

Telepítési útmutatás Windows eszközre:

Xampp telepítése:

1. Nyissa meg az Apache Friends webhelyet:
<https://www.apachefriends.org/download.html>
2. Kattintson a Letöltés gombra az XAMPP Windows-verziójához, és mentse a fájlt a számítógépére.
3. Gyors megjegyzés: A projekt a PHP 8.0.12 verzióval lett készítve és tesztelve. Amennyiben másik verzió van telepítve a gépen, telepítse a PHP 8.0.12 vagy 8.0.2 verziót, de mindenképpen a PHP 8.0.12-es verzió az ajánlott.
4. A telepítő elindításához kattintson duplán a letöltött fájlra.
5. Kattintson az OK gombra.
6. Kattintson a Tovább gombra.
7. Az XAMPP különféle telepíthető összetevőket kínál, mint például a MySQL, phpMyAdmin, PHP, Apache stb. A legtöbb esetben ezeket az összetevőket fogja használni, ami azt jelenti, hogy ajánlott hagyni az alapértelmezett beállításokat.
8. Kattintson a Tovább gombra.
9. Használja az alapértelmezett telepítési helyet. (Vagy válasszon másik mappát a szoftver telepítéséhez a „Mappa kiválasztása” mezőben.)
10. Kattintson a Tovább gombra.
11. Válassza ki az XAMPP vezérlőpult nyelvét.
12. Kattintson a Tovább gombra.
13. Törölje a További információ a Bitnami for XAMPP-ról lehetőséget.
14. Kattintson a Tovább gombra.
15. Kattintson ismét a Tovább gombra.
16. Kattintson a Hozzáférés engedélyezése gombra, hogy engedélyezze az alkalmazást a Windows tűzfalon (ha van).
17. Kattintson a Befejezés gombra.
18. A lépések végrehajtása után elindul az XAMPP Vezérlőpult, és megkezdheti a webservert környezet konfigurálását.

Az XAMPP konfigurálása Windows 10 rendszeren

1. Az XAMPP vezérlőpult három fő részből áll. A Modulok részben megtalálja az összes elérhető webszolgáltatást. Az egyes szolgáltatásokat a Start gombra kattintva indíthatja el.
2. Amikor elindít néhány szolgáltatást, beleértve az Apache-t és a MySQL-t, a jobb oldalon, az egyes szolgáltatások által használt folyamatazonosító (PID) és TCP/IP-port (Port) száma is megjelenik. Például alapértelmezés szerint az Apache a 80-as és a 443-as TCP/IP portot, míg a MySQL a 3306-os TCP/IP portot használja.
3. Az Adminisztrálás gombra kattintva elérheti az egyes szolgáltatások adminisztrációs irányítópultját, és ellenőrizheti, hogy minden megfelelően működik-e.
4. Az alapértelmezett beállításoknak működniük kell a legtöbb felhasználó számára, aki XAMPP-t használ a webhely futtatásához szükséges tesztelési környezet létrehozásához. A beállítási konfigurációtól függően azonban előfordulhat, hogy módosítania kell az Apache szerver TCP/IP portszámát, az adatbázis feltöltési méretét, vagy be kell állítania a phpMyAdmin jelszavát.
5. A beállítások módosításához a megfelelő szolgáltatás Config gombját kell használnia. Például meg kell nyitnia a httpd.conf fájlt az Apache-kiszolgáló beállításainak módosításához, a my.ini fájlt pedig a MySQL-beállítások módosításához.
6. Xampp Control panel indítása
7. Apache és MySQL rákattintani a Start gombra
8. MYSQL gombon az admin gombra kattintva eljutunk a <http://localhost/phpmyadmin/> oldalra

phpMyAdmin konfigurálás:

1. Új adatbázis létrehozása: "laravel" címen

2. Ez fontos, mivel az env állományban van meghatározva az adatbázis neve és a migrációs séma a projektben előre meg van határozva, azaz ebbe az adatbázisba fogjuk migrálni az adatbázis táblákat és azoknak a Laravel php-ban megadott relációs sémáját is.

Composer telepítése:

1. Composer telepítése exe fájl formátumban az alábbi webhelyről:
<https://getcomposer.org/download/>
2. Letöltött exe fájl futtatása és telepítése a legújabb php verzóval
3. Parancssor megnyitása és a "composer" szó beírásával majd az "Enter" szó leütésével ellenőrizzük, hogy a composer sikeresen települt a Windows 10 környezetben

Laravel telepítése:

1. composer global require laravel/installer

Projekt mappa készítése és Laravel verzió telepítése:

1. composer create-project --prefer-dist laravel/laravel ProjektNév
2. A fenti parancsot ki tudjuk egészíteni a "8.83.*" és ennek használatával biztosak lehetünk benne, hogy megkapja azt a verziót amivel a projekt készítve és tesztelve lett.
3. php artisan make:auth
4. php artisan migrate

Jetstream telepítése:

1. composer require laravel/Jetstream
2. php artisan jetstream:install livewire
3. php artisan jetstream:install inertia
4. npm install
5. npm run dev
6. php artisan migrate
7. php artisan vendor:publish --tag=jetstream-views
8. npm run dev

Laravel Sanctum telepítése:

1. `composer require laravel/sanctum`
2. `php artisan vendor:publish --
provider="Laravel\Sanctum\SanctumServiceProvider"`
3. `php artisan migrate`

Laravel Fortify telepítése

1. `composer require laravel/fortify`
2. `php artisan vendor:publish --
provider="Laravel\Fortify\FortifyServiceProvider"`
3. `php artisan migrate`

Doctrine Dbal telepítése

1. `composer require doctrine/dbal`

Clone Laravel projekt:

1. A file explorerben job egér gomb klikkel vagy Git bash here vagy open with VS code-al meg kell nyitni egy mappát és oda klónozni a git clone parancsal a project mappát avagy azt letölteni a Github URL-ről zip formátumban. A klónozáshoz új Terminál ablakot kell létesíteni a VS Code-ban vagy a projekt mappában job klikk és a GIT bash here parancsal lehet Terminált létesíteni.
2. **Egy Terminálban beírni a következő parancsokat:**
3. `git clone 'url'`
4. ha megvan a projekt a lokális mappában akkor be kell lépni az AuthFinal-HMS klónozott mappába és annak az egész tartalmát Ctrl+ A paranccsal kijelölni és Ctrl-C / Ctrl +V billentyűk segítségével át kell másolni a korábban composer-el létrehozott **ProjektNév** mappába ahol már a teljes környezet kiépítésre került (`composer create-project --prefer-dist laravel/laravel ProjektNév`)

Név	Módosítás dátuma	Típus	Méret
.git	2022. 04. 13. 11:22	Fájlmappa	
api_documentation	2022. 04. 09. 20:56	Fájlmappa	
app	2022. 03. 19. 0:26	Fájlmappa	
bootstrap	2022. 03. 19. 0:26	Fájlmappa	
config	2022. 03. 23. 22:16	Fájlmappa	
database	2022. 03. 19. 0:26	Fájlmappa	
node_modules	2022. 04. 11. 20:09	Fájlmappa	
public	2022. 03. 19. 0:57	Fájlmappa	
resources	2022. 03. 23. 21:41	Fájlmappa	
routes	2022. 03. 19. 20:53	Fájlmappa	
storage	2022. 03. 19. 0:26	Fájlmappa	
tests	2022. 03. 19. 0:26	Fájlmappa	
vendor	2022. 04. 15. 22:26	Fájlmappa	
.editorconfig	2022. 03. 19. 0:26	EDITORCONFIG fájl	1 KB
.env	2022. 04. 08. 20:31	ENV fájl	1 KB
.env.example	2022. 03. 19. 0:26	EXAMPLE fájl	1 KB
.gitattributes	2022. 03. 19. 0:26	Szöveges dokume...	1 KB
.gitignore	2022. 03. 19. 0:26	Szöveges dokume...	1 KB
.styleci.yml	2022. 03. 19. 0:26	YML fájl	1 KB
Agent	2022. 03. 19. 0:26	Fájl	0 KB
artisan	2022. 03. 19. 0:26	Fájl	2 KB
composer.json	2022. 04. 15. 22:26	Adobe After Effect...	2 KB
composer.lock	2022. 04. 15. 22:26	LOCK fájl	326 KB
laravel.sql	2022. 04. 13. 9:52	SQL fájl	28 KB
package.json	2022. 03. 23. 20:41	Adobe After Effect...	1 KB
package-lock.json	2022. 04. 11. 20:09	Adobe After Effect...	344 KB
phpunit.xml	2022. 03. 19. 0:26	XML dokumentum	2 KB
server.php	2022. 03. 19. 0:26	PHPFile	1 KB
tailwind.config.js	2022. 03. 23. 14:08	JavaScript-fájl	1 KB
~WRL0005.tmp	2022. 04. 15. 22:28	TMP fájl	137 KB
webpack.mix.js	2022. 03. 31. 23:00	JavaScript-fájl	1 KB

5. Be kell lépni a laravel **ProjektNév** mappába, amit korábban megadott a Laravel project készítésénél
6. A **ProjektNév** mappát megnyitni VS Code alkalmazással
7. Itt **New Terminal** paranccsal új terminált létesítünk
8. **composer install**
9. **npm install**
10. **npm run dev**
11. Duplikálnunk kell az **.env.example** fájlt a következő paranccsal:
 - a. **cp .env.example .env**
12. Env file konfigurálása a lejeb található env paraméterek bemásolása
13. **php artisan key:generate**
14. **php artisan migrate**
15. **php artisan serve**
16. **npm run watch**

Env file konfigurálása:

Az alábbi változók bemásolása a. env fájlba kivéve az APP_KEY változót , mivel azt a “php artisan key:generate” parancs generálja :

```
APP_NAME="HMS System"
APP_ENV=local
APP_KEY=base64:ET5PUEUNqm6u1V7I1eqgIsqTdkinieqkAhwTm6wLdKM=
APP_DEBUG=true
APP_URL=http://127.0.0.1:8000
ASSET_URL=/public/assets

LOG_CHANNEL=stack
LOG_LEVEL=debug

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=laravel
DB_USERNAME=root
DB_PASSWORD=

BROADCAST_DRIVER=log
CACHE_DRIVER=file
QUEUE_CONNECTION=sync
SESSION_DRIVER=database
SESSION_LIFETIME=120

MEMCACHED_HOST=127.0.0.1

REDIS_HOST=127.0.0.1
REDIS_PASSWORD=null
REDIS_PORT=6379

MAIL_MAILER=smtp
MAIL_HOST=smtp.mailtrap.io
MAIL_PORT=2525
MAIL_USERNAME=4c055854bb8213
```

```
MAIL_PASSWORD=89b8ac8b57a4b4
MAIL_ENCRYPTION=tls
MAIL_FROM_ADDRESS= register@budapesthotels.com
MAIL_FROM_NAME="${APP_NAME}"

AWS_ACCESS_KEY_ID=
AWS_SECRET_ACCESS_KEY=
AWS_DEFAULT_REGION=us-east-1
AWS_BUCKET=

PUSHER_APP_ID=
PUSHER_APP_KEY=
PUSHER_APP_SECRET=
PUSHER_APP_CLUSTER=mt1

MIX_PUSHER_APP_KEY="${PUSHER_APP_KEY}"
MIX_PUSHER_APP_CLUSTER="${PUSHER_APP_CLUSTER}"
```

Email küldés:

A web appon törént regisztráció után a regisztrációt jóvá kell hagyni email-en keresztül, egy felhasználó csak azután tudja használni a programot, hogyha a regisztrációt követően igazolta email címét, valamint jelszó visszaállítás, helyre állítás is emailben küldött linken keresztül működik.

Az email funkciók teszteléséhez és használatához a **MailTrap** szolgáltatását vettük igénybe.

Mailtrap bemutatása:

A Mailtrap hamis SMTP-kiszolgálót biztosít a fejlesztőcsapat számára, amellyel tesztelheti, megtekintheti és megoszthatja a gyártás előtti környezetekből küldött e-maileket, és valós adatokkal tesztelheti anélkül, hogy fennállna a valódi ügyfelek spamküldésének veszélye. A Railsware készítette, és számos fejlesztési feladathoz a Mailtrap használata ingyenes.

Lényegében regisztrál a Mailtrap szolgáltatásra, és minden e-mailt elküld a gyártás előtti környezetből a hamis Mailtrap SMTP-kiszolgálón keresztül.

A MailTrap segítségével e-maileket rögzíthet a tesztelési fejlesztői és állomásozó környezetekből

Ezután minden levele a Mailtraphez tartozik. Megtekintheti és hibakeresheti e-mailjeit a Mailtrap barátságos grafikus felületén.

A Mailtrap segítségével akár valódi felhasználói e-maileket tartalmazó kiíratásokat is elhelyezhet a termelési adatbázisában, tesztek segítségével az átmeneti kiszolgálón. Automatizált tesztjei ellentmondhatnak a valós adatoknak – ha e-mailt küldenek a Mailtrap-on keresztül, így kiküszöbölhető annak a kockázata, hogy a teszte-mailek valódi ügyfelek e-mail címére menjenek.

URL: <https://mailtrap.io/>

Ahhoz, hogy regisztrálni tudjunk a honlapon, a fenti URL-en be kell jelentkezni az alábbi paraméterekkel.

A projekthez tartozó felhasználónév és jelszó, ezekkel kell belépni a honlapon.

Felhasználónév: **citronejoe0123@gmail.com**

Jelszó: **Emailtest1234**

Adatbázis feltöltése:

1. A Laravel.sql fájlt a <http://localhost/phpmyadmin/> laravel adatbázisban az Import gombra kattintva, majd a 'Choose file' gombra kattintva be tudjuk illeszteni, majd a 'Go' gomb megnyomásával be is importáljuk az összes korábban exportált adatot.
2. Opcionálisan: A user táblát fel tudjuk tölteni random user adatokkal a AuthFinal-HMS\database\seeders\DatabaseSeeder.php elérési útvonalon található fájl alapján a faker segítségével, a "php artisan db:seed" parancsal

API dokumentációk elérése a weben:

1. Swagger:

- a. https://app.swaggerhub.com/apis-docs/wokres/product-controller_rest/1.0.0

2. Postman

- a. <https://documenter.getpostman.com/view/19126249/UVyxRZv2#ea026aee-00a1-4287-9517-85a8d58b8977>

ProductController OPENAPI 3.0 specifikáció:

```
openapi: 3.0.0
info:
  title: ProductControllerREST
  version: 1.0.0
servers:
  - url: http://localhost:8000
components:
  securitySchemes:
    bearerAuth:
      type: http
      scheme: bearer
    noauthAuth:
      type: http
      scheme: noauth
paths:
  /api/products:
    post:
      tags:
        - default
      summary: http://localhost:8000/api/products
      requestBody:
        content:
          application/x-www-form-urlencoded:
            schema:
              properties:
```

```
      name:
        type: string
        example: Gingerbread
      category:
        type: string
        example: Food
      description:
        type: string
        example: Sweet bread, allergic
      price:
        type: integer
        example: '955'
  security:
    - bearerAuth: []
  parameters:
    - name: Accept
      in: header
      schema:
        type: string
        example: application/json
  responses:
    '201':
      description: Created
      headers:
        Host:
          schema:
            type: string
            example: localhost:8000
        Date:
          schema:
            type: string
            example: Fri, 08 Apr 2022 20:04:53 GMT
        Connection:
          schema:
            type: string
            example: close
        X-Powered-By:
          schema:
            type: number
```

```
        example: PHP/8.0.12
Cache-Control:
  schema:
    type: string
    example: no-cache, private
Content-Type:
  schema:
    type: string
    example: application/json
X-RateLimit-Limit:
  schema:
    type: integer
    example: '60'
X-RateLimit-Remaining:
  schema:
    type: integer
    example: '58'
Access-Control-Allow-Origin:
  schema:
    type: string
    example: '*'
content:
  application/json:
    schema:
      type: object
    example:
      name: Ketchup
      category: Food
      description: Small portion of ketchup
      price: '248'
      updated_at: '2022-04-08T20:04:53.000000Z'
      created_at: '2022-04-08T20:04:53.000000Z'
      id: 31
'401':
  description: Unauthorized
  headers:
    Host:
      schema:
        type: string
```

```
        example: localhost:8000
Date:
  schema:
    type: string
    example: Fri, 08 Apr 2022 19:46:25 GMT
Connection:
  schema:
    type: string
    example: close
X-Powered-By:
  schema:
    type: number
    example: PHP/8.0.12
Cache-Control:
  schema:
    type: string
    example: no-cache, private
Content-Type:
  schema:
    type: string
    example: application/json
Access-Control-Allow-Origin:
  schema:
    type: string
    example: '*'
content:
  application/json:
    schema:
      type: object
    example:
      message: Unauthenticated.
'403':
  description: Forbidden
  headers:
    Host:
      schema:
        type: string
        example: localhost:8000
    Date:
```

```
    schema:
      type: string
      example: Sat, 09 Apr 2022 18:40:36 GMT
Connection:
  schema:
    type: string
    example: close
X-Powered-By:
  schema:
    type: number
    example: PHP/8.0.12
Cache-Control:
  schema:
    type: string
    example: no-cache, private
Content-Type:
  schema:
    type: string
    example: application/json
X-RateLimit-Limit:
  schema:
    type: integer
    example: '60'
X-RateLimit-Remaining:
  schema:
    type: integer
    example: '59'
Access-Control-Allow-Origin:
  schema:
    type: string
    example: '*'
content:
  application/json:
    schema:
      type: object
    example:
      message: Forbidden
get:
  tags:
```

```
- default
summary: http://localhost:8000/api/products
security:
  - bearerAuth: []
parameters:
  - name: Accept
    in: header
    schema:
      type: string
    example: application/json
responses:
  '200':
    description: OK
    headers:
      Host:
        schema:
          type: string
          example: localhost:8000
      Date:
        schema:
          type: string
          example: Fri, 08 Apr 2022 20:02:33 GMT
      Connection:
        schema:
          type: string
          example: close
      X-Powered-By:
        schema:
          type: number
          example: PHP/8.0.12
      Cache-Control:
        schema:
          type: string
          example: no-cache, private
      Content-Type:
        schema:
          type: string
          example: application/json
      X-RateLimit-Limit:
```

```
    schema:
      type: integer
      example: '60'
  X-RateLimit-Remaining:
    schema:
      type: integer
      example: '59'
  Access-Control-Allow-Origin:
    schema:
      type: string
      example: '*'
```

```
content:
  application/json:
    schema:
      type: object
    example:
      - id: 1
        name: Pina Colada
        category: Cocktail
        description: Pineapple, rum, creamy
```

cocktail

```
      price: '2399.00'
      created_at: '2022-03-20T19:10:10.000000Z'
      updated_at: '2022-03-31T21:44:06.000000Z'
    - id: 2
      name: Beef steak
      category: Food
      description: Argentinian beef
      price: '9900.00'
      created_at: null
      updated_at: '2022-03-27T15:02:11.000000Z'
    - id: 3
      name: Chicken steak
      category: Food
      description: Countryside chicken
      price: '3000.00'
      created_at: null
      updated_at: null
    - id: 4
```



```
    name: Beef burger
    category: Food
    description: 330g meat, salad, mayo, burger
    price: '3999.99'
    created_at: null
    updated_at: '2022-03-27T14:46:37.000000Z'
- id: 5
  name: Long Island
  category: Cocktail
  description: Rum, Tequila, Vodka, Coke
  price: '3999.00'
  created_at: '2022-03-21T02:03:35.000000Z'
  updated_at: '2022-03-23T08:02:03.000000Z'
- id: 8
  name: Mliano spaghetti
  category: Food
  description: Meatballs with pasta
  price: '2699.00'
  created_at: '2022-03-23T07:59:36.000000Z'
  updated_at: '2022-03-23T07:59:36.000000Z'
- id: 9
  name: Chilli con carne
  category: Food
  description: Mexican spicy minced meat
  price: '2699.00'
  created_at: '2022-03-23T08:00:06.000000Z'
  updated_at: '2022-03-23T08:00:06.000000Z'
- id: 10
  name: Tequila Sunrire
  category: Cocktail
  description: Lemony tequila cocktail
  price: '2999.00'
  created_at: '2022-03-23T08:25:54.000000Z'
  updated_at: '2022-03-23T08:25:54.000000Z'
- id: 11
  name: Sparkling water
  category: Beverage
  description: Still water sparkling
  price: '499.00'
```

```
    created_at: '2022-03-23T08:34:32.000000Z'
    updated_at: '2022-03-23T08:34:32.000000Z'
  - id: 13
    name: Coca-Cola
    category: Beverage
    description: Cola 0.5l
    price: '499.00'
    created_at: '2022-03-23T08:35:55.000000Z'
    updated_at: '2022-03-23T08:35:55.000000Z'
  - id: 14
    name: Pepsi Coke
    category: Beverage
    description: Coke 0.5l
    price: '499.00'
    created_at: '2022-03-23T08:36:09.000000Z'
    updated_at: '2022-03-23T08:36:09.000000Z'
  - id: 15
    name: Fanta Lemon
    category: Beverage
    description: Lemon drink 0.5l
    price: '499.00'
    created_at: '2022-03-23T08:36:36.000000Z'
    updated_at: '2022-03-23T08:36:36.000000Z'
  - id: 16
    name: Coca-Cola can
    category: Beverage
    description: Coke drink 0.33l
    price: '499.00'
    created_at: '2022-03-23T08:36:56.000000Z'
    updated_at: '2022-03-23T08:36:56.000000Z'
  - id: 18
    name: Lemonade
    category: Beverage
    description: Lemon juice 0.33 glass
    price: '499.00'
    created_at: '2022-03-23T08:37:46.000000Z'
    updated_at: '2022-03-23T08:37:46.000000Z'
  - id: 21
    name: Töpörtyű
```

```
category: Food
description: Hungarian pork specialty
price: '1999.00'
created_at: '2022-03-27T14:52:15.000000Z'
updated_at: '2022-03-27T14:58:50.000000Z'
- id: 22
  name: Chicken wings
  category: Food
  description: Spicy grilled wings
  price: '2350.00'
  created_at: '2022-03-27T14:53:50.000000Z'
  updated_at: '2022-03-27T14:53:50.000000Z'
- id: 23
  name: Ham and eggs
  category: Food
  description: Grilled ham and eggs
  price: '2290.00'
  created_at: '2022-03-27T14:56:18.000000Z'
  updated_at: '2022-03-27T15:00:16.000000Z'
- id: 24
  name: Vodka Soda
  category: Cocktail
  description: Kalinka Vodka and Soda 2dl
  price: '2999.00'
  created_at: '2022-03-27T15:01:10.000000Z'
  updated_at: '2022-03-27T15:01:22.000000Z'
- id: 25
  name: Goulash soup
  category: Food
  description: Pork chops, potato, vegetable,
thick soup
  price: '2999.00'
  created_at: '2022-03-27T15:13:49.000000Z'
  updated_at: '2022-03-27T15:13:49.000000Z'
- id: 26
  name: Ginger tea
  category: Drink
  description: Hot natural ginger tea
  price: '799.00'
```

```
      created_at: '2022-03-27T15:15:24.000000Z'
      updated_at: '2022-03-27T15:15:24.000000Z'
    - id: 27
      name: Black tea
      category: Drink
      description: Earl gray black tea
      price: '599.00'
      created_at: '2022-03-27T15:16:43.000000Z'
      updated_at: '2022-03-27T15:16:56.000000Z'
    - id: 29
      name: Grilled chicken
      category: Food
      description: Spicy chicken grilled
      price: '2850.00'
      created_at: '2022-04-08T19:07:48.000000Z'
      updated_at: '2022-04-08T19:07:48.000000Z'
    - id: 30
      name: Honey
      category: Food
      description: small box of honey
      price: '699.00'
      created_at: '2022-04-08T19:29:14.000000Z'
      updated_at: '2022-04-08T19:30:29.000000Z'
```

/api/logout:

post:

tags:

- default

summary: http://localhost:8000/api/logout

requestBody:

content:

application/x-www-form-urlencoded:

schema:

properties:

name:

type: string

example: Andras2

email:

type: string

example: andras2@gmail.com

```
        password:
          type: string
          example: xyz23
        password_confirmation:
          type: string
          example: xyz23
security:
  - bearerAuth: []
parameters:
  - name: Accept
    in: header
    schema:
      type: string
      example: application/json
responses:
  '200':
    description: OK
    headers:
      Host:
        schema:
          type: string
          example: localhost:8000
      Date:
        schema:
          type: string
          example: Fri, 08 Apr 2022 20:02:07 GMT
      Connection:
        schema:
          type: string
          example: close
      X-Powered-By:
        schema:
          type: number
          example: PHP/8.0.12
      Cache-Control:
        schema:
          type: string
          example: no-cache, private
      Content-Type:
```

```
    schema:
      type: string
      example: application/json
  X-RateLimit-Limit:
    schema:
      type: integer
      example: '60'
  X-RateLimit-Remaining:
    schema:
      type: integer
      example: '59'
  Access-Control-Allow-Origin:
    schema:
      type: string
      example: '*'
  content:
    application/json:
      schema:
        type: object
      example:
        message: You have succesfully logged out
'401':
  description: Unauthorized
  headers:
    Host:
      schema:
        type: string
        example: localhost:8000
    Date:
      schema:
        type: string
        example: Fri, 08 Apr 2022 20:02:23 GMT
  Connection:
    schema:
      type: string
      example: close
  X-Powered-By:
    schema:
      type: number
```

```
        example: PHP/8.0.12
Cache-Control:
  schema:
    type: string
    example: no-cache, private
Content-Type:
  schema:
    type: string
    example: application/json
Access-Control-Allow-Origin:
  schema:
    type: string
    example: '*'
content:
  application/json:
    schema:
      type: object
    example:
      message: Unauthenticated.
/api/login:
  post:
    tags:
      - default
    summary: http://localhost:8000/api/login
    requestBody:
      content:
        application/x-www-form-urlencoded:
          schema:
            properties:
              name:
                type: string
                example: Admin
              email:
                type: string
                example: admin@gmail.com
              password:
                type: string
                example: Admin111$
              password_confirmation:
```

```
        type: string
        example: Admin111$
security:
  - bearerAuth: []
parameters:
  - name: Accept
    in: header
    schema:
      type: string
      example: application/json
responses:
  '201':
    description: Created
    headers:
      Host:
        schema:
          type: string
          example: localhost:8000
      Date:
        schema:
          type: string
          example: Fri, 08 Apr 2022 19:47:16 GMT
      Connection:
        schema:
          type: string
          example: close
      X-Powered-By:
        schema:
          type: number
          example: PHP/8.0.12
      Cache-Control:
        schema:
          type: string
          example: no-cache, private
      Content-Type:
        schema:
          type: string
          example: application/json
      X-RateLimit-Limit:
```



```
    schema:
      type: integer
      example: '60'
  X-RateLimit-Remaining:
    schema:
      type: integer
      example: '59'
  Access-Control-Allow-Origin:
    schema:
      type: string
      example: '*'
  content:
    application/json:
      schema:
        type: object
      example:
        user:
          id: 1
          name: admin
          email: admin@gmail.com
          role: Admin
          email_verified_at: '2022-03-
20T19:07:07.000000Z'
          current_team_id: null
          profile_photo_path: null
          created_at: '2022-03-20T19:06:58.000000Z'
          updated_at: '2022-03-20T19:07:07.000000Z'
          address: Rakoczi utca 85
          phone_number: '+36201111111'
          profile_photo_url: >-
            https://ui-
avatars.com/api/?name=a&color=7F9CF5&background=EBF4FF
        token:
15|3XKkWLZA90yQ5sPjViOE6fUYKKNb6P4NnxPYMnCP
'401':
  description: Unauthorized
  headers:
    Host:
      schema:
```

```
    type: string
    example: localhost:8000
Date:
  schema:
    type: string
    example: Fri, 08 Apr 2022 19:48:13 GMT
Connection:
  schema:
    type: string
    example: close
X-Powered-By:
  schema:
    type: number
    example: PHP/8.0.12
Cache-Control:
  schema:
    type: string
    example: no-cache, private
Content-Type:
  schema:
    type: string
    example: application/json
X-RateLimit-Limit:
  schema:
    type: integer
    example: '60'
X-RateLimit-Remaining:
  schema:
    type: integer
    example: '60'
Access-Control-Allow-Origin:
  schema:
    type: string
    example: '*'
content:
  application/json:
    schema:
      type: object
    example:
```

```
message: Wrong credentials entered
/api/products/4:
  get:
    tags:
      - default
    summary: http://localhost:8000/api/products/25
    security:
      - bearerAuth: []
    parameters:
      - name: Accept
        in: header
        schema:
          type: string
          example: application/json
    responses:
      '200':
        description: OK
        headers:
          Host:
            schema:
              type: string
              example: localhost:8000
          Date:
            schema:
              type: string
              example: Fri, 08 Apr 2022 20:02:54 GMT
          Connection:
            schema:
              type: string
              example: close
          X-Powered-By:
            schema:
              type: number
              example: PHP/8.0.12
          Cache-Control:
            schema:
              type: string
              example: no-cache, private
          Content-Type:
```

```
      schema:
        type: string
        example: application/json
X-RateLimit-Limit:
  schema:
    type: integer
    example: '60'
X-RateLimit-Remaining:
  schema:
    type: integer
    example: '58'
Access-Control-Allow-Origin:
  schema:
    type: string
    example: '*'
content:
  application/json:
    schema:
      type: object
    examples:
      example-0:
        summary:
http://localhost:8000/api/products/25
        value:
          id: 25
          name: Goulash soup
          category: Food
          description: Pork chops, potato,
vegetable, thick soup
          price: '2999.00'
          created_at: '2022-03-27T15:13:49.000000Z'
          updated_at: '2022-03-27T15:13:49.000000Z'
      example-1:
        summary:
http://localhost:8000/api/products/4
        value:
          id: 4
          name: Beef burger
          category: Food
```

```
burger          description: 330g meat, salad, mayo,
                price: '3999.99'
                created_at: null
                updated_at: '2022-03-27T14:46:37.000000Z'
/api/products/30:
  put:
    tags:
      - default
    summary: http://localhost:8000/api/products/30
    requestBody:
      content:
        application/x-www-form-urlencoded:
          schema:
            properties:
              name:
                type: string
                example: Honey
              category:
                type: string
                example: Food
              description:
                type: string
                example: small box of honey
              price:
                type: integer
                example: '499'
    security:
      - bearerAuth: []
    parameters:
      - name: Accept
        in: header
        schema:
          type: string
          example: application/json
    responses:
      '200':
        description: OK
        headers:
```

```
Host:
  schema:
    type: string
    example: localhost:8000
Date:
  schema:
    type: string
    example: Fri, 08 Apr 2022 20:06:41 GMT
Connection:
  schema:
    type: string
    example: close
X-Powered-By:
  schema:
    type: number
    example: PHP/8.0.12
Cache-Control:
  schema:
    type: string
    example: no-cache, private
Content-Type:
  schema:
    type: string
    example: application/json
X-RateLimit-Limit:
  schema:
    type: integer
    example: '60'
X-RateLimit-Remaining:
  schema:
    type: integer
    example: '59'
Access-Control-Allow-Origin:
  schema:
    type: string
    example: '*'
content:
  application/json:
    schema:
```

```
      type: object
    example:
      id: 30
      name: Honey
      category: Food
      description: small box of honey
      price: '899'
      created_at: '2022-04-08T19:29:14.000000Z'
      updated_at: '2022-04-08T20:06:41.000000Z'
  '401':
    description: Unauthorized
    headers:
      Host:
        schema:
          type: string
          example: localhost:8000
      Date:
        schema:
          type: string
          example: Fri, 08 Apr 2022 20:06:13 GMT
      Connection:
        schema:
          type: string
          example: close
      X-Powered-By:
        schema:
          type: number
          example: PHP/8.0.12
      Cache-Control:
        schema:
          type: string
          example: no-cache, private
      Content-Type:
        schema:
          type: string
          example: application/json
      Access-Control-Allow-Origin:
        schema:
          type: string
```

```
        example: '*'
content:
  application/json:
    schema:
      type: object
    example:
      message: Unauthenticated.
'403':
  description: Forbidden
  headers:
    Host:
      schema:
        type: string
        example: localhost:8000
    Date:
      schema:
        type: string
        example: Sat, 09 Apr 2022 18:39:54 GMT
    Connection:
      schema:
        type: string
        example: close
    X-Powered-By:
      schema:
        type: number
        example: PHP/8.0.12
    Cache-Control:
      schema:
        type: string
        example: no-cache, private
    Content-Type:
      schema:
        type: string
        example: application/json
    X-RateLimit-Limit:
      schema:
        type: integer
        example: '60'
    X-RateLimit-Remaining:
```



```
        schema:
          type: integer
          example: '58'
        Access-Control-Allow-Origin:
          schema:
            type: string
            example: '*'
      content:
        application/json:
          schema:
            type: object
          example:
            message: Forbidden
delete:
  tags:
    - default
  summary: http://localhost:8000/api/products/4
  security:
    - bearerAuth: []
  parameters:
    - name: Accept
      in: header
      schema:
        type: string
        example: application/json
  responses:
    '200':
      description: OK
      headers:
        Host:
          schema:
            type: string
            example: localhost:8000
        Date:
          schema:
            type: string
            example: Fri, 08 Apr 2022 19:49:14 GMT
      Connection:
        schema:
```

```
        type: string
        example: close
X-Powered-By:
  schema:
    type: number
    example: PHP/8.0.12
Content-Type:
  schema:
    type: string
    example: text/html; charset=UTF-8
Cache-Control:
  schema:
    type: string
    example: no-cache, private
X-RateLimit-Limit:
  schema:
    type: integer
    example: '60'
X-RateLimit-Remaining:
  schema:
    type: integer
    example: '59'
Access-Control-Allow-Origin:
  schema:
    type: string
    example: '*'
content:
  text/plain:
    schema:
      type: string
      example: '1'
'401':
  description: Unauthorized
  headers:
    Host:
      schema:
        type: string
        example: localhost:8000
  Date:
```

```
    schema:
      type: string
      example: Fri, 08 Apr 2022 19:50:23 GMT
  Connection:
    schema:
      type: string
      example: close
  X-Powered-By:
    schema:
      type: number
      example: PHP/8.0.12
  Cache-Control:
    schema:
      type: string
      example: no-cache, private
  Content-Type:
    schema:
      type: string
      example: application/json
  Access-Control-Allow-Origin:
    schema:
      type: string
      example: '*'
  content:
    application/json:
      schema:
        type: object
      example:
        message: Unauthenticated.
'403':
  description: Forbidden
  headers:
    Host:
      schema:
        type: string
        example: localhost:8000
    Date:
      schema:
        type: string
```

```
        example: Sat, 09 Apr 2022 18:39:31 GMT
Connection:
  schema:
    type: string
    example: close
X-Powered-By:
  schema:
    type: number
    example: PHP/8.0.12
Cache-Control:
  schema:
    type: string
    example: no-cache, private
Content-Type:
  schema:
    type: string
    example: application/json
X-RateLimit-Limit:
  schema:
    type: integer
    example: '60'
X-RateLimit-Remaining:
  schema:
    type: integer
    example: '59'
Access-Control-Allow-Origin:
  schema:
    type: string
    example: '*'
content:
  application/json:
    schema:
      type: object
    example:
      message: Forbidden
/api/products/search/long:
  get:
    tags:
      - default
```

```
summary:
http://localhost:8000/api/products/search/chicken
security:
  - noauthAuth: []
parameters:
  - name: Accept
    in: header
    schema:
      type: string
      example: application/json
responses:
  '200':
    description: OK
    headers:
      Host:
        schema:
          type: string
          example: localhost:8000
      Date:
        schema:
          type: string
          example: Fri, 08 Apr 2022 19:32:46 GMT
      Connection:
        schema:
          type: string
          example: close
      X-Powered-By:
        schema:
          type: number
          example: PHP/8.0.12
      Cache-Control:
        schema:
          type: string
          example: no-cache, private
      Content-Type:
        schema:
          type: string
          example: application/json
      X-RateLimit-Limit:
```

```

    schema:
      type: integer
      example: '60'
  X-RateLimit-Remaining:
    schema:
      type: integer
      example: '57'
  Access-Control-Allow-Origin:
    schema:
      type: string
      example: '*'
content:
  application/json:
    schema:
      type: object
    examples:
      example-0:
        summary:
http://localhost:8000/api/products/search/chicken
        value:
          - id: 3
            name: Chicken steak
            category: Food
            description: Countryside chicken
            price: '3000.00'
            created_at: null
            updated_at: null
          - id: 22
            name: Chicken wings
            category: Food
            description: Spicy grilled wings
            price: '2350.00'
            created_at: '2022-03-
27T14:53:50.000000Z'
            updated_at: '2022-03-
27T14:53:50.000000Z'
          - id: 29
            name: Grilled chicken
            category: Food

```

```
description: Spicy chicken grilled
price: '2850.00'
created_at: '2022-04-
08T19:07:48.000000Z'
updated_at: '2022-04-
08T19:07:48.000000Z'
example-1:
  summary:
http://localhost:8000/api/products/search/chicken
  value:
    - id: 5
      name: Long Island
      category: Cocktail
      description: Rum, Tequila, Vodka, Coke
      price: '3999.00'
      created_at: '2022-03-
21T02:03:35.000000Z'
      updated_at: '2022-03-
23T08:02:03.000000Z'
  /api/register:
    post:
      tags:
        - default
      summary: http://localhost:8000/api/register
      requestBody:
        content:
          application/x-www-form-urlencoded:
            schema:
              properties:
                name:
                  type: string
                  example: tester
                email:
                  type: string
                  example: user@gmail.com
                password:
                  type: string
                  example: Testing111
                password_confirmation:
```

```
        type: string
        example: Testing111
parameters:
  - name: Accept
    in: header
    schema:
      type: string
      example: application/json
responses:
  '201':
    description: Created
    headers:
      Host:
        schema:
          type: string
          example: localhost:8000
      Date:
        schema:
          type: string
          example: Fri, 08 Apr 2022 20:01:06 GMT
      Connection:
        schema:
          type: string
          example: close
      X-Powered-By:
        schema:
          type: number
          example: PHP/8.0.12
      Cache-Control:
        schema:
          type: string
          example: no-cache, private
      Content-Type:
        schema:
          type: string
          example: application/json
      X-RateLimit-Limit:
        schema:
          type: integer
```



```
        example: '60'
X-RateLimit-Remaining:
  schema:
    type: integer
    example: '59'
Access-Control-Allow-Origin:
  schema:
    type: string
    example: '*'
content:
  application/json:
    schema:
      type: object
    example:
      user:
        name: tester
        email: tester1234@gmail.com
        updated_at: '2022-04-08T20:01:04.000000Z'
        created_at: '2022-04-08T20:01:04.000000Z'
        id: 7
        profile_photo_url: >-
          https://ui-
avatars.com/api/?name=t&color=7F9CF5&background=EBF4FF
      token:
16|13cQirhIWpgDpe7vi6ngDMAxnIsjXM0bGwmTtWct
'422':
  description: Unprocessable Content
  headers:
    Host:
      schema:
        type: string
        example: localhost:8000
    Date:
      schema:
        type: string
        example: Fri, 08 Apr 2022 19:50:43 GMT
  Connection:
    schema:
      type: string
```

```
        example: close
X-Powered-By:
  schema:
    type: number
    example: PHP/8.0.12
Cache-Control:
  schema:
    type: string
    example: no-cache, private
Content-Type:
  schema:
    type: string
    example: application/json
X-RateLimit-Limit:
  schema:
    type: integer
    example: '60'
X-RateLimit-Remaining:
  schema:
    type: integer
    example: '59'
Access-Control-Allow-Origin:
  schema:
    type: string
    example: '*'
content:
  application/json:
    schema:
      type: object
    examples:
      example-0:
        summary: http://localhost:8000/api/register
        value:
          message: The given data was invalid.
          errors:
            email:
              - The email has already been taken.
          password:
```

- The password confirmation does not match.

example-1:

summary: http://localhost:8000/api/register

value:

message: The given data was invalid.

errors:

email:

- The email has already been taken.