



የኢትዮጵያ ንግድ ባንክ
Commercial Bank of Ethiopia

Daily Exchange Rate API Document

Version 1.0

Date: October 31, 2025

Prepared by: System Development and Customization Team

1. Overview

The Commercial Bank of Ethiopia (CBE) Exchange Rate API provides official daily foreign currency exchange rates published by the bank. It allows developers, integrators, and organizations to retrieve and display live or historical exchange rate data in applications, dashboards, or websites.

This API can be integrated in two main ways:

- **Iframe Integration** — Embed a responsive exchange rate table directly into your website or app with no coding required.
- **JSON Integration** — Fetch exchange rate data programmatically for custom processing or analytics.

2. Base Endpoint

Base URL: `https://combanketh.et/cbeapi/daily-exchange-rates/?_limit=1&Date=YYYY-MM-DD`

- ✓ Replace YYYY-MM-DD with your desired date. The '_limit=1' parameter must remain unchanged.

3. Part 1 — Iframe Integration

The iframe integration option allows you to embed a ready-made exchange rate table directly on your website. This approach requires no coding, authentication, or backend setup.

Example iframe snippet:

```
<iframe src="https://combanketh.et/cbeapi/embed/exchange-rates?date=2025-10-30" width="100%" height="450" style="border:none;" loading="lazy"> </iframe>
```

When embedded, this iframe automatically retrieves and displays the daily exchange rate table for the specified date. It provides a live, formatted visualization of the same data available through the JSON API.

Part 2 — JSON Integration

The JSON integration method allows developers to directly consume exchange rate data through a RESTful endpoint. Send a GET request with the desired date to retrieve rates in JSON format. This is ideal for apps, dashboards, and financial systems that process data programmatically.

Example Request:

```
curl -X GET "https://combanketh.et/cbeapi/daily-exchange-rates/?_limit=1&Date=2025-10-30"
```

Example JSON Response:

```
{
  "date": "2025-10-30",
  "exchangeRate": [
    {
      "cashBuying": 147.9186,
      "cashSelling": 150.877,
      "transactionalBuying": 147.9186,
      "transactionalSelling": 150.877,
      "currency": {
        "CurrencyCode": "USD",
        "CurrencyName": "US DOLLAR"
      }
    }
  ]
}
```

4. Integration Examples:

- **JavaScript (Fetch):**

```
fetch('https://combanketh.et/cbeapi/daily-exchange-rates/?_limit=1&Date=2025-10-30')  
.then(res => res.json())  
.then(data => console.log(data))  
.catch(err => console.error(err));
```

- **Python (Requests):**

```
import requests  
  
url = 'https://combanketh.et/cbeapi/daily-exchange-rates/?_limit=1&Date=2025-10-30'  
response = requests.get(url)  
print(response.json())
```

- **PHP:**

```
<?php  
$url = 'https://combanketh.et/cbeapi/daily-exchange-rates/?_limit=1&Date=2025-10-30';  
$response = file_get_contents($url);  
$data = json_decode($response, true);  
print_r($data);  
?>
```

- **cURL:**

```
curl -X GET https://combanketh.et/cbeapi/daily-exchange-rates/?\_limit=1&Date=2025-10-30
```

5. Source Code

You can also embed a ready-made exchange rate widget, customize its colors, or even integrate its logic directly into your own app or website using the following code.

```
import { NextResponse } from 'next/server'

export const dynamic = 'force-dynamic' // ensures it re-renders on each request

export async function GET(request: Request) {
  const { searchParams } = new URL(request.url)
  const date = searchParams.get('date') || new Date().toISOString().slice(0, 10)
  const apiUrl = `https://combanketh.et/cbeapi/daily-exchange-rates/?_limit=1&Date=${date}`

  try {
    const response = await fetch(apiUrl, { cache: 'no-store' })
    const data = await response.json()
    const rates = data?.[0]?.ExchangeRate || data?.ExchangeRate || []

    const tableRows = rates.length
    ? rates.map((r: any) => `
      <tr>
        <td>${r.currency?.CurrencyCode || '-'}</td>
        <td>${r.currency?.CurrencyName || '-'}</td>
        <td>${r.cashBuying ?? '-'}</td>
        <td>${r.cashSelling ?? '-'}</td>
        <td>${r.transactionalBuying ?? '-'}</td>
        <td>${r.transactionalSelling ?? '-'}</td>
      </tr>
    `).join('')
    : `<tr><td colspan="6" style="text-align:center; color:#777;">No
exchange rate data for ${date}</td></tr>`

    const html = `
    <!DOCTYPE html>
    <html lang="en">
    <head>
      <meta charset="UTF-8" />
    </head>
    <body>
      ${tableRows}
    </body>
  </html>`

    return new NextResponse(html, { status: 200 })
  } catch (error) {
    return new NextResponse('Error fetching exchange rates', { status: 500 })
  }
}
```

```

    <meta name="viewport" content="width=device-width, initial-
scale=1.0" />
<title>Exchange Rates - ${date}</title>
<style>
    :root {
        --purple: #9c278B;
        --bg: #fafafa;
        --border: #e5e7eb;
    }
    body {
        font-family: Arial, sans-serif;
        background: var(--bg);
        padding: 12px;
        color: #111;
    }
    h3 {
        text-align: center;
        color: var(--purple);
        margin-bottom: 12px;
    }
    table {
        width: 100%;
        border-collapse: collapse;
        font-size: 14px;
    }
    th, td {
        border: 1px solid var(--border);
        padding: 8px;
        text-align: right;
    }
    th {
        background: var(--purple);
        color: white;
        text-align: center;
    }
    td:first-child, td:nth-child(2) {
        text-align: left;
    }
    tr:nth-child(even) {
        background: #f9f9f9;
    }
    caption {
        caption-side: bottom;
        font-size: 12px;
        color: #777;
    }

```

```

        margin-top: 6px;
    }
</style>
</head>
<body>
    <h3>Exchange Rates for ${date}</h3>
    <table>
        <thead>
            <tr>
                <th>Currency Code</th>
                <th>Currency Name</th>
                <th>Cash Buying</th>
                <th>Cash Selling</th>
                <th>Transactional Buying</th>
                <th>Transactional Selling</th>
            </tr>
        </thead>
        <tbody>${tableRows}</tbody>
    </table>
    <caption>Source: Commercial Bank of Ethiopia – Public
API</caption>
</body>
</html>
`
    return new NextResponse(html, {
        status: 200,
        headers: {
            'Content-Type': 'text/html; charset=utf-8',
            'X-Frame-Options': 'ALLOWALL'
        }
    })
} catch (err: any) {
    return new NextResponse(`
        <html>
            <body style="font-family:Arial; padding:20px; color:red;">
                <h3>Error loading exchange rate data</h3>
                <p>${err.message}</p>
            </body>
        </html>
    `, {
        status: 500,
        headers: { 'Content-Type': 'text/html' }
    })
}

```

```
}
```

6. Fields Description

The following describes each field in the JSON response and its data path for programmatic access:

Field	Path	Description
Date	data[0].Date	Date of the exchange rate data (YYYY-MM-DD).
ExchangeRate	data[0].ExchangeRate	Array of rate objects for each currency.
Currency Code	data[0].ExchangeRate[i].currency.CurrencyCode	3-letter ISO currency code (e.g., USD).
Currency Name	data[0].ExchangeRate[i].currency.CurrencyName	Full name of the currency (e.g., US DOLLAR).
Cash Buying	data[0].ExchangeRate[i].cashBuying	Cash buying exchange rate.
Cash Selling	data[0].ExchangeRate[i].cashSelling	Cash selling exchange rate.
Transactional Buying	data[0].ExchangeRate[i].transactionalBuying	Transactional (non-cash) buying rate.
Transactional Selling	data[0].ExchangeRate[i].transactionalSelling	Transactional (non-cash) selling rate.

💡 **Tip:** 'i' refers to the index of the currency in the ExchangeRate array.

7. Error Handling

If the requested date has no data or the format is invalid, the API may return an empty response []. Always check for missing or empty arrays before processing the data.

8. Notes

- The 'Date' parameter is mandatory.
- No authentication (API key) is required — the API is public.
- Data may not be available on weekends or public holidays.
- Responses are provided in application/json format.
- The iframe integration is ideal for embedding live exchange rate tables without code.

Thank you