

Django

Gál Péter

galpeter@inf.u-szeged.hu

Django

- Python
- Gyors fejlesztés
- Kevés kód – DRY – Don't repeat yourself
- Laza kapcsolódás

Django

MTV

Django

MTV

Model

Django

MTV
Template

Django

MTV

View

Django

ORM

object-relational mapper

Django

Project:

**Timetable
Poll application**

Django – Start project

```
$ django-admin.py startproject timetable
```

```
$ ls timetable/
```

```
__init__.py manage.py settings.py urls.py
```

Django – Developer server

```
$ python manage.py runserver
```

```
Validating models...
0 errors found

Django version 1.1, using settings 'timetable.settings'
Development server is running at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

It worked!

Congratulations on your first Django-powered page.

Of course, you haven't actually done any work yet. Here's what to do next:

- If you plan to use a database, edit the `DATABASE_*` settings in `timetable/settings.py`.
- Start your first app by running `python timetable/manage.py startapp [appname]`.

You're seeing this message because you have `DEBUG = True` in your Django settings file and you haven't configured any URLs. Get to work!

Django – Settings

```
$ cat settings.py
```

```
...
```

```
DATABASE_ENGINE = 'sqlite3'
```

```
DATABASE_NAME = 'db.db'
```

```
DATABASE_USER = ''
```

```
DATABASE_PASSWORD = ''
```

```
DATABASE_HOST = ''
```

```
DATABASE_PORT = ''
```

```
...
```

Django – Start application

```
$ python manage.py startapp poll
```

```
$ ls poll/
```

```
  __init__.py  models.py  tests.py  urls.py  
views.py
```

Django

Modell

Django – Modell – Poll

```
class Poll(models.Model):  
    created_by = models.ForeignKey(User)  
    title = models.CharField(max_length = 200)  
    description = models.TextField()  
    date_start = models.DateTimeField()  
    date_end = models.DateTimeField()
```

Django – SQL – Poll

```
CREATE TABLE "poll_poll" (  
    "id" integer NOT NULL PRIMARY KEY,  
    "created_by_id" integer NOT NULL  
REFERENCES "auth_user" ("id"),  
    "title" varchar(200) NOT NULL,  
    "description" text NOT NULL,  
    "date_start" datetime NOT NULL,  
    "date_end" datetime NOT NULL  
);
```

Django – Modell – Choice

```
class Choice(models.Model):  
    poll = models.ForeignKey(Poll)  
    text = models.CharField(max_length = 50)
```


Django – Modell – Vote

```
class Vote(models.Model):  
    user = models.CharField(max_length = 50)  
    choice = models.ForeignKey(Choice)  
  
    class Meta:  
        unique_together = ('user','choice')
```

Django – SQL – Vote

```
CREATE TABLE "poll_vote" (  
    "id" integer NOT NULL PRIMARY KEY,  
    "user" varchar(50) NOT NULL,  
    "choice_id" integer NOT NULL  
        REFERENCES "poll_choice" ("id"),  
    UNIQUE ("user", "choice_id")  
);
```

Django – Modell

```
$ python manage.py syncdb
```

Django

Admin

Django – Admin

```
$ cat poll/admin.py
```

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

```
from poll.models import Poll, Vote, Choice
```

```
admin.site.register(Poll)
```

```
admin.site.register(Vote)
```

```
admin.site.register(Choice)
```

Django – Admin

The screenshot shows the Django Admin interface for a poll application. The browser window title is "Poll adminisztráció | Django honlap adminisztráció - Mozilla Firefox". The address bar shows the URL "http://localhost:8000/admin/poll/". The page header includes the Django logo and the text "Django adminisztráció" on the left, and "Üdvözlöm, **electro**. Jelszó megváltoztatása / Kijelentkezés" on the right. Below the header, there is a breadcrumb trail "Kezdőlap > Poll". The main content area is titled "Poll adminisztráció" and contains a table with the following data:

Poll	
Choices	+ Új ✎ Módosítás
Polls	+ Új ✎ Módosítás
Votes	+ Új ✎ Módosítás

The status bar at the bottom of the browser window shows "Kész" on the left and "Server: WSGIServer/..." on the right.

Django – Admin

Válasszon ki egyet a módosításhoz (poll) | Django honlap adminisztráció - Mozilla

Fájl Szerkesztés Nézet Előzmények Könyvjelzők Eszközök Súgó

← → ↻ × 🏠 📄 📄 http://localhost:8000/admin/poll/poll/ ☆ Google 🔍

Válasszon ki egyet a módosít... ✕

Django adminisztráció Üdvözlöm, **electro**. Jelszó megváltoztatása / Kijelentkezés

Kezdőlap » Poll » Polls

Válasszon ki egyet a módosításhoz (poll) Új poll +

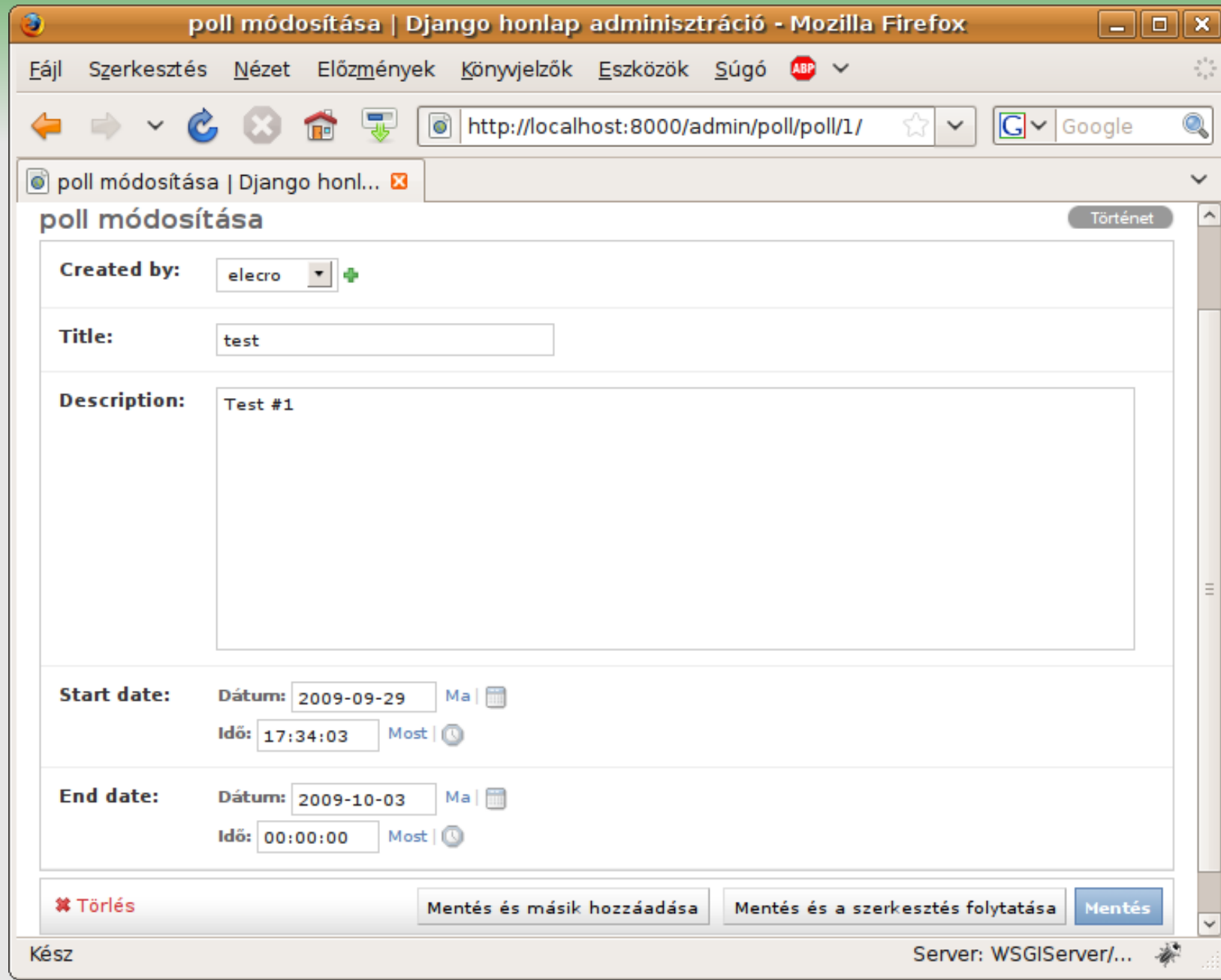
Action: -----

<input type="checkbox"/>	Poll
<input type="checkbox"/>	Test 2
<input type="checkbox"/>	test

2 polls

Kész Server: WSGIServer/...

Django – Admin



The screenshot shows a Mozilla Firefox browser window with the title "poll módosítása | Django honlap adminisztráció". The address bar shows the URL "http://localhost:8000/admin/poll/poll/1/". The page content is titled "poll módosítása" and includes a "Történet" button. The form fields are as follows:

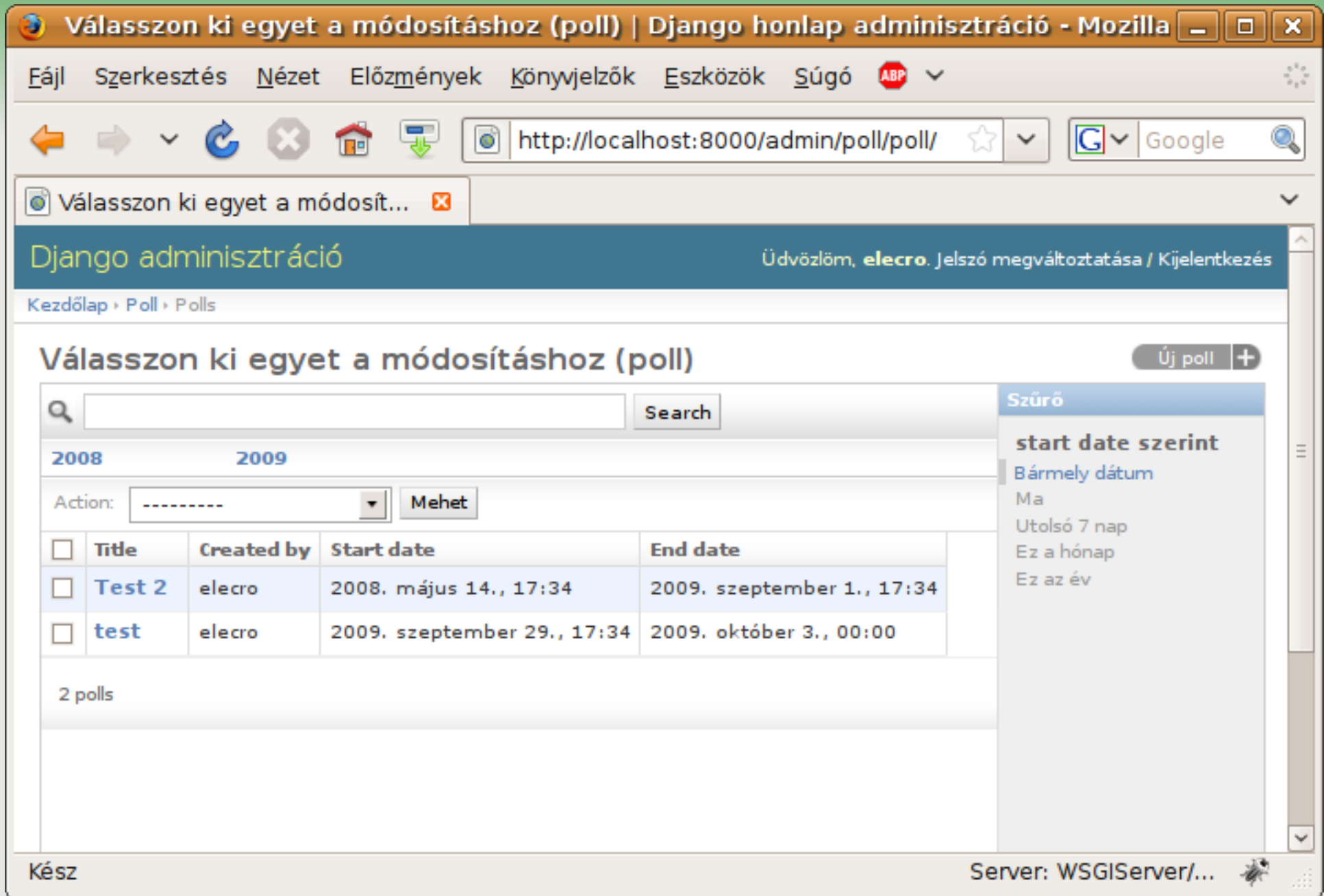
- Created by:** A dropdown menu showing "electro" with a plus sign to the right.
- Title:** A text input field containing "test".
- Description:** A large text area containing "Test #1".
- Start date:** A date field set to "2009-09-29" (Ma) and a time field set to "17:34:03" (Most).
- End date:** A date field set to "2009-10-03" (Ma) and a time field set to "00:00:00" (Most).

At the bottom of the form, there are four buttons: "✖ Törlés", "Mentés és másik hozzáadása", "Mentés és a szerkesztés folytatása", and "Mentés". The status bar at the bottom left says "Kész" and the bottom right shows "Server: WSGIServer/..." with a small bug icon.

Django – Admin v.2

```
class PollAdmin(admin.ModelAdmin):  
    date_hierarchy = 'date_start'  
    list_display = ['title', 'created_by', 'date_start',  
                  'date_end']  
    list_filter = ['created_by', 'date_start']  
    search_fields = ['title']  
  
admin.site.register(Poll, PollAdmin)
```

Django – Admin v.2



The screenshot shows the Django Admin interface in a Mozilla browser window. The page title is "Válasszon ki egyet a módosításhoz (poll) | Django honlap adminisztráció - Mozilla". The browser address bar shows the URL "http://localhost:8000/admin/poll/poll/". The page content includes a search bar, a table of polls, and a sidebar with a filter menu.

Django adminisztráció Üdvözlöm, **electro**. Jelszó megváltoztatása / Kijelentkezés

Kezdőlap » Poll » Polls

Válasszon ki egyet a módosításhoz (poll)

Új poll +

Search

2008 2009

Action: ----- Mehet

<input type="checkbox"/>	Title	Created by	Start date	End date
<input type="checkbox"/>	Test 2	electro	2008. május 14., 17:34	2009. szeptember 1., 17:34
<input type="checkbox"/>	test	electro	2009. szeptember 29., 17:34	2009. október 3., 00:00

2 polls

Szűrő

start date szerint

- Bármely dátum
- Ma
- Utolsó 7 nap
- Ez a hónap
- Ez az év

Kész Server: WSGIServer/...

Django

Poll list

Django – Poll list – URL

```
$ cat poll/url.py
```

```
from django.conf.urls.defaults import patterns, url
```

```
urlpatterns = patterns('timetable.poll.views',  
    url(r'^list/$', 'list_polls'),  
)
```

```
pl.: http://localhost:8000/poll/list/
```

Django – Poll list – View

```
from django.template import Context, loader
```

```
def list_polls(request):
```

```
    polls = Poll.objects.all()
```

```
    template = loader.get_template('poll/list.html')
```

```
    context = Context({'polls': polls})
```

```
    return HttpResponse(template.render(context))
```

Django – Poll list – View v.2

```
from django.shortcuts import render_to_response
```

```
def list_polls(request):  
    polls = Poll.objects.all()  
    return render_to_response('poll/list.html',  
                             {'polls': polls})
```

Django – Poll list – Template

```
$ cat templates/poll/list.html
```

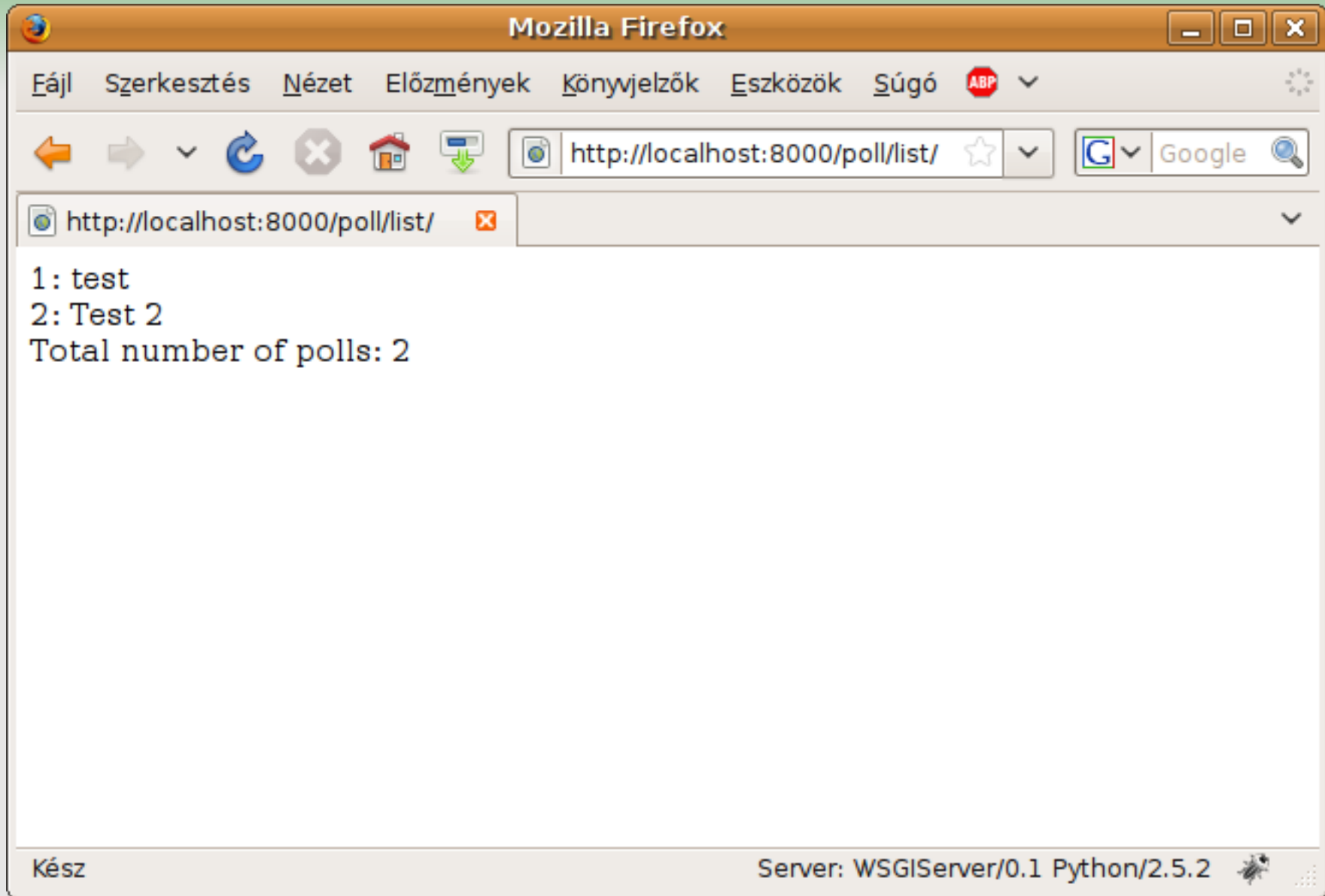
```
{% for poll in polls %}
```

```
  {{ forloop.counter }}: {{ poll }}<br/>
```

```
{% endfor %}
```

```
Total number of polls: {{ polls|length }}
```

Django – Poll list – Template



Django

Poll info

Django – Poll info – URL

```
urlpatterns = patterns('timetable.poll.views',  
...  
    url(r'^info/(?P<poll_id>\d+)/$', 'get_poll',  
        name = 'poll_info'),  
...  
)
```

pl.: <http://localhost:8000/poll/info/1/>

Django – Poll info – View

```
def get_poll(request, poll_id):  
    try:  
        poll = Poll.objects.get(pk = poll_id)  
    except Poll.DoesNotExist:  
        raise Http404  
  
    return render_to_response('poll/info.html',  
                              {'poll': poll})
```

Django – Poll info – View v.2

```
from django.shortcuts import get_object_or_404
```

```
def get_poll(request, poll_id):
```

```
    poll = get_object_or_404(Poll, pk = poll_id)
```

```
    return render_to_response('poll/info.html',  
                              {'poll': poll})
```

Django – Poll info – Template

Poll: {{ poll.title }} by {{ poll.created_by }}

Time:{{ poll.date_start }} - {{ poll.date_end }}

Description: {{ poll.description }}

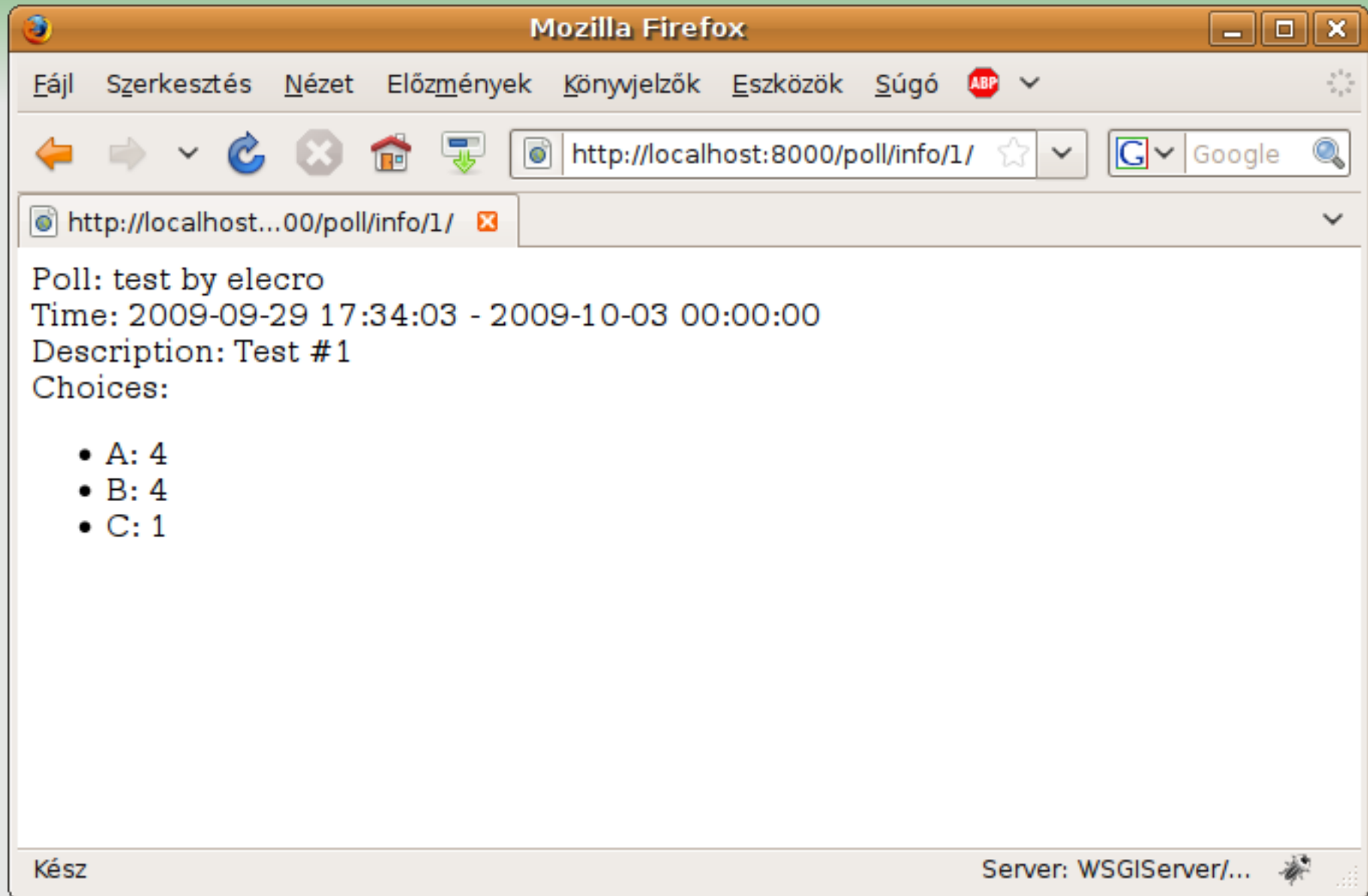
Choices:

{% for choice in poll.choice_set.all %}

{{ choice }}: {{ choice.vote_set.count }}

{% endfor %}

Django – Poll info – Template



Django

Poll vote

Django – Poll vote – Form

```
$ cat poll/forms.py
```

```
from django import forms
```

```
from poll.models import Vote
```

```
class VoteForm(forms.ModelForm):
```

```
    class Meta:
```

```
        model = Vote
```


Django – Poll vote – URL

```
urlpatterns = patterns('timetable.poll.views',  
...  
    url(r'^vote/(?P<poll_id>\d+)/$', 'vote_poll',  
        name = 'poll_vote'),  
...  
)
```

pl.: <http://localhost:8000/poll/vote/1/>

Django – Poll vote - View

```
def vote_poll(request, poll_id):
    poll = get_object_or_404(Poll, pk = poll_id)
    if request.method == "POST":
        ....
    else:
        form = VoteForm()
    return render_to_response('poll/vote.html',
                              {'voteform': form})
```

Django – Poll vote - View

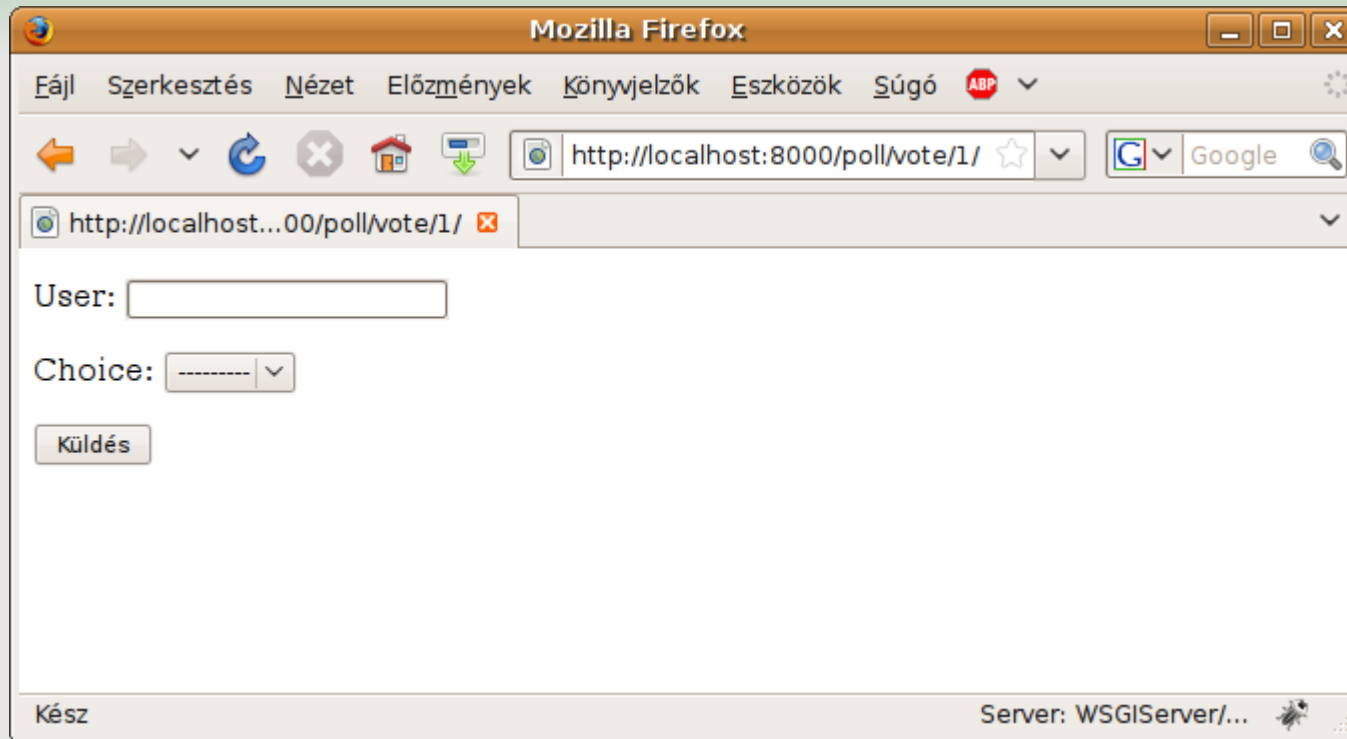
```
def vote_poll(request, poll_id):
    poll = get_object_or_404(Poll, pk = poll_id)
    if request.method == "POST":
        form = VoteForm(request.POST)
        if form.is_valid():
            form.save()
            return HttpResponseRedirect(
reverse('poll_info', kwargs = {'poll_id': poll.id}) )
```

.....

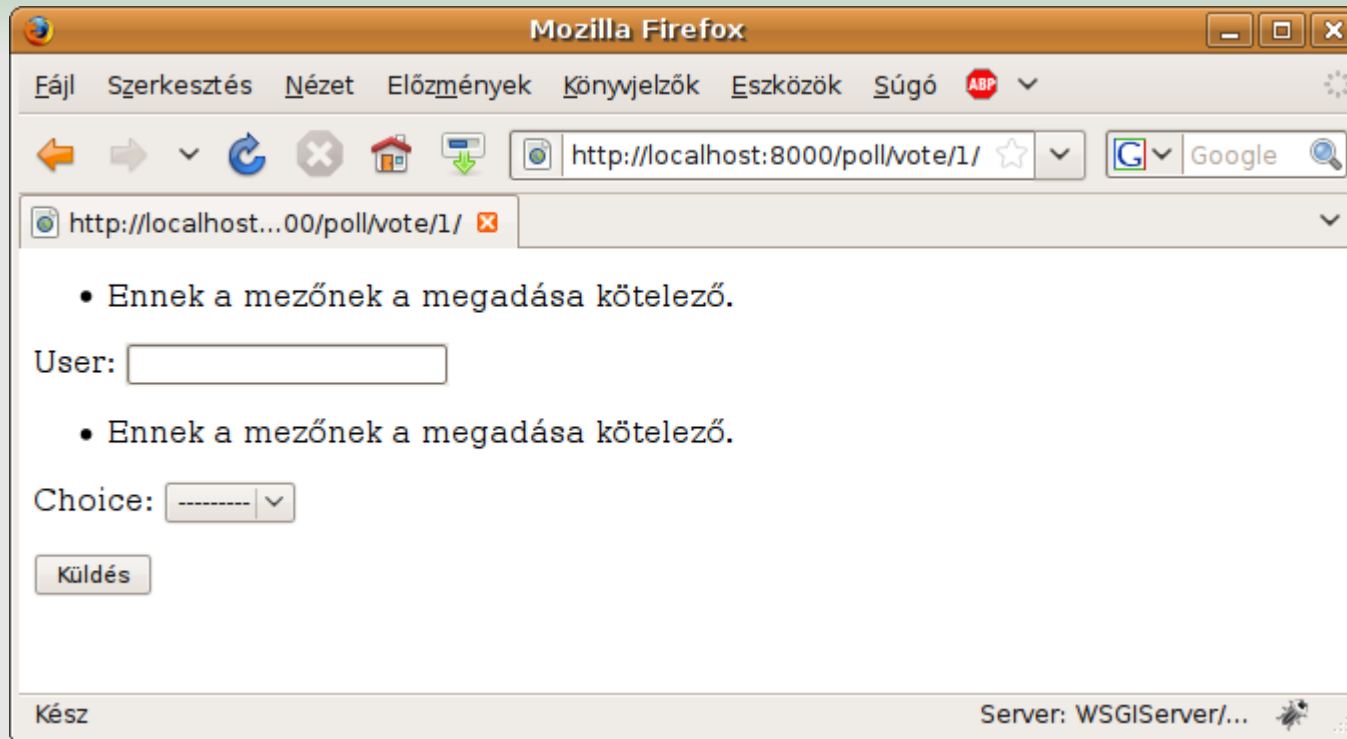
Django – Poll vote – Template

```
<form method="POST">  
{{ voteform.as_p }}  
<input type="submit">  
</form>
```

Django – Poll vote – Template



Django – Poll vote – Template



Mozilla Firefox

Fájl Szerkesztés Nézet Előzmények Könyvjelzők Eszközök Súgó ABP

← → ↻ × 🏠 📄 http://localhost:8000/poll/vote/1/ ☆ Google 🔍

http://localhost...00/poll/vote/1/ ✕

- Ennek a mezőnek a megadása kötelező.

User:

- Ennek a mezőnek a megadása kötelező.

Choice: ▾

Kész Server: WSGIServer/...

Django – egyéb szolgáltatások

- auth
- admin
- flatpages
- humanize
- syndication
- ...

Köszönöm a figyelmet!

Gál Péter

galpeter@inf.u-szeged.hu

Django

Gál Péter
galpeter@inf.u-szeged.hu

Django

- Python
- Gyors fejlesztés
- Kevés kód – DRY – Don't repeat yourself
- Laza kapcsolódás

2

- Python-ban irt keretrendszer
- Gyors fejlesztést próbálja meg lehetővé tenni
- Kevés kód felhasználásával, DRY, nem ismételjük kódot, mindent lehetőleg csak egy helyen írjunk le
- Főbb részei között laza kapcsolat, amennyire csak lehet

Főbb részek?

Django

MTV

3

3 fő réteg

Django

MTV
Model

4

Model réteg:

az adattal kapcsolatos információkat tartalmazza, hogyan kell elérni, ellenőrizni/validálni, milyen függvényei vannak valamint a kapcsolatok ezek között

M stands for “Model,” the data access layer. This layer contains anything and everything about the data: how to access it, how to validate it, which behaviors it has, and the relationships between the data.

Django

MTV

Template

5

Template vagy sablon réteg:
a megjelenést határozza meg, hogyan nézzen ki pl.:
a honlap

T stands for “Template,” the presentation layer. This layer contains presentation-related decisions: how something should be displayed on a Web page or other type of document.

Django

MTV
View

6

View réteg:

„üzleti logika” található itt. Lekérdezni a szükséges adatokat majd a templattal (sablonnal) megjeleníti.

az adatokat (a modell rétegből) köti össze a templattal (sablonnal)

V stands for “View,” the business logic layer. This layer contains the logic that access the model and defers to the appropriate template(s). You can think of it as the bridge between models and templates.

Django

ORM

object-relational mapper

7

Python osztály és adatbázis közti kommunikációt végzi.

Django

Project:
**Timetable
Poll application**

8

Célunk: Naptár projektünkhöz egy szavazó alkalmazás fejlesztése. Tfh.: ez az első komponens.

Django – Start project

```
$ django-admin.py startproject timetable  
$ ls timetable/  
__init__.py manage.py settings.py urls.py
```

9

Projekt létrehozásakor kapott fájlok.
Már ki is próbálható a „üres” projektünk

Django – Developer server

```
$ python manage.py runserver
```

```
Validating models...
0 errors found

Django version 1.1, using settings 'timetable.settings'
Development server is running at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

It worked!

Congratulations on your first Django-powered page.

Of course, you haven't actually done any work yet. Here's what to do next:

- If you plan to use a database, edit the `DATABASE_*` settings in `timetable/settings.py`.
- Start your first app by running `python timetable/manage.py startapp [appname]`.

You're seeing this message because you have `DEBUG = True` in your Django settings file and you haven't configured any URLs. Get to work!

Tartalmaz beépített fejlesztői szerveret, nem kell webszerver

Django – Settings

```
$ cat settings.py
...
DATABASE_ENGINE = 'sqlite3'
DATABASE_NAME = 'db.db'
DATABASE_USER = ''
DATABASE_PASSWORD = ''
DATABASE_HOST = ''
DATABASE_PORT = ''
...
```

11

Legfontosabb beállítás az adatbázis kapcsolat

ENGINE: PostgreSQL, MySQL, Sqlite3 vagy Oracle

egyéb lehetőséget: nyelvi lokalizáció, template könyvtár, időformátum stb..

Django – Start application

```
$ python manage.py startapp poll
```

```
$ ls poll/
```

```
__init__.py models.py tests.py urls.py  
views.py
```

12

Alkalmazás létrehozásakor kapott fájlok

model:

tests: unittestek

urls: milyen url milyen view-t használjon

views: a view-k

Django

Modell

13

Adatbázis sémánk definiálása, Python osztályokkal

Django – Modell – Poll

```
class Poll(models.Model):  
    created_by = models.ForeignKey(User)  
    title = models.CharField(max_length = 200)  
    description = models.TextField()  
    date_start = models.DateTimeField()  
    date_end = models.DateTimeField()
```

14

Egy szavazás információi: ki hozta létre, címe, rövid leírás, kezdő és befejező időpont

Django – SQL – Poll

```
CREATE TABLE "poll_poll" (  
    "id" integer NOT NULL PRIMARY KEY,  
    "created_by_id" integer NOT NULL  
    REFERENCES "auth_user" ("id"),  
    "title" varchar(200) NOT NULL,  
    "description" text NOT NULL,  
    "date_start" datetime NOT NULL,  
    "date_end" datetime NOT NULL  
);
```

15

ORM által generált tábla.

Django – Modell – Choice

```
class Choice(models.Model):  
    poll = models.ForeignKey(Poll)  
    text = models.CharField(max_length = 50)
```

16

Szavazáshoz tartozó opciók/lehetőségek modellje
hivatkozás a Poll modellre

Django – Modell – Vote

```
class Vote(models.Model):
    user = models.CharField(max_length = 50)
    choice = models.ForeignKey(Choice)

    class Meta:
        unique_together = ('user', 'choice')
```

17

felhasználó szavazata, hivatkozás a választott opcióra

Megköveteljük hogy páronként különbözőek legyenek az elemek

Django – SQL – Vote

```
CREATE TABLE "poll_vote" (  
    "id" integer NOT NULL PRIMARY KEY,  
    "user" varchar(50) NOT NULL,  
    "choice_id" integer NOT NULL  
        REFERENCES "poll_choice" ("id"),  
    UNIQUE ("user", "choice_id")  
);
```

18

Generált tábla SQL, látható a UNIQUE megszorítás

Django – Modell

```
$ python manage.py syncdb
```

19

Táblák, indexek létrehozása az adatbázisban

Django

Admin

20

Az adatok gyors megjelenítését biztosítja az „automatikus” admin felület

Django – Admin

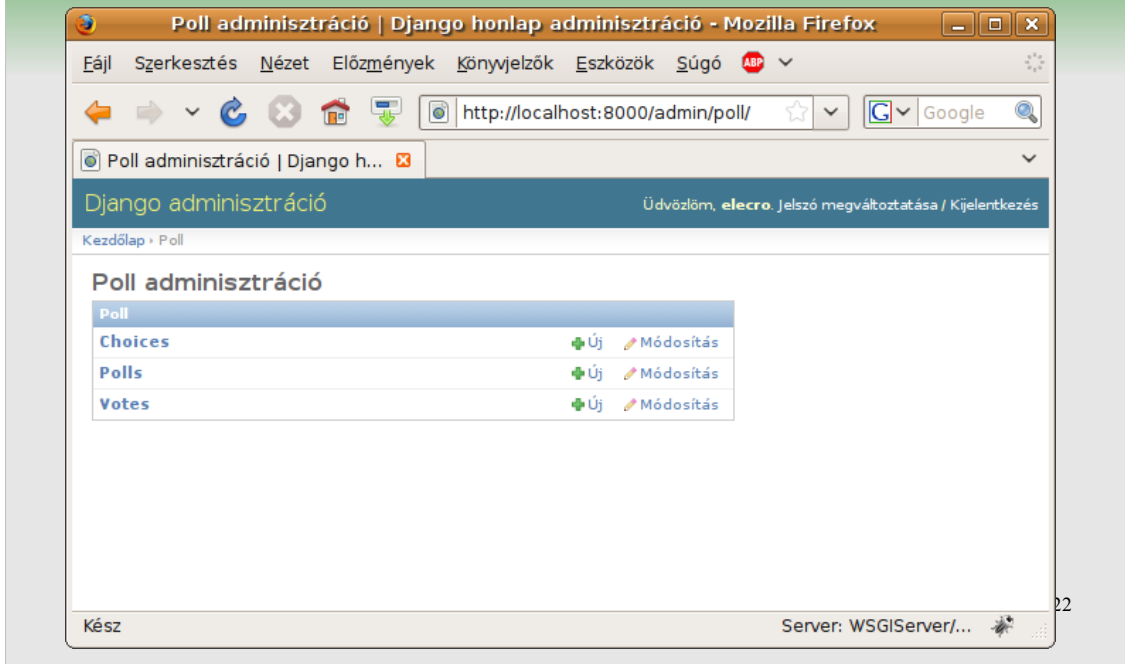
```
$ cat poll/admin.py
from django.contrib import admin
from poll.models import Poll, Vote, Choice

admin.site.register(Poll)
admin.site.register(Vote)
admin.site.register(Choice)
```

21

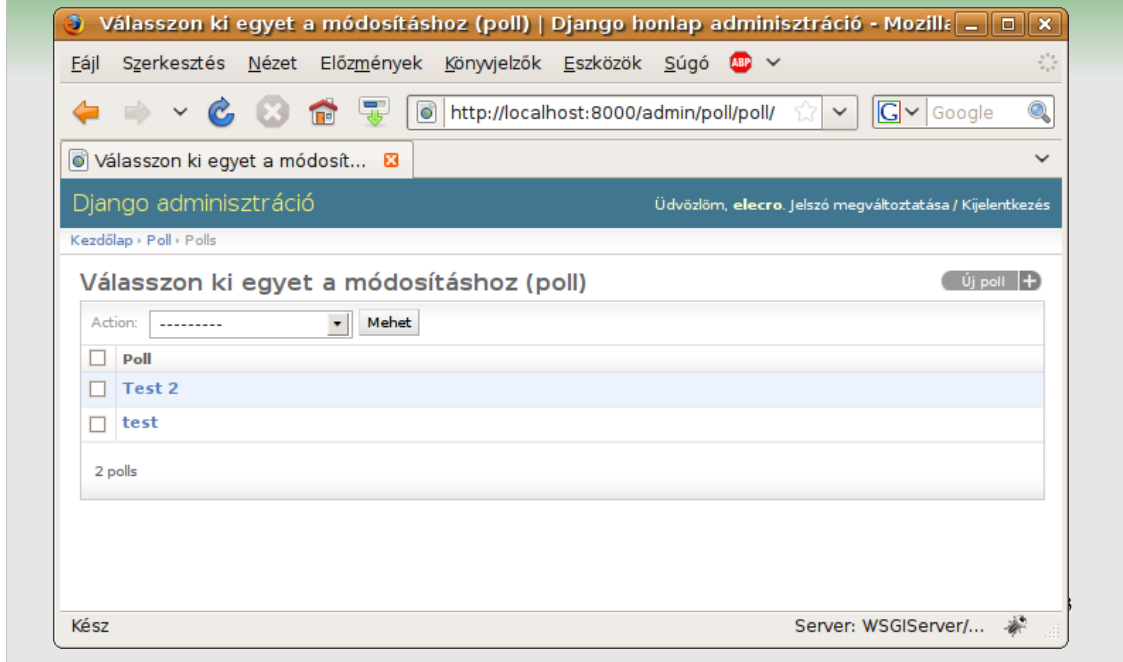
Beregisztráljuk a modelleket az admin felületre

Django – Admin



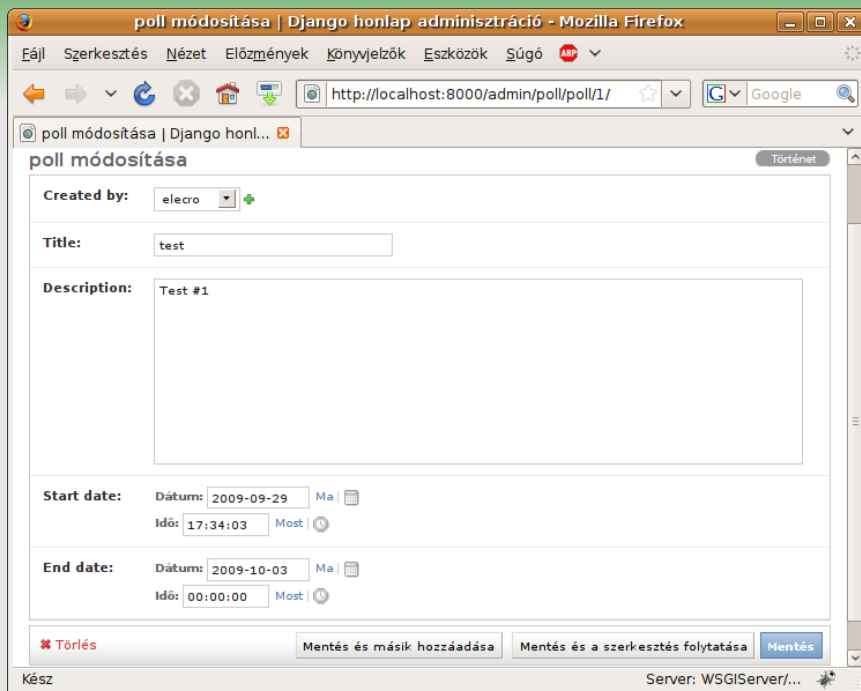
A megkapott admin felület, összes adott alkalmazásban beregisztrált modellel

Django – Admin



A szavazások listája

Django – Admin



The screenshot shows the Django Admin interface in a Mozilla Firefox browser window. The title bar reads "poll módosítása | Django honlap adminisztráció - Mozilla Firefox". The address bar shows "http://localhost:8000/admin/poll/poll/1/". The page title is "poll módosítása". The form contains the following fields:

- Created by:** A dropdown menu with "elecro" selected and a plus sign to add a new user.
- Title:** A text input field containing "test".
- Description:** A large text area containing "Test #1".
- Start date:** A date field set to "2009-09-29" and a time field set to "17:34:03".
- End date:** A date field set to "2009-10-03" and a time field set to "00:00:00".

At the bottom of the form, there are four buttons: "Törölés" (Delete), "Mentés és másik hozzáadása" (Save and add another), "Mentés és a szerkesztés folytatása" (Save and continue editing), and "Mentés" (Save). The status bar at the bottom left says "Kész" and the bottom right says "Server: WSGIServer/...".

24

Egy szavazás szerkesztése

Django – Admin v.2

```
class PollAdmin(admin.ModelAdmin):
    date_hierarchy = 'date_start'
    list_display = ['title', 'created_by', 'date_start',
                  'date_end']
    list_filter = ['created_by', 'date_start']
    search_fields = ['title']

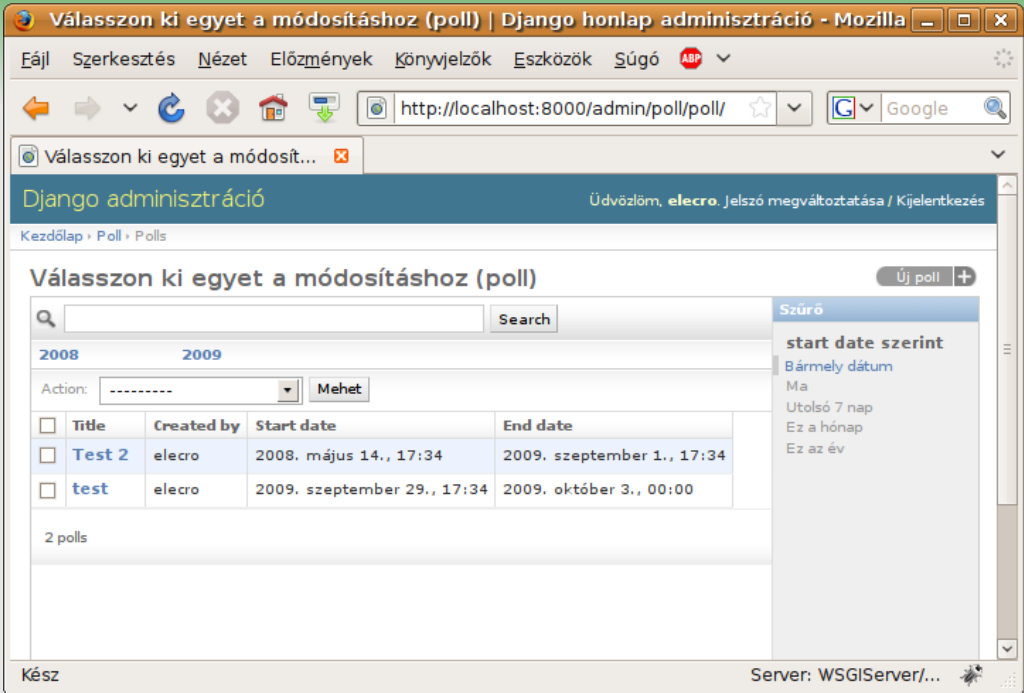
admin.site.register(Poll, PollAdmin)
```

25

Kis tuningolás:

- `date_hierarchy`: dátum szerinti bontás az adott mező felhasználva
- `list_display`: oszlopok melyek megjelenjenek a listában
- `list_filter`: szűrés lehetőségek
- `search_fields`: keresés az adott mezők között

Django – Admin v.2



The screenshot shows the Django Admin interface in a Mozilla browser window. The page title is "Válasszon ki egyet a módosításhoz (poll) | Django honlap adminisztráció - Mozilla". The browser address bar shows "http://localhost:8000/admin/poll/poll/". The page content includes a search bar, a table of polls, and a filter sidebar.

Válasszon ki egyet a módosításhoz (poll)

2008 2009

Action: ----- Mehet

<input type="checkbox"/>	Title	Created by	Start date	End date
<input type="checkbox"/>	Test 2	electro	2008. május 14., 17:34	2009. szeptember 1., 17:34
<input type="checkbox"/>	test	electro	2009. szeptember 29., 17:34	2009. október 3., 00:00

2 polls

Szűrő

- start date szerint
- Bármely dátum
- Ma
- Utolsó 7 nap
- Ez a hónap
- Ez az év

Kész Server: WSGIServer/...

eredmény

Django

Poll list

27

Feladat:

Listázzuk ki az összes elérhető szavazást

Django – Poll list – URL

```
$ cat poll/url.py
from django.conf.urls.defaults import patterns, url

urlpatterns = patterns('timetable.poll.views',
    url(r'^list/$', 'list_polls'),
)

pl.: http://localhost:8000/poll/list/
```

28

Url:

- Első paraméter: reguláris kifejezés
- Második param.: függvény neve (view), relatív a patterns első paraméteréhez képest

Django – Poll list – View

```
from django.template import Context, loader

def list_polls(request):
    polls = Poll.objects.all()
    template = loader.get_template('poll/list.html')
    context = Context({'polls': polls})
    return HttpResponse(template.render(context))
```

29

Első view-unk, yupp!!!.:

függvény első paramétere HttpRequest

összes poll lekérése

html template betöltése

context létrehozása

templatet és contextet legeneráljuk az oldalt

eredmény visszaküldése

Django – Poll list – View v.2

```
from django.shortcuts import render_to_response
```

```
def list_polls(request):
```

```
    polls = Poll.objects.all()
```

```
    return render_to_response('poll/list.html',  
                             {'polls': polls})
```

30

Előző egyszerűbben, 2 sorral kevesebb :)

`render_to_response`:

- Első param.: template utvonala
- Második param.: context dictionary

Django – Poll list – Template

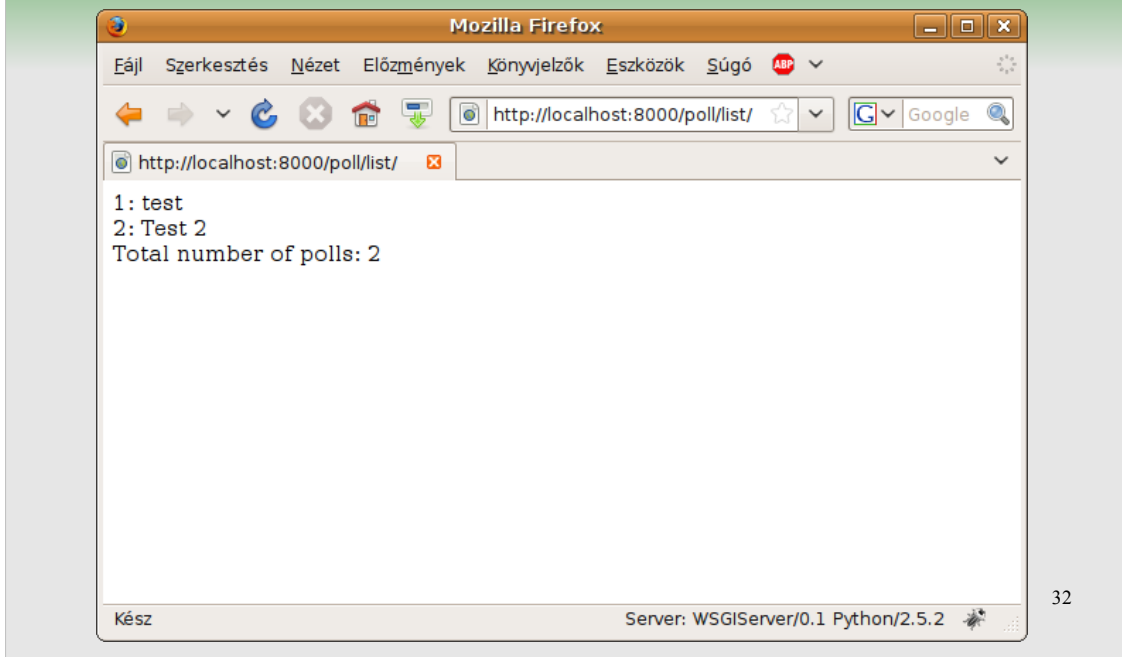
```
$ cat templates/poll/list.html
```

```
{% for poll in polls %}  
  {{ forloop.counter }}: {{ poll }}<br/>  
{% endfor %}  
Total number of polls: {{ polls|length }}
```

31

forloop.counter: for tag által biztosított számláló
polls|length: szavazások számát adja meg

Django – Poll list – Template



View és templet eredmény

Django

Poll info

33

Feladat:

Egy szavazás adatainak megjelenítése

Django – Poll info – URL

```
urlpatterns = patterns('timetable.poll.views',  
    ...  
    url(r'^info/(?P<poll_id>\d+)/$', 'get_poll',  
        name = 'poll_info'),  
    ...  
)
```

pl.: <http://localhost:8000/poll/info/1/>

34

Url-ben:

- Reguláris kifejezés: poll_id néven egy szám
- name: ezen a néven lehet majd később hivatkozni, nem kell url-el

Django – Poll info – View

```
def get_poll(request, poll_id):
    try:
        poll = Poll.objects.get(pk = poll_id)
    except Poll.DoesNotExist:
        raise Http404

    return render_to_response('poll/info.html',
                              {'poll': poll})
```

35

Egy szavazás lekérése elsődleges kulcs alapján,
amit paraméterként kapunk

Django – Poll info – View v.2

```
from django.shortcuts import get_object_or_404

def get_poll(request, poll_id):
    poll = get_object_or_404(Poll, pk = poll_id)

    return render_to_response('poll/info.html',
                              {'poll': poll})
```

36

Előző egyszerűbben, -3 sor

Django – Poll info – Template

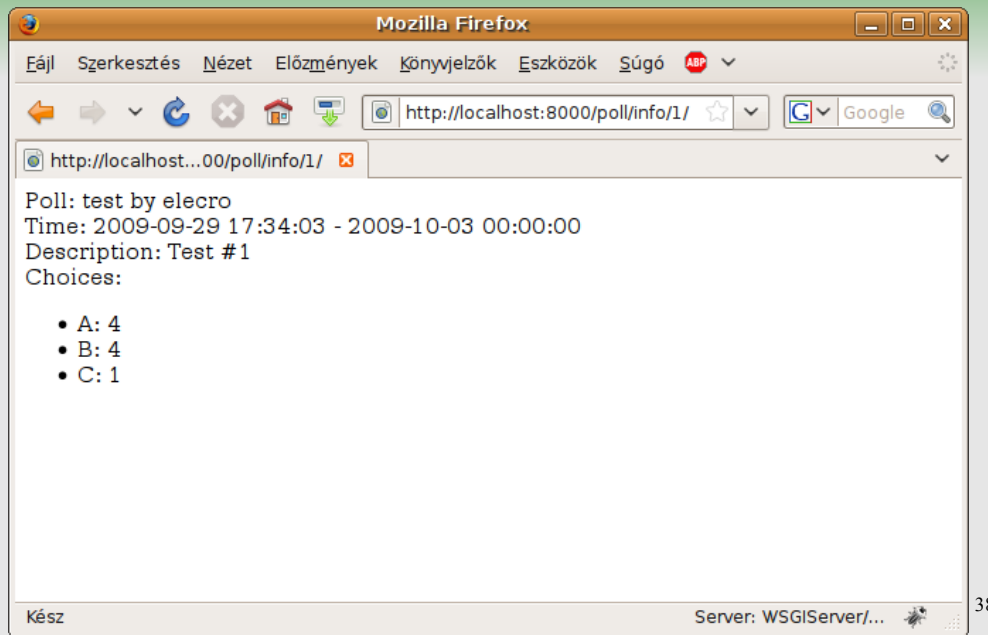
```
Poll: {{ poll.title }} by {{ poll.created_by }}<br/>
Time:{{ poll.date_start }} - {{ poll.date_end }}<br/>
Description: {{ poll.description }}<br/>
Choices:
<ul>
  {% for choice in poll.choice_set.all %}
  <li>{{ choice }}: {{ choice.vote_set.count }}</li>
  {% endfor %}
</ul>
```

37

Szavazás adatai.

Poll-hoz tartozó opciók lekérdezése és listázása,
valamint az egyes opciókhoz tartozó szavazatok
számlálása

Django – Poll info – Template



38

A legenerált template

Django

Poll vote

39

Legfontosabb rész. Szavazás form(űrlap) létrehozása

Django – Poll vote – Form

```
$ cat poll/forms.py
from django import forms
from poll.models import Vote

class VoteForm(forms.ModelForm):
    class Meta:
        model = Vote
```

40

Modellhez tartozó form, a modellben megadott mezőből generálja le majd a formot

Django – Poll vote – URL

```
urlpatterns = patterns('timetable.poll.views',
...
    url(r'^vote/(?P<poll_id>\d+)/$', 'vote_poll',
        name = 'poll_vote'),
...
)
```

pl.: <http://localhost:8000/poll/vote/1/>

41

Szavazás urlje, id-val hivatkozva

Django – Poll vote - View

```
def vote_poll(request, poll_id):
    poll = get_object_or_404(Poll, pk = poll_id)
    if request.method == "POST":
        ....
    else:
        form = VoteForm()
    return render_to_response('poll/vote.html',
                              {'voteform': form})
```

42

Szavazás view-a, 2db dia.

Ha nem post-al jutottunk az oldalra, létrehozunk a formot és átadjuk a templatenek

Django – Poll vote - View

```
def vote_poll(request, poll_id):
    poll = get_object_or_404(Poll, pk = poll_id)
    if request.method == "POST":
        form = VoteForm(request.POST)
        if form.is_valid():
            form.save()
            return HttpResponseRedirect(
                reverse('poll_info', kwargs = {'poll_id': poll.id}) )
```

43

.....

Ha POST-al jutottunk az oldalra, az adatokat ellenőrizzük a form-al, ha minden rendben elmentjük az adatbázisba és átirányítjuk az adott szavazás oldalára

Django – Poll vote – Template

```
<form method="POST">  
{{ voteform.as_p }}  
<input type="submit">  
</form>
```

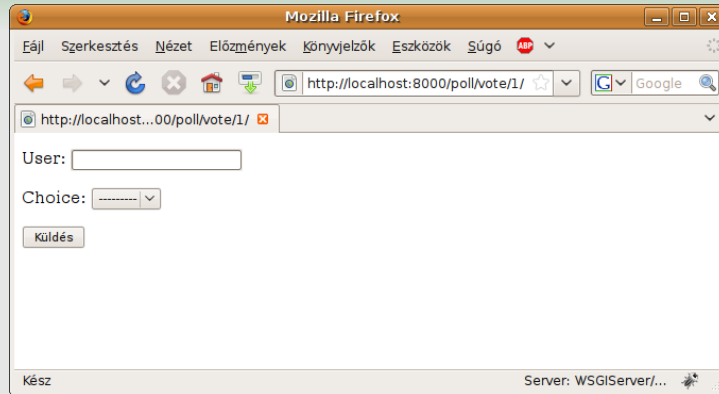
44

Form template

as_p: paragraph-ba generálja le

egyéb lehetőség: as_table (alapértelmezett)

Django – Poll vote – Template



The screenshot shows a Mozilla Firefox browser window with the address bar displaying `http://localhost:8000/poll/vote/1/`. The page content includes a form with the following elements:

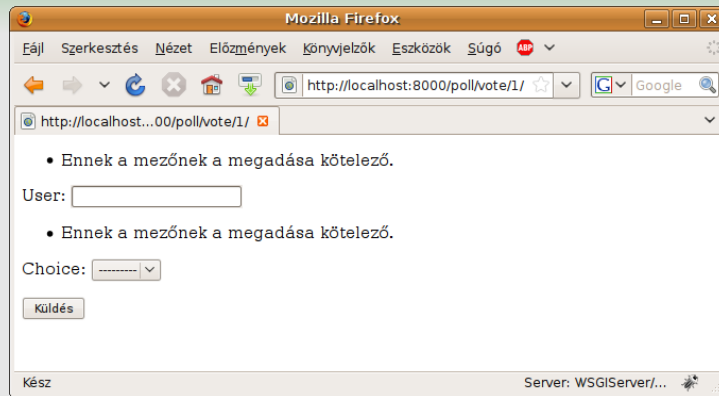
- A text input field labeled "User:".
- A dropdown menu labeled "Choice:".
- A button labeled "Küldés" (Submit).

The status bar at the bottom of the browser window shows "Kész" and "Server: WSGIServer/..."

45

És kész is a form

Django – Poll vote – Template



46

Hibás kitöltés esetén

Django – egyéb szolgáltatások

- auth
- admin
- flatpages
- humanize
- syndication
- ...

47

- Auth: autentikációs rész
- admin: automatikus
- flatpages: HTML-ek az adatbázisban
- humanize: template filterek gyűjtemények, barátságosabbá teszi az adatokat (fájlméret, számformázások, stb.)
- Syndication: Rss és Atomfeedek(hírfolyamok) létrehozása
- stb...

Köszönöm a figyelmet!

Gál Péter
galpeter@inf.u-szeged.hu