ELSEWEDY UNIVERSITY OF TECHNOLOGY
- POLYTECHNIC OF EGYPT -

Web and Security Technologies

Chapter 6: Advanced Authentication Systems

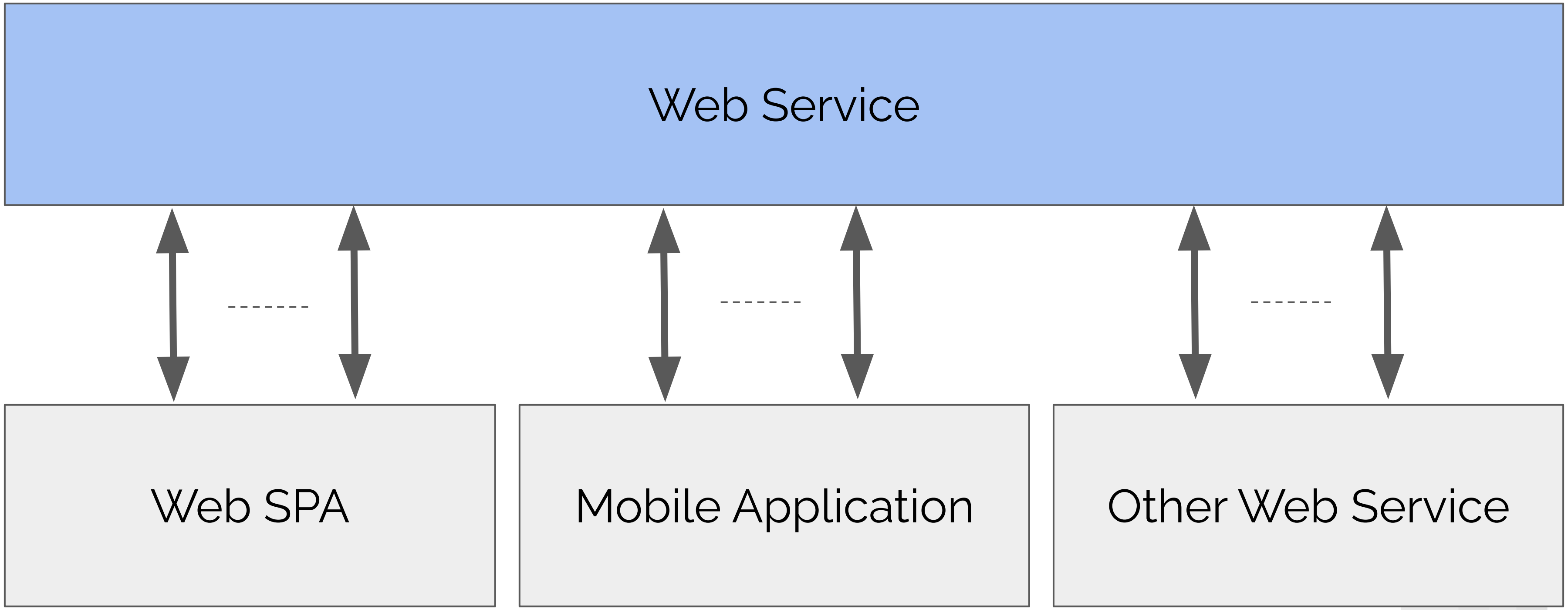
Agenda

- ❑ Token Based Authentication
- ❑ External Authentication Providers

ELSEWEDY UNIVERSITY OF TECHNOLOGY
- POLYTECHNIC OF EGYPT -

Token Based Authentication

Building Application Programming Interface (APIs) for Web



Building Application Programming Interface (APIs) for Web

- Modern Web Based System are divided into two separate tears server side code (Web Service) and front side interface (HTM/JS Pages, SPA)
- Mobile applications uses (Web Service) as to exchange information and execute commands on server side.
- Even (Web Services) can talk to each other to exchange information.
- Web Application can have multiple interfaces (End Points) to talk to provide multiple services.
- Web Application can have integrated UI like blade in addition to Web Service interface.

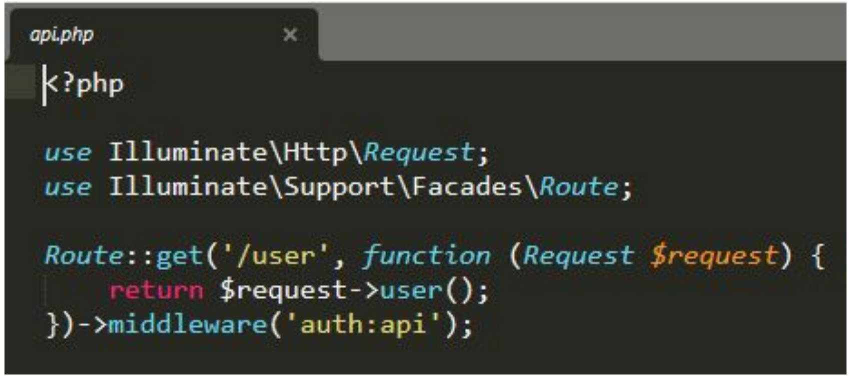
Application Programming Interface (APIs)

- APIs is a set of http requests.
- Normally APIs uses JSON format to produce results.
- APIs has it own security model.
- The authentication began after login which produce Access Token.
- The Access Token is used later in any API that required authenticated user.
- The Access Token may remain till logout, or expires.
- It is possible to Refresh the Access Token to extend the life time.
- There are many types of Access Token like Bearer and JWT

Install Passport

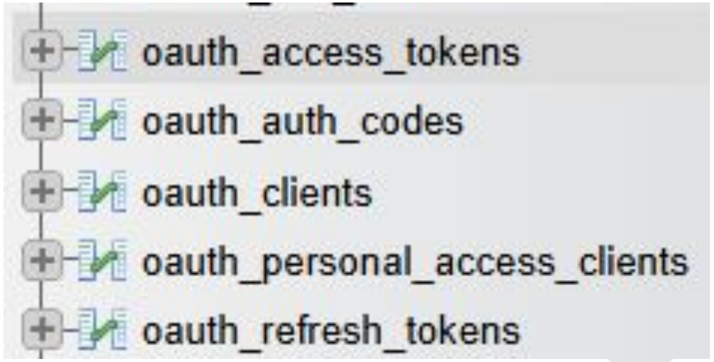
Laravel Passport is an OAuth2 server and API authentication package for Laravel applications. Essentially, it provides a straightforward way to implement secure API authentication using tokens.

```
php artisan install:api --passport
```



api.php

```php
<?php

use Illuminate\Http\Request;
use Illuminate\Support\Facades\Route;

Route::get('/user', function (Request $request) {
    return $request->user();
})->middleware('auth:api');
```



oauth_access_tokens
oauth_auth_codes
oauth_clients
oauth_personal_access_clients
oauth_refresh_tokens

Add and Configure Api in to Support Passport

config\auth.php

```
'guards' => [
    'web' => [
        'driver' => 'session',
        'provider' => 'users',
    ],
    'api' => [
        'driver' => 'passport',
        'provider' => 'users',
    ],
],
```

Modify User Model

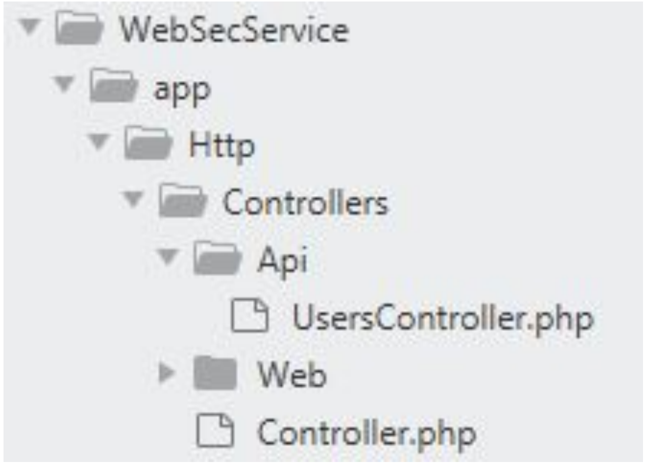
app\Models\User.php

```
<?php
...
use Laravel\Passport\HasApiTokens;

class User extends Authenticatable
{
    use HasApiTokens;
    ...
}
```

Make Api Controllers

app\Http\Controllers\Api\UsersController.php

```php
<?php
namespace App\Http\Controllers\Api;

use Illuminate\Foundation\Validation\ValidatesRequests;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use Artisan;

use App\Http\Controllers\Controller;
use App\Models\User;

class UsersController extends Controller {
    public function login(Request $request) { }
    public function users(Request $request) { }
    public function logout(Request $request) { }
}
```



Modify api routes

routes\api.php

```php
<?php

use Illuminate\Http\Request;
use Illuminate\Support\Facades\Route;
use App\Http\Controllers\Api\UsersController;

Route::post('login', [UsersController::class, 'login']);

Route::get('/users', [UsersController::class, 'users'])->middleware('auth:api');
Route::get('/logout', [UsersController::class, 'logout'])->middleware('auth:api');
```

Make Simple Login

app\Http\Controllers\Api\UsersController.php

```php
public function login(Request $request) {

    if(!Auth::attempt(['email' => $request->email, 'password' => $request->password])) {

        return response()->json(['error' => 'Invalid login info.'], 401);
    }

    $user = User::where('email', $request->email)
    ->select('id', 'name', 'email')->first();

    return response()->json(['user'=>$user->getAttributes()]);
}
```

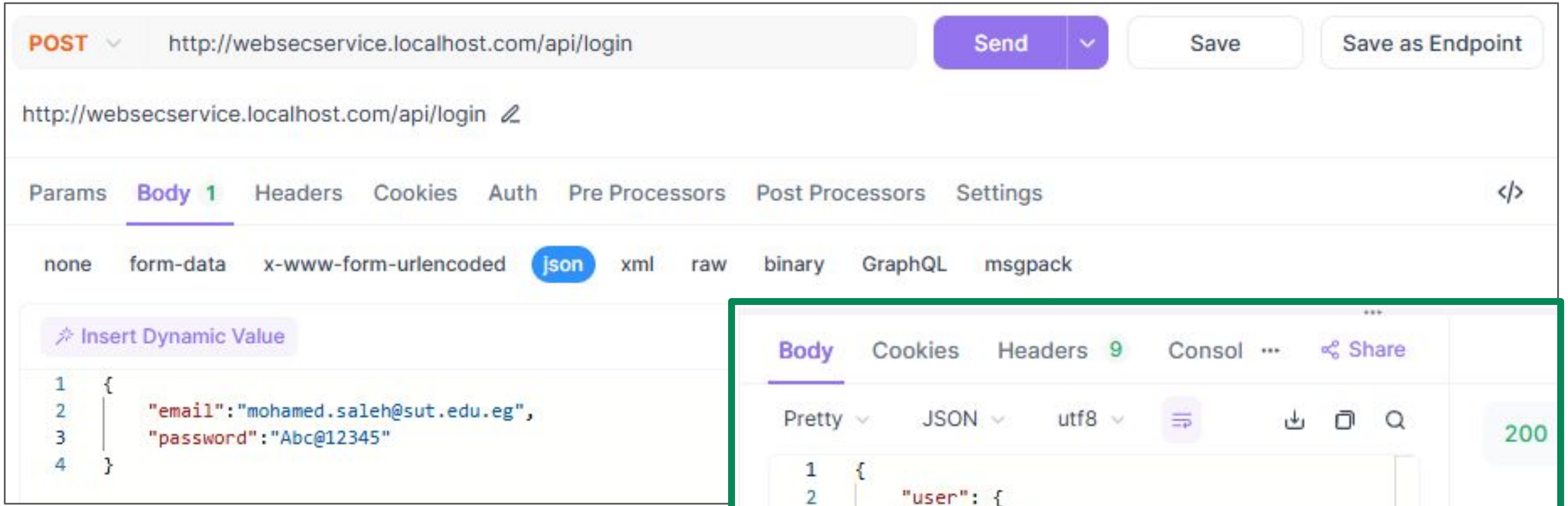
Make User Listing

app\Http\Controllers\Api\UsersController.php

```
public function users(Request $request) {

    $users = User::select('id', 'name', 'email')->get()->toArray();

    return response()->json(['users'=>$users]);
}
```

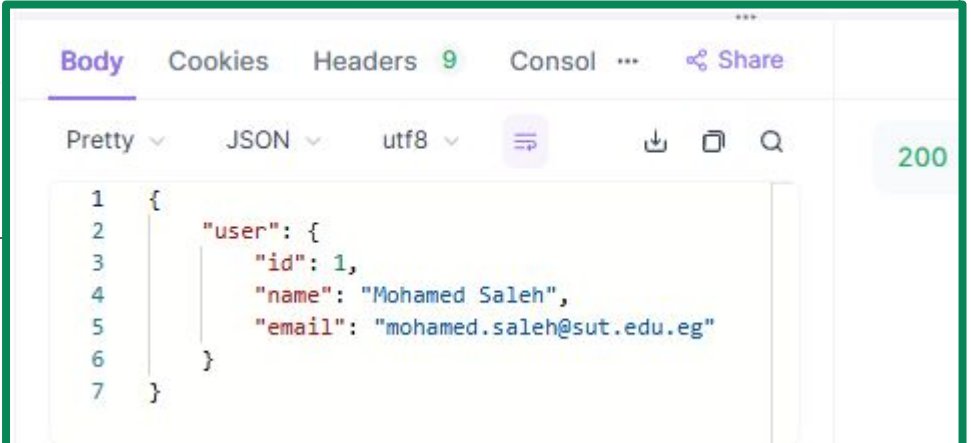
Try Your First Api using External Api Caller (postman, apidog)

https://app.apidog.com



http://websecservice.localhost.com/api/login

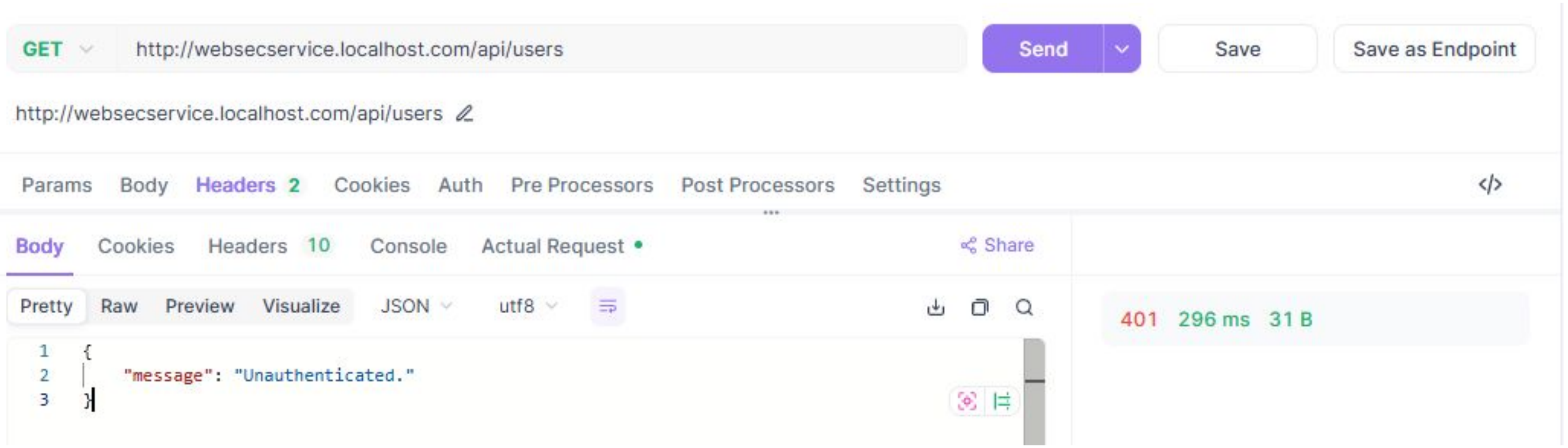
```
{
    "email":"mohamed.saleh@sut.edu.eg",
    "password":"Abc@12345"
}
```



```
{
    "user": {
        "id": 1,
        "name": "Mohamed Saleh",
        "email": "mohamed.saleh@sut.edu.eg"
    }
}
```

Try users Api

https://app.apidog.com



http://websecservice.localhost.com/api/users

http://websecservice.localhost.com/api/users

```
{
    "message": "Unauthenticated."
}
```

401 296 ms 31 B

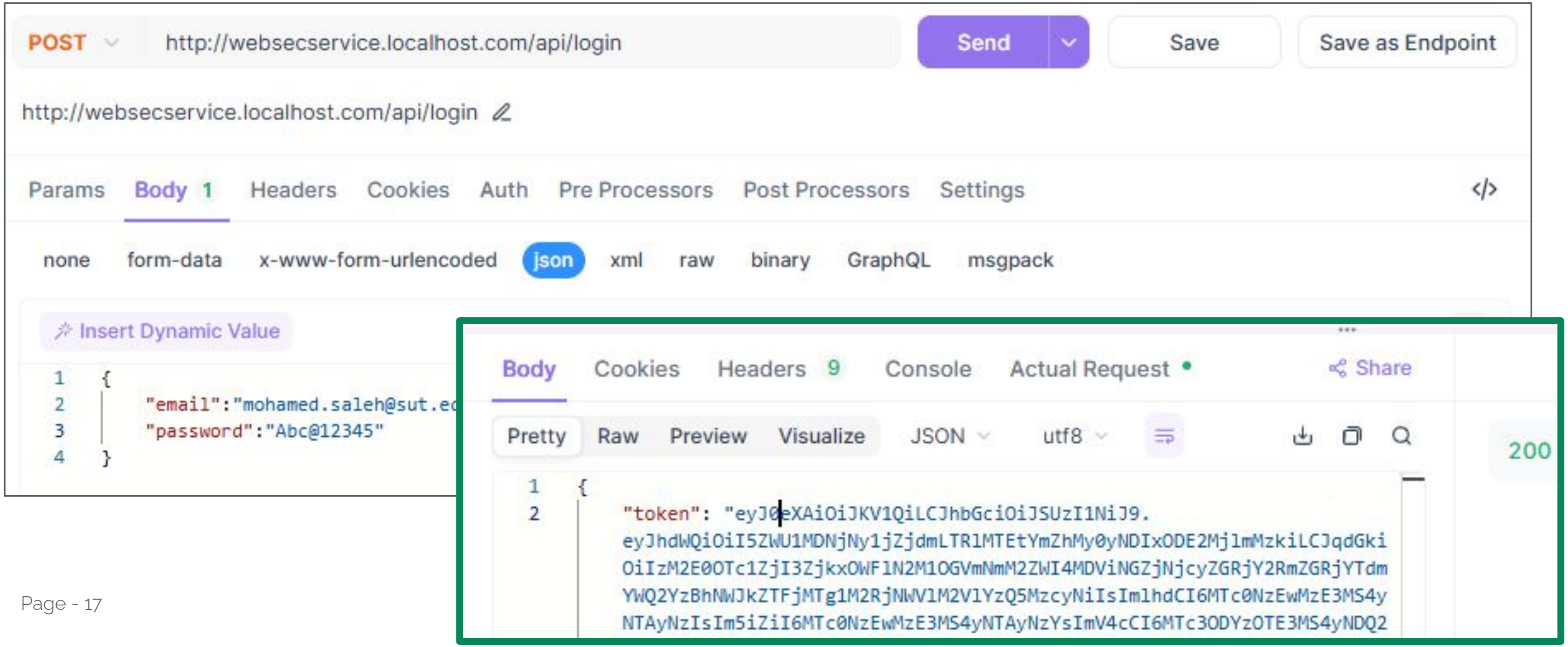
Modify login Api to Generate Passport Token

app\Http\Controllers\Api\UsersController.php

```
public function login(Request $request) {

    if(!Auth::attempt(['email' => $request->email, 'password' => $request->password])) {

        return response()->json(['error' => 'Invalid login info.'], 401);
    }

    $user = User::where('email', $request->email)->select('id', 'name', 'email')->first();

    $token = $user->createToken('app');

    return response()->json(['token'=>$token->accessToken, 'user'=>$user->getAttributes()]);
}
```

Try loing Api again and copy the token

https://app.apidog.com



POST http://websecservice.localhost.com/api/login

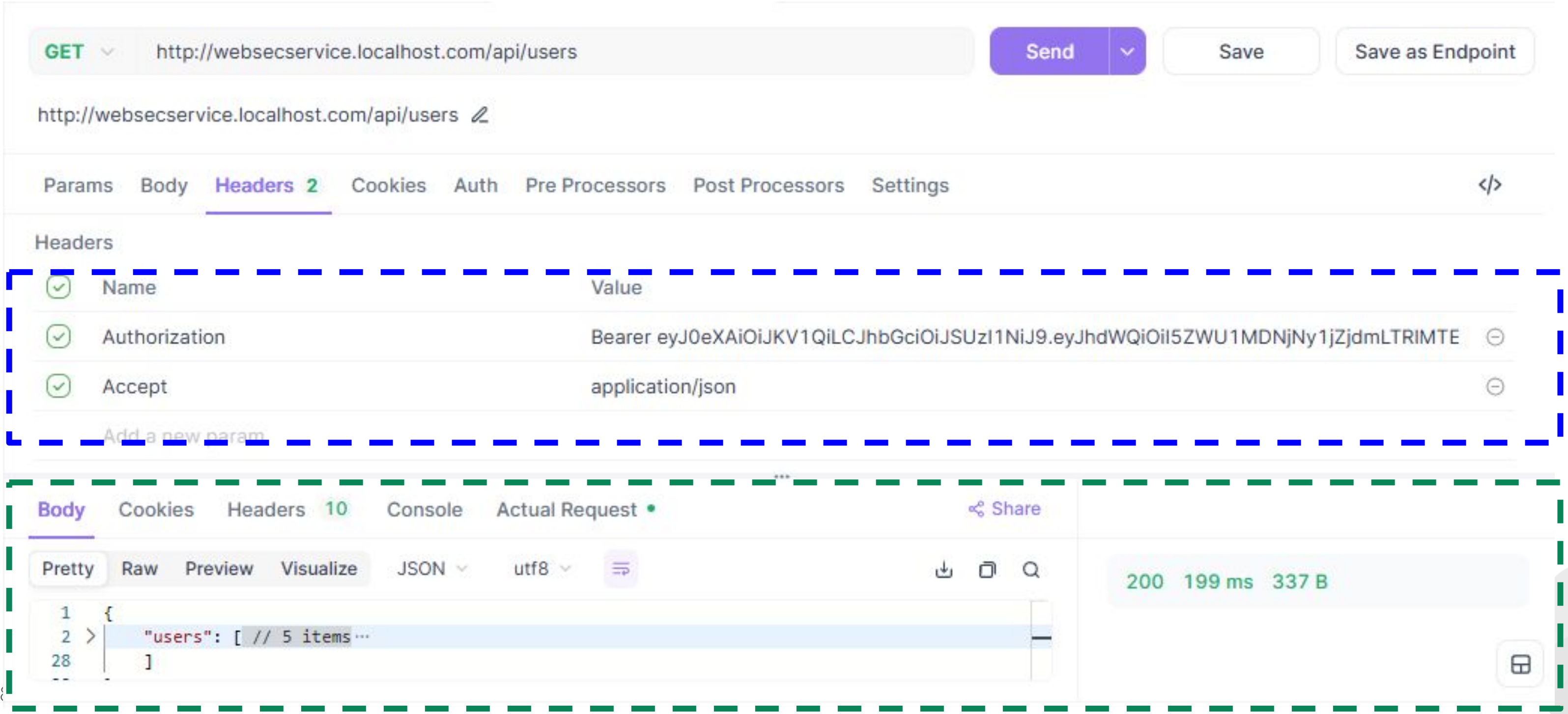
http://websecservice.localhost.com/api/login

```
{
    "email":"mohamed.saleh@sut.ed
    "password":"Abc@12345"
}
```

```
{
    "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.
    eyJhdWQiOiI5ZWU1MDNjNy1jZjdmLTR1MTEtYmZhMy0yNDIxODE2MjlmMzkiLCJqdGki
    OiIzM2E0OTc1ZjI3ZjkxOWF1N2M1OGVmNmM2ZWI4MDViNGZjNjcyZGRjY2RmZGRjYTdm
    YWQ2YzBhNWJkZTFjMTg1M2RjNWV1M2V1YzQ5MzcyNiIsImlhdCI6MTc0NzEwMzE3MS4y
    NTAyNzIsIm5iZiI6MTc0NzEwMzE3MS4yNTAyNzYsImV4cCI6MTc3ODYzOTE3MS4yNDQ2
```

Try users Api and with token

https://app.apidog.com



GET http://websecservice.localhost.com/api/users

http://websecservice.localhost.com/api/users

Headers

Name	Value
Authorization	Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJhdWQiOiI5ZWU1MDNjNy1jZjdmLTRlMTE
Accept	application/json

```
{
  "users": [ // 5 items ···
  ]
```

200 199 ms 337 B

Make logout Api to Revoke the Token

app\Http\Controllers\Api\UsersController.php

```
public function logout(Request $request) {

    auth()->user()->token()->revoke();
}
```

ELSEWEDY UNIVERSITY OF TECHNOLOGY
- POLYTECHNIC OF EGYPT -

External Authentication Providers

Identity Provider Server (IDP)

An IDP server, or Identity Provider server, is a system that manages and verifies digital identities. Think of it as a digital passport agency for the internet. Its primary functions include:

- Creating, storing, and managing digital identities
- Authenticating users and issuing security tokens
- Enhanced Security
- Single Sign-On (SSO) and Federated Identity

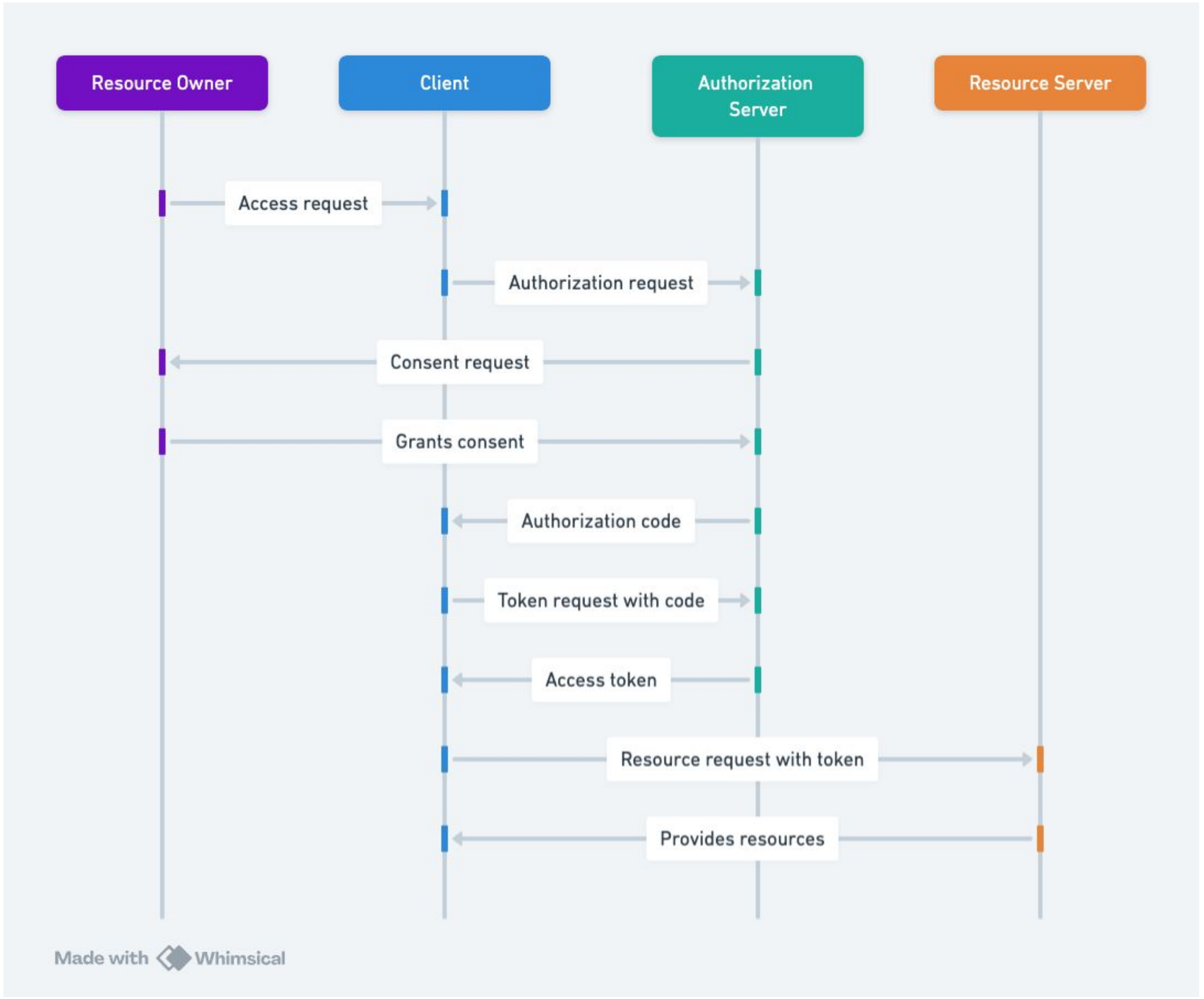
Example: Open ID based on (OAuth 2.0), SAML

OAuth 2.0 Authorization

Ah, the OAuth 2.0 Authorization Server! Think of it as the wise gatekeeper of user data, standing between applications that want access and the users who own that information. It's the central authority that manages identities and decides who gets permission to do what. Core Responsibilities:

- Authentication
- Authorization
- Issuing Tokens
- Managing Tokens

OAuth 2.0 Authorization Sequence



OAuth 2.0 Authorization Sequence

- Client Initiates the Authorization Request
  - response_type=code, client_id, redirect_uri, scope, state
- Authorization Server Issues an Authorization Code
  - code, state
- Client Requests an Access Token
  - grant_type=authorization_code, code, redirect_uri, client_id, client_secret
- Authorization Server Issues Access and Refresh Tokens
  - access_token, token_type, expires_in, refresh_token, scope
- Client Accesses Protected Resources

OAuth 2.0 Authorization Sequence

- Refreshing the Access Token
  - grant_type=refresh_token, refresh_token, client_id, client_secret, scope
- The Authorization Server verifies the refresh_token and the client's credentials. If valid, it issues:
  - new access_token (and potentially a new refresh_token)

Self Study for Teams

- Implement OAuth 2.0 Authorization.
- Use it in with Test Client application to gain access to resource.
- Project Teams should demonstrate their achievement during next lecture.
- Teams with successful demo take 5 degree bonus in course work.