

Copilot

Yes, you can integrate a payment gateway into your booking application using both React Native for the frontend and Django for the backend. Payment gateways like Stripe, Razorpay, and Paytm offer APIs and SDKs that you can use to handle payments securely. Here's a general approach to integrating a payment gateway:

1. **Choose a Payment Gateway:** Select a payment gateway that supports the currencies and payment methods you want to offer, and that provides good documentation and SDKs for both Django and React Native.
2. **Set Up the Payment Gateway:** Sign up for an account with the payment gateway provider and obtain the necessary API keys for integration.
3. **Backend Integration (Django):**
 - Install the payment gateway's Python package.
 - Configure the payment gateway with your API keys in your Django settings.
 - Create views and endpoints to handle payment-related actions, such as creating charges and handling webhooks for payment status updates.
4. **Frontend Integration (React Native):**
 - Install the payment gateway's React Native package.
 - Use the provided components or APIs to create a payment form where users can enter their payment details.
 - Handle the submission of payment information and communicate with your Django backend to process the payment.
5. **Dynamic Pricing:** To set specific prices for each facility, you can implement dynamic pricing in your Django models. You can define a price field in your facility model and update it based on various factors like demand, time of day, or special events.
6. **Secure Transactions:** Ensure that all payment data is handled securely. Sensitive information should never be stored on your servers and should be processed directly by the payment gateway.
7. **Testing:** Use the payment gateway's test mode to thoroughly test the payment process before going live.
8. **Go Live:** After testing, switch to live mode with your real API keys and start accepting payments.

Here's an example of how you might define the price for each facility in your Django model:

```
class Facility(models.Model):
    name = models.CharField(max_length=100)
    description = models.TextField()
    price = models.DecimalField(max_digits=6, decimal_places=2) # Price for booking the facility
```

And here's a simplified example of a Django view that creates a charge:

```
import stripe
from django.conf import settings
from rest_framework.views import APIView
from rest_framework.response import Response

stripe.api_key = settings.STRIPE_SECRET_KEY

class CreateCharge(APIView):
    def post(self, request, *args, **kwargs):
        # Extract payment details from request
        token = request.data.get('stripeToken')
        amount = request.data.get('amount') # Amount should be in cents

        try:
```

```
# Create a charge
charge = stripe.Charge.create(
    amount=amount,
    currency='usd',
    description='Facility Booking Fee',
    source=token
)
return Response(charge)
except stripe.error.StripeError as e:
    return Response({'error': str(e)}, status=400)
```

Remember to replace 'stripeToken' and 'amount' with the actual data from your frontend, and to handle the conversion of your facility price to the smallest currency unit (e.g., cents for USD).

By following these steps, you can integrate a payment gateway into your app and set dynamic pricing for each facility. Make sure to comply with all legal and regulatory requirements related to online payments in your region.