

PROGRAMAÇÃO BACK-END

The Django logo, featuring the word "django" in a white, lowercase, sans-serif font with a small dot over the 'i', set against a dark green rectangular background.

The web framework for
perfectionists with deadlines.

Django – Usando Ajax

PROGRAMAÇÃO BACK-END

Usando Ajax

Nesta apresentação será apresentado o uso de AJAX em aplicação do Django.

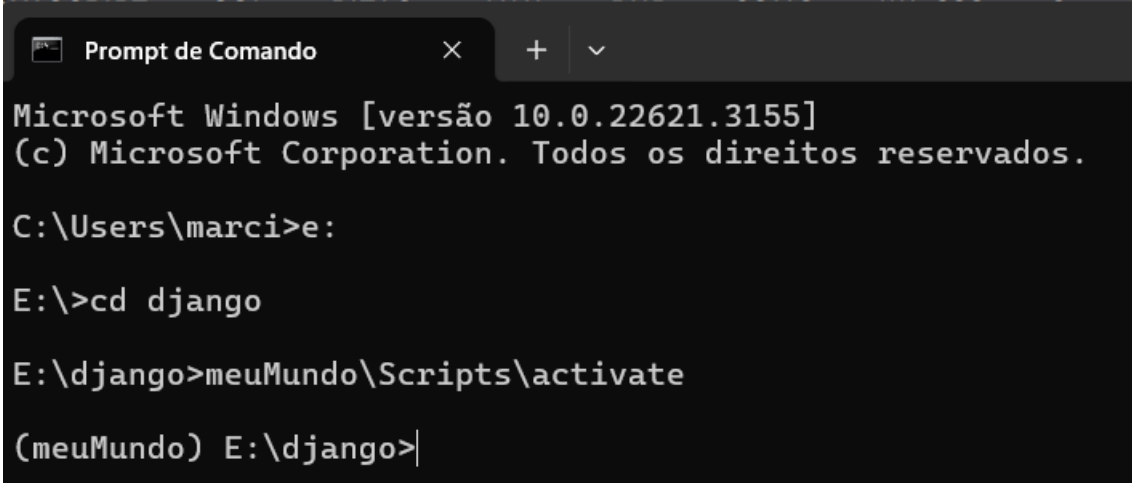
AJAX (Asynchronous JavaScript and XML) é usado no HTML para permitir que páginas web se comuniquem com o servidor sem recarregar a página inteira. Isso melhora a experiência do usuário, tornando a navegação mais rápida e interativa.

PROGRAMAÇÃO BACK-END

Iniciando

PROGRAMAÇÃO BACK-END

Rodar o script activate.bat



```
Prompt de Comando
Microsoft Windows [versão 10.0.22621.3155]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Users\marci>e:

E:\>cd django

E:\django>meuMundo\Scripts\activate

(meuMundo) E:\django>
```


Este passo é importante para configurar as variáveis de ambiente.

PROGRAMAÇÃO BACK-END

Iniciar Servidor Postgresql

PROGRAMAÇÃO BACK-END

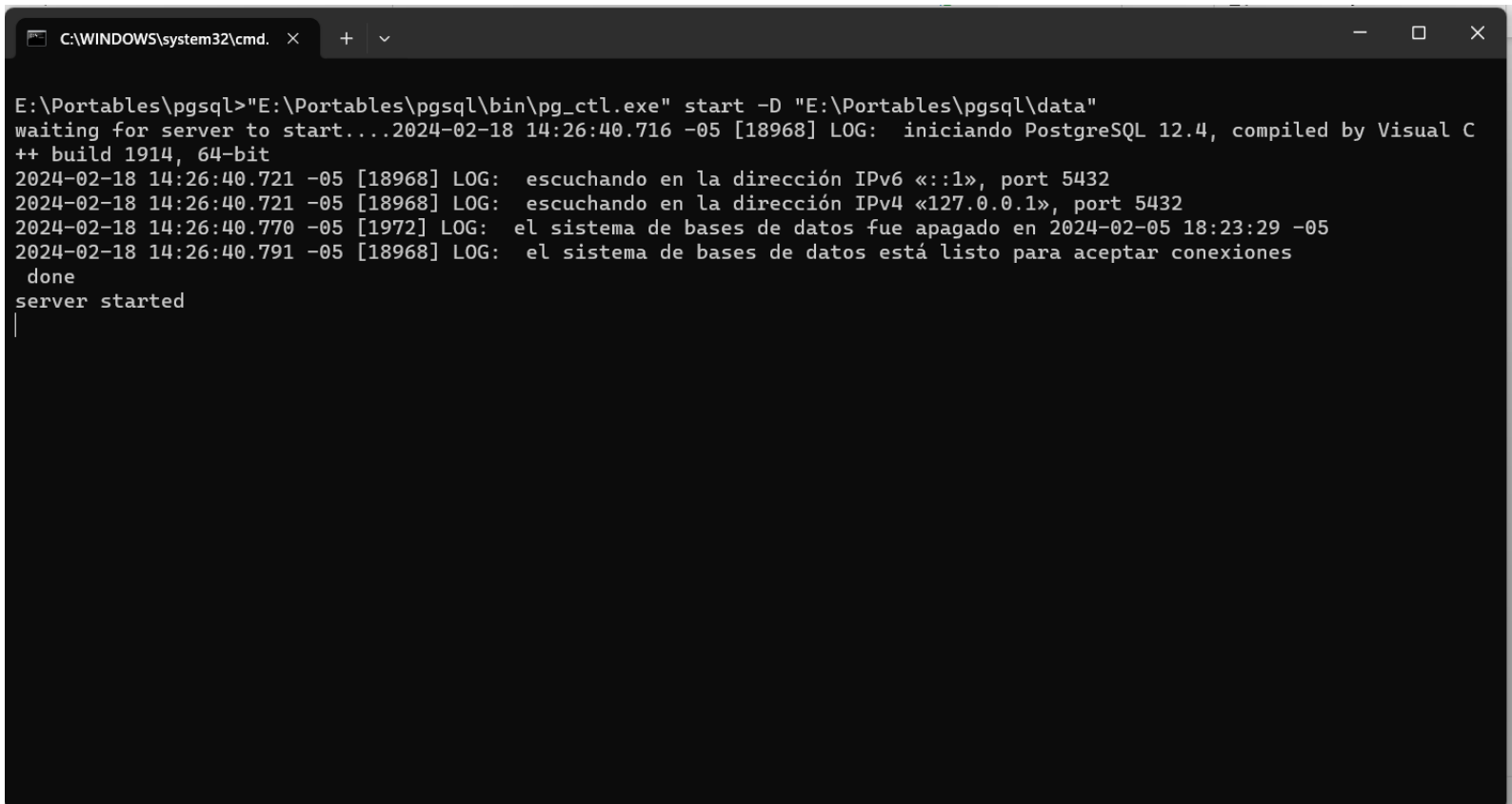
Na pasta `../pgsql` execute o arquivo **PostgreSQL-Start.bat** para iniciar o servidor



Nome	Data de modificação	Tipo	Tamanho
bin	11/08/2020 08:41	Pasta de arquivos	
data	18/02/2024 16:26	Pasta de arquivos	
doc	11/08/2020 08:41	Pasta de arquivos	
include	11/08/2020 08:41	Pasta de arquivos	
lib	11/08/2020 08:41	Pasta de arquivos	
pgAdmin 4	11/08/2020 08:41	Pasta de arquivos	
share	11/08/2020 08:41	Pasta de arquivos	
StackBuilder	11/08/2020 08:41	Pasta de arquivos	
Make Cluster.bat	09/01/2017 14:12	Arquivo em Lotes ...	1 KB
PgAdmin4.bat	28/04/2017 22:29	Arquivo em Lotes ...	1 KB
PostgreSQL-Restart.bat	06/04/2012 09:42	Arquivo em Lotes ...	1 KB
PostgreSQL-Start.bat	28/04/2017 21:46	Arquivo em Lotes ...	1 KB
PostgreSQL-Stop.bat	06/04/2012 09:43	Arquivo em Lotes ...	1 KB
Service - Delete.bat	29/08/2020 23:25	Arquivo em Lotes ...	1 KB
Service - Maker.bat	29/08/2020 23:26	Arquivo em Lotes ...	1 KB

PROGRAMAÇÃO BACK-END

Será aberta uma janela em Texto informando que o servidor iniciou (**server started**).
Não feche esta janela



```
C:\WINDOWS\system32\cmd. x + v
E:\Portables\pgsql>"E:\Portables\pgsql\bin\pg_ctl.exe" start -D "E:\Portables\pgsql\data"
waiting for server to start...2024-02-18 14:26:40.716 -05 [18968] LOG:  iniciando PostgreSQL 12.4, compiled by Visual C
++ build 1914, 64-bit
2024-02-18 14:26:40.721 -05 [18968] LOG:  escuchando en la dirección IPv6 «::1», port 5432
2024-02-18 14:26:40.721 -05 [18968] LOG:  escuchando en la dirección IPv4 «127.0.0.1», port 5432
2024-02-18 14:26:40.770 -05 [1972] LOG:  el sistema de bases de datos fue apagado en 2024-02-05 18:23:29 -05
2024-02-18 14:26:40.791 -05 [18968] LOG:  el sistema de bases de datos está listo para aceptar conexiones
done
server started
|
```


Templates

PROGRAMAÇÃO BACK-END

Utilizando a App **dvdrentalapp**. Na pasta **templates** altere o arquivo **all_customers.html** digite o seguinte conteúdo:

```
dvdrentalapp > templates > <> all_customers.html > ...
1  {% load static %}
2  <!DOCTYPE html>
3  <html>
4  <head>
5      <title>Customers</title>
6      <!-- Bootstrap CSS -->
7      <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
8      <!-- Custom CSS -->
9      <link rel="stylesheet" href="{% static 'styles.css' %}">
10 </head>
11 <body>
12     <h1>Customers</h1>
13     <ul>
14         {% for x in mycustomer %}
15             <li>
16                 {{ x.customer_id }}
17                 <a href="#" class="customer-link" data-customer-id="{{ x.customer_id }}">
18                     {{ x.first_name }} {{ x.last_name }}
19                 </a>
20             </li>
21         {% endfor %}
22     </ul>
```

PROGRAMAÇÃO BACK-END

Continuação de all_customers.html

```
23
24     <!-- Modal -->
25     <div class="modal fade" id="rentalModal" tabindex="-1" aria-labelledby="rentalModalLabel" aria-hidden="true">
26         <div class="modal-dialog modal-lg">
27             <div class="modal-content">
28                 <div class="modal-header">
29                     <h5 class="modal-title" id="rentalModalLabel">Rental Details</h5>
30                     <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
31                 </div>
32                 <div class="modal-body" id="modalContent">
33                     <!-- Conteúdo será carregado via AJAX -->
34                 </div>
35                 <div class="modal-footer">
36                     <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Close</button>
37                 </div>
38             </div>
39         </div>
40     </div>
41
42     <!-- Bootstrap JS and Popper.js -->
43     <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
44     <!-- jQuery -->
45     <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
46     <!-- Custom JS -->
47     <script src="{% static 'customer.js' %}"></script>
48 </body>
49 </html>
```

PROGRAMAÇÃO BACK-END

e rental_modal_content.html

```
dvdrentalapp > templates > <> rental_modal_content.html > table.table.table-bordered
1  <h2>Detalhes para {{ customer_name }}</h2>
2  <table class="table table-bordered">
3      <thead>
4          <tr>
5              <th>rental_id</th>
6              <th>rental_date</th>
7              <th>inventory_id</th>
8          </tr>
9      </thead>
10     <tbody>
11         {% for x in myDetalhe %}
12             <tr>
13                 <td><a href="/payment/{{ x.rental_id }}">{{ x.rental_id }}</a></td>
14                 <td>{{ x.rental_date }}</td>
15                 <td>{{ x.inventory_id }}</td>
16             </tr>
17         {% endfor %}
18     </tbody>
19 </table>
```

PROGRAMAÇÃO BACK-END

e payment.html

```
dvdrentalapp > templates > <> payment.html > html > body > table >
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Payments</title>
6  </head>
7  <body>
8      <h1>Payment Details</h1>
9
10     <p>Back to <a href="/customer">Customers</a></p>
11
12     <table border="1">
13         <thead>
14             <tr>
15                 <th>payment_id</th>
16                 <th>Amount</th>
17                 <th>Payment Date</th>
18             </tr>
19         </thead>
20         <tbody>
21             {% for x in myPayment %}
22                 <tr>
23                     <td>{{ x.payment_id }}</td>
24                     <td>{{ x.amount }}</td>
25                     <td>{{ x.payment_date }}</td>
26                 </tr>
27             {% endfor %}
28         </tbody>
29     </table>
30 </body>
31 </html>
32
```

PROGRAMAÇÃO BACK-END

Arquivos `views.py` e `urls.py`

PROGRAMAÇÃO BACK-END

Na pasta **dvdrentalapp** adicione o seguinte conteúdo no arquivo **views.py**

```
dvdrentalapp > 📁 views.py > ...
1  from django.shortcuts import render, get_object_or_404, redirect
2  # Create your views here.
3  from django.http import HttpResponse
4  from django.template import loader
5  from .models import Customer, Rental, Payment, Film, FilmActor, Actor
6  from django.http import JsonResponse
7
8  def customer(request):
9      mycustomers = Customer.objects.all().values()
10     template = loader.get_template('all_customers.html')
11     context = {
12         'mycustomer': mycustomers,
13     }
14     return HttpResponse(template.render(context, request))
15
```

PROGRAMAÇÃO BACK-END

Continuação do `views.py`

```
16 def get_rental_details(request, id):
17     myDetalhes = Rental.objects.filter(customer_id=id)
18     customer = get_object_or_404(Customer, pk=id)
19     template = loader.get_template('rental_modal_content.html')
20     context = {
21         'myDetalhe': myDetalhes,
22         'customer_name': f"{customer.first_name} {customer.last_name}",
23     }
24     rendered_content = template.render(context, request)
25     return JsonResponse({'html': rendered_content})
26
27 def payment(request, id):
28     myDetalhes = Payment.objects.filter(rental_id=id)
29     template = loader.get_template('payment.html')
30     context = {
31         'myPayment' : myDetalhes,
32     }
33     return HttpResponse(template.render(context, request))
```


PROGRAMAÇÃO BACK-END

Na pasta **dvdrentalapp** adicione o seguinte conteúdo no arquivo **urls.py**

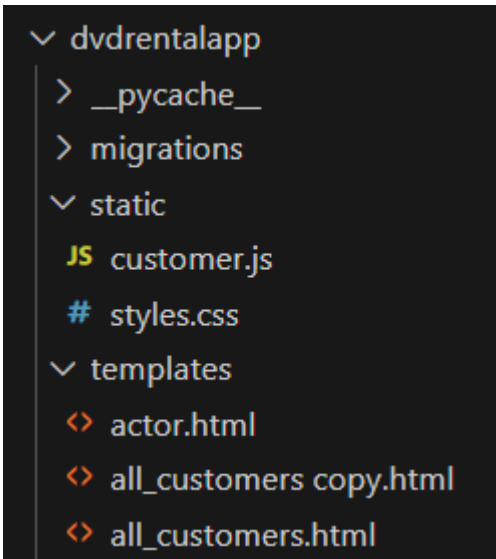
```
dvdrentalapp > 📄 urls.py > ...  
1  from django.urls import path  
2  from . import views  
3  
4  urlpatterns = [  
5      path('customer/', views.customer, name="customer"),  
6      path('get_rental_details/<int:id>', views.get_rental_details, name='get_rental_details'),  
7      path('payment/<int:id>', views.payment, name= 'payment' ),  
8  ]
```

PROGRAMAÇÃO BACK-END

Arquivos customer.js e styles.css

PROGRAMAÇÃO BACK-END

Na pasta **static** em **dvdrentalapp** crie e adicione o seguinte conteúdo no arquivo **customer.js**.

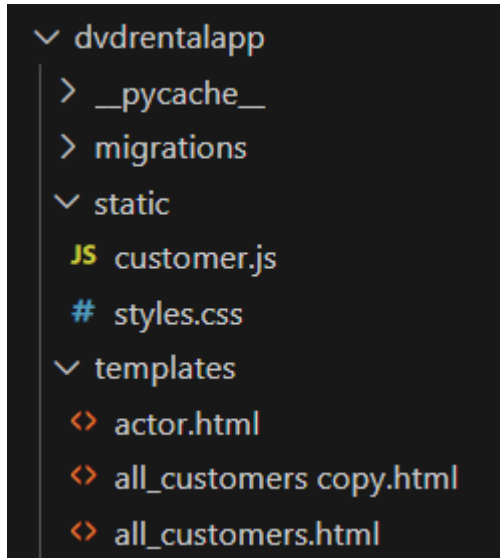


```

dvdrentalapp > static > JS customer.js > ...
1  $(document).ready(function() {
2      $('.customer-link').click(function(e) {
3          e.preventDefault();
4          var customerId = $(this).data('customer-id');
5
6          $.ajax({
7              url: '/get_rental_details/' + customerId,
8              type: 'GET',
9              success: function(data) {
10                  $('#modalContent').html(data.html);
11                  $('#rentalModal').modal('show');
12              },
13              error: function(xhr, status, error) {
14                  console.error('Error:', error);
15                  alert('Failed to load rental details.');
```

PROGRAMAÇÃO BACK-END

Na pasta **static** em **dvdrentalapp** crie e adicione o seguinte conteúdo no arquivo **styles.css**.



```

dvdrentalapp > static > # styles.css > .table
1  body {
2    |   padding: 20px;
3  }
4
5  h1 {
6    |   margin-bottom: 20px;
7  }
8
9  ul {
10   |   list-style-type: none;
11   |   padding: 0;
12  }
13
14  li {
15   |   margin: 10px 0;
16  }
17
18  .table {
19   |   margin-top: 20px;
20  }

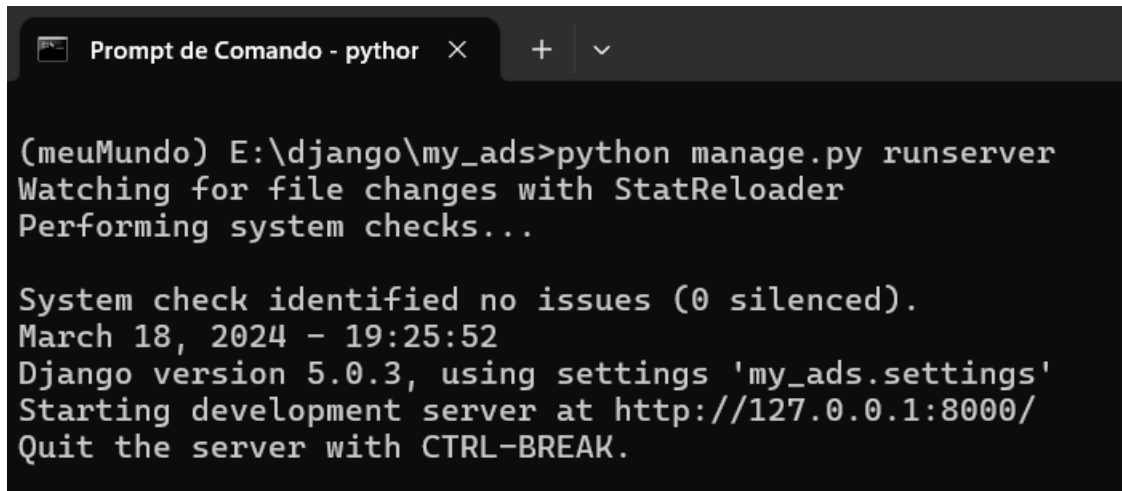
```

**Iniciando o servidor
e testando**

PROGRAMAÇÃO BACK-END

Execute na pasta **my_ads** o comando:

python manage.py runserver



```
Prompt de Comando - pythor  X  +  v

(meuMundo) E:\django\my_ads>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

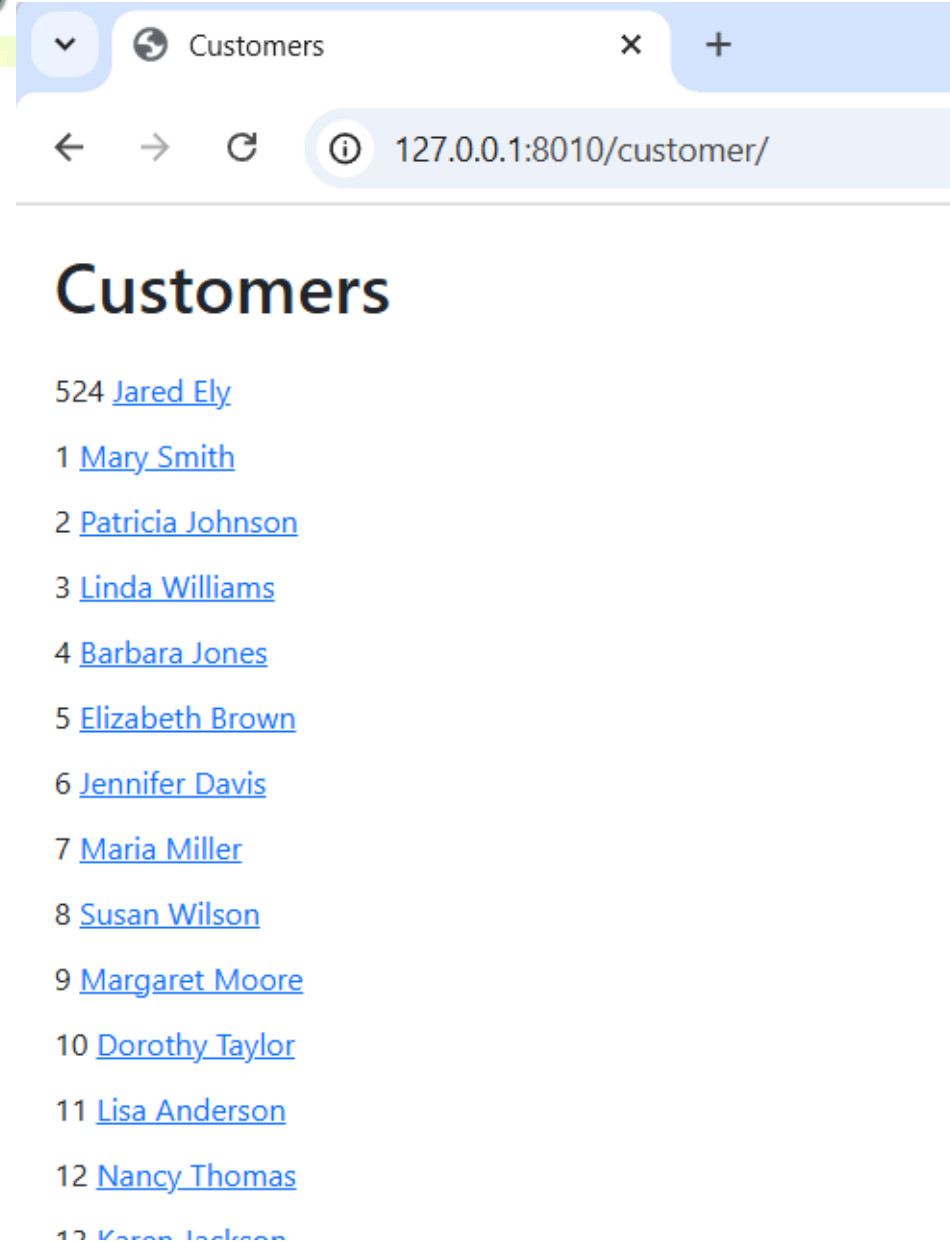
System check identified no issues (0 silenced).
March 18, 2024 - 19:25:52
Django version 5.0.3, using settings 'my_ads.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Vá ao navegador e digite:

<http://127.0.0.1:8000/customer/>

PROGRAMAÇÃO BACK-END

O resultado da página customer.



PROGRAMAÇÃO BACK-END

Selecione um **customer** será apresentado um **modal Rental Details**.

Customers

127.0.0.1:8010/customer/

Customers

524 [Jared Ely](#)

1 [Mary Smith](#)

2 [Patricia Johnson](#)

3 [Linda Williams](#)

4 [Barbara Jones](#)

5 [Elizabeth Brown](#)

6 [Jennifer Davis](#)

7 [Maria Miller](#)

8 [Susan Wilson](#)

9 [Margaret Moore](#)

10 [Dorothy Taylor](#)

11 [Lisa Anderson](#)

12 [Nancy Thomas](#)

13 [Karen Jackson](#)

14 [Betty White](#)

15 [Helen Harris](#)

16 [Sandra Martin](#)

17 [Donna Thompson](#)

18 [Carol Garcia](#)

19 [Ruth Martinez](#)

Rental Details

Detalhes para Mary Smith

rental_id	rental_date	inventory_id
76	May 25, 2005, 11:30 a.m.	3021
573	May 28, 2005, 10:35 a.m.	4020
1185	June 15, 2005, 12:54 a.m.	2785
1422	June 15, 2005, 6:02 p.m.	1021
1476	June 15, 2005, 9:08 p.m.	1407
1725	June 16, 2005, 3:18 p.m.	726
2308	June 18, 2005, 8:41 a.m.	197
2363	June 18, 2005, 1:33 p.m.	3497
3284	June 21, 2005, 6:24 a.m.	4566
4526	July 8, 2005, 3:17 a.m.	1443
4611	July 8, 2005, 7:33 a.m.	3486
5244	July 9, 2005, 1:24 p.m.	3726
5326	July 9, 2005, 4:38 p.m.	797
6163	July 11, 2005, 10:13 a.m.	1330