

PMS-v2 API Documentation

/api/users

GET /get-all :

- Protection: none
- Request: no data required
- Response: returns all registered users

GET /logout :

- Protection: none
- Request: no data required
- Response: removes cookies: **REFRESH_TOKEN**, **ACCESS_TOKEN**

GET /auth :

- Protection: auth with **JWT**
- Request: no data required
- Response: returns logged **User** data

POST /register-admin :

- Protection: none
- Request:
 - required: **name(max:50)**,
surname(max:50),
password(min:8)
- Response: create admin account if not exist

POST /login-user :

- Protection: none
- Request:
 - required: **idUser**, **password**
- Response: returns user payload and creates **REFRESH_TOKEN** cookie

POST /insert :

- Protection: auth with **JWT**, specific role auth: **ADMIN**
- Request:
 - required: **name(max:50)**,
surname(max:50),
password(min:8),
role(["ADMIN", "SEKRETARIAT", "KSIEGOWOSC", "TEREN"])
- Response: creates new user

PUT /update :

- Protection: auth with **JWT**, specific role auth: **ADMIN**
- Request:
 - required: **idUser**,
name(max:50),
surname(max:50),
role(["ADMIN", "SEKRETARIAT", "KSIEGOWOSC", "TEREN"])
- Response: updates user

PUT /update-password :

- Protection: auth with **JWT**, specific role auth: **ADMIN**
- Request:
 - Required: **idUser**, **password(min:8)**
- Response: updates user password

DELETE /delete:

- Protection: auth with **JWT**, specific role auth: **ADMIN**

- Request:
 - Required: **idUser**
- Response: deletes user

/api/mpzp

GET /get-all :

- Protection: auth with **JWT**
- Request: no data required
- Reponse: returns all available **MPZP**

POST /insert :

- Protection: auth with **JWT**, specific role auth: **ADMIN**
- Request:
 - Required: **code(max:5)**, **description(max:70)**
- Response: creates new **MPZP**

POST /insert-file:

- Protection: auth with **JWT**, specific role auth: **ADMIN**
- Request: no data required
- Response: creates new **MPZP** from mpzp.json file

PUT /update:

- Protection: auth with **JWT**, specific role auth: **ADMIN**
- Request:
 - Required: **idMpzp**, **code(max:5)**, **description(max:70)**
- Response: updates **MPZP**

DELETE /delete :

- Protection: auth with **JWT**, specific role auth: **ADMIN**
- Request:
 - Required: **idMpzp**
- Response: deletes **MPZP**

/api/general-plans

GET /get-all :

- Protection: auth with **JWT**
- Request: no data required
- Reponse: returns all available **General Plans**

POST /insert :

- Protection: auth with **JWT**, specific role auth: **ADMIN**
- Request:
 - Required: **code(max:5)**, **description(max:70)**
- Response: creates new **General Plan**

POST /insert-file:

- Protection: auth with **JWT**, specific role auth: **ADMIN**
- Request: no data required
- Response: creates new **General Plans** from generalPlans.json file

PUT /update:

- Protection: auth with **JWT**, specific role auth: **ADMIN**
- Request:
 - Required: **idGeneralPlan**, **code(max:5)**, **description(max:70)**
- Response: updates **General Plan**

DELETE /delete :

- Protection: auth with **JWT**, specific role auth: **ADMIN**
- Request:
 - Required: **idGeneralPlan**
- Response: deletes **General Plan**

/api/land-purposes

GET /get-all :

- Protection: auth with **JWT**
- Request: no data required
- Response: returns all available **Land Purposes**

POST /insert :

- Protection: auth with **JWT**, specific role auth: **ADMIN**
- Request:
 - Required: **type(max:50)**
- Response: creates new **Land Purpose**

POST /insert-file:

- Protection: auth with **JWT**, specific role auth: **ADMIN**
- Request: no data required
- Response: creates new **Land Purposes** from landPurposes.json file

PUT /update:

- Protection: auth with **JWT**, specific role auth: **ADMIN**
- Request:
 - Required: **idLandPurpose, type(max:50)**
- Response: updates **Land Purpose**

DELETE /delete :

- Protection: auth with **JWT**, specific role auth: **ADMIN**
- Request:
 - Required: **idLandPurpose**
- Response: deletes **Land Purpose**

/api/land-types

GET /get-all :

- Protection: auth with **JWT**
- Request: no data required
- Response: returns all available **Land Types**

POST /insert :

- Protection: auth with **JWT**, specific role auth: **ADMIN**
- Request:
 - Required: **type(max:50)**
- Response: creates new **Land Type**

POST /insert-file:

- Protection: auth with **JWT**, specific role auth: **ADMIN**
- Request: no data required
- Response: creates new **Land Types** from landTypes.json file

PUT /update:

- Protection: auth with **JWT**, specific role auth: **ADMIN**
- Request:
 - Required: **idLandType, type(max:50)**
- Response: updates **Land Type**

DELETE /delete :

- Protection: auth with **JWT**, specific role auth: **ADMIN**
- Request:
 - Required: **idLandType**
- Response: deletes **Land Type**

/api/owners

GET /get :

- Protection: auth with **JWT**
- Request:
 - Optional: **nameFilter**, **limit(int min:0)**
- Reponse: returns filtered owners and their lands

POST /insert :

- Protection: auth with **JWT**
- Request:
 - Required: **name(max:100)**
 - Optional: **phone(polish phone format)**
- Response: creates new **Owner**

PUT /update:

- Protection: auth with **JWT**
- Request:
 - Required: **idOwner**, **name(max:100)**
 - Optional: **phone(polish phone format)**
- Response: updates **Owner**

DELETE /delete :

- Protection: auth with **JWT**, specific role auth: **ADMIN**
- Request:
 - Required: **idOwner**
- Response: deletes **Owner**

/api/renters

GET /get-all :

- Protection: auth with **JWT**
- Request: no data required
- Reponse: returns all **Renters**

GET /get :

- Protection: auth with **JWT**
- Request:
 - Optional: **monthFilter(int)**, **nameFilter**, **endYearFilter(int)**, **ownerNameFilter**, **showExpired(bool)**, **limit(int, min:0)**
- Reponse: returns filtered renters and their rents

POST /insert :

- Protection: auth with **JWT**
- Request:
 - Required: **name(max:100)**, **phone(polish phone format)**
- Response: creates new **Renter**

PUT /update:

- Protection: auth with **JWT**
- Request:
 - Required: **idRenter**, **name(max:100)**, **phone(polish phone format)**
- Response: updates **Renter**

DELETE /delete :

- Protection: auth with **JWT**, specific role auth: **ADMIN**
- Request:
 - Required: **idRenter**
- Response: deletes **Renter**

/api/locations

GET /get-towns :

- Protection: auth with **JWT**
- Request:
 - Optional: **town, commune, district, province**
- Response: returns filtered **Towns**

GET /get :

- Protection: auth with **JWT**, specific role auth: **SEKRETARIAT, KSIEGOWOSC**
- Request:
 - Optional: **taxDistrict(int), agriculturalTax(„0” or „1”), forestTax(„0” or „1”), commune, district, province, limit(int, min:0)**
- Response: returns filtered **Locations**

PUT /update:

- Protection: auth with **JWT**, specific role auth: **SEKRETARIAT, KSIEGOWOSC**
- Request:
 - Required: **idLocation**
 - Optional: **taxDistrict(int), agriculturalTax(float, between 0 and 9999), forestTax(float, between 0 and 9999)**
- Response: updates **Location**

PUT /update-all:

- Protection: auth with **JWT**, specific role auth: **SEKRETARIAT, KSIEGOWOSC**
- Request:
 - Optional: **agriculturalTax(float, between 0 and 9999), forestTax(float, between 0 and 9999)**
- Response: updates all **Locations**

/api/files

GET /get/:idLand/:filename :

- Protection: auth with **JWT**
- Request:
 - Required: **idLand, filename**
- Response: returns a **file** (connected with specific land)

POST /upload/:idLand :

- Protection: auth with **JWT**
- Request:
 - Required: **idLand**
 - FormData: **files(multiple files)**
- Response: upload **files** connected to specific land

DELETE /delete :

- Protection: auth with **JWT**
- Request:
 - Required: **idLand, filename**
- Response: deletes file

/api/ground-classes

GET /get :

- Protection: auth with **JWT**
- Request: no data required
- Reponse: returns **Ground Classes** data

GET /count :

- Protection: auth with **JWT**
- Request:
 - Required: **groundClass**
- Reponse: returns count of filtered **Ground Classes**

GET /unique :

- Protection: auth with **JWT**
- Request: no data required
- Reponse: returns unique groundClasses by class

POST /insert :

- Protection: auth with **JWT**, specific role auth: **SEKRETARIAT, KSIEGOWOSC**
- Request:
 - Required: **groundClass(max:10), convertersData(array length:4, element:float, between 0 and 9), tax(„rolny” or „lesny” or „brak”), released(boolean)**
- Response: creates new **Ground Class**

PUT /update :

- Protection: auth with **JWT**, specific role auth: **SEKRETARIAT, KSIEGOWOSC**
- Request:
 - Required: **idGroundClass, groundClass(max:10), convertersData(array length:4, element:float, between 0 and 9) tax(„rolny” or „lesny” or „brak”), released(boolean)**
- Response: updates **Ground Class**

DELETE /delete :

- Protection: auth with **JWT**
- Request:
 - Required: **idGroundClass**
- Response: deletes **Ground Class**

/api/areas

POST /insert :

- Protection: auth with **JWT**, specific role auth: **SEKRETARIAT, KSIEGOWOSC**
- Request:
 - Required: **idLand(length:21), areasData(array,element: {idGroundClass(length:21), area(float, between 0 and 9999)})**
- Response: creates new **Land Areas**

PUT /update :

- Protection: auth with **JWT**, specific role auth: **SEKRETARIAT, KSIEGOWOSC**
- Request:
 - Required: **areasData(array, element:{idArea, idGroundClass(length:21), area(float, between 0 and 9999)})**
- Response: updates **Land Areas**

DELETE /delete :

- Protection: auth with **JWT**, specific role auth: **SEKRETARIAT, KSIEGOWOSC**
- Request:
 - Required: **areasIds(array)**
- Response: deletes **Land Area**

/api/rents

POST /insert :

- Protection: auth with **JWT**
- Request:
 - Required: **idRenter(length:21), idLand(length:21), startDate(date), endDate(date), rental(int, max:8 millions), issueRentalFactureDate(date)**
- Response: creates new **Rent**

PUT /update :

- Protection: auth with **JWT**
- Request:
 - Required: **idRent, idRenter(length:21), startDate(date), endDate(date), rental(int, max:8 millions), issueRentalFactureDate(date)**
- Response: updates **Rent**

DELETE /delete :

- Protection: auth with **JWT**, specific role auth: **ADMIN**
- Request:
 - Required: **idRent**
- Response: deletes **Rent**

/api/lands

GET /serial-exist :

- Protection: auth with **JWT**
- Request:
 - Required: **serialNumber**
- Reponse: returns land serial number existing

GET /get-one :

- Protection: auth with **JWT**
- Request:
 - Required: **idLand**
- Reponse: returns one **Land** data

GET /get :

- Protection: auth with **JWT**
- Request:
 - Optional: **serialFilter, purchaseYearFilter(int), lowSellDateFilter(date), highSellDateFilter(date), ownerFilter, purposeFilter, rentFilter(boolean), lowAreaFilter(float), highAreaFilter(float), communeFilter, districtFilter, provinceFilter, townFilter, landNumberFilter, groundClassFilter, limit(int, min:0)**
- Reponse: returns one **Land** data

GET /get-rent :

- Protection: auth with **JWT**
- Request: no data required
- Reponse: returns **Lands** intended for rent

GET /get-insertion-data :

- Protection: auth with **JWT**
- Request: no data required
- Reponse: returns data for land insertion:
owners, landTypes, landPurposes, generalPlans, MPZP

POST /insert :

- Protection: auth with **JWT**
- Request:
 - Required: **landNumber(max:7), area(float,between 0 and 9999), town(max:50), commune(max:50), district(max:50), province(max:50), idOwner(length:21), propertyTax(boolean)**
 - Optional: **serialNumber(max:20), registerNumber(max:15), mortgage(boolean, default:false), idLandType(length:21), description(max:65535), idLandPurpose(length:21), idMpzp(length:21), idGeneralPlan(length:21), waterCompany(boolean, default:false), purchaseDate(date), purchaseActNumber(max:21), seller(max:50), purchasePrice(int), sellDate(date), sellActNumber(max:21), buyer(max:50), sellPrice(int)**
- Response: creates new **Land** and others objects connected with **Land**

PUT /update :

- Protection: auth with **JWT**
- Request:
 - Required: **idLand, landNumber(max:7), area(float, between 0 and 9999), town(max:50), commune(max:50), district(max:50), province(max:50), idOwner(length:21), propertyTax(boolean)**
 - Optional: **serialNumber(max:20), registerNumber(max:15), mortgage(boolean, default:false), idLandType(length:21), description(max:65535), idLandPurpose(length:21), idMpzp(length:21), idGeneralPlan(length:21), waterCompany(boolean, default:false), purchaseDate(date), purchaseActNumber(max:21), seller(max:50), purchasePrice(int), sellDate(date), sellActNumber(max:21), buyer(max:50), sellPrice(int)**
- Response: updates **Land** and others things connected with **Land**

DELETE /delete :

- Protection: auth with **JWT**, specific role auth: **ADMIN**
- Request:
 - Required: **idLand**
- Response: deletes **Land**