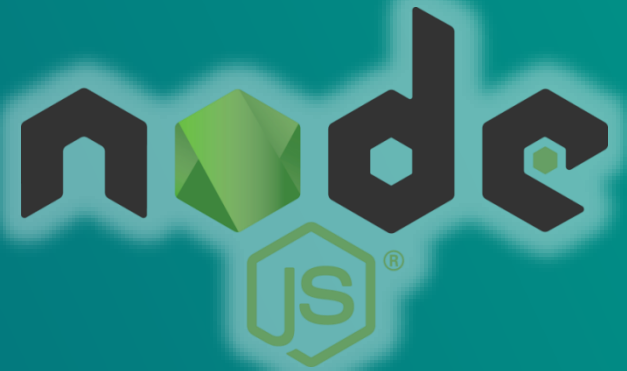


Back-End Development

# Exploring

## JSON Web Token



# jsonwebtoken

JSON Web Token (JWT) is an open standard ([RFC 7519](#)) that defines a compact and self-contained way for securely transmitting information between parties as a JSON object.

Info Link : <https://jwt.io/introduction/>

Documentation Link : <https://www.npmjs.com/package/jsonwebtoken>

# Installation

- Run (npm i express cors body-parser jsonwebtoken express-bearer-token) in your backend project, after you run (npm init), make sure the package.json already exist first.

```
{
  "name": "api-jwt",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "Baron",
  "license": "ISC",
  "dependencies": {
    "body-parser": "^1.19.0",
    "cors": "^2.8.5",
    "express": "^4.17.1",
    "express-bearer-token": "^2.4.0",
    "jsonwebtoken": "^8.5.1"
  }
}
```

← Libraries Installed

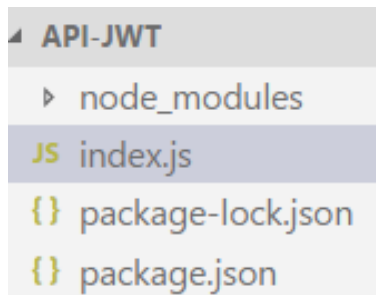
# Create index.js

```
const express = require('express')
const cors = require('cors')
const bodyParser = require('body-parser')
const bearerToken = require('express-bearer-token');

const port = 1997
const app = express()
app.use(cors())
app.use(bearerToken())
app.use(bodyParser.json())

app.get('/', (req, res) => {
  res.status(200).send('<h1>Selamat datang di API Test JWT!</h1>')
})

app.listen(port, () => console.log(`API aktif di port ${port}`))
```



**Selamat datang di API Test JWT!**

# Create jwt.js

```
const jwt = require ('jsonwebtoken');

module.exports = {
  createJWTToken(payload){
    return jwt.sign(payload, "puripuriprisoner", { expiresIn : '12h' })
  }
}
```

# Create auth.js

```
const jwt = require ('jsonwebtoken');

module.exports = {
  auth : (req, res, next) => {
    if (req.method !== "OPTIONS") {
      jwt.verify(req.token, "puripuriprisoner", (error, decoded) => {
        if (error) {
          return res.status(401).json({ message: "User not
            authorized.", error: "User not authorized." });
        }
        req.user = decoded;
        next();
      });
    } else {
      next();
    }
  }
}
```

## Edit index.js, add code below

```
const { createJWTToken } = require('./jwt')

•
•

app.post('/gettoken', (req, res) => {
  const token = createJWTToken({ userId: 2, username: 'baron197' })
  res.status(200).send({ theToken: token })
})
```

# Create React UI Project

Now run `create-react-app ui-jwt` in the terminal, with the path same as where api-jwt is located, so now you have both folder (api & ui)

 api-jwt	16/09/2019 17:13	File folder
 ui-jwt	17/09/2019 16:22	File folder



# Install axios in ui project

npm install axios

```
"dependencies": {  
  "axios": "^0.19.0",  
  "react": "^16.9.0",  
  "react-dom": "^16.9.0",  
  "react-scripts": "3.1.1"  
},
```

# Change App.js Content

```
import React, { Component } from 'react';
import axios from 'axios';

class App extends Component {
  state = { token: '' }

  requestToken = () => {
    axios.post('http://localhost:1997/gettoken')
      .then((res) => {
        this.setState({ token: res.data.theToken })
      })
  }

  render() {
    return(
      <div>
        <h1>Ini Home Page</h1>
        <h3>Token : {this.state.token}</h3>
        <input type="button" value="Request Token" onClick={this.requestToken} />
      </div>
    )
  }
}

export default App;
```

# Ini Home Page

Token :

Request Token



Sehabis button di click

# Ini Home Page

Token : eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiOi

Request Token

# Edit API index.js, add code below

```
const { auth } = require('./auth')
```

```
•  
•  
•
```

```
app.get('/testauth', auth, (req, res) => {  
  console.log(req.user)  
  res.status(200).send({ message: 'Request Accepted' })  
})
```

Edit state in App.js in the UI Project like this (add response prop) :

```
state = { token: '', response: '' }
```

And then make new arrow function called testAuth in App.js too :

```
testAuth = () => {  
  var options = {  
    headers: {  
      'Authorization': `Bearer ${this.state.token}`  
    }  
  }  
  
  axios.get('http://localhost:1997/testauth', options)  
    .then((res) => {  
      this.setState({ response: res.data.message })  
    }).catch((err) => {  
      this.setState({ response: err.response.data.message })  
    })  
}
```

Then edit function render in App.js too :

```
<div>  
  <h1>Ini Home Page</h1>  
  <h3>Token : {this.state.token}</h3>  
  <input type="button" value="Request Token" onClick={this.requestToken} />  
  <input type="button" value="Test Auth" onClick={this.testAuth} />  
  <h3>Response : {this.state.response}</h3>  
</div>
```

Now you can see the difference in response from the API, if you have the token and if you not have the token.

## Ini Home Page

Token :



Without Token

Request Token Test Auth

Response : User not authorized.

## Ini Home Page

Token : eyJhbGciOiJIUzI1NiIsInR



With Token

Request Token Test Auth

Response : Request Accepted

Back-End Development

# Exploring

*JSON Web Token*

