

# User

Django uživatel bez dalších fieldů. Uživatelé se dělí na normální uživatele, na administrátory (`is_staff=True`) a na superusery (`is_superuser=True`). Administrátory záměrně nelze modifikovat přes API, musí se to dělat buď v databázi, nebo v Django administraci (což zvládne pouze superuser).

## Retrieve fields:

Název pole	Popis	Oprávnění
django fieldy kromě password	RTFD	Admin nebo sám sobě
courses	Kurzy, kterých je členem	Admin nebo sám sobě
parallels	Paralelky, kterých je členem	Admin nebo sám sobě
jobs	Joby, které spustil	Admin nebo sám sobě
assignments	Úlohy, které pro něj byly vygenerovány	Admin nebo sám sobě
coursesAsStudent	Kurzy, kterých je studentem	Admin nebo sám sobě
coursesAsTeacher	Kurzy, kterých je učitelem	Admin nebo sám sobě
parallelsAsStudent	Paralelky, kterých je studentem	Admin nebo sám sobě
parallelsAsTeacher	Paralelky, kterých je učitelem	Admin nebo sám sobě

## Akce:

Název akce	Popis	Vstup	Výstup	Oprávnění
List				Admin
Detail				Přihlášený uživatel
Create	Heslo se nastaví zvlášť (aby si jej uživatel mohl stejnou akcí měnit)	Username, first_name, last_name, email		Admin
Update	Heslo se nastaví zvlášť (aby si jej uživatel mohl stejnou akcí měnit)	first_name, last_name, email		Admin
SetPassword	Nastavení/změna hesla	Password (plaintext)	Uživatel, jehož heslo bylo změněno	Admin nebo sám sobě
Myself	Vrátí detail vlastního profilu	-	Uživatel, který provedl request	Přihlášený uživatel

# Job

Objekt reprezentující úlohu. Používá se při asynchronních requestech. Slouží k ukládání dat i metadat úlohy.

## Retrieve fields:

Název pole	Popis	Oprávnění
user	Kdo úlohu spustil	Admin nebo autor úlohy
createTime	Kdy byla úloha vytvořena	Admin nebo autor úlohy
description	Textový popis úlohy	Admin nebo autor úlohy
endTime	Kdy byla úloha ukončena	Admin nebo autor úlohy
error	Chybový výstup úlohy	Admin nebo autor úlohy
result	Výstup úlohy	Admin nebo autor úlohy
startTime	Kdy byla úloha spuštěna	Admin nebo autor úlohy
running	Běží úloha?	Admin nebo autor úlohy
finished	Byla úloha ukončena?	Admin nebo autor úlohy
duration	Doba běhu úlohy	Admin nebo autor úlohy
status	RUNNING/FAILED/PENDING/COMPLETED	Admin nebo autor úlohy

## Akce:

Název akce	Popis	Vstup	Výstup	Oprávnění
List				Admin
Detail				Admin nebo autor úlohy

# Course

Objekt reprezentující kurz. Paralelky a uživatele je třeba buď naimportovat z KOS, nebo přidat ručně.

## Retrieve fields:

Název pole	Popis	Oprávnění
kosSemester	Semestr z KOS	Přihlášený uživatel
kosTag	Tag z KOS	Přihlášený uživatel
name	Jméno kurzu	Přihlášený uživatel
parallels	Paralelky kurzu	Přihlášený uživatel
assignments	Úlohy (nevygenerované) v kurzu	Admin nebo učitel
noParallel	Paralelka pro lidi bez paralelky	Admin, student nebo učitel
teachers	Učitelé kurzu	Přihlášený uživatel
students	Studenti kurzu	Admin nebo učitel
members	Členové kurzu	Admin nebo učitel

## Akce:

Název akce	Popis	Vstup	Výstup	Oprávnění
List				Přihlášený uživatel
Detail				Přihlášený uživatel
Create				Admin
Update				Admin
ImportFromKos	Import kurzu z KOS, <b>async</b>	course_tag, semester (např. BI-PS1, B181)	Job, který vrátí ID kurzu	Admin

# Parallel

Objekt reprezentující paralelku. Paralelka pro členství v kurzu paralelky nemá jméno (každá jiná jej mít musí), když se zakládá paralelka ručně, je vždy potřeba jí nějaké jméno nastavit.

## Retrieve fields:

Název pole	Popis	Oprávnění
course	Kurz, ke kterému paralelka patří	Přihlášený uživatel
day	Kdy?	Přihlášený uživatel
kosId	ID paralelky v KOS	Přihlášený uživatel
name	Jméno paralelky	Přihlášený uživatel
parity	V sudý/lichý týden?	Přihlášený uživatel
room	Kde?	Přihlášený uživatel
startTime	Kdy?	Přihlášený uživatel
teachers	Učitelé paralelky	Přihlášený uživatel
students	Studenti paralelky	Admin nebo učitel
members	Členové paralelky	Admin nebo učitel

## Akce:

Název akce	Popis	Vstup	Výstup	Oprávnění
Detail				Přihlášený uživatel
Create				Admin
Update				Admin

# ParallelMembership

Objekt reprezentující členství v paralelce. Nelze editovat, vždy je potřeba smazat a vytvořit nový. Pro smazání je třeba List, filtr pomocí paralelky a uživatele, získat tak id ParallelMembership a toto potom použít pro Delete.

## Retrieve fields:

Název pole	Popis	Oprávnění
parallel		Admin
user		Admin
level	Student nebo učitel?	Admin

## Akce:

Název akce	Popis	Vstup	Výstup	Oprávnění
List				Admin
Create				Admin
Delete				Admin

# Service

Objekt reprezentující externí službu, buď generátor zadání, nebo opravovač. Jméno je pouze pro potřeby identifikace v aplikaci, url je base URL služby (musí být úplná, včetně specifikace protokolu). Jak se má služba chovat je popsáno v readme projektu. Při přidání nové služby systém otestuje, zda služba splňuje požadavky, a pokud ne, nepřidá ji. Automaticky se uloží očekávané formáty dat.

## Retrieve fields:

Název pole	Popis	Oprávnění
correctionSchema	JSONschema dat předávaných zadávajícím opravovači	Admin
generatorSchema	JSONschema dat předávaných zadávajícím generátoru	Admin
isGenerator	Ternární boolean: ano/ne, je opravovač/ještě není ověřena (job nedoběhl)	Přihlášený uživatel
name		Přihlášený uživatel
submissionSchema	JSONschema dat předávaných studentem opravovači	Admin
url		Admin
generator_for	Šablony, pro které je tato služba generátorem	Admin
correction_for	Šablony, pro které je tato služba opravovačem	Admin

## Akce:

Název akce	Popis	Vstup	Výstup	Oprávnění
List				Přihlášený uživatel
Detail				Přihlášený uživatel
Create	<b>async</b>	name, url	Job, který vrátí ID služby	Admin

## Poznámky:

Jméno služby a její typ musí vidět každý přihlášený uživatel, jinak by učitel nemohl zadávat úlohy (toto se asi časem změní – přiřazení služeb ke kurzům)

# AssignmentTemplate

Definuje, jaké služby budou používány pro opravu úloh.

## Retrieve fields:

Název pole	Popis	Oprávnění
correctionService		Admin
generatorService	Nepovinné: když chybí, všichni dostanou stejné zadání (jako doposud na learnshellu)	Admin
name		Přihlášený uživatel
assignments		Admin

## Akce:

Název akce	Popis	Vstup	Výstup	Oprávnění
List				Přihlášený uživatel
Detail				Přihlášený uživatel
Create				Admin

## Poznámky:

Jméno šablony musí vidět každý přihlášený uživatel, jinak by učitel nemohl zadávat úlohy (toto se asi časem změní – přiřazení služeb ke kurzům)

# Assignment

Konkrétní úloha zadaná učitelem. Každá úloha má svého vlastníka – uživatele, který ji jako jediný může editovat (aby se předešlo data race). Rozpracovanou úlohu lze otestovat. Když je úloha považována za hotovou a otestovanou, tak se publikuje, poté již nemůže být editována. Zároveň s publikací se vygeneruje zadání pro každého studenta.

Učitel vyplní data pro generátor a opravovač. Formáty jsou definované ve službách. Mimo klasické JSONschema definují atribut widget, který klientovi říká, jaký druh formulářového pole by bylo dobré použít.

V zadání úlohy (description) a jménu úlohy je možno použít jazyk Jinja2. Do šablony se vyrenderují následující data:

```
{
    "env": data prostředí, kde je student přihlášen (momentálně pouze username)
    "vars": data z generátoru zadání
}
```

## Retrieve fields:

Název pole	Popis	Oprávnění
course		Admin nebo učitel
owner		Admin nebo učitel
template		Admin nebo učitel
correctionData	JSONschema odpovídající službě opravovače	Admin nebo učitel
description	Jinja2	Admin nebo učitel
generatorData	JSONschema odpovídající službě generátoru	Admin nebo učitel
name	Jinja2	Admin nebo učitel
published	Je pulikovaná?	Admin nebo učitel
generatedAssignments		Admin nebo učitel
generatorSchema	Shortcut na schéma dat služby	Admin nebo učitel
correctionSchema	Shortcut na schéma dat služby	Admin nebo učitel

## Akce:

Název akce	Popis	Vstup	Výstup	Oprávnění
List				Přihlášený uživatel
Detail				Přihlášený uživatel
Create		Owner a published se nastaví samo		Admin nebo učitel



Update	Pouze nepublikovaná	Owner a published se modifikuje zvlášť		Vlastník úlohy
Delete	Pouze nepublikovaná			Vlastník úlohy
ChangeOwner	Změna vlastníka úlohy	ID úlohy, ID nového vlastníka		Admin nebo vlastník úlohy
Publish	Publikuje úlohu <b>async</b>	ID úlohy	job: job, který vrátí, pro koho se vygenerování podařilo, pro koho ne, a počty object: return self	Vlastník úlohy
Test	Vygeneruje úlohu k otestování, předchozí vygenerovaná úloha se smaže <b>async</b>	ID úlohy	job, který vrátí ID vygenerované úlohy	Admin nebo učitel

# Generated Assignment

Zadání úlohy vygenerované pro konkrétního studenta. Nebo učitele pro testování.

## Retrieve fields:

Název pole	Popis	Oprávnění
assignment		Admin, učitel nebo student, pro kterého se úloha vygenerovala
student	Uživatel, pro kterého se úloha vygenerovala	Admin, učitel nebo student, pro kterého se úloha vygenerovala
data	Data z generátoru zadání	Admin, učitel nebo student, pro kterého se úloha vygenerovala
submissions		Admin, učitel nebo student, pro kterého se úloha vygenerovala
name	Vyrenderované jméno úlohy	Admin, učitel nebo student, pro kterého se úloha vygenerovala
descriptionMd	Zadání v MD	Admin, učitel nebo student, pro kterého se úloha vygenerovala
descriptionHtml	Zadání v HTML	Admin, učitel nebo student, pro kterého se úloha vygenerovala
score	Dosažený počet bodů	Admin, učitel nebo student, pro kterého se úloha vygenerovala

## Akce:

Název akce	Popis	Vstup	Výstup	Oprávnění
Detail				Přihlášený uživatel
List				Přihlášený uživatel

# Submission

Odevzdání úlohy. Obsahuje pouze data, která jsou známa ještě před opravením, data opravy jsou uložena v OneToOne relationship: Correction.

## Retrieve fields:

Název pole	Popis	Oprávnění
generatedAssignment		Admin, učitel nebo student, pro kterého se úloha vygenerovala
submissionData	Musí odpovídat submission schematu	Admin, učitel nebo student, pro kterého se úloha vygenerovala
createdAt	Vyplní se automaticky	Admin, učitel nebo student, pro kterého se úloha vygenerovala
correction		Admin, učitel nebo student, pro kterého se úloha vygenerovala
submissionSchema	Shortcut na schéma dat služby	Admin, učitel nebo student, pro kterého se úloha vygenerovala

## Akce:

Název akce	Popis	Vstup	Výstup	Oprávnění
List				Přihlášený uživatel
Detail				Přihlášený uživatel
Create	Zároveň s odevzdáním vyrobí job na opravu <b>async</b>		Job: vrátí ID Correction object: return this	učitel nebo student, pro kterého se úloha vygenerovala
Delete				Admin

# Correction

Část odevzdání obsahující data o opravě úlohy. Body se zatím berou staticky podle fieldu score v příchozím JSONu.

## Retrieve fields:

Název pole	Popis	Oprávnění
submission		Admin, učitel nebo student, pro kterého se úloha vygenerovala
data	To, co přišlo zpět z opravovače	Admin, učitel nebo student, pro kterého se úloha vygenerovala
createdAt	Vyplní se automaticky	Admin, učitel nebo student, pro kterého se úloha vygenerovala
score	Dosažené body	Admin, učitel nebo student, pro kterého se úloha vygenerovala
generatedAssignment	Úloha, ke které oprava patří	Admin, učitel nebo student, pro kterého se úloha vygenerovala

## Akce:

Název akce	Popis	Vstup	Výstup	Oprávnění
------------	-------	-------	--------	-----------

## Poznámky:

Zatím nedokáže říct, jakým způsobem se mají data vykreslovat. V budoucnu by to chtělo nejspíš další schéma.