

# Identity Hungary Informatikai Korlátolt Felelősségű Társaság

Profilunk a **PHP fejlesztés** és web alapú termékek előállítására.



6722 Szeged, Attila utca 11. 1. em. 1.  
<http://identity-hungary.hu>  
+36-62 452-239

# Garnet API backend

## 1. Headers

### Szükséges header-ek:

- Accept : application/json
- X-Requested-With : XMLHttpRequest
- Content-Type : application/json

Amennyiben az adott endpoint bejelentkezéshez kötött, abban az esetben token küldése szükséges a következő formátumban:

- Authentication : tokenType token [ Bearer {token here} ]

## 2. Belépés, regisztráció:

A bejelentkezés email / jelszó páros megadásával történik. Sikeres bejelentkezés után a garNet backend rendszer egy, a bejelentkezett felhasználó adatait tartalmazó objektummal tér vissza frontend rendszer részére. Ez az objektum tartalmaz egy úgynevezett AccessToken-t, amellyel a token lejáratáig a felhasználó azonosítható és a különböző védett kérésekhez a hozzáférése szabályozható. Az autentikációt a Passport nevű open source csomag segítségével valósítjuk meg, amely a Google OAuth Personal Access token-re épül.

### 2.1: Belépés:

#### Endpoint:

Method	Endpoint	Headers
POST	api/user/login	Alapértelmezett (token nélkül)

#### Body parameters:

Param	Type	Status	Description
email	string	required	Email
password	string	required	Jelszó
remember_me	boolean	optional	Emlékeztessen később



```

    "pivot": {
      "role_id": 3,
      "permission_id": 3
    }
  ]
}

```

## 2.2: Regisztráció:

A regisztráció folyamatának első lépéseként a regisztrációhoz szükséges adatok megadása szükséges. Az adatok megadása után a garNet backend rendszer a regisztráció során megadott postafiók címre kiküld egy emailt, amiben a regisztráció véglegesítéséhez szükséges link található. A linkben szerepel egy 60 karakter hosszúságú speciális hash, ami a regisztrált felhasználóhoz kötődik és egyértelműen azonosítja őt. Az adatok megadása után a frontend a szükséges endpoint felé továbbítja az adatokat.

### Endpoint:

Method	Endpoint	Headers
POST	api/user/signup	Alapértelmezett

### Body parameters:

Param	Type	Status	Description
email	string	required	Email
password	string	required	Jelszó
password_confirmation	string	required	Jelszó ismét
role_id	integer	required	Jogosultsági csoport

### Handled responses:

Status	StatusCode
Success	200
Internal Server Error	500

**Regisztráció megerősítése:**

A linkre kattintva a felhasználó véglegesítheti a regisztrációt (amennyiben a biztonsági feltételeknek megfelel). Amíg a regisztráció megerősítése nem történik meg, addig a felhasználó nem tud bejelentkezni a garNet rendszerbe. Az adatok megadása után a frontend a szükséges endpoint felé továbbítja az adatokat.

**Regisztráció véglegesítésekor elvégzett biztonsági ellenőrzések:**

- **a linkben szereplő aktivációs kód szerepel az adatbázisban**
- **a linkben szereplő aktivációs kód valós aktivációs kód**
- **a linkben szereplő aktivációs kód érvényes**

A garNet backend rendszerben a felhasználó egy adott típusú felhasználóként kerül rögzítésre. A regisztráció alapadatai egy közös adatbázis táblába (users tábla), a különböző típusú felhasználókhöz tartozó speciális adatok külön – külön adatbázis táblákban kerülnek rögzítésre.

**Endpoint:**

Method	Endpoint	Headers
GET	api/user/signup/activate/{token}	Alapértelmezett

**Query parameters:**

Param	Type	Status	Description
token	string	required	Token

**Handled responses:**

Status	StatusCode
Success	200
Internal Server Error	500

**Sample success response:**

**Status: success StatusCode: 200 Response: JSON**

```
{
  "id": 16,
  "email": "john.doe@example.com",
  "email_verified_at": {
    "date": "2019-01-22 09:23:47.007925",
    "timezone_type": 3,
    "timezone": "UTC"
  },
  "active": true,
  "default_language": "hu",
  "created_at": "2019-01-22 08:17:39",
  "updated_at": "2019-01-22 09:23:47",
  "deleted_at": null
}
```

### 2.3: Kilépés:

Kijelentkezés esetén a garNet backend rendszer a bejelentkezett felhasználó AccessToken-ét törli a rendszerből, ezzel biztosítva a sikeres kijelentkezést. Az adatok megadása után a frontend a szükséges endpoint felé továbbítja az adatokat.

#### Endpoint:

Method	Endpoint	Headers
POST	api/user/logout	Alapértelmezett

#### Body parameters:

Param	Type	Status	Description
---	---	---	---

#### Handled responses:

Status	StatusCode
Success	200
Internal Server Error	500

#### Sample responses:

**Status: success StatusCode: 200 Response: JSON**

```
{  
  "message": "Successfully logged out"  
}
```

## 3. Számla:

A folyamat úgy épül fel, hogy a bejelentkezett felhasználó első lépésben rögzíti a rendszerben a számla alapadatait, majd ehhez a számlához rögzíti a számlán található termékek adatait.

**Számlával kapcsolatos endpoint-ok:****3.1: A bejelentkezett felhasználó által rögzített számlák listázása:**

Endpoint:

Method	Endpoint	Headers
GET	api/public/customers/{user_id}/invoices	Alapértelmezett

Handled responses:

Status	StatusCode
Success	200
Unauthorized	401
Internal Server Error	500

**Response:****Status: success StatusCode: 200 Response: JSON**

```
{
  "success": [
    {
      "id": 1,
      "invoice_no": "ABC123/2019",
      "user_id": 1,
      "seller_id": 1,
      "purchased": "2019-05-09",
      "ap_no": "AP12345678",
      "created_at": "2019-05-09 12:09:11",
      "updated_at": "2019-05-09 12:09:11",
      "deleted_at": null
    },
    {
      "id": 2,
      "invoice_no": "ABC12345/2019",
      "user_id": 1,
      "seller_id": 2,
      "purchased": "2019-05-09",
      "ap_no": "AP12345678",
      "created_at": "2019-05-09 12:09:11",
      "updated_at": "2019-05-09 12:09:11",
      "deleted_at": null
    }
  ]
}
```

**3.2: A bejelentkezett felhasználó által rögzített adott számla megjelenítése ID alapján:**

Endpoint:

Method	Endpoint	Headers
GET	api/public/customers/{user_id}/invoices/{invoice_id}	Alapértelmezett

Handled responses:

Status	StatusCode
Success	200
Unauthorized	401
Not found	404
Internal Server Error	500

**Response:****Status: success StatusCode: 200 Response: JSON**

```
{
  "success": [
    {
      "id": 1,
      "invoice_no": "ABC123/2019",
      "user_id": 1,
      "seller_id": 1,
      "purchased": "2019-05-09",
      "ap_no": "AP12345678",
      "created_at": "2019-05-09 12:09:11",
      "updated_at": "2019-05-09 12:09:11",
      "deleted_at": null
    }
  ]
}
```



**3.3: Számla rögzítése:**

Endpoint:

Method	Endpoint	Headers
POST	api/public/customer/{user_id}/invoices	Alapértelmezett

Body parameters:

Param	Type	Status	Description
seller_id	string	required integer	Eladó azonosító
invoice_no	string	required	Számla sorszám
user_id	string	required integer	Felhasználó azonosító
purchased	string	required	Vásárlás dátuma
ap_no	string	sometimes	Pénztárgép azonosító
files	string	required	Csatolt file-ok

Handled responses:

Status	StatusCode
Success	200
Unauthorized	401
Internal Server Error	500

**Response:****Status: success StatusCode: 200 Response: JSON**

```
{
  "success": true
}
```

**3.4: Számla módosítása:**

Endpoint:

Method	Endpoint	Headers
PUT	api/public/customer/{user_id}/invoices/{invoice_id}	Alapértelmezett

Body parameters:

Param	Type	Status	Description
seller_id	string	required integer	Eladó azonosító
invoice_id	string	required integer	Számla azonosító
invoice_no	string	required	Számla sorszám
user_id	string	required integer	Felhasználó azonosító
purchased	string	required	Vásárlás dátuma
ap_no	string	sometimes	Pénztárgép azonosító

Handled responses:

Status	StatusCode
Success	200
Unauthorized	401
Not found	404
Security error	422
Internal Server Error	500

**Response:****Status: success StatusCode: 200 Response: JSON**

```
{
  "success": true
}
```

**3.5: Számla törlése:**

Meglévő számla típusú adat státuszának módosítása. A státusz aktív vagy inaktív lehet, inaktív státusz esetén a számla alapértelmezetten nem kerül listázásra a garNet rendszerbe.

Endpoint:

Method	Endpoint	Headers
PUT	api/admin/customers/{id}/status	Alapértelmezett

Body parameters:

Param	Type	Status	Description
invoice_id	integer	required	Számla azonosító
user_id	integer	required	Felhasználó azonosító

Handled responses:

Status	StatusCode
Success	200
Unauthorized	401
Not found	404
Security error	422
Internal Server Error	500

**Response:**

**Status: success StatusCode: 200 Response: JSON**

```
{
  "success": true
}
```

## 4. Termékek:

### Termékekkel kapcsolatos endpoint-ok:

#### 4.1: A bejelentkezett felhasználó által rögzített számlákhoz rendelt termékek listázása.

Endpoint:

Method	Endpoint	Headers
GET	api/public/customer/{user_id}/invoices/{invoice_id}/items	Alapértelmezett

Handled responses:

Status	StatusCode
Success	200
Unauthorized	401
Internal Server Error	500

### Response:

**Status: success StatusCode: 200 Response: JSON**

```
{
  "success": [
    {
      "id": 1,
      "invoice_id": 1,
      "name": "Sample item1",
      "product_category_id": 1,
      "product_subcategory_id": 2,
      "manufacture": "11",
      "type_no": "11",
      "barcode": "111",
      "price": 10000,
      "warranty_in_month": 12,
      "warranty": 0,
      "created_at": "2019-05-14 10:38:44",
      "updated_at": "2019-05-14 10:38:44",
      "deleted_at": null
    },
    {
      "id": 2,
      "invoice_id": 1,
      "name": "Sample item2",
      "product_category_id": 1,
      "product_subcategory_id": 2,
      "manufacture": "11",

```

```

    "type_no": "11",
    "barcode": "111",
    "price": 20000,
    "warranty_in_month": 24,
    "warranty": 0,
    "created_at": "2019-05-14 10:38:44",
    "updated_at": "2019-05-14 10:38:44",
    "deleted_at": null
  }
]
}

```

#### 4.2: A bejelentkezett felhasználó által rögzített adott számlához rendelt adott termék megjelenítése ID alapján

Endpoint:

Method	Endpoint	Headers
GET	api/public/customer/{user_id}/invoices/{invoice_id}/items/{item_id}	Alapértelmezett

Handled responses:

Status	StatusCode
Success	200
Unauthorized	401
Not found	404
Internal Server Error	500

#### Response:

**Status: success StatusCode: 200 Response: JSON**

```

{
  "success": [
    {
      "id": 1,
      "invoice_id": 1,
      "name": "Sample item1",
      "product_category_id": 1,
      "product_subcategory_id": 2,
      "manufacture": "11",
      "type_no": "11",
      "barcode": "111",
      "price": 10000,
    }
  ]
}

```

```

    "warranty_in_month": 12,
    "warranty": 0,
    "created_at": "2019-05-14 10:38:44",
    "updated_at": "2019-05-14 10:38:44",
    "deleted_at": null
  }
]
}

```

#### 4.3: A bejelentkezett felhasználóhoz rendelt összes termék listázása user azonosító alapján:

Endpoint:

Method	Endpoint	Headers
GET	api/public/customer/{user_id}/items/	Alapértelmezett

Handled responses:

Status	StatusCode
Success	200
Unauthorized	401
Not found	404
Internal Server Error	500

#### Response:

**Status: success StatusCode: 200 Response: JSON**

```

{
  "success": [
    {
      "id": 1,
      "invoice_id": 1,
      "name": "Sample item1",
      "product_category_id": 1,
      "product_subcategory_id": 2,
      "manufacture": "11",
      "type_no": "11",
      "barcode": "23456789",
      "price": 10000,
      "warranty_in_month": 12,
      "warranty": 0,
      "created_at": "2019-05-14 10:38:44",
      "updated_at": "2019-05-14 10:38:44",
      "deleted_at": null
    }
  ]
}

```

```

    },
    {
      "id": 2,
      "invoice_id": 2,
      "name": "Sample item2",
      "product_category_id": 1,
      "product_subcategory_id": 2,
      "manufacture": "11",
      "type_no": "11",
      "barcode": "23456789",
      "price": 10000,
      "warranty_in_month": 12,
      "warranty": 0,
      "created_at": "2019-05-14 10:38:44",
      "updated_at": "2019-05-14 10:38:44",
      "deleted_at": null
    }
  ]
}

```

#### 4.4: A bejelentkezett felhasználó által rögzített adott számlához tartozó lejárt termékek listázása:

Endpoint:

Method	Endpoint	Headers
GET	api/public/customer/{user_id}/expiredItems	Alapértelmezett

#### 4.5: A bejelentkezett felhasználó által rögzített adott számlához történő termék rögzítése:

Endpoint:

Method	Endpoint	Headers
POST	api/public/customer/{user_id}/invoices/{invoice_id}/items	Alapértelmezett

Body parameters:

Param	Type	Status	Description
invoice_id	integer	required	Számla azonosító
name	string	required	Termék megnevezés
product_category_id	integer	required	Termék kategória azonosító
product_subcategory_id	integer	required	Termék alkategória azonosító
manufacture	string	sometimes	Gyártó megnevezés
type_no	string	required	Típusszám
barcode	string	sometimes	Vonalkód
price	integer	required	Ár
warranty_in_month	integer	required	Garancia hónapokban
warranty	boolean	required	Feltöltött dokumentum csatolva
has_extra_form_item	boolean	required	Extra form elem

Handled responses:

Status	StatusCode
Success	200
Unauthorized	401
Internal Server Error	500

**Response:**

**Status: success StatusCode: 200 Response: JSON**

```
{
  "success": true
}
```



#### 4.6: A bejelentkezett felhasználó által rögzített adott számlához történő termék adatainak módosítása:

Endpoint:

Method	Endpoint	Headers
PUT	customer/{user_id}/invoices/{invoice_id}/items/{item_id}	Alapértelmezett

Body parameters:

Param	Type	Status	Description
invoice_id	integer	required	Számla azonosító
name	string	required	Termék megnevezés
product_category_id	integer	required	Termék kategória azonosító
product_subcategory_id	integer	required	Termék alkategória azonosító
manufacture	string	sometimes	Gyártó megnevezés
type_no	string	required	Típuszám
barcode	string	sometimes	Vonalkód
price	integer	required	Ár
warranty_in_month	integer	required	Garancia hónapokban
warranty	boolean	required	Feltöltött dokumentum csatolva
has_extra_form_item	boolean	required	Extra form elem

Handled responses:

Status	StatusCode
Success	200
Unauthorized	401
Not found	404
Internal Server Error	500

**Response:****Status: success StatusCode: 200 Response: JSON**

```
{
  "success": true
}
```

**4.7: Termék törlése:**

A bejelentkezett felhasználó által rögzített adott számlához történő termék státuszának módosítása. A státusz aktív vagy inaktív lehet. Inaktív státusz esetén a számlához tartozó termék alapértelmezetten nem kerül listázásra a garNet rendszerbe.

Endpoint:

Method	Endpoint	Headers
PUT	api/public/customer/{user_id}/invoices/{invoice_id}/items/{item_id}	Alapértelmezett

Body parameters:

Param	Type	Status	Description
user_id	integer	required	Felhasználó azonosító
invoice_id	integer	required	Számla azonosító
item_id	integer	required	Termék azonosító

Handled responses:

Status	StatusCode
Success	200
Unauthorized	401
Not found	404
Security error	422
Internal Server Error	500

**Response:****Status: success StatusCode: 200 Response: JSON**

```
{
  "success": true
}
```

## 5. Termék kategóriák:

### 5.1: Az összes productCategory listázása **(csak adminisztrátori jogosultsággal lehetséges!)**:

Az összes productCategory listázása adminisztráció céljából. Az alapértelmezett listázás paginate nélkül kerül megvalósításra. A withPaginate query modifier használatával a garNet backend rendszer az eredményt alapértelmezetten olyan formátumban adja vissza, ami tartalmaz minden szükséges adatot egy paginator létrehozásához. A withTrashed query modifier használatával a garNet backend rendszer az eredményt alapértelmezetten a törölt adatokkal együtt adja vissza. A withRelation query modifier használatával a garNet backend rendszer az eredményt alapértelmezetten az adott modellhez kapcsolódó relációkkal együtt adja vissza.

Endpoint:

Method	Endpoint	Headers
GET	api/admin/products	Alapértelmezett
GET	api/admin/products? withTrashed=true&withPaginate=true&withRelation=true	With query modifiers

Query parameters:

Param	Type	Default	Status	Description
\$withTrashed	boolean	false	optional	query modifier (add softDeleted rows to result)
\$withPaginate	boolean	false	optional	query modifier (return with paginated result)
\$withRelation	boolean	false	optional	query modifier (add relation(s) to result)

Handled responses:

Status	StatusCode
Success	200
Unauthorized	401
Internal Server Error	500

**Response:****Status: success StatusCode: 200 Response: JSON**

```

{
  "success": [
    {
      "id": 1,
      "name": "Sample product1",
      "parent_id": null,
      "created_at": "2019-04-02 12:08:44",
      "updated_at": "2019-04-02 12:08:44",
      "deleted_at": null
    },
    {
      "id": 2,
      "name": "Sample product2",
      "parent_id": 1,
      "created_at": "2019-04-02 12:08:44",
      "updated_at": "2019-04-02 12:08:44",
      "deleted_at": null
    },
    {
      "id": 3,
      "name": "Sample product3",
      "parent_id": 1,
      "created_at": "2019-04-02 12:08:44",
      "updated_at": "2019-04-02 12:08:44",
      "deleted_at": null
    }
  ]
}

```

**5.2: Adott (aktív és törölt állapotban lévő ) productCategory típusú adat listázása ID alapján (csak adminisztrátori jogosultsággal lehetséges!):**

Adott (aktív és törölt állapotban lévő ) productCategory típusú adat listázása ID alapján. A garNet backend rendszer az eredményt alapértelmezetten olyan formátumban adja vissza, ami tartalmaz minden szükséges adatot. Az alapértelmezett listázás paginate nélkül kerül megvalósításra. A withPaginate query modifier használatával a garNet backend rendszer az eredményt alapértelmezetten olyan formátumban adja vissza, ami tartalmaz minden szükséges adatot egy paginator létrehozásához. A withTrashed query modifier használatával a garNet backend rendszer az eredményt alapértelmezetten a törölt adatokkal együtt adja vissza. A withRelation query modifier használatával a garNet backend rendszer az eredményt alapértelmezetten az adott modellhez kapcsolódó relációkkal együtt adja vissza.

Endpoint:

Method	Endpoint	Headers
GET	api/admin/products/{id}	Alapértelmezett
GET	api/admin/products/{id}?withTrashed=true&withPaginate=true&withRelation=true	With query modifiers

Query parameters:

Param	Type	Default	Status	Description
id	integer	---	required	ID
\$withTrashed	boolean	false	optional	query modifier (add softDeleted rows to result)
\$withPaginate	boolean	false	optional	query modifier (return with paginated result)
\$withRelation	boolean	false	optional	query modifier (add relation(s) to result)

Handled responses:

Status	StatusCode
Success	200
Unauthorized	401
Data not found	404
Internal Server Error	500

### Response:

**Status: success StatusCode: 200 Response: JSON**

```
{
  "success": [
    {
      "id": 1,
      "name": "Sample product1",
      "parent_id": null,
      "created_at": "2019-05-08 09:50:13",
      "updated_at": "2019-05-08 09:50:13",
      "deleted_at": null
    }
  ]
}
```

### 5.3: Termék kategória rögzítése (csak adminisztrátori jogosultsággal lehetséges!)

Endpoint:

Method	Endpoint	Headers
POST	api/admin/products	Alapértelmezett

Body parameters:

Param	Type	Status	Description
name	string	required	Category name
parent_id	integer	required	Parent Id

Handled responses:

Status	StatusCode
Success	200
Unauthorized	401
Internal Server Error	500

### Response:

**Status: success StatusCode: 200 Response: JSON**

```
{
  "success": true
}
```

### 5.4: Termék kategória módosítása **(csak adminisztrátori jogosultsággal lehetséges!)**

Endpoint:

Method	Endpoint	Headers
PUT	api/admin/products/{id}	Alapértelmezett

Body parameters:

Param	Type	Status	Description
id	integer	required integer	Item Id
name	string	required	Category name
parent_id	integer	required integer	Parent Id

Handled responses:

Status	StatusCode
Success	200
Unauthorized	401
Not found	404
Security error	422
Internal Server Error	500

### Response:

**Status: success StatusCode: 200 Response: JSON**

```
{
  "success": true
}
```

### 5.5: Termék kategória törlése **(csak adminisztrátori jogosultsággal lehetséges!)**

Meglévő ProductCategory típusú adat státuszának módosítása. A státusz aktív vagy inaktív lehet. inaktív státusz esetén nem kerül alapértelmezetten listázva.

Endpoint:

Method	Endpoint	Headers
PUT	api/admin/products/{id}/status	Alapértelmezett

Body parameters:

Param	Type	Status	Description
id	integer	required	Azonosító
active	boolean	required	Állapot

Handled responses:

Status	StatusCode
Success	200
Unauthorized	401
Not found	404
Security error	422
Internal Server Error	500

**Response:**

**Status: success StatusCode: 200 Response: JSON**

```
{  
  "success": true  
}
```