deti universidade de aveiro
departamento de electrónica,
telecomunicações e informática

Tecnologias e Programação Web 2019/2020

A Plataforma Django

Plataforma Django

Envio/Receção de Dados

Forms

Receção de Dados

- O objeto “request” do tipo “HttpRequest” permite aceder a um conjunto vasto de dados que são recebidos pelo *web server*
- Esses dados podem ser acedidos diretamente através de alguns atributos e métodos dedicados, como:
  - request.path, request.get_host(), request.is_secure()
- Ou podem ser acedidos através do dicionário “request.META” que contém toda a informação presente no *header* do protocolo HTTP

Receção de Dados

- Exemplo de uma *view* para mostrar todos os dados presentes no *header* HTTP

views.py

```
from django.shortcuts import render, render_to_response
from django.http import HttpResponse

def info(request):
    values = request.META.items()
    html = []
    for k, v in values:
        html.append('<tr><td>%s</td><td>%s</td></tr>' % (k, v))
    return HttpResponse('<table>%s</table>' % '\n'.join(html))
```

Forms

- Os *Forms* são os elementos HTML por excelência para o envio/receção de dados do cliente para o servidor
- Do lado browser (cliente) o *Form* pode usar o método *Get* ou o método *Post*, para enviar os dados nele contidos
- Do lado do servidor, a *view* invocada pode recorrer aos dicionários “request.GET” e “request.POST” para aceder aos dados recebidos

Forms - exemplo

- Exemplo da criação de um *Form* para pesquisa pelos títulos dos livros no modelo de dados anteriormente criado
- Definição da URL:

urls.py

```
17  from django.contrib import admin
18  from django.urls import path
19
20  from app import views
21
22  urlpatterns = [
23      path('booksearch/', views.booksearch, name='booksearch'),
24      path('insauthor/', views.authorins, name='authorins'),
25
```

Forms - exemplo

- Definição da *view*:

views.py

```
from app.models import Author, Publisher, Book


def booksearch(request):
    if 'query' in request.POST:
        query = request.POST['query']
        if query:
            books = Book.objects.filter(title__icontains=query)
            return render(request, 'booklist.html',{'boks':books, 'query':query})
        else:
            return render(request, 'booksearch.html', {'error':True})
    else:
        return render(request, 'booksearch.html', {'error':False})
```

Forms - exemplo

- Definição da *template* para pesquisa:

booksearch.html

```
{% extends "layout.html" %}

{% block content %}

{% if error %}
    <p style="...">ERROR: Insert a query term.</p>
{% endif %}

<form action="." method="post">
    {% csrf_token %}
    <input type="text" name="query">
    <input type="submit" value="Search">
</form>

{% endblock %}
```

Forms - exemplo

- Definição da *template* para os resultados:

booklist.html

```
{% extends "layout.html" %}
{% block content %}
<p>Search by: <strong>{{ query }}</strong></p>
{% if boks %}
<p>Found {{ boks|length }} book{{ boks|pluralize }}.</p>
<ul>
    {% for bok in boks %}
        <li>{{ bok.title }}</li>
        <ul>
            <li>{{ bok.publisher }}</li>
            <li>{{ bok.date }}</li>
            <ul>
            {% for aut in bok.authors.all %}
                <li>{{ aut }}</li>
            {% endfor %}
            </ul>
        </ul>
    {% endfor %}
</ul>
{% else %}
    <p>Not found any result.</p>
{% endif %}
{% endblock %}
```