



Node JWT Middleware

By Laeeq Khan Niazi



Task



- Create a Create API store following information of the Card
- Cart type (Debit/Credit)
- Card Bank
- Card Holder Name
- Card Number
- Card Expiry
- Card CVC

Learning Objective



- Student should be able to write and implement the middleware's in express
- Students should be able to protect the APIs using JWT base token

Create Signup API

- Create an API to store username and password in the database using API



Install Package

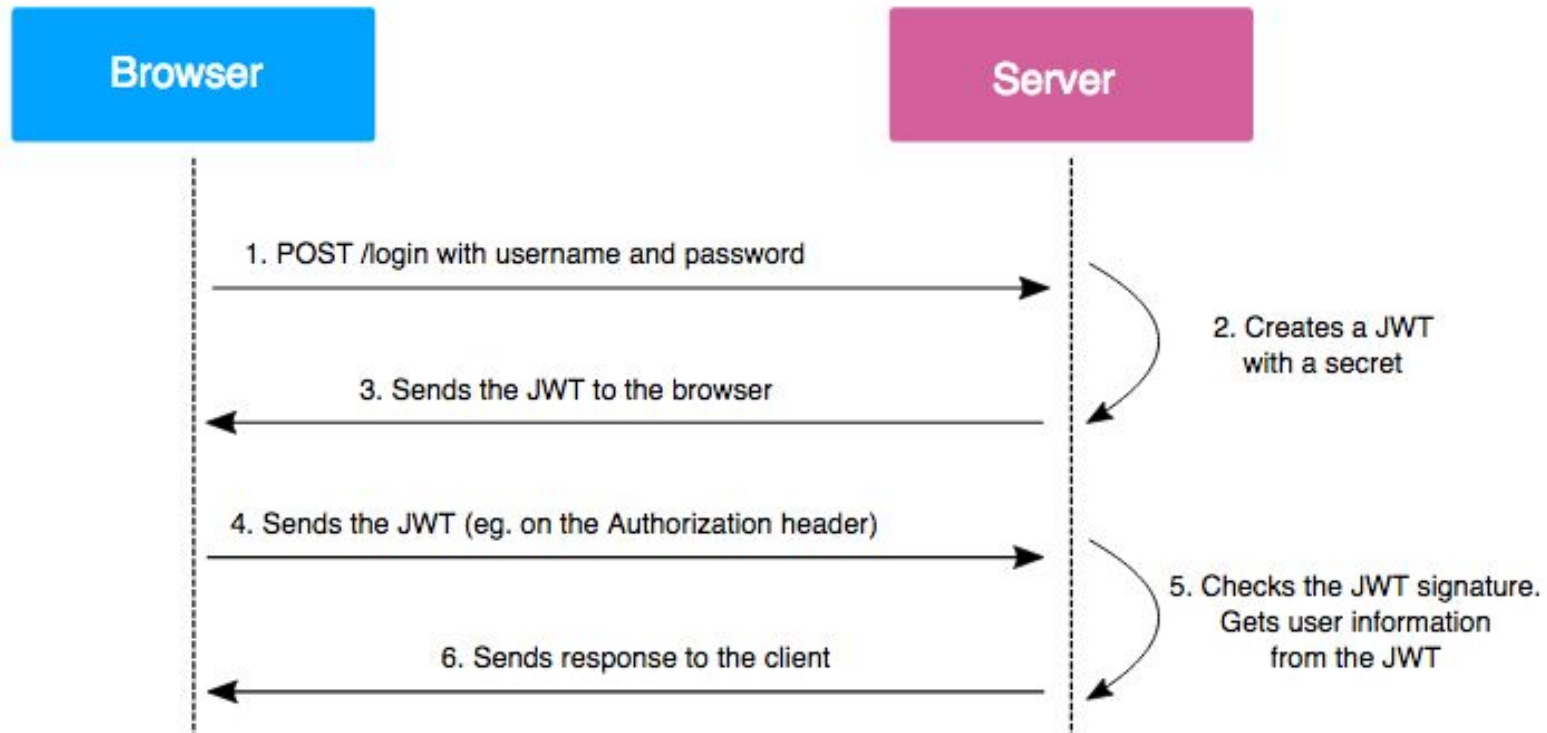
- `npm install jsonwebtoken`

Create Signup API

```
async function login(req, res, next) => {  
  const { email, password } = req.body;  
  try {  
    const user = await User.findOne({ email });  
    if (!user) return res.status(404).json({ error: 'User not found' });  
    return res.status(200).json({  
      message: 'Logged in successfully',  
      email: email,  
      fullname: user.fullname,  
      userid: user.id,  
    });  
  } catch (err) {  
    return res.status(500).json({ message: err });  
  }  
};
```



How JWT Works



Now let's get the token on the login

```
function GenerateToken(user) => {  
  const payload = {  
    role: user.role,  
    id: user._id,  
  };  
  const token = jwt.sign(payload, process.env.ACCESS_TOKEN_SECRET);  
  return token;  
};
```


Send back the token to the Login Response

```
async function login (req, res, next) {  
  const { email, password } = req.body;  
  try {  
    const user = await User.findOne({ email });  
    if (!user) return res.status(404).json({ error: 'User not found' });  
    if (!isMatch) return res.status(401).json({ error: 'Invalid credentials' });  
    var token = generateLoginToken(user);  
    return res.status(200).json({  
      message: 'Logged in successfully',  
      email: email,  
      fullname: user.fullname,  
      userid: user.id,  
      token: token,  
    });  
  } catch (err) {  
    return res.status(500).json({ message: err });  
  }  
};
```

Make middleware to verify the token

- Create a middleware file in the Utils folder

Middleware Function

```
async function SecureAPI(req, res, next) => {
  const token = req.body.token;
  if (!token) {
    return res
      .status(401)
      .json({ message: 'Access denied. Please login to access these pages' });
  }
  const secretkey = process.env.ACCESS_TOKEN_SECRET;
  try {
    const decoded = jwt.verify(token, secretkey);
    req.body.role = decoded.role;
    req.body.userid = decoded.id;
    next();
  } catch (ex) {
    return res.status(400).json({ message: 'Invalid token.' });
  }
};

module.exports = { SecureAPI };
```

Include that middleware in Routes

```
const { SecureAPI } = require('../utils/authorization_middleware');
```

Call middleware function as 2nd argument in your routes calling function like

```
myRouter.post('/admin', SecureAPI, controller.adminDashboard);
```