

# Příloha REST API DOKUMENTACE

## OBSAH

1	UVOD.....	1
2	Role a oprávnění .....	1
2.1.1	Systém.....	1
2.1.2	Projekt.....	1
2.1.3	Dataset .....	2
3	Kontroléry.....	2
3.1	Auth Controller .....	2
3.2	User controller .....	3
3.3	Project Controller.....	4
3.4	Dataset Controller .....	8
3.5	Image Controller .....	11

## 1 UVOD

Tento dokument je věnován popisu API aplikace "Mikroslužba pro správu multimediální dat pro trénování neuronových sítí", kde podrobně popsány požadavky aplikace, typy dat požadavků a jejich odpovědí.

## 2 Role a oprávnění

Systém umožňuje přiřazovat každému prvku konkrétní roli pro určitého uživatele.

### 2.1.1 Systém

Pro ovládání systému následující role

**Admin** – hlavní admin, který může vše. Spravovat uživatele, projekt apod.

**Researcher** – uživatel, který je schopen založit projekt

**User** – běžný uživatel, který může otevírat projekty, ke kterým má přístup

### 2.1.2 Projekt

Pro projekt následující role

**Owner** – uživatel, který vytvořil projekt. Má právo ho upravovat, mazat projekt. Upravovat oprávnění

**Dataset** – uživatel, který má právo administrovat datasety – vytvářet, měnit a nastavovat práva, kopírovat datasety

### 2.1.3 Dataset

**Owner** – uživatel, který administruje daný dataset. V rámci datasetu vytváří kategorie, názvy objektů, exportuje a importuje datasety, vkládá a maže obrázky. apod.

**Label** – uživatel, který může v rámci projektu u obrázků označovat kategorie a označovat objekty

**Viewer** – pouze nahlíží

## 3 Kontroléry

Pro interakci s aplikací jsou zodpovědné následující kontroléry:

**AuthController** — zajišťuje autorizaci a registraci nových uživatelů;

**UserController** — spravuje údaje uživatelů, jako jsou přihlašovací jméno, heslo a role;

**ProjectController** — spravuje údaje o projektech uživatelů;

**DatasetController** — spravuje údaje o datasetech uživatelů;

**ImageController** — spravuje uživatelské obrázky.

### 3.1 Auth Controller

Tento kontrolér zajišťuje autorizaci a registraci nových uživatelů. Po provedení požadavku na registraci je do databáze zapsán objekt s následujícími vlastnostmi:

```
public class AuthUser implements UserDetails {  
    @Id  
    @GeneratedValue(strategy = GenerationType.IDENTITY)  
    private String id;  
    @Indexed(unique = true)  
    private String username;  
    private String email;  
    private String password;  
    @Enumerated(EnumType.STRING)  
    private SystemRole role;  
}
```

Obrázek 1: ukázka kódu třídy AuthUser

**POST /api/auth/register** — Registrace nového uživatele.

Příklad požadavku: { "username": "admin1", "password": "password" }.

Příklad odpovědi: { "jwtResponse": "string" }.

**POST /api/auth/login** — Přihlášení uživatele.

Příklad požadavku: { "username": "admin1", "password": "password" }.

Příklad odpovědi: { "jwtResponse": "string" }.

### 3.2 User controller

Pro zajištění autorizace, každá metoda je označena anotací `PreAuthorize` která omezuje skupinu uživatelů které mohou mají přístup k požadavku.

**PUT /api/users/updateUsername** — Aktualizace uživatelského jména.

**PUT /api/users/updateRole** — Aktualizace role uživatele.

**PUT /api/users/updatePassword** — Aktualizace hesla uživatele.

**PUT /api/users/updateEmail** — Aktualizace emailové adresy uživatele.

Příklad požadavku: { "userId": "string", "newUsername": "string" }.

Příklad odpovědi: { "id": "string", "username": "string", "email": "string", "password": "string", "role": "string" }

Vrátí 404 not found pokud vybraný uživatel nebyl nalezen.

**PUT /api/users/update/{id}** — Aktualizace všech údajů uživatele.

Příklad požadavku: { "id": "string", "username": "string", "email": "string", "password": "string", "role": "string" }.

Příklad odpovědi: { "id": "string", "username": "string", "email": "string", "password": "string", "role": "string" }.

**POST /api/users/createUser** — Vytvoření nového uživatele.

Příklad požadavku: { "id": "string", "username": "string", "email": "string", "password": "string", "role": "string" }.

Příklad odpovědi: { "id": "string", "username": "string", "email": "string", "password": "string", "role": "string"}.

**GET /api/users/getByUsername/{username}** — Získání uživatele podle uživatelského jména.

Parametry: username (řetězec) - uživatelské jméno (v cestě).

Příklad odpovědi: { "id": "string", "username": "string", "email": "string ", "password": "string", "role": "string"}.

**GET /api/users/getById/{id}** — Získání uživatele podle identifikátoru.

Parametry: id (řetězec) - identifikátor uživatele (v cestě).

Příklad odpovědi: { "id": "string", "username": "string", "email": "string ", "password": "string", "role": "string"}.

**GET /api/users/getAll** — Získání seznamu všech uživatelů.

Příklad odpovědi: pole uživatelů.

**DELETE /api/users/deleteById/{id}** — Smazání uživatele podle identifikátoru.

Parametry: id (řetězec) - identifikátor uživatele (v cestě).

Příklad odpovědi: { "id": "string", "username": "string", "email": "string", "password": "string", "role": "string"}.

### 3.3 Project Controller

Tento kontrolér je zodpovědný za zpracování uživatelských projektů, vytváření, přidávání a změny projektů, správa uživatelů, kteří mají k projektu přístup.

Pro zajištění autorizace, každá metoda je označena anotací `PreAuthorize` která omezuje skupinu uživatelů které mohou mají přístup k požadavku. Všechny metody v parametru požadavku mají parametr `UserDetails` označený anotací `@AuthenticationPrincipal`, která umožňuje získat data o oprávněném uživateli pomocí frameworku `SpringSecurity`.

```

public class Project {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private String id;
    private String name;
    private String ownerId;
    private Date creationTimestamp;
}

```

*Obrázek 2: ukázka kódu třídy Project*

```

public class UserProject {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private String id;
    private String userId;
    private String projectId;
    private SystemRole userRole;
}

```

*Obrázek 3: ukázka kódu třídy projektu UserProject*

**PUT /api/projects/update/{id}** — Aktualizace projektu.

Parametry: id (řetězec) - identifikátor projektu (v cestě).

Příklad požadavku: { "id": "string", "name": "string", "ownerId": "string", "creationTimestamp": "date"}.

Příklad odpovědi: { "id": "string", "name": "string", "ownerId": "string", "creationTimestamp": "date"}.

Vrátí 403 forbidden, pokud uživatel není autorizován nebo nemá přístup k vybranému projektu.

**POST /api/projects/users/addUserToProject** — Přidání uživatele do projektu.

Příklad požadavku: { "id": "string", "userId": "string", "projectId": "string", "userRole": "string"}.

Příklad odpovědi: { "user": { "id": "string", "username": "string", "email": "string", "password": "string", "role": "string"} },

"userProject": {"id": "string", "userId": "string", "projectId": "string", "userRole": "string" }}

Vrátí 403 forbidden, pokud uživatel není autorizován nebo nemá přístup k vybranému projektu.

**POST /api/projects/create** — Vytvoření nového projektu.

Příklad požadavku: { "id": "string", "name": "string", "ownerId": "string", "creationTimestamp": "date" }.

Příklad odpovědi: { "id": "string", "name": "string", "ownerId": "string", "creationTimestamp": "date" }.

Vrátí 403 forbidden, pokud uživatel není autorizován nebo uživatel nemá oprávnění správce.

**GET /api/projects/users/getByUserIdAndProjectId** — Získání uživatelů podle identifikátoru uživatele a projektu.

Parametry: "userId"(řetězec) - identifikátor uživatele, "projectId"(řetězec) - identifikátor projektu

Příklad odpovědi: pole: {"user": {"id": "string", "username": "string", "email": "string", "password": "string", "userRole": "string"}, "id": "string", "userId": "string", "projectId": "string", "userRole": "string" }

Vrátí 403 forbidden, pokud uživatel není autorizován nebo nemá přístup k vybranému projektu nebo uživatel nemá oprávnění správce.

**GET /api/projects/users/getByProjectId/{projectId}** — Získání uživatelů podle identifikátoru projektu.

Parametry: projectId (řetězec) - identifikátor projektu (v cestě)

Příklad odpovědi: pole: {"user": {"id": "string", "username": "string", "email": "string", "password": "string", "userRole": "string"}, "id": "string", "userId": "string", "projectId": "string", "userRole": "string" }

Vrátí 403 forbidden, pokud uživatel není autorizován nebo nemá přístup k vybranému projektu nebo uživatel nemá oprávnění správce.

**GET /api/projects/getById/{projectId}** — Získání projektu podle identifikátoru.

Parametry: projectId (řetězec) - identifikátor projektu (v cestě)

Příklad odpovědi: pole projektů.

Vrátí 403 forbidden, pokud uživatel není autorizován nebo nemá přístup k vybranému projektu nebo uživatel nemá oprávnění správce.

**GET /api/projects/getAll** — Získání seznamu všech projektů.

Příklad odpovědi: pole: { "id": "string", "name": "string", "ownerId": "string", "creationTimestamp": "date" }.

Vrátí 403 forbidden, pokud uživatel nemá oprávnění správce.

**GET /api/projects/getAllByOwnerId/{id}** — Získání seznamu všech projektů podle identifikátoru vlastníka.

Parametry: id (řetězec) - identifikátor vlastníka (v cestě).

Příklad odpovědi: pole projektů.

Vrátí 403 forbidden, pokud uživatel není autorizován nebo autorizovaný uživatel má jiné ID nebo uživatel nemá oprávnění správce.

**DELETE /api/projects/users/delete** — Smazání uživatele z projektu.

Příklad požadavku: { "userId": "string", "projectId": "string", "userRole": "SYSTEM\_USER" }.

Příklad odpovědi: { "user": { "id": "string", "username": "string", "email": "string", "password": "string", "userRole": "string" },

userProject: { "id": "string", "userId": "string", "projectId": "string", "userRole": "string" } }.

Vrátí 403 forbidden, pokud uživatel není autorizován nebo nemá přístup k vybranému projektu nebo uživatel nemá oprávnění správce.

**DELETE /api/projects/delete/{id}** — Smazání projektu podle identifikátoru.

Parametry: id (řetězec) - identifikátor projektu (v cestě).

Příklad odpovědi: { "id": "string", "name": "string", "ownerId": "string", "creationTimestamp": "date" }.

Vrátí 403 forbidden, pokud uživatel není autorizován nebo nemá přístup k vybranému projektu nebo uživatel nemá oprávnění správce.

### 3.4 Dataset Controller

Tento kontrolér je zodpovědný za zpracování uživatelských datasetů, vytváření, přidávání a změny, správa uživatelů, kteří mají k datasetům přístup.

Pro zajištění autorizace, každá metoda je označena anotací `PreAuthorize` která omezuje skupinu uživatelů které mohou mají přístup k požadavku. Všechny metody v parametru požadavku mají parametr `UserDetails` označený anotací `@AuthenticationPrincipal`, která umožňuje získat data o oprávněném uživateli pomocí frameworku `SpringSecurity`.

```
public class Dataset {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private String id;
    private String name;
    private String projectId;
    private String ownerId;
    private Date creationTimestamp;
    private String category;
}
```

Obrázek 4: ukázka kódu třídy `Dataset`

```
public class UserDataset {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private String id;
    private String userId;
    private String datasetId;
    private SystemRole userRole;
}
```

Obrázek 5: ukázka kódu třídy `UserDataset`

**PUT /api/datasets/update/{id}** — Aktualizace datasetu.

Parametry: id (řetězec): Identifikátor datasetu (v cestě).

Příklad požadavku: { "id": "string", "name": "string", "projectId": "string", "ownerId": "string", "creationTimestamp": "date", "category": "string"}.

Příklad odpovědi: { "id": "string", "name": "string", "projectId": "string", "ownerId": "string", "creationTimestamp": "date", "category": "string"}.



Vrátí 403 forbidden, pokud uživatel není autorizován nebo nemá přístup k vybranému datasetu nebo uživatel nemá oprávnění správce.

**POST /api/datasets/users/addUserToDataset** — Přidání uživatele do datasetu.

Příklad požadavku: { "id": "string", "userId": "string", "datasetId": "string", "userRole": "string" }.

Vrátí 403 forbidden, pokud uživatel není autorizován nebo nemá přístup k vybranému datasetu nebo uživatel nemá oprávnění správce.

**POST /api/datasets/create** — Vytvoření nového datasetu.

Příklad požadavku: { "id": "string", "name": "string", "ownerId": "string", "projectId": "string", "creationTimestamp": "date" }

Vrátí 403 forbidden, pokud uživatel není autorizován nebo nemá přístup k vybranému projektu nebo uživatel nemá oprávnění správce.

**GET /api/datasets/users/getByDatasetIdAndUserId** — Získání uživatelů podle identifikátoru uživatele a datasetu.

Parametry: id (řetězec): Identifikátor uživatele.

userId (řetězec): Identifikátor datasetu.

Příklad odpovědi: { "user": { "id": "string", "username": "string", "email": "string", "password": "string", "userRole": "string" },

"userDataset": { "id": "string", "userId": "string", "datasetId": "string", "userRole": "string" } }

Vrátí 403 forbidden, pokud uživatel není autorizován nebo nemá přístup k vybranému datasetu nebo uživatel nemá oprávnění správce.

**GET /api/datasets/users/getByDatasetId/{datasetId}** — Získání uživatelů podle identifikátoru datasetu.

Parametry: datasetId (řetězec): Identifikátor datasetu (v cestě).

Příklad odpovědi: pole datasetů,

Vrátí 403 forbidden, pokud uživatel není autorizován nebo nemá přístup k vybranému datasetu nebo uživatel nemá oprávnění správce.

**GET /api/datasets/getById/{id}** — Získání datasetu podle identifikátoru.

Parametry:id (řetězec): Identifikátor datasetu (v cestě).

Příklad odpovědi: { "id": "string", "name": "string", "ownerId": "string", "projectId": "string", "creationTimestamp": "date" }

Vrátí 403 forbidden, pokud uživatel není autorizován nebo nemá přístup k vybranému datasetu nebo uživatel nemá oprávnění správce.

**GET /api/datasets/getAll** — Získání seznamu všech datasetů.

Příklad odpovědi: pole datasetů

Vrátí 403 forbidden, pokud nemá oprávnění správce

**GET /api/datasets/getAllByProjectId/{projectId}** — Získání seznamu všech datasetů podle identifikátoru projektu.

Parametry:projectId (řetězec): Identifikátor projektu (v cestě).

Příklad odpovědi: pole datasetů.

Vrátí 403 forbidden, pokud uživatel není autorizován nebo nemá přístup k vybranému projektu nebo nemá oprávnění správce.

**GET /api/datasets/getAllByOwnerId/{ownerId}** — Získání seznamu všech datasetů podle identifikátoru vlastníka.

Parametry:ownerId (řetězec): Identifikátor vlastníka (v cestě).

Příklad odpovědi: pole datasetů

Vrátí 403 forbidden, pokud uživatel není autorizován nebo když přihlášený uživatel se pokusí získat data jiného uživatele nebo uživatel nemá oprávnění správce.

**GET /api/datasets/getAllByName/{name}** — Získání seznamu všech datasetů podle jména.

Parametry:name (řetězec): Jméno datasetu (v cestě).

Příklad odpovědi: pole datasetů

Vrátí 403 forbidden, pokud uživatel není autorizován nebo nemá oprávnění správce.

**DELETE /api/datasets/users/deleteUserFromDataset** — Smazání uživatele z datasetu.

Příklad požadavku: { "id": "string", "userId": "string", "datasetId": "string", "userRole": "string" }

Příklad odpovědi: { "user": { "id": "string", "username": "string", "email": "string", "password": "string", "userRole": "string" },

"userDataset": { "id": "string", "userId": "string", "datasetId": "string", "userRole": "string" } }

Vrátí 403 forbidden, pokud uživatel není autorizován nebo nemá přístup k vybranému datasetu nebo nemá oprávnění správce.

**DELETE /api/datasets/deleteAllByProjectId/{id}** — Smazání všech datasetů podle identifikátoru projektu.

Parametry: id (řetězec): Identifikátor projektu (v cestě).

Příklad odpovědi: pole datasetu

Vrátí 403 forbidden, pokud uživatel není autorizován nebo nemá přístup k vybranému projektu nebo nemá oprávnění správce.

**DELETE /api/datasets/delete/{id}** — Smazání datasetu podle identifikátoru.

Parametry: id (řetězec): Identifikátor datasetu (v cestě).

Vrátí 403 forbidden, pokud uživatel není autorizován nebo nemá přístup k vybranému datasetu nebo nemá oprávnění správce.

### 3.5 Image Controller

Tento kontrolér zajišťuje zprávu obrazu uložených uživatelem a objekty pro labelování obrazu.

Pro zajištění autorizace, každá metoda je označena anotací `PreAuthorize` která omezuje skupinu uživatelů které mohou mají přístup k požadavku. Všechny metody v parametru požadavku mají parametr `UserDetails` označený anotací `@AuthenticationPrincipal`, která umožňuje získat data o oprávněném uživateli pomocí frameworku `SpringSecurity`.

```

public class Image {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private String id;

    private String name;
    private String format;
    private byte[] data;
    private String datasetId;
    @ElementCollection
    private List<String> categories;
}

```

Obrázek 6: ukázka kódu třídy Image

```

public class ImageObject {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private String id;
    private String name;
    private String imageId;
}

```

Obrázek 7: ukázka kódu třídy ImageObject

```

public class ObjectPoint {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private String id;
    private String imageObjectId;
    private String x;
    private String y;
}

```

Obrázek 8: ukázka kódu třídy ObjectPoint

**PUT /api/images/objects/update/{id}** — Aktualizace objektu obrazu.

Parametry: id (řetězec): Identifikátor objektu obrazu (v cestě).

Příklad požadavku: { "id": "string", "name": "string", "imageId": "string" }

Příklad odpovědi: { "id": "string", "name": "string", "imageId": "string" }

Vrátí 403 forbidden, pokud uživatel není autorizován nebo nemá přístup k vybranému datasetu nebo nemá oprávnění správce.

**POST /api/images/save** — Uložení obrazu.

Parametry: name (řetězec): Název obrazu (v dotazu),

datasetId (řetězec): Identifikátor datasetu (v dotazu),

categories (pole stringů): Kategorie obrazu (v dotazu),

data (MultipartFile): obsah soubrou.

Příklad odpovědi: {"id": "string", "name": "string", "format": "string", "data": "byte[]" ,  
"datasetId": "string", "categories": "string[]"}

Vrátí 403 forbidden, pokud uživatel není autorizován nebo nemá přístup k vybranému datasetu nebo nemá oprávnění správce.

**POST /api/images/objects/save** — Uložení objektu obrazu.

Příklad požadavku: { "id": "string", "name": "string", "imageId": "string" }

Příklad odpovědi: { "id": "string", "name": "string", "imageId": "string" }

Vrátí 403 forbidden, pokud uživatel není autorizován nebo nemá přístup k vybranému datasetu nebo nemá oprávnění správce.

**POST /api/images/objects/points/save** — Uložení bodů objektu obrazu.

Příklad požadavku: { "id": "string", "imageObjectId": "string", "x": "string", "y": "string" }

Příklad odpovědi: { "id": "string", "imageObjectId": "string", "x": "string", "y": "string" }

Vrátí 403 forbidden, pokud uživatel není autorizován nebo nemá přístup k vybranému datasetu nebo nemá oprávnění správce.

**GET /api/images/objects/points/getAll** — Získání všech bodů objektů obrazu.

Příklad odpovědi: pole bodu { "id": "string", "imageObjectId": "string", "x": "string", "y":  
"string" }

Vrátí 403 forbidden, pokud uživatel nemá oprávnění správce.

**GET /api/images/objects/points/getAllByObjectId/{id}** — Získání všech bodů objektů obrazu podle identifikátoru objektu.

Parametry: id (řetězec): Identifikátor objektu obrazu (v cestě).

Příklad odpovědi: pole bodu { "id": "string", "imageObjectId": "string", "x": "string", "y":  
"string" }

Vrátí 403 forbidden, pokud uživatel není autorizován nebo nemá přístup k vybranému datasetu nebo nemá oprávnění správce.

**GET /api/images/objects/findAll** — Získání všech objektů obrazu.

Příklad odpovědi: pole objektu { "id": "string", "name": "string", "imageId": "string" }

Vrátí 403 forbidden, pokud uživatel nemá oprávnění správce.

**GET /api/images/objects/findAllByImageId/{imageId}** — Získání všech objektů obrazu podle identifikátoru obrazu.

Parametry:imageId (řetězec): Identifikátor obrazu (v cestě).

Příklad odpovědi: pole objektu { "id": "string", "name": "string", "imageId": "string" }

Vrátí 403 forbidden, pokud uživatel není autorizován nebo nemá přístup k vybranému datasetu nebo nemá oprávnění správce.

**GET /api/images/getById/{imageId}** — Získání obrazu podle identifikátoru.

Parametry:imageId (řetězec): Identifikátor obrazu (v dotazu).

Příklad odpovědi: { "id": "string", "name": "string", "format": "string", "data": "byte[]" , "datasetId": "string", "categories": "string[]" }

Vrátí 403 forbidden, pokud uživatel není autorizován nebo nemá přístup k vybranému datasetu nebo nemá oprávnění správce.

**GET /api/images/getAll** — Získání seznamu všech obrazů.

Příklad odpovědi: pole obrazu { "id": "string", "name": "string", "format": "string", "data": "byte[]" , "datasetId": "string", "categories": "string[]" }

Vrátí 403 forbidden, pokud uživatel nemá oprávnění správce.

**GET /api/images/getAllByDatasetId/{datasetId}** — Získání seznamu všech obrazů podle identifikátoru datasetu.

Parametry:datasetId (řetězec): Identifikátor datasetu (v dotazu).

Příklad odpovědi: pole obrazu { "id": "string", "name": "string", "format": "string", "data": "byte[]" , "datasetId": "string", "categories": "string[]" }

Vrátí 403 forbidden, pokud uživatel není autorizován nebo nemá přístup k vybranému datasetu nebo nemá oprávnění správce.

**GET /api/images/findAllByCategoryContains** — Získání seznamu všech obrazů obsahujících zadanou kategorii.

Parametry:category (řetězec): Kategorie (v dotazu).

Příklad odpovědi: pole obrazu {"id": "string", "name": "string","format": "string","data":"byte[]" , "datasetId": "string", "categories": "string[]"} }

Vrátí 403 forbidden, pokud uživatel nemá oprávnění správce.

**GET /api/images/findAllByCategories** — Získání seznamu všech obrazů podle zadaných kategorií.

Parametry:categories (pole stringů): Kategorie (v dotazu).

Příklad odpovědi: pole obrazu {"id": "string", "name": "string","format": "string","data":"byte[]" , "datasetId": "string", "categories": "string[]"} }

Vrátí 403 forbidden, pokud uživatel nemá oprávnění správce.

**GET /api/images/findAllByCategoriesAndDatasetId** — Získání seznamu všech obrazů podle zadaných kategorií a identifikátoru datasetu.

Parametry:datasetId (řetězec): Identifikátor datasetu (v dotazu).

categories (pole stringů): Kategorie (v dotazu).

Příklad odpovědi: pole obrazu {"id": "string", "name": "string","format": "string","data":"byte[]" , "datasetId": "string", "categories": "string[]"} }

Vrátí 403 forbidden, pokud uživatel není autorizován nebo nemá přístup k vybranému datasetu nebo nemá oprávnění správce.

**DELETE /api/images/objects/points/deleteAllByObjectId/{id}** — Smazání všech bodů objektů obrazu podle identifikátoru objektu.

Parametry:id (řetězec): Identifikátor objektu (v cestě).

Příklad odpovědi: pole bodu { "id": "string", "imageObjectId": "string", "x": "string", "y": "string"} }

Vrátí 403 forbidden, pokud uživatel není autorizován nebo nemá přístup k vybranému datasetu nebo nemá oprávnění správce.

**DELETE /api/images/objects/points/delete/{id}** — Smazání bodu objektu obrazu podle identifikátoru.

Parametry:id (řetězec): Identifikátor bodu (v cestě).

Příklad odpovědi: { "id": "string", "imageObjectId": "string", "x": "string", "y": "string" }

Vrátí 403 forbidden, pokud uživatel není autorizován nebo nemá přístup k vybranému datasetu nebo nemá oprávnění správce.

**DELETE /api/images/objects/deleteById/{id}** — Smazání objektu obrazu podle identifikátoru.

Parametry:id (řetězec): Identifikátor objektu (v cestě).

Příklad odpovědi: { "id": "string", "name": "string", "ownerId":"string","creationTimestamp": "date" }

Vrátí 403 forbidden, pokud uživatel není autorizován nebo nemá přístup k vybranému datasetu nebo nemá oprávnění správce.

**DELETE /api/images/objects/deleteAllByImageId/{id}** — Smazání všech objektů obrazu podle identifikátoru obrazu.

Parametry:id (řetězec): Identifikátor obrazu (v cestě).

Příklad odpovědi: pole objektu{ "id": "string", "name": "string", "ownerId":"string","creationTimestamp": "date" }

Vrátí 403 forbidden, pokud uživatel není autorizován nebo nemá přístup k vybranému datasetu nebo nemá oprávnění správce.

**DELETE /api/images/deleteById/{imageId}** — Smazání obrazu podle identifikátoru.

Parametry:imageId (řetězec): Identifikátor obrazu (v cestě).

Příklad odpovědi: { "id": "string", "name": "string","format": "string","data":"byte[]" , "datasetId": "string", "categories": "string[]" }



Vrátí 403 forbidden, pokud uživatel není autorizován nebo nemá přístup k vybranému datasetu nebo nemá oprávnění správce.

**DELETE /api/images/deleteAllByDatasetId/{datasetId}** — Smazání všech obrazů podle identifikátoru datasetu.

Parametry:datasetId (řetězec): Identifikátor datasetu (v cestě).

Příklad odpovědi: pole obrazu {"id": "string", "name": "string","format": "string","data":"byte[]" , "datasetId": "string", "categories": "string[]"} }

Vrátí 403 forbidden, pokud uživatel není autorizován nebo nemá přístup k vybranému datasetu nebo nemá oprávnění správce.