса и дать доступ к своему приложению. Для этого он использует специальный секретный ключ и идентификатор приложения. Каждый раз при необходимости использования каких-нибудь данных пользователя, который авторизовался с помощью OAuth2, Passport JS отправляет запрос вместе с этими ключом и идентификатором. Сервис проводит валидацию запроса и передает данные пользователя приложению. При регистрации пользователь должен будет войти в свой аккаунт стороннего сервиса и дать разрешение на использование его данных приложению.

Всё описанное выше программисты разрабатывали сами, но с приходом Passport JS это всё стало автоматизированной системой. Требуется лишь реализовать стратегию сохранения данных пользователя в базе данных или в оперативной памяти.

Passport JS является очень хорошим инструментом для облегчения и упрощения работы разработчика и сокращения времени на написание объемного кода.

Литература

1. *Этан Броун* Web-development with Node & Express – Москва, 2010.