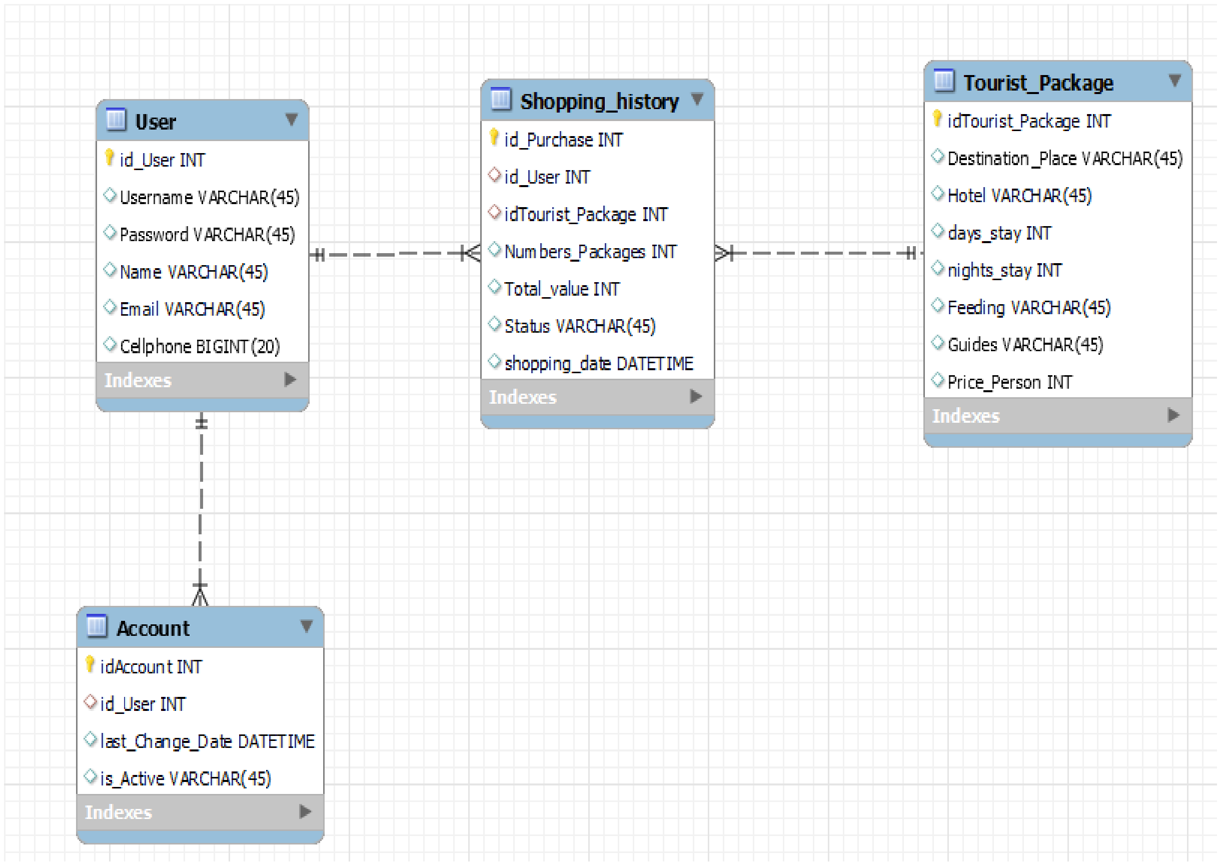
SPRINT #2

Modelo Relacional, sólo para basarnos (no tener en cuenta las relaciones entre modelos).



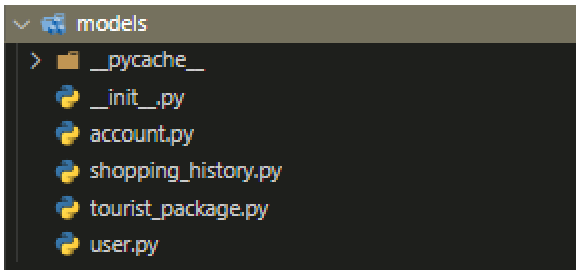
Funcionalidades del microservicio:

Sign Up-Account:
Tienen como función todo lo referente al proceso de registro y autenticación, basado en el JWT Token.

Shopping History:
Este cumple dos tareas, la primera es mantener el registro de compras realizadas por todos los usuarios por lo que tiene como llave foránea el id del usuario y por otro lado el id del paquete que compro. La segunda tarea que realiza gracias a una vista es filtrar el historial de compras de un usuario en particular.

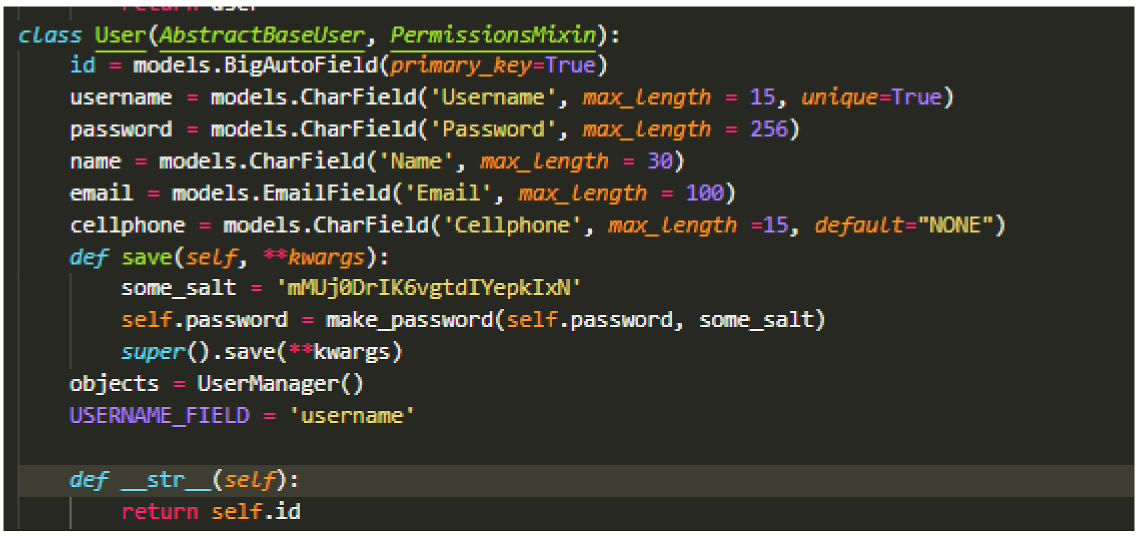
Tourist Package:
La función de este modelo es poder mantener una especie de inventario con los paquetes que ofrece la empresa. Cada paquete contiene su información correspondiente.

Modelos:



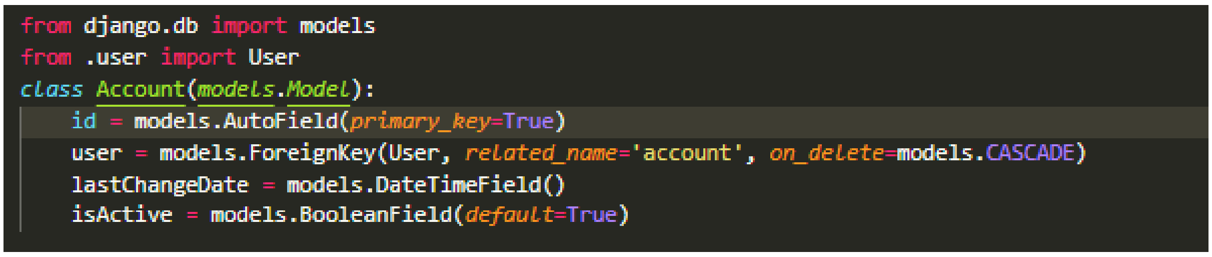
```
models
  > __pycache__
    __init__.py
    account.py
    shopping_history.py
    tourist_package.py
    user.py
```

User:



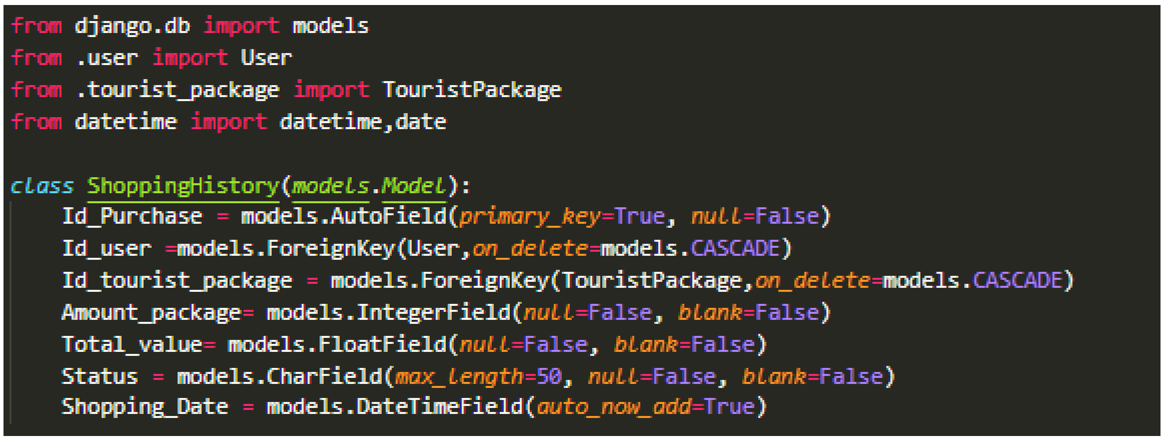
```python
class User(AbstractBaseUser, PermissionsMixin):
    id = models.BigAutoField(primary_key=True)
    username = models.CharField('Username', max_length = 15, unique=True)
    password = models.CharField('Password', max_length = 256)
    name = models.CharField('Name', max_length = 30)
    email = models.EmailField('Email', max_length = 100)
    cellphone = models.CharField('Cellphone', max_length =15, default="NONE")
    def save(self, **kwargs):
        some_salt = 'mMUj0DrIK6vgtdIYepkIxN'
        self.password = make_password(self.password, some_salt)
        super().save(**kwargs)
    objects = UserManager()
    USERNAME_FIELD = 'username'

    def __str__(self):
        return self.id
```

Account:



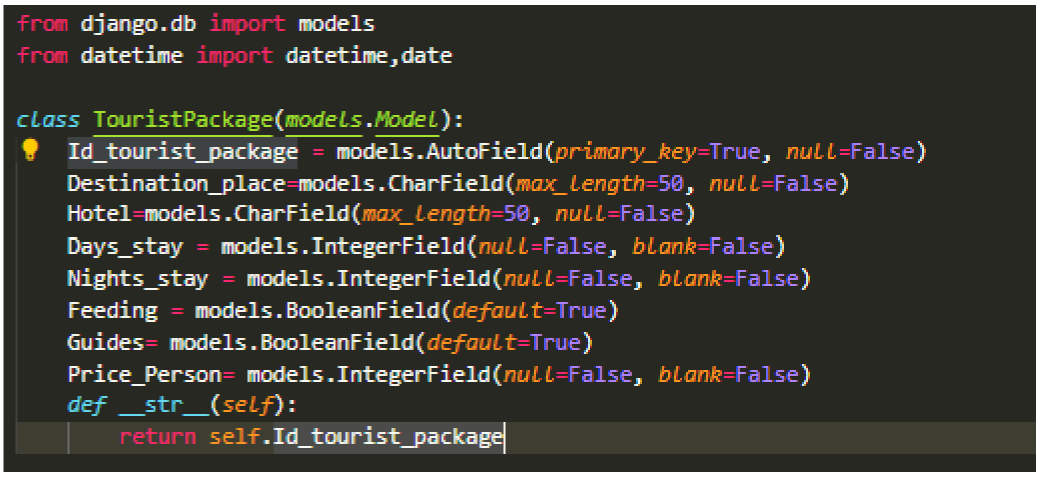
```python
from django.db import models
from .user import User
class Account(models.Model):
    id = models.AutoField(primary_key=True)
    user = models.ForeignKey(User, related_name='account', on_delete=models.CASCADE)
    lastChangeDate = models.DateTimeField()
    isActive = models.BooleanField(default=True)
```

Shopping_history:



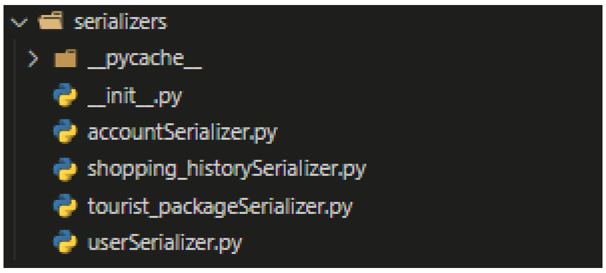
```python
from django.db import models
from .user import User
from .tourist_package import TouristPackage
from datetime import datetime,date

class ShoppingHistory(models.Model):
    Id_Purchase = models.AutoField(primary_key=True, null=False)
    Id_user =models.ForeignKey(User,on_delete=models.CASCADE)
    Id_tourist_package = models.ForeignKey(TouristPackage,on_delete=models.CASCADE)
    Amount_package= models.IntegerField(null=False, blank=False)
    Total_value= models.FloatField(null=False, blank=False)
    Status = models.CharField(max_length=50, null=False, blank=False)
    Shopping_Date = models.DateTimeField(auto_now_add=True)
```

Tourist_package:



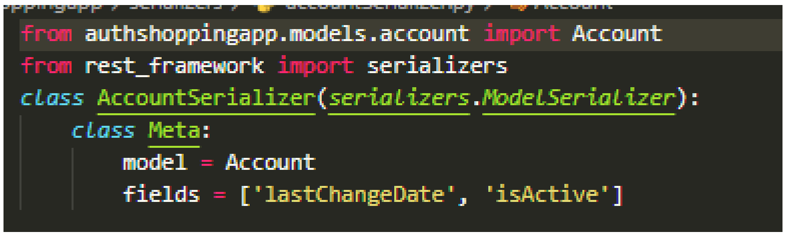
```python
from django.db import models
from datetime import datetime,date

class TouristPackage(models.Model):
    Id_tourist_package = models.AutoField(primary_key=True, null=False)
    Destination_place=models.CharField(max_length=50, null=False)
    Hotel=models.CharField(max_length=50, null=False)
    Days_stay = models.IntegerField(null=False, blank=False)
    Nights_stay = models.IntegerField(null=False, blank=False)
    Feeding = models.BooleanField(default=True)
    Guides= models.BooleanField(default=True)
    Price_Person= models.IntegerField(null=False, blank=False)
    def __str__(self):
        return self.Id_tourist_package
```

Serializers:



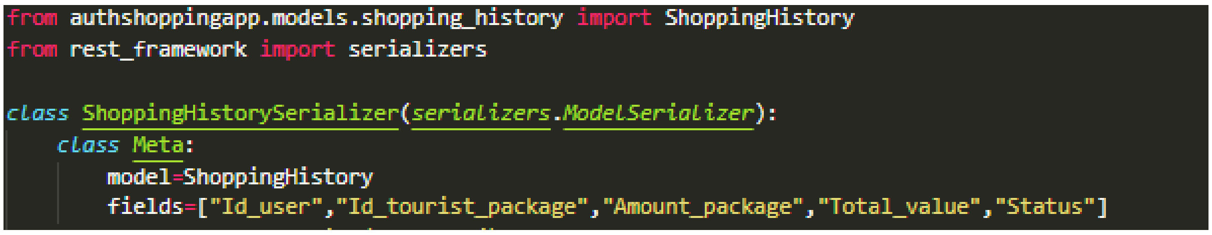
```
serializers
  > _pycache_
    _init_.py
    accountSerializer.py
    shopping_historySerializer.py
    tourist_packageSerializer.py
    userSerializer.py
```

AccountSerializer:



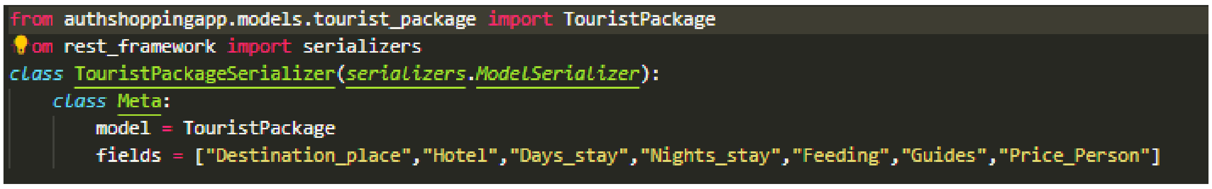
```python
from authshoppingapp.models.account import Account
from rest_framework import serializers
class AccountSerializer(serializers.ModelSerializer):
    class Meta:
        model = Account
        fields = ['lastChangeDate', 'isActive']
```

Shopping_historySerializer:



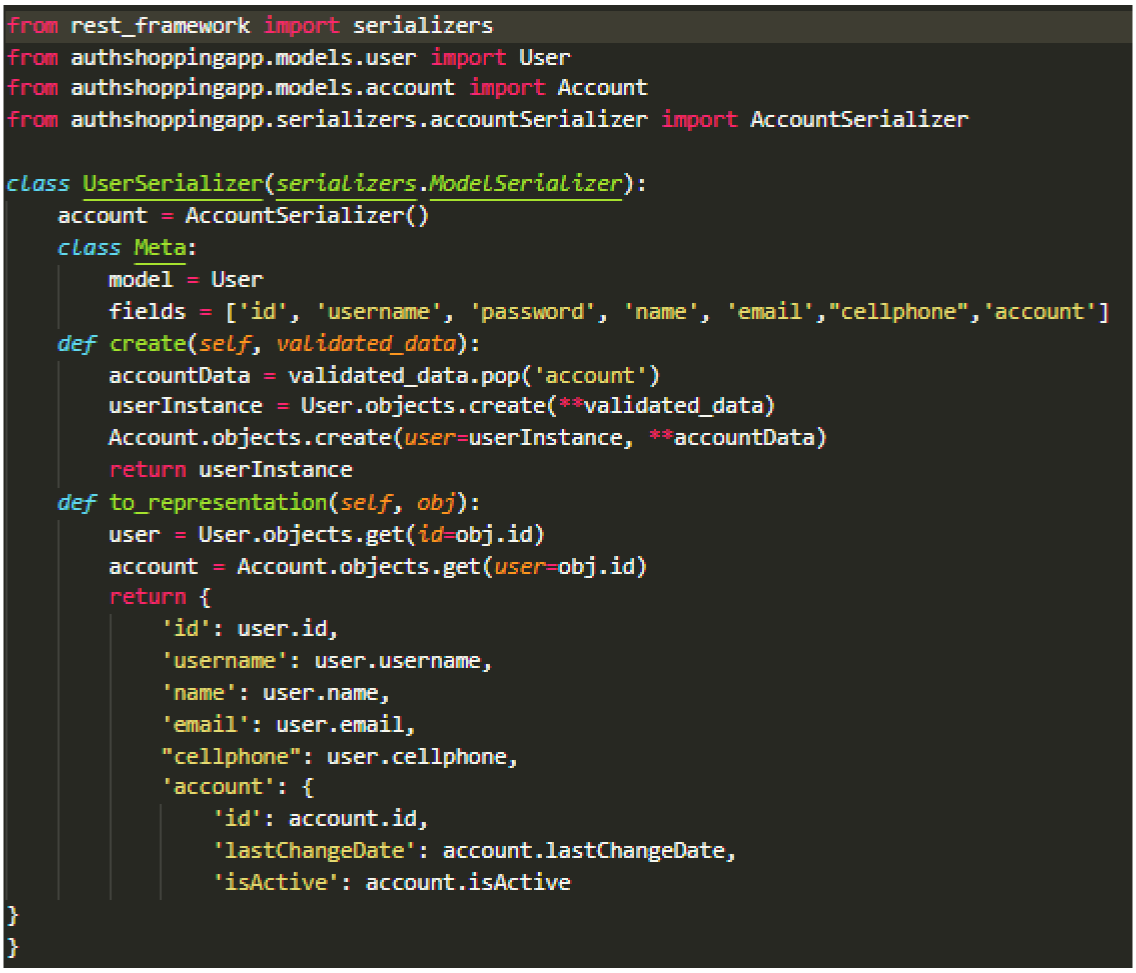
```python
from authshoppingapp.models.shopping_history import ShoppingHistory
from rest_framework import serializers

class ShoppingHistorySerializer(serializers.ModelSerializer):
    class Meta:
        model=ShoppingHistory
        fields=["Id_user","Id_tourist_package","Amount_package","Total_value","Status"]
```

Tourist_packageSerializer:



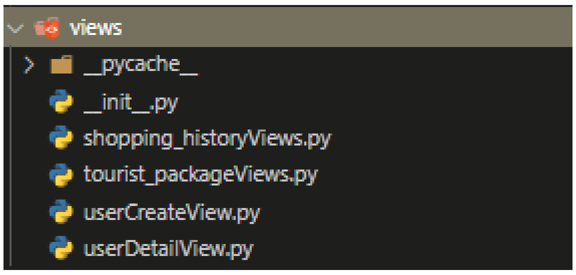
```python
from authshoppingapp.models.tourist_package import TouristPackage
om rest_framework import serializers
class TouristPackageSerializer(serializers.ModelSerializer):
    class Meta:
        model = TouristPackage
        fields = ["Destination_place","Hotel","Days_stay","Nights_stay","Feeding","Guides","Price_Person"]
```

UserSerializer:



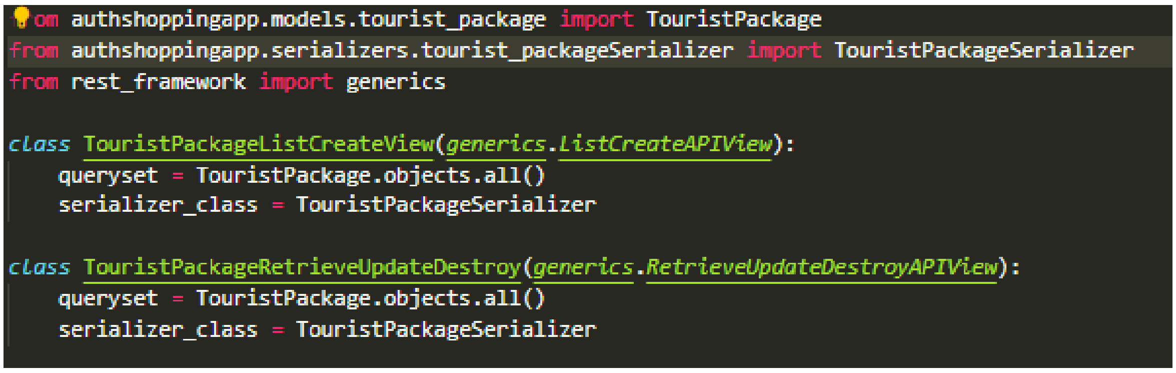
```
from rest_framework import serializers
from authshoppingapp.models.user import User
from authshoppingapp.models.account import Account
from authshoppingapp.serializers.accountSerializer import AccountSerializer

class UserSerializer(serializers.ModelSerializer):
    account = AccountSerializer()
    class Meta:
        model = User
        fields = ['id', 'username', 'password', 'name', 'email',"cellphone",'account']
    def create(self, validated_data):
        accountData = validated_data.pop('account')
        userInstance = User.objects.create(**validated_data)
        Account.objects.create(user=userInstance, **accountData)
        return userInstance
    def to_representation(self, obj):
        user = User.objects.get(id=obj.id)
        account = Account.objects.get(user=obj.id)
        return {
            'id': user.id,
            'username': user.username,
            'name': user.name,
            'email': user.email,
            "cellphone": user.cellphone,
            'account': {
                'id': account.id,
                'lastChangeDate': account.lastChangeDate,
                'isActive': account.isActive
}
}
```

Vistas:



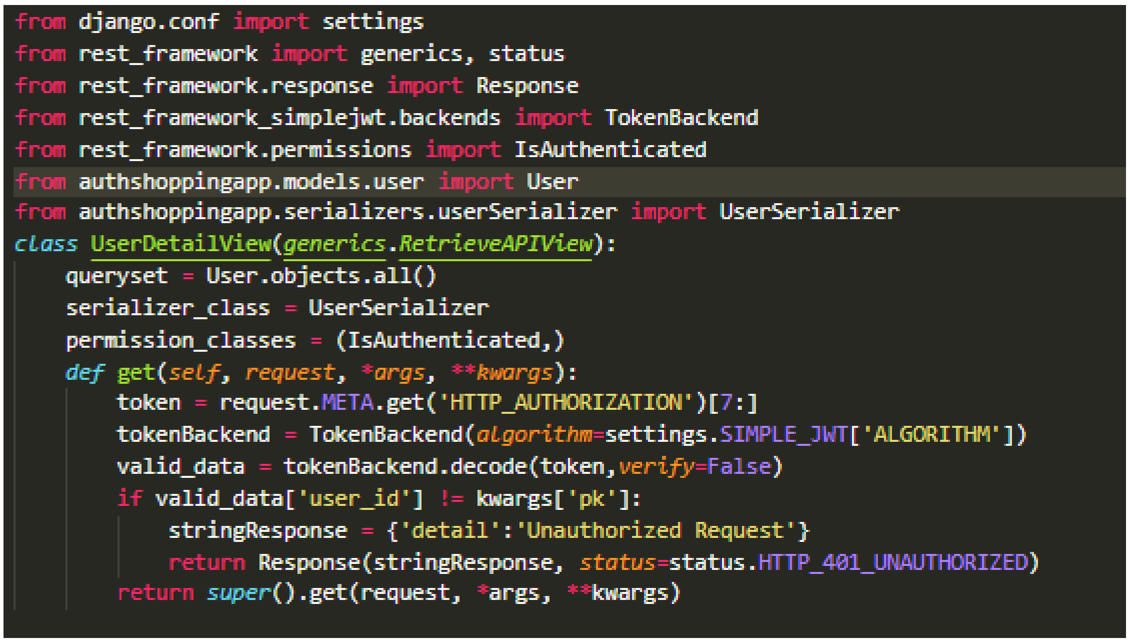
```
views
  __pycache__
  __init__.py
  shopping_historyViews.py
  tourist_packageViews.py
  userCreateView.py
  userDetailView.py
```

Tourist_packageViews:



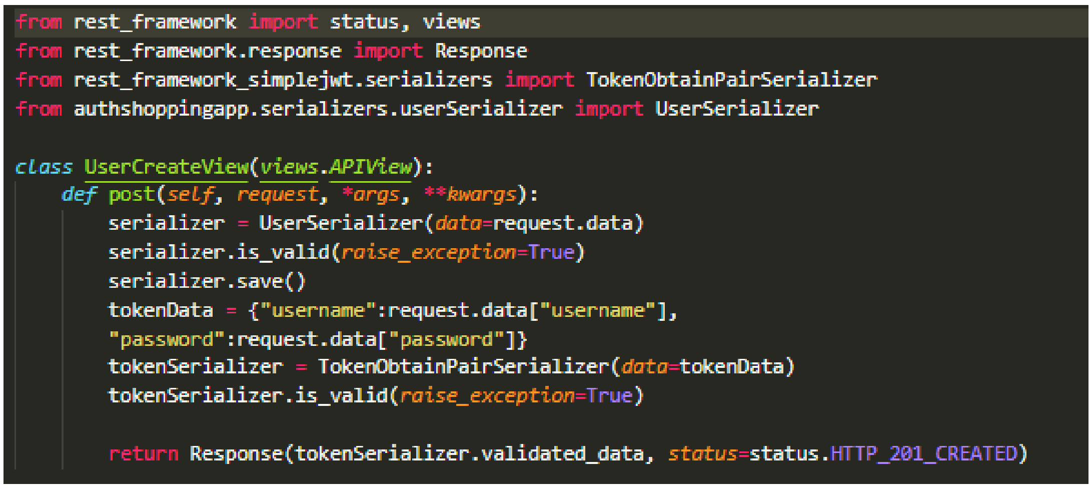
```
om authshoppingapp.models.tourist_package import TouristPackage
from authshoppingapp.serializers.tourist_packageSerializer import TouristPackageSerializer
from rest_framework import generics

class TouristPackageListCreateView(generics.ListCreateAPIView):
    queryset = TouristPackage.objects.all()
    serializer_class = TouristPackageSerializer

class TouristPackageRetrieveUpdateDestroy(generics.RetrieveUpdateDestroyAPIView):
    queryset = TouristPackage.objects.all()
    serializer_class = TouristPackageSerializer
```

UserDetailView:



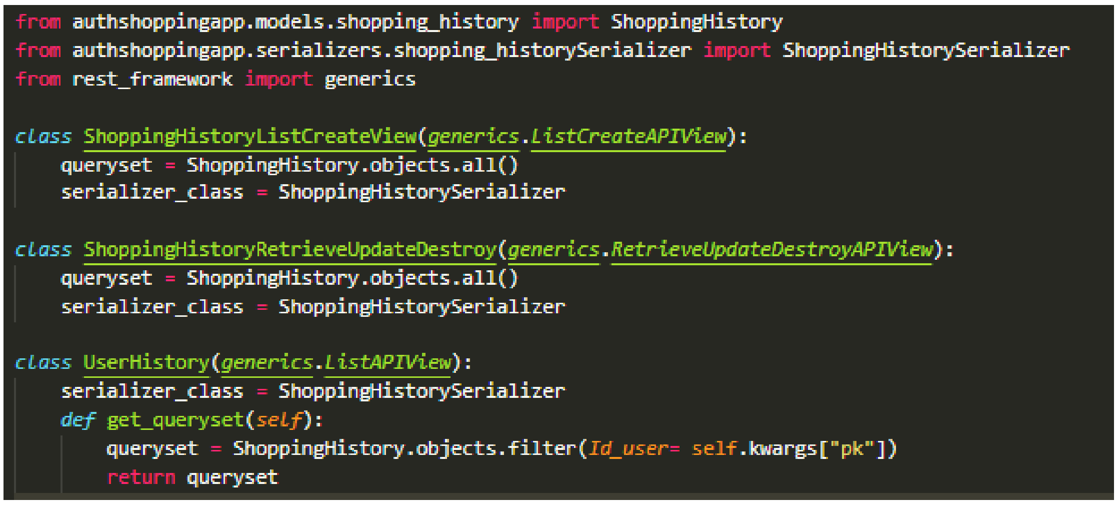
```python
from django.conf import settings
from rest_framework import generics, status
from rest_framework.response import Response
from rest_framework_simplejwt.backends import TokenBackend
from rest_framework.permissions import IsAuthenticated
from authshoppingapp.models.user import User
from authshoppingapp.serializers.userSerializer import UserSerializer
class UserDetailView(generics.RetrieveAPIView):
    queryset = User.objects.all()
    serializer_class = UserSerializer
    permission_classes = (IsAuthenticated,)
    def get(self, request, *args, **kwargs):
        token = request.META.get('HTTP_AUTHORIZATION')[7:]
        tokenBackend = TokenBackend(algorithm=settings.SIMPLE_JWT['ALGORITHM'])
        valid_data = tokenBackend.decode(token,verify=False)
        if valid_data['user_id'] != kwargs['pk']:
            stringResponse = {'detail':'Unauthorized Request'}
            return Response(stringResponse, status=status.HTTP_401_UNAUTHORIZED)
        return super().get(request, *args, **kwargs)
```

UserCreateView:



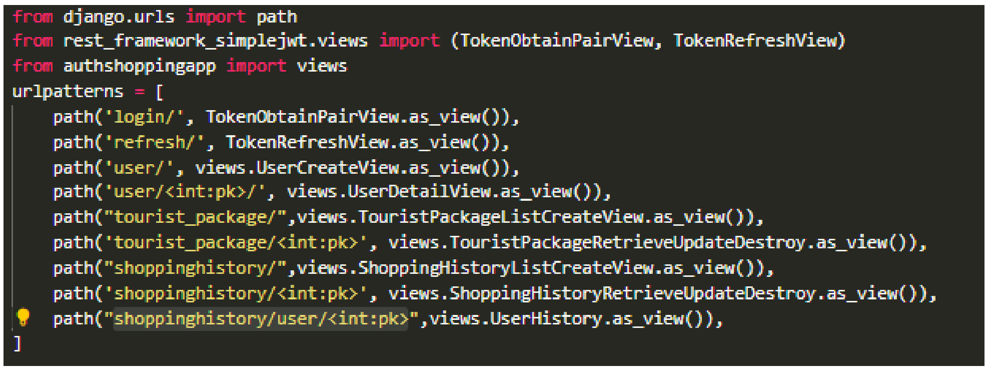
```python
from rest_framework import status, views
from rest_framework.response import Response
from rest_framework_simplejwt.serializers import TokenObtainPairSerializer
from authshoppingapp.serializers.userSerializer import UserSerializer

class UserCreateView(views.APIView):
    def post(self, request, *args, **kwargs):
        serializer = UserSerializer(data=request.data)
        serializer.is_valid(raise_exception=True)
        serializer.save()
        tokenData = {"username":request.data["username"],
        "password":request.data["password"]}
        tokenSerializer = TokenObtainPairSerializer(data=tokenData)
        tokenSerializer.is_valid(raise_exception=True)

        return Response(tokenSerializer.validated_data, status=status.HTTP_201_CREATED)
```

ShoppingHistoryViews:



```python
from authshoppingapp.models.shopping_history import ShoppingHistory
from authshoppingapp.serializers.shopping_historySerializer import ShoppingHistorySerializer
from rest_framework import generics

class ShoppingHistoryListCreateView(generics.ListCreateAPIView):
    queryset = ShoppingHistory.objects.all()
    serializer_class = ShoppingHistorySerializer

class ShoppingHistoryRetrieveUpdateDestroy(generics.RetrieveUpdateDestroyAPIView):
    queryset = ShoppingHistory.objects.all()
    serializer_class = ShoppingHistorySerializer

class UserHistory(generics.ListAPIView):
    serializer_class = ShoppingHistorySerializer
    def get_queryset(self):
        queryset = ShoppingHistory.objects.filter(Id_user= self.kwargs["pk"])
        return queryset
```

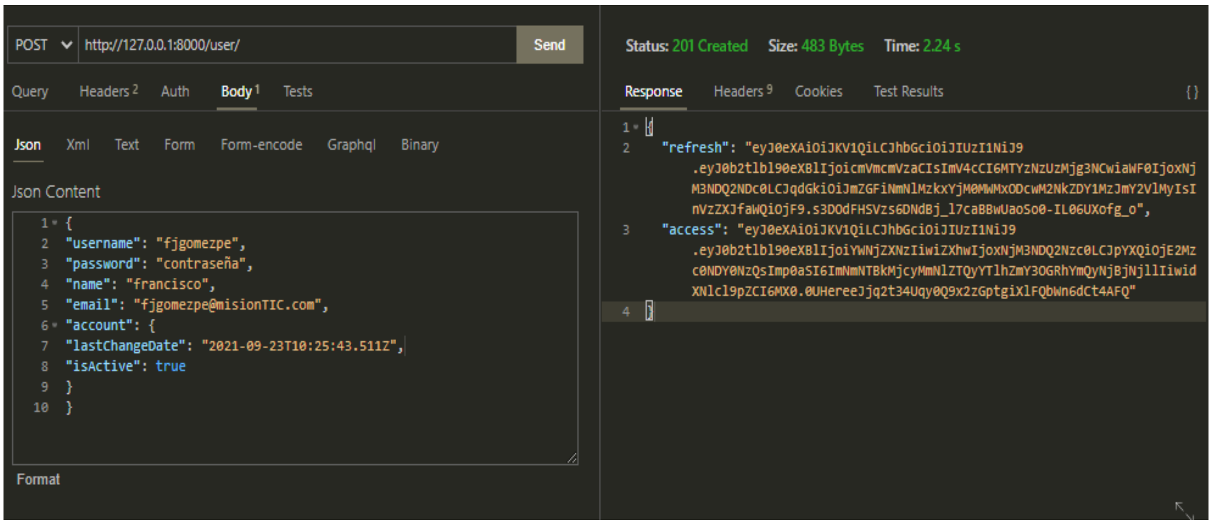
URLS:



```python
from django.urls import path
from rest_framework_simplejwt.views import (TokenObtainPairView, TokenRefreshView)
from authshoppingapp import views
urlpatterns = [
    path('login/', TokenObtainPairView.as_view()),
    path('refresh/', TokenRefreshView.as_view()),
    path('user/', views.UserCreateView.as_view()),
    path('user/<int:pk>/', views.UserDetailView.as_view()),
    path("tourist_package/",views.TouristPackageListCreateView.as_view()),
    path('tourist_package/<int:pk>', views.TouristPackageRetrieveUpdateDestroy.as_view()),
    path("shoppinghistory/",views.ShoppingHistoryListCreateView.as_view()),
    path('shoppinghistory/<int:pk>', views.ShoppingHistoryRetrieveUpdateDestroy.as_view()),
    path("shoppinghistory/user/<int:pk>",views.UserHistory.as_view()),
]
```

Pruebas en los endpoints:

Registro de Usuario:

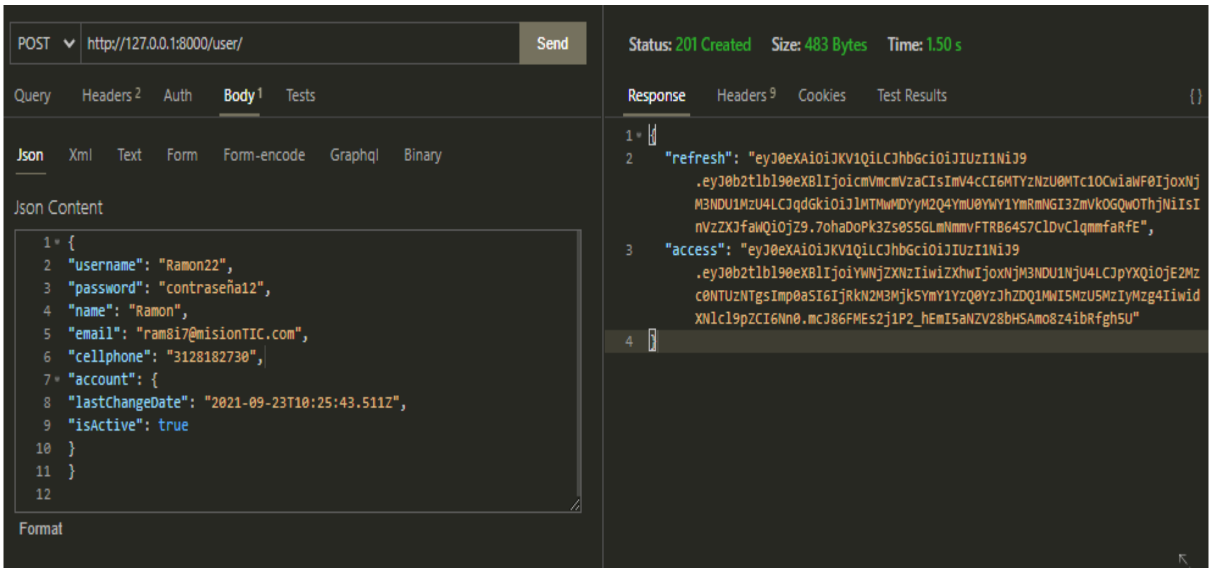


```
http://127.0.0.1:8000/user/
```

```json
{
"username": "fjgomezpe",
"password": "contraseña",
"name": "francisco",
"email": "fjgomezpe@misionTIC.com",
"account": {
"lastChangeDate": "2021-09-23T10:25:43.511Z",
"isActive": true
}
}
```

Status: 201 Created   Size: 483 Bytes   Time: 2.24 s

```json
{
    "refresh": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9
        .eyJ0b2tlbl90eXBlIjoicmVmcmVzaCIsImV4cCI6MTYzNzUzMjg3NCwiaWF0IjoxNj
        M3NDQ2NDc0LCJqdGkiOiJmZGFiNmNlMzkxYjM0MWMxODcwM2NkZDY1MzJmY2VlMyIsI
        nVzZXJfaWQiOjF9.s3DOdFHSVzs6DNdBj_l7caBBwUaoSo0-IL06UXofg_o",
    "access": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9
        .eyJ0b2tlbl90eXBlIjoiYWNjZXNzIiwiZXhwIjoxNjM3NDQ2Nzc0LCJpYXQiOjE2Mz
        c0NDY0NzQsImp0aSI6ImNmNTBkMjcyMmNlZTQyYTlhZmY3OGRhYmQyNjBjNjllIiwid
        XNlcl9pZCI6MX0.0UHereeJjq2t34Uqy0Q9x2zGptgiXlFQbWn6dCt4AFQ"
}
```

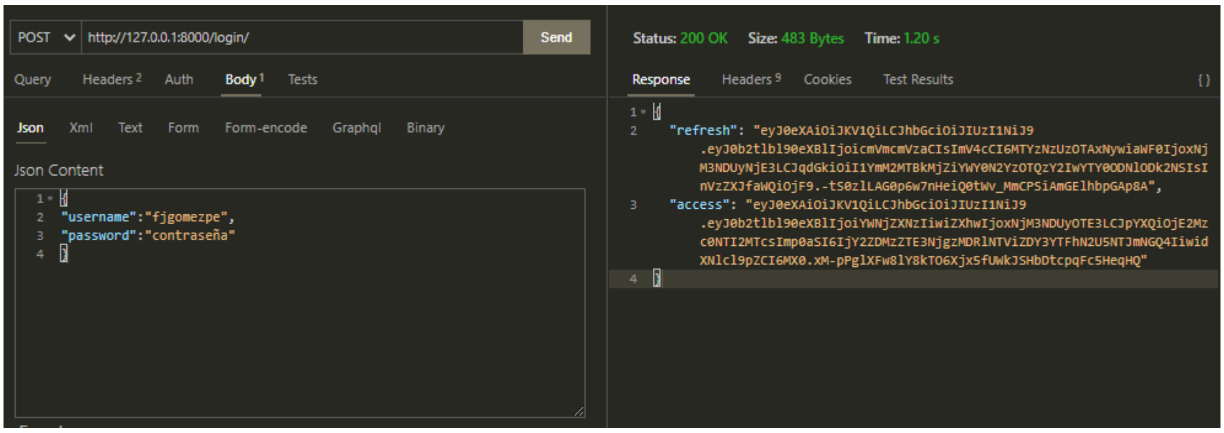


```
http://127.0.0.1:8000/user/
```

```json
{
"username": "Ramon22",
"password": "contraseña12",
"name": "Ramon",
"email": "ram8i7@misionTIC.com",
"cellphone": "3128182730",
"account": {
"lastChangeDate": "2021-09-23T10:25:43.511Z",
"isActive": true
}
}

```

Status: 201 Created   Size: 483 Bytes   Time: 1.50 s

```json
{
    "refresh": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9
        .eyJ0b2tlbl90eXBlIjoicmVmcmVzaCIsImV4cCI6MTYzNzU0MTc1OCwiaWF0IjoxNj
        M3NDU1MzU4LCJqdGkiOiJlMTMwMDYyM2Q4YmU0YWY1YmRmNGI3ZmVkOGQwOThjNiIsI
        nVzZXJfaWQiOjZ9.7ohaDoPk3Zs0S5GLmNmmvFTRB64S7ClDvClqmmfaRfE",
    "access": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9
        .eyJ0b2tlbl90eXBlIjoiYWNjZXNzIiwiZXhwIjoxNjM3NDU1NjU4LCJpYXQiOjE2Mz
        c0NTUzNTgsImp0aSI6IjRkN2M3Mjk5YmY1YzQ0YzJhZDQ1MWI5MzU5MzIyMzg4Iiwid
        XNlcl9pZCI6Nn0.mcJ86FMEs2j1P2_hEmI5aNZV28bHSAmo8z4ibRfgh5U"
}
```



http://127.0.0.1:8000/login/

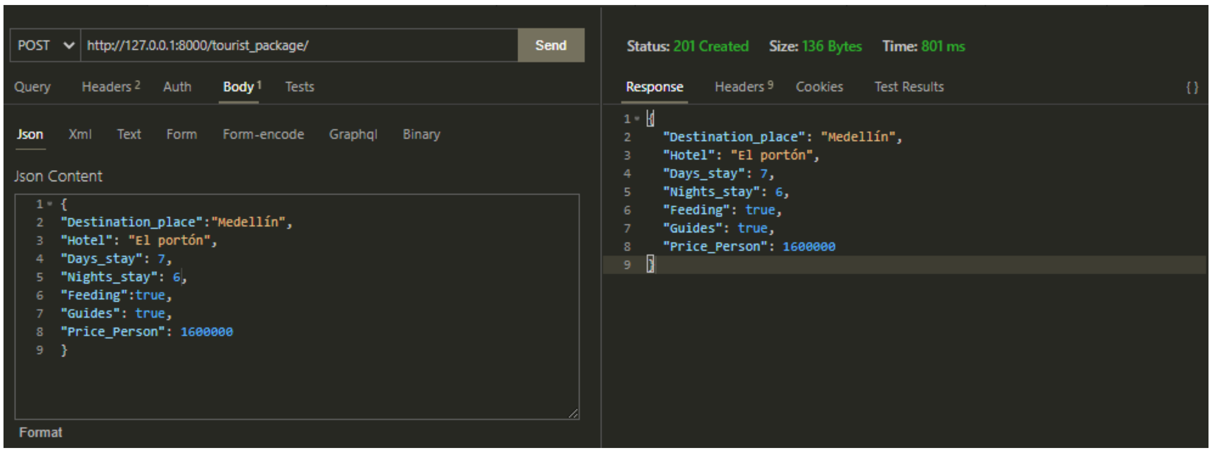
Json Content

```
{
"username":"fjgomezpe",
"password":"contraseña"
}
```

Status: 200 OK Size: 483 Bytes Time: 1.20 s

```
{
  "refresh": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9
    .eyJ0b2tlbl90eXBlIjoicmVmcmVzaCIsImV4cCI6MTYzNzUzOTAxNywiaWF0IjoxNj
    M3NDUyNjE3LCJqdGkiOiI1YmM2MTBkMjZiYWY0N2YzOTQzY2IwYTY0ODNlODk2NSIsI
    nVzZXJfaWQiOjF9.-tS0zlLAG0p6w7nHeiQ0tWv_MmCPSiAmGElhbpGAp8A",
  "access": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9
    .eyJ0b2tlbl90eXBlIjoiYWNjZXNzIiwiZXhwIjoxNjM3NDUyOTE3LCJpYXQiOjE2Mz
    c0NTI2MTcsImp0aSI6IjY2ZDMzZTE3NjgzMDRlNTViZDY3YTFhN2U5NTJmNGQ4Iiwid
    XNlcl9pZCI6MX0.xM-pPglXFw8lY8kTO6Xjx5fUWkJSHbDtcpqFc5HeqHQ"
}
```

Registro de Paquetes turísticos:



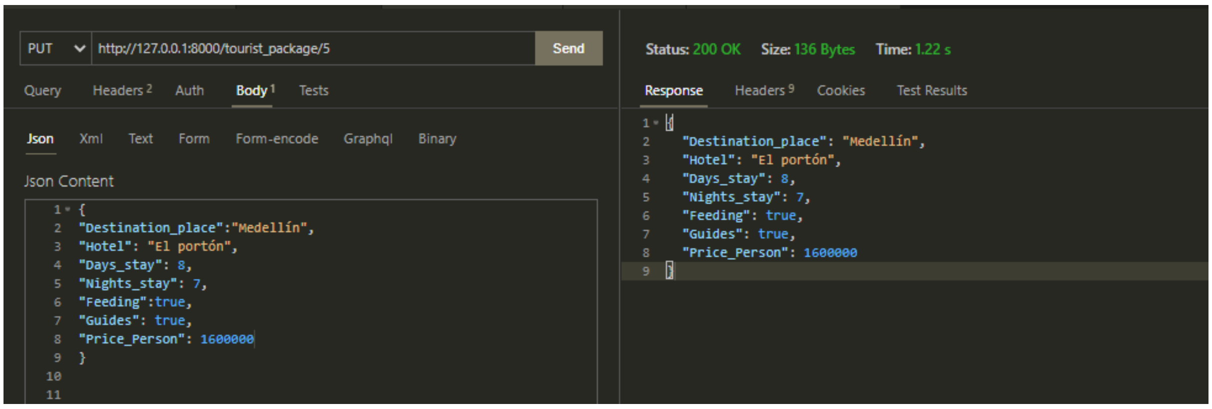
http://127.0.0.1:8000/tourist_package/

Json Content

```
{
"Destination_place":"Medellín",
"Hotel": "El portón",
"Days_stay": 7,
"Nights_stay": 6,
"Feeding":true,
"Guides": true,
"Price_Person": 1600000
}
```

Status: 201 Created Size: 136 Bytes Time: 801 ms

```
{
  "Destination_place": "Medellín",
  "Hotel": "El portón",
  "Days_stay": 7,
  "Nights_stay": 6,
  "Feeding": true,
  "Guides": true,
  "Price_Person": 1600000
}
```



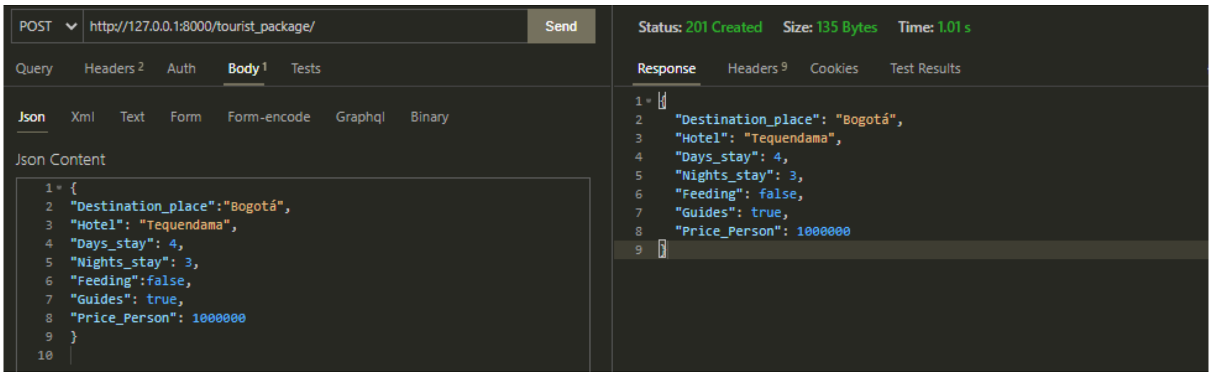
http://127.0.0.1:8000/tourist_package/5

Json Content

```
{
"Destination_place":"Medellín",
"Hotel": "El portón",
"Days_stay": 8,
"Nights_stay": 7,
"Feeding":true,
"Guides": true,
"Price_Person": 1600000
}
```

Status: 200 OK Size: 136 Bytes Time: 1.22 s

```
{
  "Destination_place": "Medellín",
  "Hotel": "El portón",
  "Days_stay": 8,
  "Nights_stay": 7,
  "Feeding": true,
  "Guides": true,
  "Price_Person": 1600000
}
```



http://127.0.0.1:8000/tourist_package/

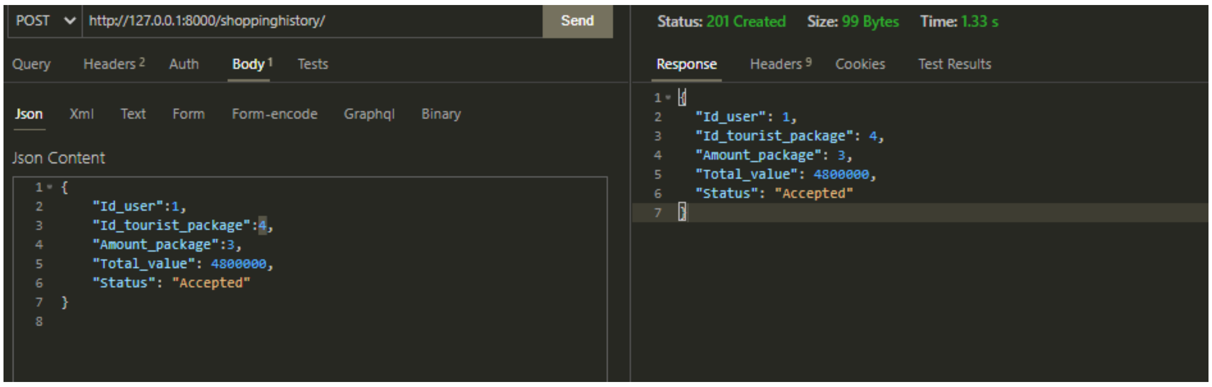
Json Content

```
{
"Destination_place":"Bogotá",
"Hotel": "Tequendama",
"Days_stay": 4,
"Nights_stay": 3,
"Feeding":false,
"Guides": true,
"Price_Person": 1000000
}
```

Status: 201 Created Size: 135 Bytes Time: 1.01 s

```
{
  "Destination_place": "Bogotá",
  "Hotel": "Tequendama",
  "Days_stay": 4,
  "Nights_stay": 3,
  "Feeding": false,
  "Guides": true,
  "Price_Person": 1000000
}
```

Registro de Compra:



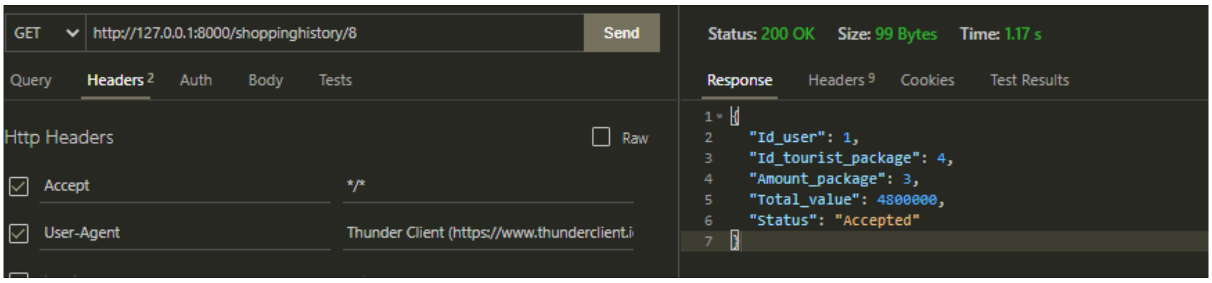
POST http://127.0.0.1:8000/shoppinghistory/

Status: 201 Created Size: 99 Bytes Time: 1.33 s

Json Content

```
{
    "Id_user":1,
    "Id_tourist_package":4,
    "Amount_package":3,
    "Total_value": 4800000,
    "Status": "Accepted"
}
```

```
{
  "Id_user": 1,
  "Id_tourist_package": 4,
  "Amount_package": 3,
  "Total_value": 4800000,
  "Status": "Accepted"
}
```



GET http://127.0.0.1:8000/shoppinghistory/8

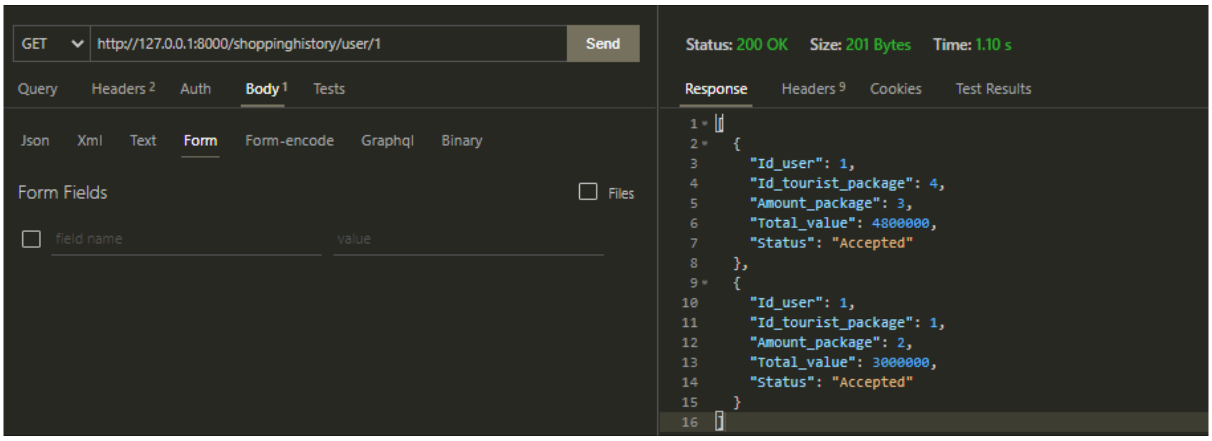
Status: 200 OK Size: 99 Bytes Time: 1.17 s

Http Headers

✓ Accept	*/*
✓ User-Agent	Thunder Client (https://www.thunderclient.i

```
{
  "Id_user": 1,
  "Id_tourist_package": 4,
  "Amount_package": 3,
  "Total_value": 4800000,
  "Status": "Accepted"
}
```

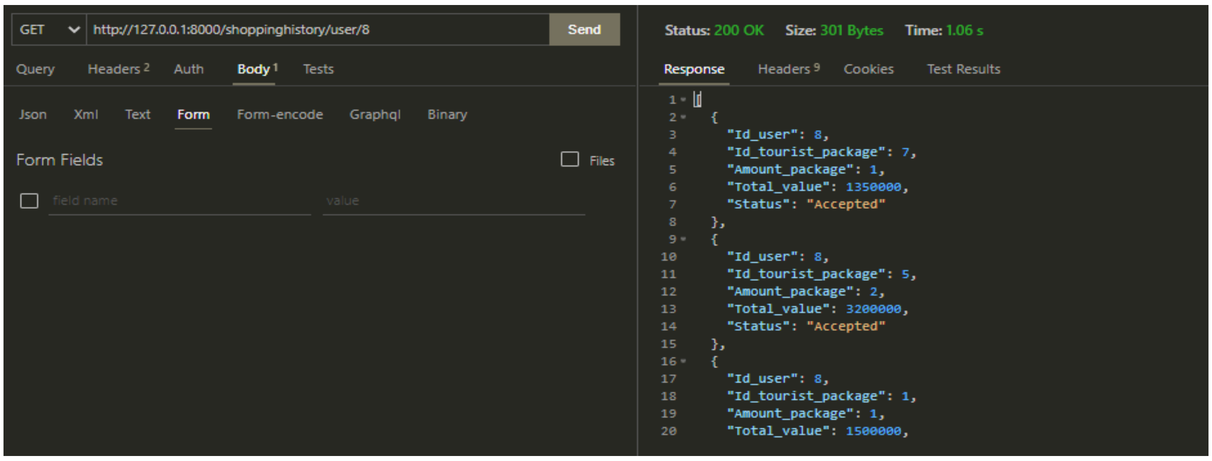
Vista Especial para obtener el Historial de Compras de un usuario en específico:



GET http://127.0.0.1:8000/shoppinghistory/user/1

Status: 200 OK Size: 201 Bytes Time: 1.10 s

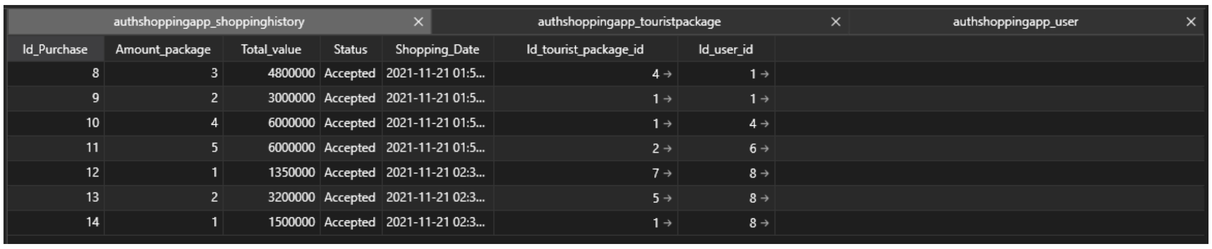
```
[
  {
    "Id_user": 1,
    "Id_tourist_package": 4,
    "Amount_package": 3,
    "Total_value": 4800000,
    "Status": "Accepted"
  },
  {
    "Id_user": 1,
    "Id_tourist_package": 1,
    "Amount_package": 2,
    "Total_value": 3000000,
    "Status": "Accepted"
  }
]
```



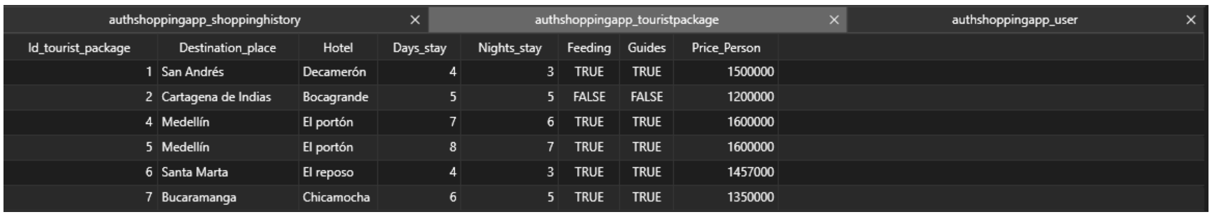
GET http://127.0.0.1:8000/shoppinghistory/user/8

Status: 200 OK Size: 301 Bytes Time: 1.06 s

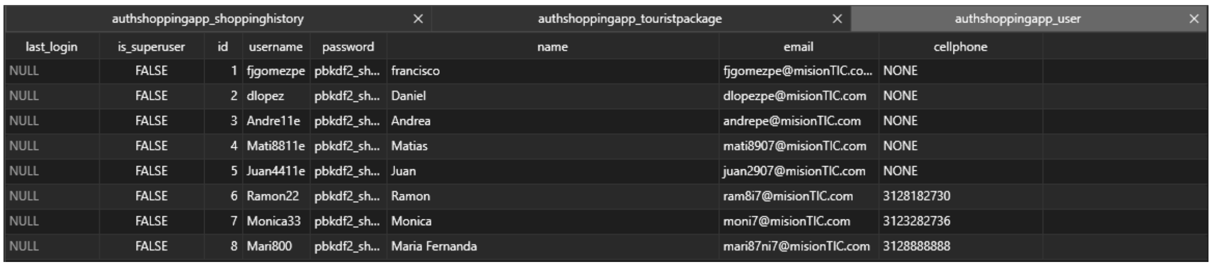
```
[
  {
    "Id_user": 8,
    "Id_tourist_package": 7,
    "Amount_package": 1,
    "Total_value": 1350000,
    "Status": "Accepted"
  },
  {
    "Id_user": 8,
    "Id_tourist_package": 5,
    "Amount_package": 2,
    "Total_value": 3200000,
    "Status": "Accepted"
  },
  {
    "Id_user": 8,
    "Id_tourist_package": 1,
    "Amount_package": 1,
    "Total_value": 1500000,
```


authshoppingapp_shoppinghistory
authshoppingapp_touristpackage
authshoppingapp_user

Id_Purchase	Amount_package	Total_value	Status	Shopping_Date	Id_tourist_package_id	Id_user_id
8	3	4800000	Accepted	2021-11-21 01:5...	4 →	1 →
9	2	3000000	Accepted	2021-11-21 01:5...	1 →	1 →
10	4	6000000	Accepted	2021-11-21 01:5...	1 →	4 →
11	5	6000000	Accepted	2021-11-21 01:5...	2 →	6 →
12	1	1350000	Accepted	2021-11-21 02:3...	7 →	8 →
13	2	3200000	Accepted	2021-11-21 02:3...	5 →	8 →
14	1	1500000	Accepted	2021-11-21 02:3...	1 →	8 →

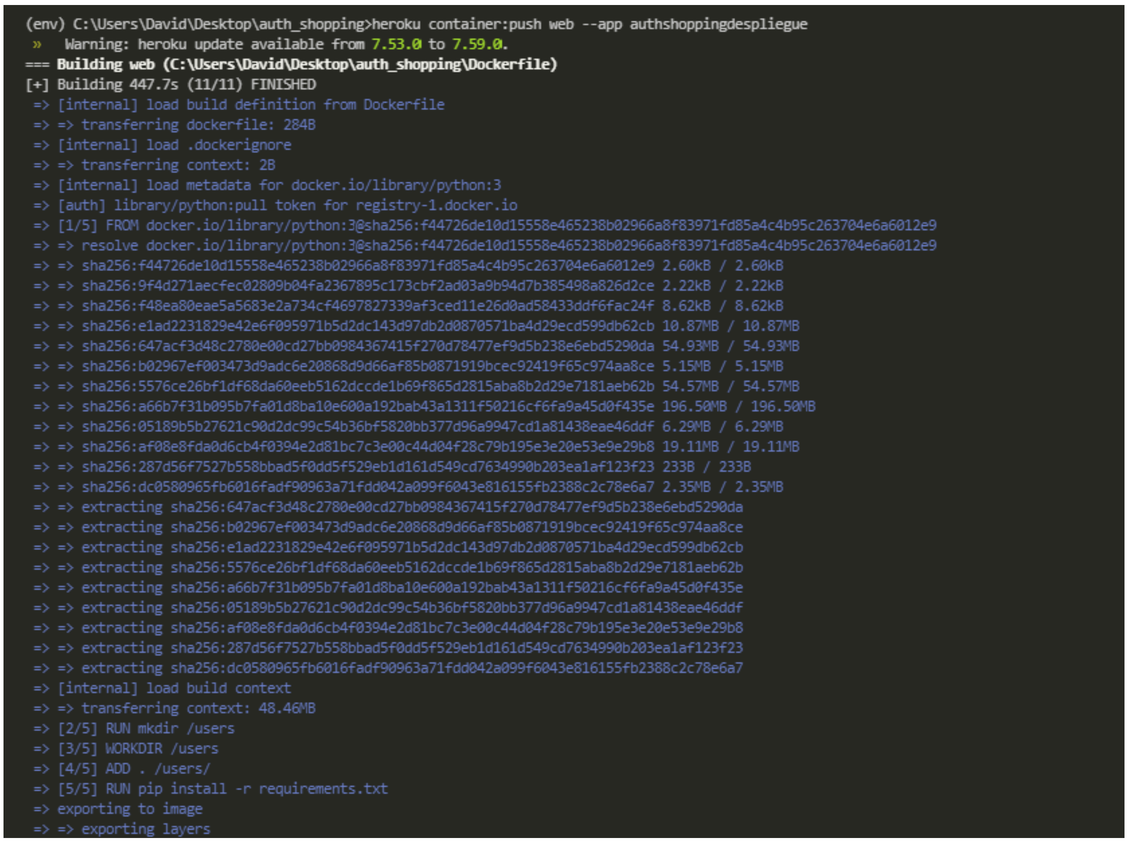

authshoppingapp_shoppinghistory
authshoppingapp_touristpackage
authshoppingapp_user

Id_tourist_package	Destination_place	Hotel	Days_stay	Nights_stay	Feeding	Guides	Price_Person
1	San Andrés	Decamerón	4	3	TRUE	TRUE	1500000
2	Cartagena de Indias	Bocagrande	5	5	FALSE	FALSE	1200000
4	Medellín	El portón	7	6	TRUE	TRUE	1600000
5	Medellín	El portón	8	7	TRUE	TRUE	1600000
6	Santa Marta	El reposo	4	3	TRUE	TRUE	1457000
7	Bucaramanga	Chicamocha	6	5	TRUE	TRUE	1350000

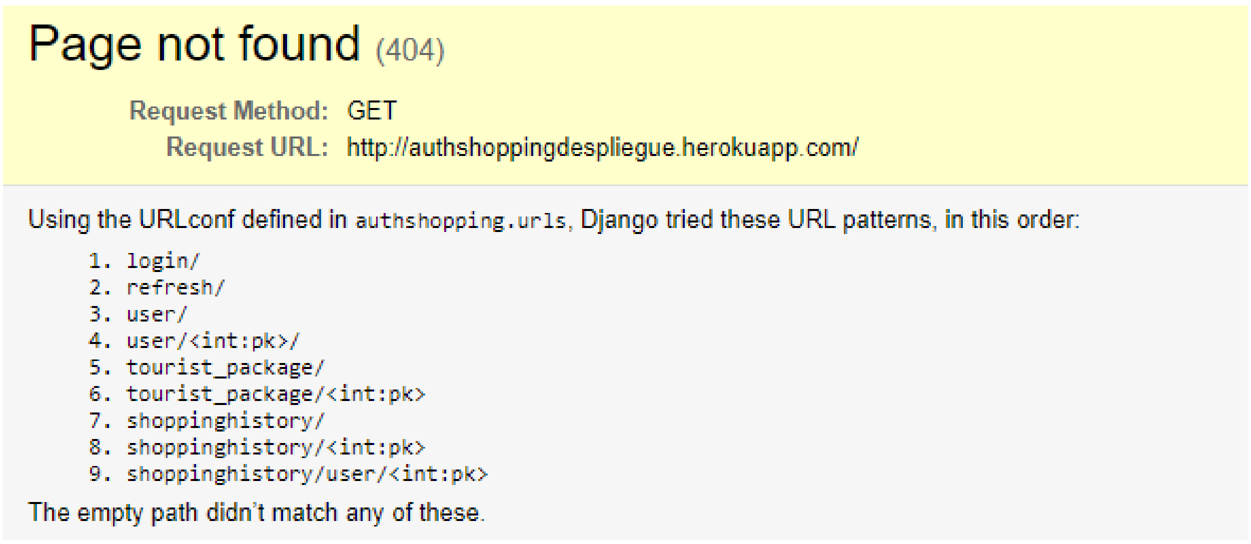

authshoppingapp_shoppinghistory
authshoppingapp_touristpackage
authshoppingapp_user

last_login	is_superuser	id	username	password	name	email	cellphone
NULL	FALSE	1	fjgomezpe	pbkdf2_sh...	francisco	fjgomezpe@misionTIC.co...	NONE
NULL	FALSE	2	dlopez	pbkdf2_sh...	Daniel	dlopezpe@misionTIC.com	NONE
NULL	FALSE	3	Andre11e	pbkdf2_sh...	Andrea	andrepe@misionTIC.com	NONE
NULL	FALSE	4	Mati8811e	pbkdf2_sh...	Matias	mati8907@misionTIC.com	NONE
NULL	FALSE	5	Juan4411e	pbkdf2_sh...	Juan	juan2907@misionTIC.com	NONE
NULL	FALSE	6	Ramon22	pbkdf2_sh...	Ramon	ram8i7@misionTIC.com	3128182730
NULL	FALSE	7	Monica33	pbkdf2_sh...	Monica	moni7@misionTIC.com	3123282736
NULL	FALSE	8	Mari800	pbkdf2_sh...	Maria Fernanda	mari87ni7@misionTIC.com	3128888888

Despliegue con docker:



```
(env) C:\Users\David\Desktop\auth_shopping>heroku container:push web --app authshoppingdespliegue
 »   Warning: heroku update available from 7.53.0 to 7.59.0.
=== Building web (C:\Users\David\Desktop\auth_shopping\Dockerfile)
[+] Building 447.7s (11/11) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 284B
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [internal] load metadata for docker.io/library/python:3
 => [auth] library/python:pull token for registry-1.docker.io
 => [1/5] FROM docker.io/library/python:3@sha256:f44726de10d15558e465238b02966a8f83971fd85a4c4b95c263704e6a6012e9
 => => resolve docker.io/library/python:3@sha256:f44726de10d15558e465238b02966a8f83971fd85a4c4b95c263704e6a6012e9
 => => sha256:f44726de10d15558e465238b02966a8f83971fd85a4c4b95c263704e6a6012e9 2.60kB / 2.60kB
 => => sha256:9f4d271aecfec02809b04fa2367895c173cbf2ad03a9b94d7b385498a826d2ce 2.22kB / 2.22kB
 => => sha256:f48ea80eae5a5683e2a734cf4697827339af3ced11e26d0ad58433ddf6fac24f 8.62kB / 8.62kB
 => => sha256:e1ad2231829e42e6f095971b5d2dc143d97db2d0870571ba4d29ecd599db62cb 10.87MB / 10.87MB
 => => sha256:647acf3d48c2780e00cd27bb0984367415f270d78477ef9d5b238e6ebd5290da 54.93MB / 54.93MB
 => => sha256:b02967ef003473d9adc6e20868d9d66af85b0871919bcec92419f65c974aa8ce 5.15MB / 5.15MB
 => => sha256:5576ce26bf1df68da60eeb5162dccde1b69f865d2815aba8b2d29e7181aeb62b 54.57MB / 54.57MB
 => => sha256:a66b7f31b095b7fa01d8ba10e600a192bab43a1311f50216cf6fa9a45d0f435e 196.50MB / 196.50MB
 => => sha256:05189b5b27621c90d2dc99c54b36bf5820bb377d96a9947cd1a81438eae46ddf 6.29MB / 6.29MB
 => => sha256:af08e8fda0d6cb4f0394e2d81bc7c3e00c44d04f28c79b195e3e20e53e9e29b8 19.11MB / 19.11MB
 => => sha256:287d56f7527b558bbad5f0dd5f529eb1d161d549cd7634990b203ea1af123f23 233B / 233B
 => => sha256:dc0580965fb6016fadf90963a71fdd042a099f6043e816155fb2388c2c78e6a7 2.35MB / 2.35MB
 => => extracting sha256:647acf3d48c2780e00cd27bb0984367415f270d78477ef9d5b238e6ebd5290da
 => => extracting sha256:b02967ef003473d9adc6e20868d9d66af85b0871919bcec92419f65c974aa8ce
 => => extracting sha256:e1ad2231829e42e6f095971b5d2dc143d97db2d0870571ba4d29ecd599db62cb
 => => extracting sha256:5576ce26bf1df68da60eeb5162dccde1b69f865d2815aba8b2d29e7181aeb62b
 => => extracting sha256:a66b7f31b095b7fa01d8ba10e600a192bab43a1311f50216cf6fa9a45d0f435e
 => => extracting sha256:05189b5b27621c90d2dc99c54b36bf5820bb377d96a9947cd1a81438eae46ddf
 => => extracting sha256:af08e8fda0d6cb4f0394e2d81bc7c3e00c44d04f28c79b195e3e20e53e9e29b8
 => => extracting sha256:287d56f7527b558bbad5f0dd5f529eb1d161d549cd7634990b203ea1af123f23
 => => extracting sha256:dc0580965fb6016fadf90963a71fdd042a099f6043e816155fb2388c2c78e6a7
 => [internal] load build context
 => => transferring context: 48.46MB
 => [2/5] RUN mkdir /users
 => [3/5] WORKDIR /users
 => [4/5] ADD . /users/
 => [5/5] RUN pip install -r requirements.txt
 => exporting to image
 => => exporting layers
```


Page not found (404)

Request Method: GET
Request URL: http://authshoppingdespliegue.herokuapp.com/

Using the URLconf defined in `authshopping.urls`, Django tried these URL patterns, in this order:

```
1. login/
2. refresh/
3. user/
4. user/<int:pk>/
5. tourist_package/
6. tourist_package/<int:pk>
7. shoppinghistory/
8. shoppinghistory/<int:pk>
9. shoppinghistory/user/<int:pk>
```

The empty path didn't match any of these.