# Transaction Controller Documentation

This file handles the transaction-related operations for a payment gateway. It is responsible for initiating transactions, fetching transaction data, processing commissions, updating transaction statuses, and generating settlement logs.

---

## Dependencies

1. **Models**:
   - `ICICIQRModel`, `Transaction`, `Merchant`, `Psp`, `apiKeyModel`, `walletModel`, `pricingModel`, `AffiliateMerchants`, `AffiliatesCommission`, `MerchantAmountSettlement`, `MerchantAmountSettlementLogs`, `BackendOperations`, `affiliateUsersModels`
2. **Utilities**:
   - `APIFeatures`: Utility for filtering, sorting, and paginating query results.
   - `json2csv`, `path`, `fs`, `moment`, `ExcelJS`: Used for data manipulation and exporting.
3. **Controllers**:
   - `decentroController`, `payUController`, `easeBuzzController`, `razorpayController`, `apiKeyController`, `whitelistController`, `alertController`.
4. **Other**:
   - `SibApiV3Sdk`, `axios`

---

## Key Functions

1. **`initiateTransaction`**:

   - Initializes a new transaction by validating the provided API key and secret.
   - Checks for IP whitelisting and merchant activation status.
   - Calculates transaction fees, affiliate commissions, and profit.
   - Returns a transaction object and a payment link.

2. **`getAllTransactions`**:

   - Fetches all transactions for a merchant or admin user.
   - Supports filtering by search parameters, status, and pagination.

3. **`getSingleTransaction`**:

   - Retrieves details of a single transaction by its ID.
   - Includes both transaction and associated merchant information.

4. **`processAffiliateCommission`**:

   - Processes affiliate commissions for a transaction.
   - Fetches merchant pricing and affiliate details.
   - Calculates commission based on the affiliate's percentage.

5. **`updateTransactionStatus`**:

   - Updates the status of a transaction (success or failed) based on user input.
   - Sends a callback with the updated status.

6. `deleteAllTransaction`:

   - Deletes all transaction records in the database.

7. `getTransactionStats`:

   - Aggregates transaction statistics, including today's and this month's transaction volumes and profits.
   - Available for both admin and non-admin merchants.

8. `progressBar`:

   - Fetches progress bar data based on completed or successful transactions.
   - Admins can view all transactions, while merchants only see their own.

9. `getTransactionStatus`:

   - Returns the current status of a specific transaction.

10. `updateTransactionStatusICICI`:

    - Updates the transaction status in ICICI's system.

11. `collectRequest`:

    - Initiates a collect request for a UPI transaction.
    - Verifies the merchant's wallet balance and status before proceeding.

12. `requestTransaction`:

    - Requests a transaction using either a UPI collect or QR method.
    - Verifies if the wallet balance is sufficient for the transaction.

13. `updateTransactionStatusCompleted`:

    - Marks a transaction as completed once the payment is successful.

14. `fetchAllTransactionStatus`:

    - Retrieves all distinct statuses of the transactions.

15. `getRedirectUrlAsPerTransactionStatus`:

    - Determines and returns the appropriate redirect URL based on the transaction status.

16. `generateMerchantSettlement`:

    - Generates merchant settlement based on completed transactions within specified time slots (morning or evening).

17. `generateMerchantSettlementCustom`:

- Creates a custom merchant settlement log with provided details like merchant ID, amount, and remarks.

18. `getAllMerchantSettlementLog`:

    - Fetches all settlement logs for merchants.

19. `fetchUniqueMerchantsInSettlementLogs`:

    - Retrieves a list of unique merchants involved in settlement logs.

20. `updateCommissionAndFeesForEveryRecord`:

    - Updates commission and fees for all completed transactions.

21. `markTransactionAsFailed`:

    - Marks a transaction as failed if not completed within a 10-minute window.

22. `getAllTransactionsJsonByMerchantId`:

    - Fetches all transactions for a specific merchant in JSON format for export.

---

## Additional Information

- The file contains logic for handling UPI payments, generating QR codes, and initiating collect requests from different payment operators such as Razorpay, Decentro, and EaseBuzz.
- **Affiliate commission processing** is an integral part of the transaction flow, ensuring that affiliates receive a percentage of the transaction fees.

---

This controller provides comprehensive functionality to manage the entire transaction lifecycle within the payment gateway system.