UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Programador Web Avanzado

Clase N° 6: Mongoose y JWT

Profesor: Ing. Leandro Rodolfo Gil Carrano
Email: leangilutn@gmail.com

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Subdocumentos

Mongoose – Subdocumentos

Permite definir un esquema dentro de otro, es decir un documento hijo dentro de un documento padre.

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Mongoose – Subdocumentos

Definición Schema

```
const Schema = mongoose.Schema;
var childSchema = new Schema({ name: 'string' });
//Define a schema

const ProductSchema = new Schema({
 name: {
  type: String,
  trim: true,
  required: true,
 },
 sku: {
  type: String,
  trim: true,
  required: true
 },
 price: {
  type: Number,
  trim: true,
  required: true
 },
 categoria: {type:Schema.ObjectId, ref:"categorias"},
 relacionados:[childSchema]
});
module.exports  =  mongoose.model('products', ProductSchema);
```

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Mongoose – Subdocumentos

Insertar datos

```
save: async function(req, res, next) {
    var product = new productModel({
        name: req.body.name,
        sku: req.body.sku,
        price: req.body.price,
        categoria:req.body.categoria
    });
    console.log(req.body.relacionados)
    product.relacionados.push(req.body.relacionados)
    var result = await product.save()

    res.status(200).json({status: "success", message: "Product added successfully!!!", data: result});
}
```

Se hace un push sobre el subdocumento relacionado.

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Mongoose – Subdocumentos

Leer un subdocumento. Aplicando un find para consultar todos los productos (documentos), el detalle del subdocumento se visualiza de la siguiente manera


```
{
    "_id": "5d8049528f038622202b0f4f",
    "name": "Test1",
    "sku": "123132",
    "price": 2000,
    "categoria": {
        "_id": "5d80488c30cd71ec5f53549b",
        "name": "Celulares"
    },
    "relacionados": [
        {
            "_id": "5d8049528f038622202b0f50",
            "name": "rel1"
        }
    ],
    "__v": 0
}
```

UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Queries

Mongoose – Queries

```
Model.deleteMany()
Model.deleteOne()
Model.find()
Model.findById()
Model.findByIdAndDelete()
Model.findByIdAndRemove()
Model.findByIdAndUpdate()
Model.findOne()
Model.findOneAndDelete()
Model.findOneAndRemove()
Model.findOneAndReplace()
Model.findOneAndUpdate()
Model.replaceOne()
Model.updateMany()
Model.updateOne()
```

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Mongoose – Queries

findbyIdandUpdate

```
update: async function(req, res, next) {
    var data = await productModel.findByIdAndUpdate(req.params.id, { $set: { name: req.body.name }})
    res.status(200).json({status: "success", message: "Product added successfully!!!", data: data});

},
```

Ver todos los queries

https://mongoosejs.com/docs/queries.html

UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Validations

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Mongoose – Queries

findbyIdandUpdate

```
update: async function(req, res, next) {
    var data = await productModel.findByIdAndUpdate(req.params.id, { $set: { name: req.body.name }})
    res.status(200).json({status: "success", message: "Product added successfully!!!", data: data});

},
```

Ver todos los queries

https://mongoosejs.com/docs/queries.html

JWT

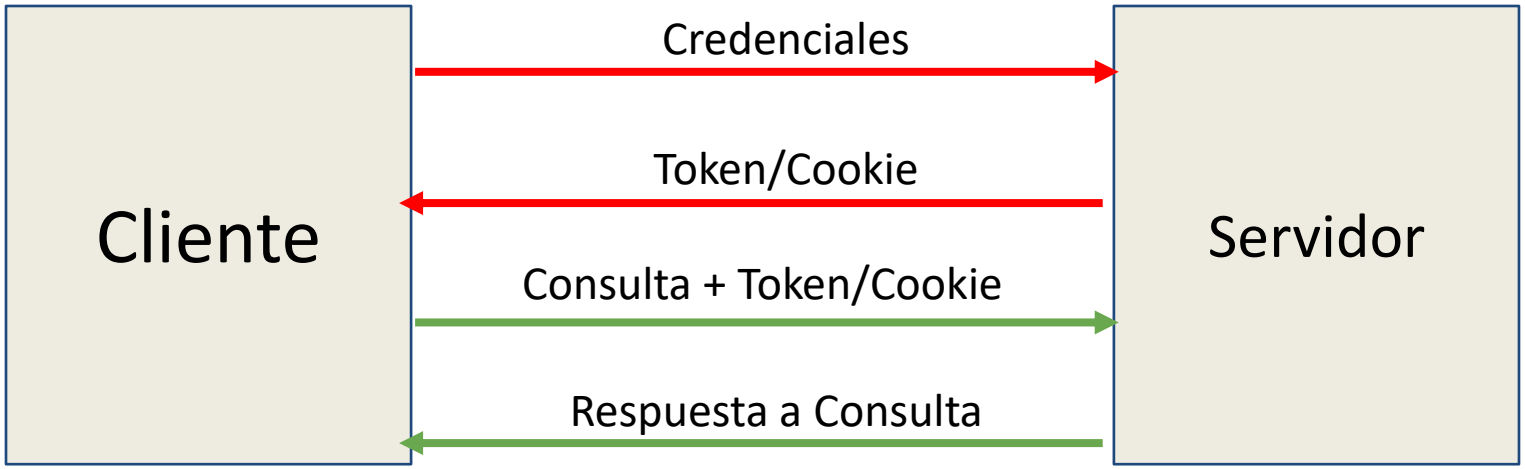
UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Autenticación y Autorización

Definición de conceptos

- **Autenticación:** Proceso para identificar que “algo” o “alguien” es quien dice ser.

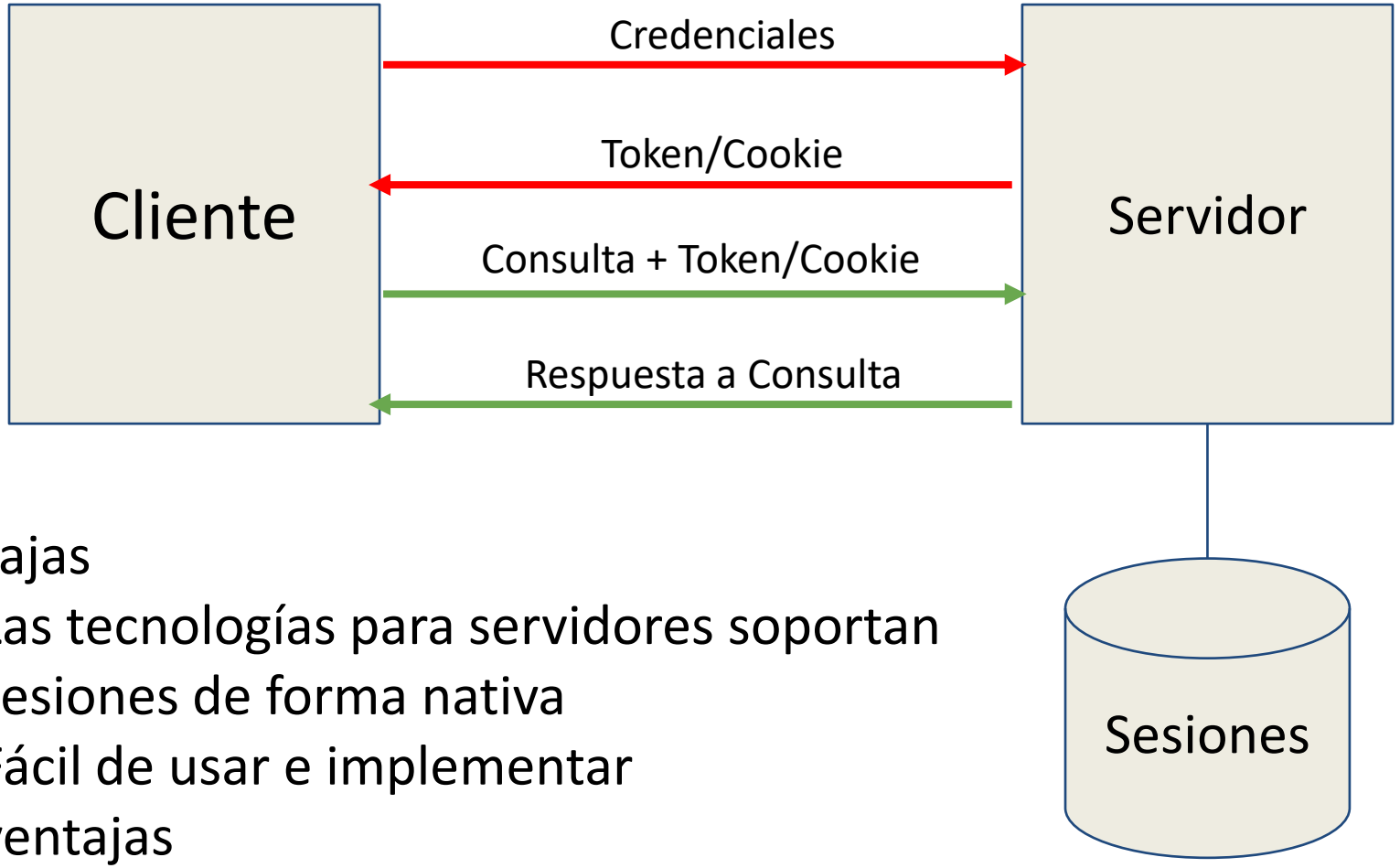
- **Autorización:** Proceso por el cual se determina la posibilidad de obtener algún tipo de información.



UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Métodos

Sesiones



Ventajas

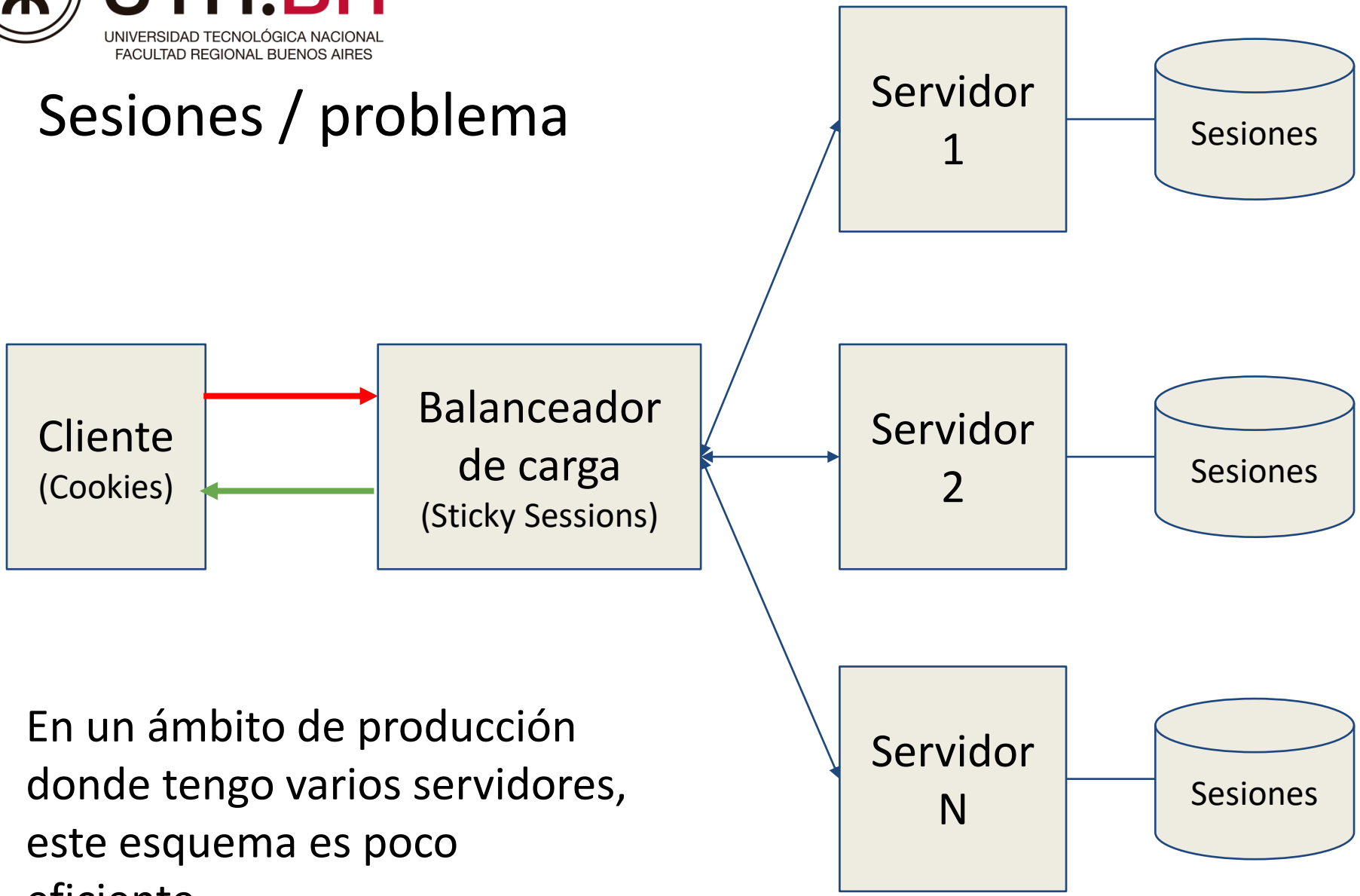
- Las tecnologías para servidores soportan sesiones de forma nativa
- Fácil de usar e implementar

Desventajas

- No es eficiente para sistemas distribuidos

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

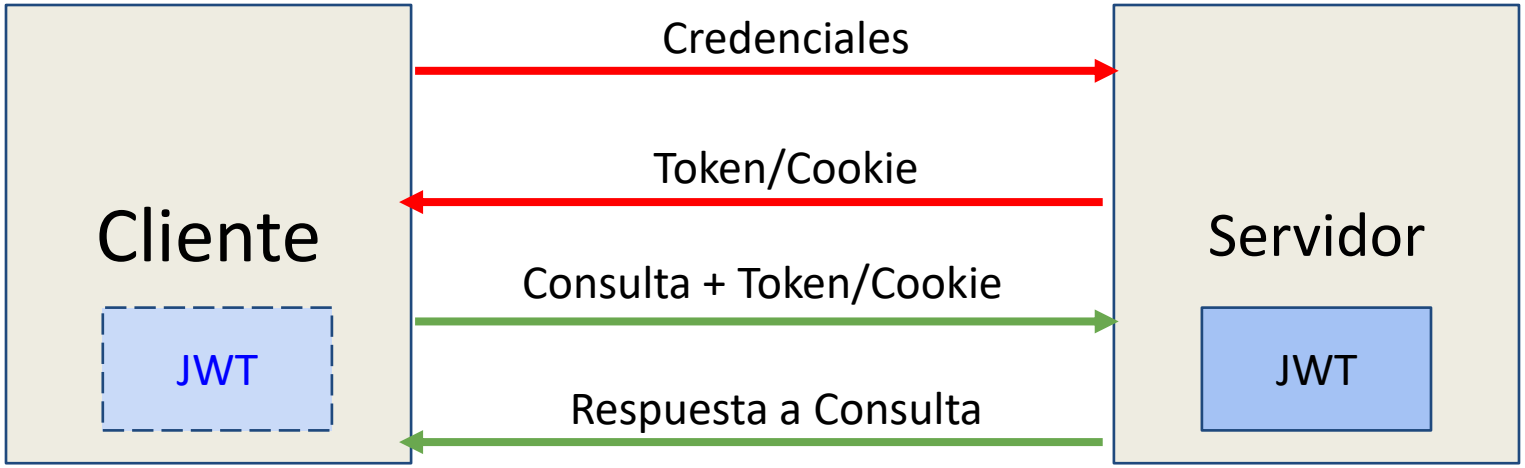
Sesiones / problema



En un ámbito de producción donde tengo varios servidores, este esquema es poco eficiente.

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

JWT



Ventajas

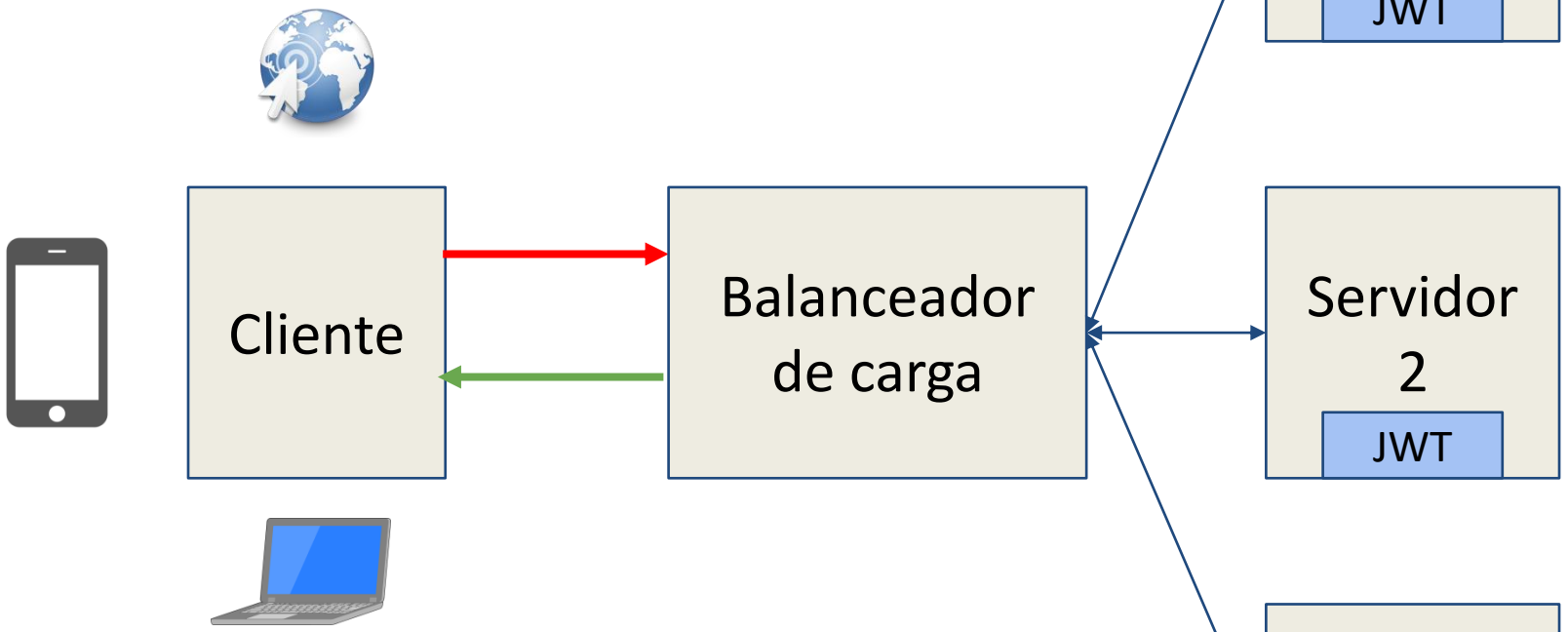
- No existe el manejo de sesiones, cada request es independiente.
- Independencia entre plataformas

Desventajas

- Cross Domain y CORS

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

JWT / solución



La utilización de un token con “información autocontenida”, es más eficiente en arquitecturas distribuidas, además que permite independencia en las plataformas cliente.

UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

JSON WEB TOKEN

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

¿Que es JWT?

JSON Web Token (**JWT**) es un **standard abierto** (RFC 7519) que **define** una forma **compacta** y **autocontenida** para la **transmisión** segura de información entre partes por medio de un objeto **JSON**.

Esta información puede ser verificada y validada porque se encuentra firmada digitalmente usando una clave secreta.

Adicionalmente, los JWT pueden ser encriptados para ocultar que la información contenida sea inaccesible.

JSON:

- Header
- Payload
- Signature

hhhh.pppppp.ssssss

Header

El header típicamente contiene dos partes: El tipo de token, que es JWT y algoritmo de hash que se utiliza para firmar el token, como HMAC, SHA256 o RSA.

Ejemplo:

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

Payload

La segunda parte del token es el payload, que contiene los “claims”. Los “claims” son valores acerca de la entidad (generalmente el usuario que se está logueando) y además puede contener datos adicionales.

Existen 3 tipos de “claims”:

- Registered
  iss (issuer), **exp** (expiration time), **sub** (subject), etc.

- Public

- Private

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

JWT / Eejemplo

HEADER:ALGORITHM & TOKEN TYPE

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

PAYLOAD:DATA

```
{
 "sub": "1234567890",
 "name": "John Doe",
 "iat": 1516239022,
 "dat":{
    "user_role": "admin",
    "user_email": "user@user.com"
 }
}
```

VERIFY SIGNATURE

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

) secret base64 encoded
```

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyLCJkYXQiOnsidXNlcl9yb2xlIjoiYWRtaW4iLCJ1c2VyX2VtYWlsIjoidXNlckB1c2VyLmNvbSJ9fQ.UVPhWil_QqdjOvv7vpxdfgRjVW8kBeLxUJ6jc3aEYIU

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Librerías

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

JWT / Librerías

- .NET
- Python
- **NodeJs**
- Java
- **JavaScript**
- Perl
- Ruby
- Elixir
- Erlang
- Go
- Groovy
- Haskell
- Haxe
- Rust
- Lua
- Scala
- D
- Cloujure
- Objetive-C
- Swift
- C
- C++
- kdb+/Q
- Delphi
- **PHP**
- Crystal
- 1C
- PostgressSQL

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Instalación – NodeJs

```
npm install jsonwebtoken
```

Uso básico

```
var jwt = require('jsonwebtoken');
var token = jwt.sign({ foo: 'bar' }, 'clave_secreta');
```

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

NodeJs - JWT / Crear nuevo token

```
var jwt = require('jsonwebtoken');
var token = jwt.sign({
  exp: Math.floor(Date.now() / 1000) + (60 * 60),
  data: 'foobar'
}, 'clave_secreta');
```

- El “claim” **exp**, define el tiempo de expiración del token
- El “claim” **data** es privado y puede utilizarse para pasar cualquier tipo de información, incluso otro JSON.

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

NodeJs - JWT / Verificar un token

```
jwt.verify(token, 'clave_secreta', function(err,
decoded) {
  console.log(decoded.data) // foobar
});
```

```
try {
  var decoded = jwt.verify(token, 'wrong-secret');
} catch(err) {
  // err
}
```

UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

BCRYPT

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

JWT - Bcrypt

Modulo utilizar para encriptar el password. Debemos ejecutar por consola:

npm install bcrypt

```
D:\pwaapp>npm install bcrypt

> bcrypt@3.0.6 install D:\pwaapp\node_modules\bcrypt
> node-pre-gyp install --fallback-to-build

node-pre-gyp WARN Using needle for node-pre-gyp https download
[bcrypt] Success: "D:\pwaapp\node_modules\bcrypt\lib\binding\bcrypt_lib.node" is installed via remote
+ bcrypt@3.0.6
added 60 packages from 47 contributors and audited 304 packages in 49.749s
found 1 moderate severity vulnerability
  run `npm audit fix` to fix them, or `npm audit` for details
```

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Aplicación

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

JWT - JWT

App.js

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

var usersRouter = require('./routes/users');
var productosRouter = require('./routes/productos');
var autenticacionRouter = require('./routes/autentication');

//Definicion de secretKey
app.set('secretKey', 'nodeRestApi'); // jwt secret token

app.use('/users', usersRouter);
app.use('/autentication', autenticacionRouter);
app.use('/products', validateUser,productosRouter);

function validateUser(req, res, next) {
  jwt.verify(req.headers['x-access-token'], req.app.get('secretKey'), function(err, decoded) {
    if (err) {
      res.json({status:"error", message: err.message, data:null});
    }else{
      // add user id to request
      req.body.userId = decoded.id;
      next();
    }
  });

}
```

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

JWT - JWT

Se incluye modulo de jwt

Se define secret key. **set(“secretKey”,”nodeAPi”)**

A los métodos privados (que solo pueden ser accedidos por usuarios autenticados) se les aplica middleware para validar su token. Metodo **validateUser**

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

JWT - JWT

Route de autenticación

```
var express = require('express');
var router = express.Router();
var autentication = require("../controllers/autentication")

/* GET home page. */
router.post('/registrar', autentication.save);
router.post('/login', autentication.login);
module.exports = router;
```

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

JWT - JWT

Controller de autenticación

```
var express = require('express');
var router = express.Router();
const bcrypt = require('bcrypt');
var autenticationModel = require("../models/autenticationModel")
const jwt = require('jsonwebtoken');
```

Require de modulos

- bcrypt
- jwt
- autenticationModel

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

JWT - JWT

Controller de autenticación

```
module.exports = {
  save: async function(req, res, next) {
    var data = await autenticationModel.create({ name: req.body.name, usuario: req.body.usuario, password: req.body.password });
    res.json({status: "success", message: "User added successfully!!!", data: data});
  },
```

Registro de usuarios. Aplica método create de mongoose

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

JWT - JWT

Controller de autenticación

```
login: async function(req, res, next) {
  var usuario = await autenticationModel.findOne({usuario:req.body.usuario});
    if (usuario) {
      if(bcrypt.compareSync(req.body.password, usuario.password)) {
        const token = jwt.sign({id: usuario._id}, req.app.get('secretKey'), { expiresIn: '1h' });
        console.log(token,usuario)
        res.json({status:"success", message: "user found!!!", data:{user: usuario, token:token}});
      }else{
        res.json({status:"error", message: "Invalid user/password!!!", data:null});
      }


    }else{
     res.json({status:"not_found", message: "user not found!!!", data:null});
    }
}
```

Login de usuarios. Si el usuario existe y el password es correcto (comparado con bcrypt porque esta encriptado) entonces se genera token

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

JWT - JWT

Model de autenticación

```
const mongoose = require('../bin/mongodb');
const bcrypt = require('bcrypt');
var UsuariosSchemna = mongoose.Schema({
    name:String,
    usuario:{
        type: String,
        required: true
        },
    password:{
        type: String,
        trim: true,
        required: true
        }
})
UsuariosSchemna.pre('save', function(next){
    this.password = bcrypt.hashSync(this.password, 10);
    next();
});
module.exports  =  mongoose.model('users',UsuariosSchemna)
```

Se aplica middleware de tipo “pre” -> “save”. Toma el password sin encriptar y lo encripta para ser almacenado