



JSON Web Token (JWT) es un estándar abierto (RFC-7519 request for comment) basado en JSON para crear un token que sirva para enviar datos entre aplicaciones o servicios y garantizar que sean válidos y seguros.

Datos seguros en APIs

Válidos y Seguros

Uso:

Manejar la autenticación en aplicaciones móviles o web.

Sistema de login, autenticación, autorización de usuario por roles, accesos y permisos.

JSON Web Tokens escalables

Usuario envía un código alfanumérico al servidor y el servidor se encarga de descifrar decodificar valida y el usuario existe si tiene permiso a través de sus roles.

Implementa seguridad en nuestras aplicaciones RESTful api.

Características:

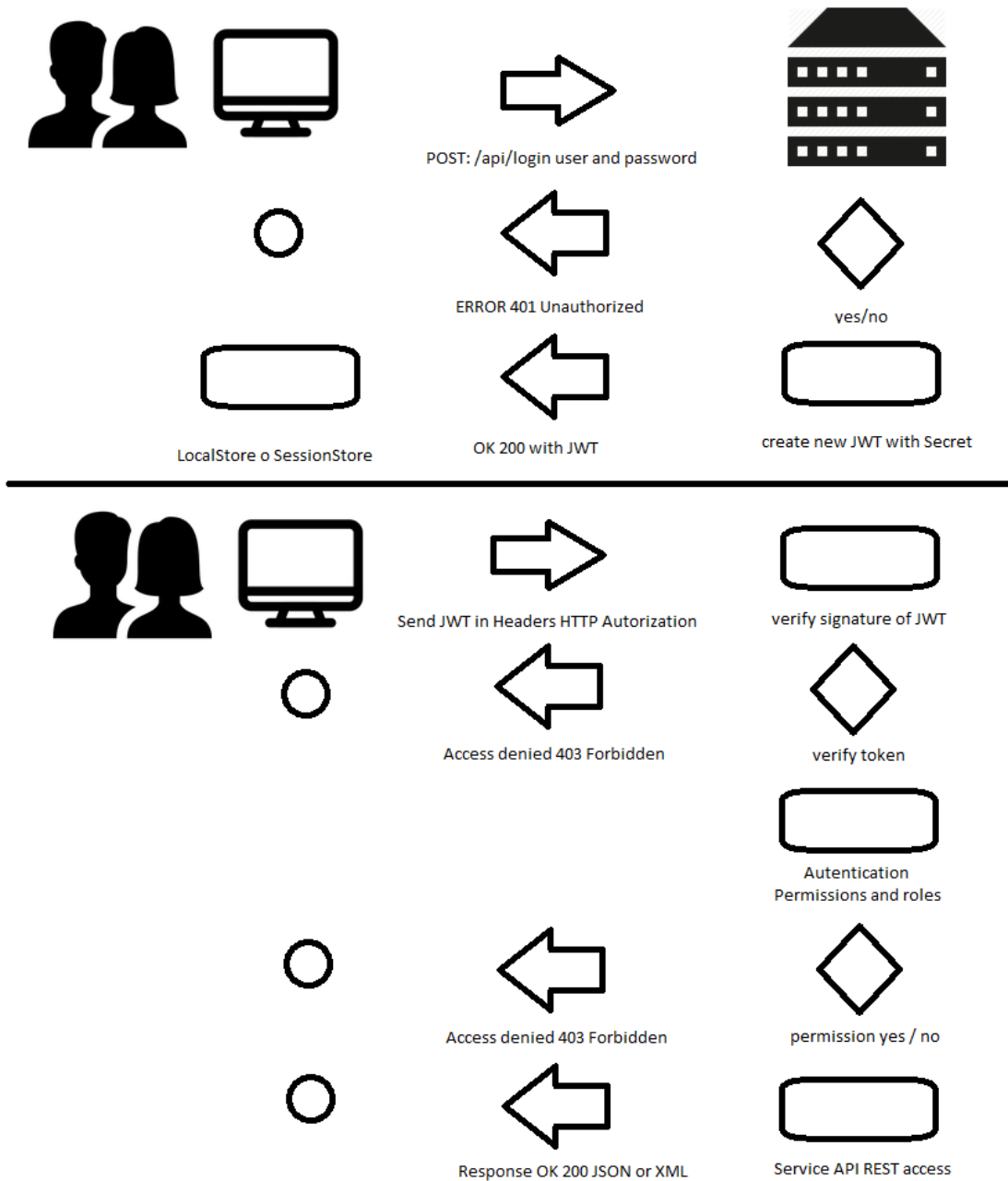
Compacto, enviar cabeceras http authorization.

Es autónomo, no se guarda nada en el servidor.

Token firmado, codificado en doble vía.

Tiempo por defecto es ilimitado.

Metodo de autentiucacion por Sesiones es Statefull, JWT Stales





```
#mkdir jwt-simple
```

```
#npm init -yes
```

```
#npm i express jsonwebtoken
```

```
/index.js
```

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

app.get('/api', (req, res) => {
  res.json({
    text: 'api works!'
  });
});

app.post('/api/login', (req, res) => {
  // auth user
  const user = {id: 3};
  const token = jwt.sign({ user }, 'my_secret_key');
  res.json({token});
});

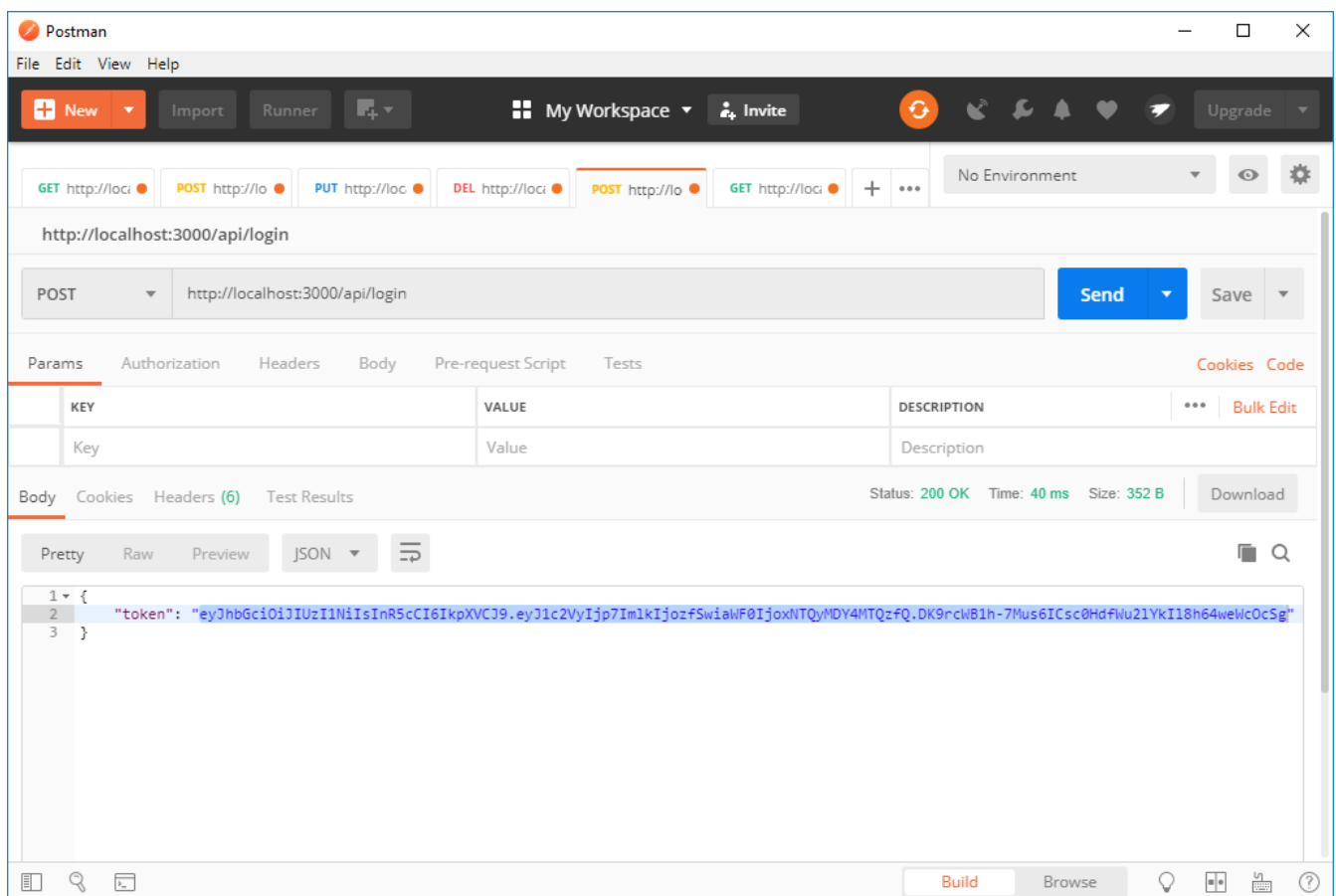
app.get('/api/protected', ensureToken ,(req, res) => {
  jwt.verify(req.token, 'my_secret_key', (err, data) => {
    if(err) {
      res.sendStatus(403);
    } else {
      res.json({
        text: 'protected Response!',
        data
      });
    }
  });
});

function ensureToken(req, res, next) {
  const bearerHeader = req.headers["authorization"];
  console.log(bearerHeader)
  if(typeof bearerHeader !== 'undefined') {
    const bearer = bearerHeader.split(" ");
    const bearerToken = bearer[1];
```

```
    req.token = bearerToken;
    next();
  } else {
    res.sendStatus(403);
  }
}

app.listen(3000, () => {
  console.log('App on port 3000')
});
```

#node .





Postman

File Edit View Help

New Import Runner My Workspace Invite

GET http://lo POST http://lo PUT http://loc DEL http://loc POST http://lo GET http://loc

No Environment

http://localhost:3000/api/protected

GET http://localhost:3000/api/protected Send Save

Params Authorization Headers (1) Body Pre-request Script Tests Cookies Code

KEY	VALUE	DESCRIPTION
Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2Vyljpw7ImIkljozfSwiaWF0IjoxNTQyMDY4MTQzfQ.DK9rcWB1h-7Mus6lCsc0HdfWu2lYklI8h64weWcOcSg	
Key		Description

Body Cookies Headers (6) Test Results Status: 200 OK Time: 25 ms Size: 284 B Download

Pretty Raw Preview JSON

```
1 {
2   "text": "protected Response!",
3   "data": {
4     "user": {
5       "id": 3
6     },
7     "iat": 1542068143
8   }
9 }
```

Build Browse