

III. TVORBA MODELU

1. Modely v aplikaci Django

Webová aplikace v Django provádí správu dat prostřednictvím tzv. modelu. Model definuje strukturu uložených dat včetně datových typů, klíčů a parametrů atributů (sloupců) - například maximální délky, výchozí hodnoty, seznamu přípustných hodnot, ale také pojmenování prvku ve formuláři nebo textu nápovědy. Základem modelu jsou třídy vytvářené podle zásad objektového programování v jazyce Python a Django s daty pracuje jako s objekty nezávisle na použitém typu databáze. Využívá k tomu programovací techniku známou jako ORM (Object-relational mapping neboli objektově-relační mapování). Díky tomu můžeme hotovou aplikaci nasadit v kombinaci s různými databázovými systémy, aniž by bylo nutné provádět složité změny v programovém kódu. Zpravidla stačí pouze zásah do konfiguračního souboru a přepsání nastavení databáze.

Podrobněji o principech ORM zde: <https://www.fullstackpython.com/object-relational-mappers-orms.html>

Východiskem pro tvorbu modelu v Django by měla být datová analýza zadaného problému a výsledkem pak konkrétní představa o datovém modelu aplikace. Pro naši jednoduchou filmovou databázi si můžeme připravit například následující datový model filmové databáze:

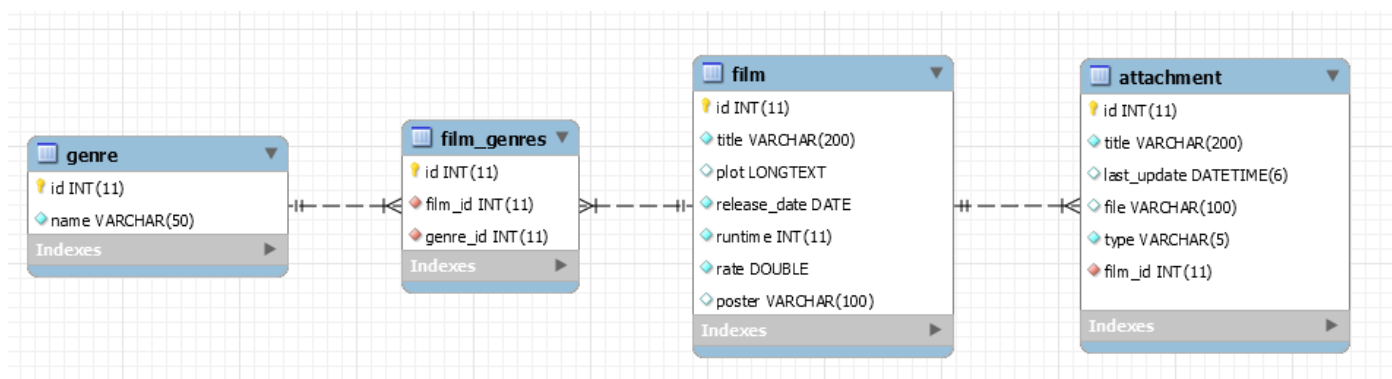


Schéma naznačuje, že nejdůležitější údaje budou uloženy v tabulce film (název filmu, děj, datum vydání filmu do distribuce, délka filmu, hodnocení, cesta k uloženému obrázku s plakátem filmu). Do tabulky genre budou ukládány názvy filmových žánrů (např. komedie, drama, krimi, sci-fi atd.). Protože k filmu můžeme někdy přiřadit více žánrů, je mezi tabulkami film a genre vztah M:N. V důsledku toho vznikne v relačních databázích pomocná propojovací tabulka film_genres, v níž jsou použity cizí klíče z obou zmíněných tabulek. Jiný typ vztahu - 1:N - je použit mezi tabulkou film a tabulkou attachment. V této tabulce budeme registrovat všechny přílohy, které k danému filmu v aplikaci uložíme (příkladem může být recenze ve formátu PDF, rozhlasový rozhovor s režisérem ve formátu MP3 nebo trailer čili ukázka z filmu v některém z oblíbených videoformátů). U příloh zapisujeme kromě jejího názvu také čas poslední aktualizace, název souboru a jeho typ.

Všimněte si, že ve všech tabulkách je primární klíč označen identifikátorem id. Jak si ukážeme dále, Django se o vytvoření primárních klíčů postará automaticky, pokud tedy nechceme modelu "vnutit" nějaké vlastní primární klíče, které je samozřejmě také možné definovat. Automatický přístup má ale některé své přednosti (například si ušetříme psaní několika řádků kódu :-)) a obvykle se doporučuje ponechat tuto starost frameworku.

2. Definice prvního modelu (soubor movies\models.py)

Pokud jsme si dobře promysleli datovou strukturu naší aplikace, můžeme přistoupit k samotné tvorbě modelů a začít editovat soubor models.py ve složce movies.

Podrobná dokumentace k tvorbě modelů v Django i s četnými příklady naleznete zde: <https://docs.djangoproject.com/en/3.0/topics/db/models/>

Začínáme nejprve od modelů, které neobsahují cizí klíče, a naopak prostřednictvím svého primárního klíče jsou zdrojem dat pro jiné tabulky. V našem případě můžeme na prvním místě vytvořit model pro žánry - tedy třídu Genre:

```
from django.db import models
```

""" Třída Genre je modelem pro databázový objekt (tabulku), v němž budou ukládány názvy žánrů filmů

V závorce za názvem třídy je uvedena třída-předek Model pocházející z knihovny (balíku, modulu) models (models.Model)

Je zde využit princip dědičnosti - všechny naše vlastní modely pocházejí ze společného předka třídy Model

a dědí její vlastnosti """

```
class Genre(models.Model):
```

"""Fields - definice jednotlivých polí/sloupců modelu/tabulky

Každé pole modelu (budoucí tabulky v databázi) je uloženo do vhodně pojmenované proměnné/atributu - zde "name"

Vzniká jako instance určité třídy (zde models.CharField), která rozhoduje o datovém typu pole a o jeho vlastnostech

V tomto případě bude pole "name" obsahovat maximálně 50 znaků (parametr max_length), bude obsahovat unikátní hodnoty (parametr unique),

ve formuláři se bude zobrazovat pod označením "Genre name" (parametr verbose_name), a uživateli se jako nápověda nabídne text uvedený v parametru help_text """

```
name = models.CharField(max_length=50, unique=True, verbose_name="Genre name", help_text='Enter a film genre (e.g. sci-fi, comedy)')
```

""" Metadata - slouží ke specifikaci některých dalších vlastností modelu, jež ale už mohou být v řadě případů závislé

i na konkrétním typu použité databáze """

```
class Meta:
```

atribut ordering definuje upřednostňovaný způsob řazení - zde vzestupně podle pole/sloupce name

```
ordering = ["name"]
```

""" Methods - definují chování objektu v určité situaci """

```
def __str__(self):
```

""" Řetězec, který se používá jako textová reprezentace objektu (například v administraci aplikace).

V našem případě bude objekt (žánr) reprezentován výpisem obsahu pole name """

```
return self.name
```

Vložené komentáře by vám měly objasnit význam jednotlivých řádků kódu.

3. Přidání modelu do administrace Django

Abychom mohli zjistit, jak model v praxi funguje, můžeme k tomu využít administrační prostředí, které nám nabízí framework Django. Už jsme se s ním setkali v okamžiku, kdy jsme vytvořili účet tzv. **superuživatele** a ověřovali jeho přihlášení.

Nyní do administrace webu přidáme i náš první model **Genre**. Provedeme to jednoduchou úpravou souboru **movies\admin.py**:

```
from django.contrib import admin
```

```
# Import všech modelů, které obsahuje models.py
```

```
from .models import *
```

```
# Registrace modelů v administraci aplikace
```

```
admin.site.register(Genre)
```

Před samotným spuštěním webového serveru musíme opět provést migrace:

```
py manage.py makemigrations
py manage.py migrate
```

Spustíme webový server:

```
py manage.py runserver
```

Do adresního řádku prohlížeče zadáme URL adresu pro vstup do administrace našeho webu (<http://127.0.0.1:8000/admin/>) a přihlásíme se pod účtem superuživatele.

Poté bychom měli spatřit tuto základní obrazovku:

The screenshot shows the Django Admin interface. The browser address bar displays `127.0.0.1:8000/admin/`. The page header includes the title "Správa systému Django" and navigation links: "VÍTEJTE, UŽIVATELI HILDA. ZOBRAZENÍ WEBU / ZMĚNIT HESLO / ODHLÁSIT SE". The main content area is titled "Správa webu" and contains two sections: "AUTENTIZACE A AUTORIZACE" with links for "Skupiny" and "Uživatelé" (each with "+ Přidat" and "Změnit" buttons), and "MOVIES" with a link for "Genres" (also with "+ Přidat" and "Změnit" buttons). On the right, there is a sidebar with "Nedávné akce" and "Moje akce" (showing "Nic").

Vidíme, že k odkazům týkajícím se správy uživatelských účtů přibýly nově odkazy pro správu žánrů v aplikaci movies. Můžeme si vyzkoušet přidání nového žánru:

The screenshot shows the "Add Genre" form in the Django Admin interface. The browser address bar displays `127.0.0.1:8000/admin/movies/genre/add/`. The page header is the same as the previous screenshot. The breadcrumb trail at the top reads "Domů > Movies > Genres > genre: přidat". The main content area is titled "genre: přidat" and features a form with a label "Genre name:" and a text input field containing "sci-fi". Below the input field is a hint: "Enter a film genre (e.g. sci-fi, comedy)". At the bottom of the form, there are three buttons: "Uložit a přidat další položku", "Uložit a pokračovat v úpravách", and "ULOŽIT".

Nyní vidíme, jak se některé parametry vytvořeného modelu promítají do podoby formuláře: název formulářového pole (v modelu parametr `verbose_name`) nebo nápověda pod formulářovým prvkem (`help_text`).

Když postupně přidáváme další žánry, objevují se v podobě seznamu.

← → ↻ 127.0.0.1:8000/admin/movies/genre/ ☆ M Pozastaveno !

Správa systému Django

VÍTEJTE, UŽIVATELI HILDA. [ZOBRAZENÍ WEBU](#) / [ZMĚNIT HESLO](#) / [ODHLÁSIT SE](#)

Domů > Movies > Genres

Vyberte položku genre ke změně GENRE: PŘIDAT +

Operace: Provést Vybranych je 0 položek z celkem 2.

Genre	Action
<input type="checkbox"/> komedie	
<input type="checkbox"/> sci-fi	

2 genres

Můžeme ještě ověřit, zda funguje unikátnost zadání jména žánru. Pokusíme-li se o opětovné přidání již dříve zadaného žánru, validace formulářů nahlásí chybu:

← → ↻ 127.0.0.1:8000/admin/movies/genre/add/ ☆ M Pozastaveno !

Správa systému Django

VÍTEJTE, UŽIVATELI HILDA. [ZOBRAZENÍ WEBU](#) / [ZMĚNIT HESLO](#) / [ODHLÁSIT SE](#)

Domů > Movies > Genres > genre: přidat

genre: přidat

Opravte níže uvedenou chybu.

Položka Genre s touto hodnotou v poli Genre name již existuje.

Genre name:

Enter a film genre (e.g. sci-fi, comedy)

Uložit a přidat další položku Uložit a pokračovat v úpravách ULOŽIT

4. Vytvoření modelu Film (soubor movies\models.py)

V následujícím kroku vytvoříme klíčový model naší malé filmové databáze - třídu Film. Potřebný kód budeme opět psát do souboru **movies\models.py**. Na začátku souboru nejprve vložíme potřebné importy:

```
from django.db import models
# Import metod MaxValueValidator, MinValueValidator z balíku django.core.validators
from django.core.validators import MaxValueValidator, MinValueValidator
# Import metody reverse z balíku django.urls - zajistí generování URL obrácením URL
patterns
from django.urls import reverse
```

Poté pokračujeme psaním kódu pod již existující model Genre:

```

""" Třída Film je modelem pro databázový objekt (tabulku), který bude obsahovat základní údaje o filmech """

class Film(models.Model):
    # Fields
    # Znakové pole o maximální délce 200 znaků pro vložení názvu filmu
    title = models.CharField(max_length=200, verbose_name="Title")
    # Textové pole pro vložení delšího textu popisujícího děj filmu
    # Formulářový prvek může zůstat prázdný - parametry blank=True, null=True
    plot = models.TextField(blank=True, null=True, verbose_name="Plot")
    # Pole obsahuje datum uvedení filmu, které musí být zadáno v náležitém tvaru (YYYY-MM-DD); nepovinný údaj
    release_date = models.DateField(blank=True, null=True,
                                     help_text="Please use the following format: <em>YYYY-MM-DD</em>.",
                                     verbose_name="Release date")
    # Celočíslné pole pro nepovinné zadání stopáže (délky) filmu v minutách
    runtime = models.IntegerField(blank=True, null=True,
                                   help_text="Please enter an integer value (minutes)",
                                   verbose_name="Runtime")
    # Pole pro zadání desetinného čísla vyjadřujícího hodnocení filmu v rozsahu 1.0 až 10.0
    # Výchozí hodnota je nastavena na 5.0
    # K validaci hodnot jsou použity metody z balíku/knihovny django.core.validators
    rate = models.FloatField(default=5.0,
                              validators=[MinValueValidator(1.0), MaxValueValidator(10.0)],
                              null=True,
                              help_text="Please enter an float value (range 1.0 - 10.0)",
                              verbose_name="Rate")
    # Vytvoří vztah mezi modely Film a Genre typu M:N
    genres = models.ManyToManyField(Genre, help_text='Select a genre for this film')

    # Metadata
    class Meta:
        # Záznamy budou řazeny primárně sestupně (znaménko minus) podle data uvedení,
        # sekundárně vzestupně podle názvu
        ordering = ["-release_date", "title"]

    # Methods
    def __str__(self):
        """Součástí textové reprezentace filmu bude jeho název, rok uvedení a hodnocení"""
        return f"{self.title}, year: {str(self.release_date.year)}, rate: {str(self.rate)}"

    def get_absolute_url(self):
        """Metoda vrací URL stránky, na které se vypisují podrobné informace o filmu"""
        return reverse('film-detail', args=[str(self.id)])

```

Projděte si opět vložené komentáře, abyste lépe pochopili principy tvorby modelu ve frameworku Django.

Abychom funkčnost modelu ověřili v administraci aplikace, musíme opakovat kroky z předchozí podkapitoly:

1. Zaregistrujeme nový model v souboru **movies\admin.py**.
2. Příkazy provedeme migrace.

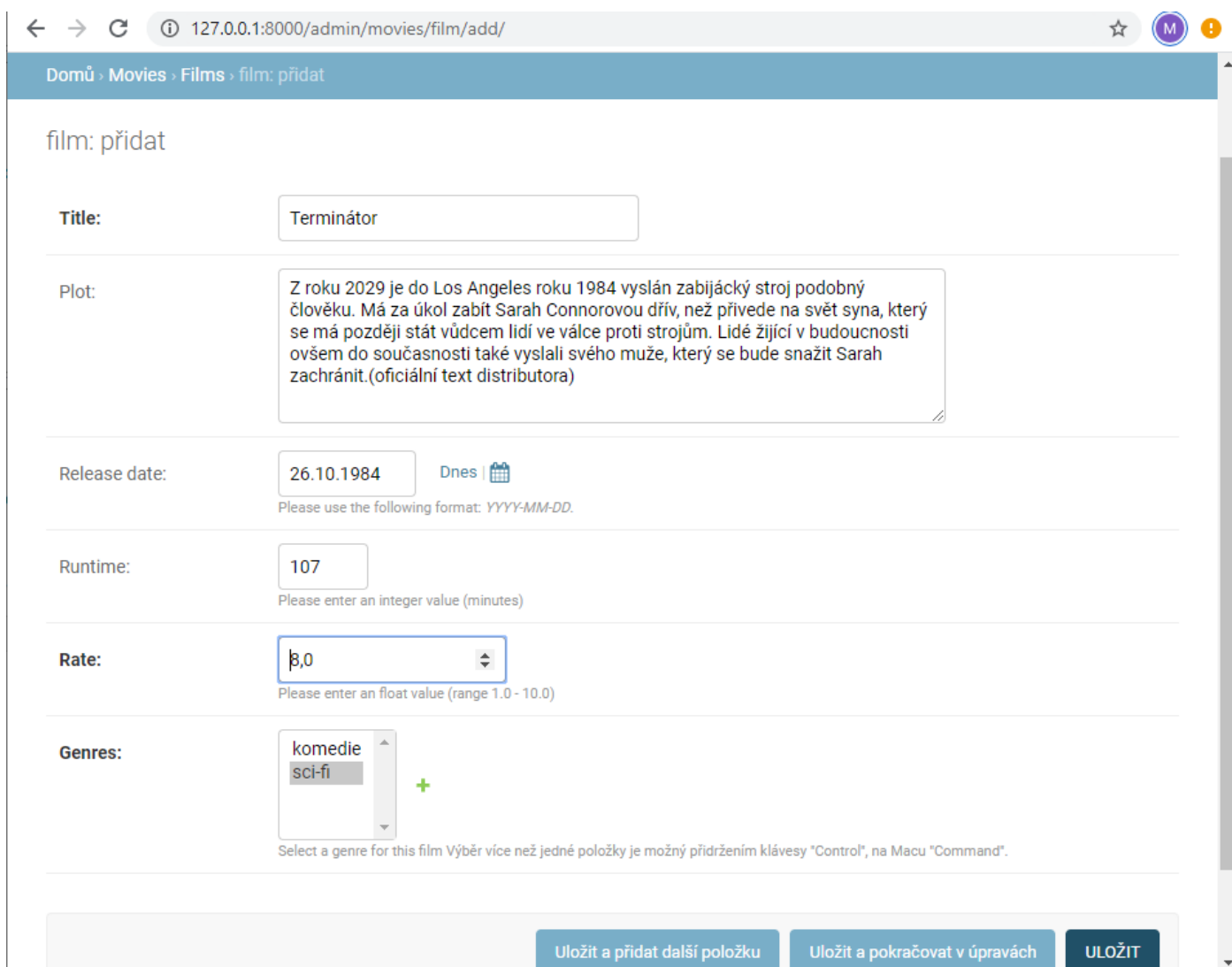
3. Spustíme webový server.
4. V prohlížeči se přihlásíme do administrace.

Ve správě systému Django by se nyní měl objevit nový řádek označený Films:



The screenshot shows the Django Admin interface. The top navigation bar includes the Django logo and links for 'VÍTEJTE, UŽIVATELI HILDA', 'ZOBRAZENÍ WEBU', 'ZMĚNIT HESLO', and 'ODHLÁSIT SE'. The main content area is titled 'Správa webu' (Website Management). Under the 'AUTENTIZACE A AUTORIZACE' (Authentication and Authorization) section, there are links for 'Skupiny' (Groups) and 'Uživatelé' (Users), each with '+ Přidat' and 'Změnit' (Edit) options. The 'MOVIES' section is highlighted, showing 'Films' and 'Genres', each with '+ Přidat' and 'Změnit' options. On the right, the 'Nedávné akce' (Recent Actions) section shows 'Moje akce' (My Actions) with a list of actions: 'komedie' (Genre) and 'sci-fi' (Genre).

Pomocí odkazů můžeme opět přidat několik testovacích záznamů.



The screenshot shows the Django Admin interface for adding a new film. The breadcrumb trail is 'Domů > Movies > Films > film: přidat'. The form is titled 'film: přidat'. It includes the following fields:

- Title:** Terminátor
- Plot:** Z roku 2029 je do Los Angeles roku 1984 vyslán zabijácký stroj podobný člověku. Má za úkol zabít Sarah Connorovou dřív, než přivede na svět syna, který se má později stát vůdcem lidí ve válce proti strojům. Lidé žijící v budoucnosti ovšem do současnosti také vyslali svého muže, který se bude snažit Sarah zachránit. (oficiální text distributora)
- Release date:** 26.10.1984 (Dnes) Please use the following format: YYYY-MM-DD.
- Runtime:** 107 Please enter an integer value (minutes)
- Rate:** 8,0 Please enter a float value (range 1.0 - 10.0)
- Genres:** komedie, sci-fi Select a genre for this film. Výběr více než jedné položky je možný přidržetím klávesy "Control", na Macu "Command".

At the bottom, there are three buttons: 'Uložit a přidat další položku', 'Uložit a pokračovat v úpravách', and 'ULOŽIT'.

Všimněme si řešení propojení modelu Film s modelem Genre (viz výběrový seznam Genres).

Můžeme úmyslnou (nebo neúmyslnou) chybou prověřit funkčnost validace dat ve formuláři:

127.0.0.1:8000/admin/movies/film/add/

Opravte níže uvedenou chybu.

Title:

12 opic

Plot:

James Cole (Bruce Willis) prožil převážnou většinu svého života v podzemí. Nyní, v roce 2025, žije v ponuré, tmavé, zasněžené přítomnosti. Začalo to už na konci dvacátého století, kdy během dvou let (1996 a 1997) zemřelo příčiněm neznámého bídačka, který vypustil do ovzduší nebezpečný vir, pět miliard obyvatel zeměkoule. Zůstalo jen jedno procento lidí, kteří se teď krčí jako krtci pod zemí. Hrůza lidí se s tím ale nehodlá jenom tak smířit, a proto vysílá do minulosti prostřednictvím stroje času "nedobrovolné dobrovolníky" z

Release date:

Dnes

Please use the following format: YYYY-MM-DD.

Runtime:

124

Please enter an integer value (minutes)

Rate:

10,1

Hodnota musí být menší nebo rovna 10.0.

Please enter an float value (range 1.0 - 10.0)

Genres:

komedie

sci-fi

+

Select a genre for this film Výběr více než jedné položky je možný přidáním klávesy "Control", na Macu "Command".

Uložit a přidat další položku

Uložit a pokračovat v úpravách

ULOŽIT

V seznamu filmů vidíme, jak se výpis filmů ovlivňuje úprava tzv. magické metody `__str__()` a současně jak se projevuje nastavení proměnné `ordering` - filmy jsou primárně seřazeny od nejnovějších po nejstarší:

Domů · Movies · Films

✓

Položka typu film "Smrtící epidemie, year: 1995, rate: 7.3" byla úspěšně přidána.

Vyberte položku film ke změně

FILM: PŘIDAT

Operace:

Provést

Vybraných je 0 položek z celkem 6.

FILM

12 opic, year: 1995, rate: 8.7

Smrtící epidemie, year: 1995, rate: 7.3

Terminátor, year: 1984, rate: 8.7

Zítra vstanu a opařím se čajem, year: 1977, rate: 7.8

Bílá paní, year: 1965, rate: 8.2

Bílá nemoc, year: 1937, rate: 8.4

6 films

5. Model Attachment a ukládání externích příloh

Naši malou filmovou databázi chceme ještě vylepšit o možnost vkládání různých typů příloh - médií (textů, obrázků, zvuků, videa apod.) - k jednotlivým filmům. Přílohy nebudou ukládány v databázi, ale ve speciální složce určené pro uploady souborů (media) a pro přehlednost ještě v podsložkách film, podsložce symbolicky pojmenované podle id filmu a podsložce attachments. Do databáze vložíme jen záznam o každém takovém souboru a prostřednictvím cizího klíče i propojení s konkrétním filmem.

Na začátku souboru **movies\models.py** si připravíme speciální metodu, jejímž úkolem bude vracet správnou cestu k uploadované příloze:

```
# Import metody reverse z balíku django.urls - zajistí generování URL obrácením URL patterns
from django.urls import reverse

""" Metoda vrací cestu k uploadovaným souborům - přílohám filmů.
    Cesta má obecnou podobu: film/id-filmu/attachments/nazev-souboru.
    Parametr instance odkazuje na instanci (objekt) filmu.
    Parametr filename obsahuje název uploadovaného souboru. """

def attachment_path(instance, filename):
    return "film/" + str(instance.film.id) + "/attachments/" + filename
```

Na konci souboru **movies\models.py** vytvoříme model Attachment:

```
""" Třída Attachment je modelem pro databázový objekt (tabulku), který bude obsahovat údaje o přílohách filmů """

class Attachment(models.Model):
    # Fields
    # Povinný titulek přílohy - text do délky 200 znaků
    title = models.CharField(max_length=200, verbose_name="Title")
    # Časový údaj o poslední aktualizaci přílohy - automaticky se ukládá aktuální čas
    last_update = models.DateTimeField(auto_now=True)
    # Pole pro upload souboru
    # Parametr upload_to zajistí uložení souboru do složky specifikované v návratové hodnotě metody attachment_path
    file = models.FileField(upload_to=attachment_path, null=True, verbose_name="File")

    # Konstanta, v níž jsou ve formě n-tic (tuples) předdefinovány různé typy příloh
    TYPE_OF_ATTACHMENT = (
        ('audio', 'Audio'),
        ('image', 'Image'),
        ('text', 'Text'),
        ('video', 'Video'),
        ('other', 'Other'),
    )

    # Pole s definovanými předvolbami pro uložení typu přílohy
    type = models.CharField(max_length=5, choices=TYPE_OF_ATTACHMENT, blank=True, default='image', help_text='Select allowed attachment type', verbose_name="Attachment type")

    # Cizí klíč, který zajišťuje propojení přílohy s daným filmem (vztah N:1)
```



```

    # Parametr on_delete slouží k zajištění tzv. referenční integrity - v případě
    odstranění filmu
    # budou odstraněny i všechny jeho přílohy (models.CASCADE)
    film = models.ForeignKey(Film, on_delete=models.CASCADE)

    # Metadata
    class Meta:
        # Primární seřazeno podle poslední aktualizace souborů, sekundárně podle typu
        přílohy
        ordering = ["-last_update", "type"]

    # Methods
    def __str__(self):
        """ Textová reprezentace objektu """
        return f"{self.title}, ({self.type})"

```

Abychom zajistili správné ukládání uploadovaných souborů, provedeme nejprve změnu v konfiguračním souboru **settings.py**:

```

# Umístění statických souborů (CSS, JavaScript, Images)

STATIC_URL = '/static/'
STATIC_ROOT = os.path.join(BASE_DIR, 'static')

# Umístění uploadovaných souborů

MEDIA_URL = '/media/'
MEDIA_ROOT = os.path.join(BASE_DIR, 'media')

```

Pro uploadované soubory jsme tím vymezili složku media, která se vytvoří v základní složce našeho webu (v tomto případě hildaweb).

Zbývá ještě namapovat složku media v souboru **hildaweb/urls.py**:

```

# Mapování URL adres webu - propojení URL s jejich obsluhou
urlpatterns = [
    # URL "admin/" je namapováno na administrační rozhraní webu
    path('admin/', admin.site.urls),
    # URL "movies/" je namapováno na aplikaci movies a její vlastní soubor 'movies.urls'
    path('movies/', include('movies.urls')),
    # domovská stránka webu (') je automaticky přesměrována na url z předešlého řádku
    ('movies')
    path('', RedirectView.as_view(url='movies/')),
] + static(settings.STATIC_URL, document_root=settings.STATIC_ROOT) +
static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
# Metodou static() zajistíme správné mapování adres statických souborů webu (css, js,
obrázků atd.).
# Jejich umístění bude definováno konstantami STATIC_URL a STATIC_ROOT v souboru
settings.py

```

Nakonec ještě zopakujeme již nacvičené úkony - tedy registraci modelu Attachment v souboru **movies\admin.py**, provedeme migraci, spustíme webový server a přihlásíme se do administrace systému. Můžeme zkusit přidat několik příloh k různým filmům:

←

→

↺

127.0.0.1:8000/admin/movies/attachment/add/

☆

M

!

Správa systému Django

VÍTEJTE, UŽIVATELI HILDA. ZOBRAZENÍ WEBU / ZMĚNIT HESLO / ODHLÁSIT SE

Domů > Movies > Attachments > attachment: přidat

attachment: přidat

Title:

Bílá nemoc - knižní vydání divadelní hry

File:

Vybrat soubor

bila-nemoc.pdf

Attachment type:



Text

▼

Select allowed attachment type

Film:

Bílá nemoc, year: 1937, rate: 8.4

▼  

Uložit a přidat další položku

Uložit a pokračovat v úpravách

ULOŽIT

A ověříme si, zda se ukládají na správné místo ve struktuře aplikace:

```
▼ DJANGO
├── .idea
├── .vscode
├── docs
├── hildaweb
│   ├── __pycache__
│   ├── .idea
│   ├── __init__.py
│   ├── asgi.py
│   ├── settings.py
│   ├── urls.py
│   └── wsgi.py
├── media
│   ├── film
│   │   ├── 1
│   │   │   ├── attachments
│   │   │   │   ├── Terminator_-_trailer.mp4
│   │   │   │   ├── terminator-01.jpg
│   │   │   │   ├── terminator-02.jpg
│   │   │   │   └── terminator-03.jpg
│   │   └── 2
│   │       ├── attachments
│   │       │   ├── 12_opic_-_trailer.mp4
│   │       └── 5
│   │           ├── attachments
│   │           │   ├── bila-nemoc-trailer.mp4
│   │           │   └── bila-nemoc.pdf
│   └── movies
```

6. Doplnění modelu Film o upload plakátu

Abychom dokončili tvorbu modelů naší aplikace podle původního návrhu, chybí nám ještě dořešit upload obrázku s plakátem v modelu Film. Podobně jako v předchozí podkapitole si i nyní vytvoříme pomocnou metodu, kterou umístíme na začátek souboru **models.py**:

```
""" Metoda vrací cestu k uploadovanému plakátu. """  
  
def poster_path(instance, filename):  
    return "film/" + str(instance.id) + "/poster/" + filename
```

Do modelu Film poté přidáme nové pole poster:

```
rate = models.FloatField(default=5.0,  
                           validators=[MinValueValidator(1.0), MaxValueValidator(10.0)],  
                           null=True,  
                           help_text="Please enter an float value (range 1.0 - 10.0)",  
                           verbose_name="Rate")  
  
# Pole typu image, které umožňuje upload obrázku s plakátem filmu  
poster = models.ImageField(upload_to=poster_path, blank=True, null=True,  
                           verbose_name="Poster")  
  
# Vytvoří vztah mezi modely Film a Genre typu M:N  
genres = models.ManyToManyField(Genre, help_text='Select a genre for this film')
```

Když se nyní pokusíme aktualizovat migrace, objeví se následující chyba:

```
(django) C:\hilda\django>py manage.py makemigrations  
SystemCheckError: System check identified some issues:  
  
ERRORS:  
movies.Film.poster: (fields.E210) Cannot use ImageField because Pillow is not installed.  
    HINT: Get Pillow at https://pypi.org/project/Pillow/ or run command "python -m pip  
install Pillow".
```

Hned pod ní je uvedeno řešení - je třeba doinstalovat balík Pillow, který se v Pythonu používá k práci s obrázky:

```
(django) C:\hilda\django>python -m pip install Pillow  
Collecting Pillow  
  Downloading Pillow-7.1.1-cp37-cp37m-win_amd64.whl (2.0 MB)  
    |████████████████████████████████████████| 2.0 MB 939 kB/s  
Installing collected packages: Pillow  
Successfully installed Pillow-7.1.1
```

Po úspěšné instalaci chybějícího balíku provedeme znovu migrace, spustíme server a můžeme k jednotlivým filmům vložit obrázky s plakáty:

127.0.0.1:8000/admin/movies/film/2/change/

M

Pozastaveno

Title:

12 opic

Plot:

James Cole (Bruce Willis) prožil převážnou většinu svého života v podzemí. Nyní, v roce 2025, žije v ponuré, tmavé, zasněžené přítomnosti. Začalo to už na konci dvacátého století, kdy během dvou let (1996 a 1997) zemřelo přičiněním neznámého bídaáka, který vypustil do ovzduší nebezpečný vir, pět miliard obyvatel zeměkoule. Zůstalo jen jedno procento lidí, kteří se teď krčí jako krtci pod zemí. Hrstka lidí se s tím ale nehodlá jenom tak smířit, a proto vsílá do minulosti prostřednictvím stroje času "nedobrovolné dobrovolníky" z

Release date:

08.12.1995

Dnes

Please use the following format: YYYY-MM-DD.

Runtime:

129

Please enter an integer value (minutes)

Rate:

8,7

Please enter an float value (range 1.0 - 10.0)

Poster:

Vybrat soubor

12-opic-plakat.jpg

Genres:

komedie

sci-fi

+

Select a genre for this film Výběr více než jedné položky je možný přidržením klávesy "Control", na Macu "Command".

Odstranit

Uložit a přidat další položku

Uložit a pokračovat v úpravách

ULOŽIT

Můžeme se přesvědčit, že se obrázky skutečně ukládají do podsložky **poster** ve složce daného filmu.

Na závěr celé této kapitoly provedeme zálohu na GitHub.