

API Conversión

GET/Monedas

Obtener Todas Las Monedas

```
app.get('/monedas', async (req, res) => {  
  try {  
    const data = await  
monedas.findAll();  
    res.json(data);  
  } catch (error) {  
    next(error);  
  } });
```

Este endpoint se encarga de recuperar y retornar la lista completa de todas las monedas actualmente almacenadas en la base de datos. Es útil para obtener un catálogo general de monedas disponibles en el sistema, ya sea para fines de visualización, selección o procesamiento adicional por parte del cliente.

Agregar Nueva Moneda

```
app.post('/monedas', validarMoneda, async (req, res, next) => {  
  try {  
    const { origen, destino, valor } = req.body;  
    // Verifica si ya existe la moneda  
    const existente =  
await monedas.findOne({  
      where: {  
        [Op.and]: [{ origen }, { destino }],  
      }  
    });  
    // Si no existe, la crea  
    const  
nuevaMoneda = await monedas.create({  
      origen,  
      destino,  
      valor  
    });  
    res.status(201).json(nuevaMoneda);  
  } catch (error) {  
    next(error);  
  }  
});
```

Este endpoint permite agregar nuevas monedas al sistema. Por ejemplo, se puede utilizar para registrar una nueva conversión, como del Dólar estadounidense al Peso Mexicano. Es útil cuando se desea ampliar la lista de monedas soportadas por la aplicación, incluyendo sus respectivas tasas de cambio.

Agregar Moneda (PUT/Monedas/Id)

```
app.put('/monedas/:id', validarMoneda, async (req, res, next) => {
  try {
    const { id } = req.params;
    const { origen, destino, valor } = req.body;
    // Actualiza los valores de la moneda
    const actualizada = await moneda.update({
      origen, destino, valor
    });
    res.json({
      mensaje: 'Moneda actualizada correctamente',
      data: actualizada
    });
  } catch (error) {
    next(error);
  }
});
```

Este endpoint se utiliza para actualizar el valor de una conversión de moneda ya existente en el sistema. Es útil cuando se requiere modificar la tasa de cambio previamente registrada entre dos monedas, asegurando que la información esté actualizada y refleje las condiciones actuales del mercado.

Realizar Conversión (POST/Convertir)

```
app.post('/convertir', async (req, res, next) => {
  try {
    const { origen, destino, cantidad } = req.body;
    // Busca la tasa de conversión
    const data = await monedas.findOne({
      where: {
        [Op.and]: [{ origen }, { destino }]
      }
    });
    // Realiza la conversión
    const { valor } = data;
    const resultado = Number((cantidad * valor).toFixed(4));
    res.json({
      origen, destino, cantidad, resultado,
      tasa: valor, timestamp: new Date().toISOString()
    });
  } catch (error) {
    next(error);
  }
});
```

Este endpoint principal se encarga de realizar las conversiones entre diferentes monedas. A partir de los valores registrados en el sistema, calcula y devuelve el monto equivalente en la moneda de destino según la tasa de cambio correspondiente.

Base De Datos

```

const monedas = sequelize.define('monedas', {
  origen: {
    type: DataTypes.STRING(10),
    allowNull: false,
    validate: {
      isUppercase: true,
      // Solo mayúsculas
      len: [2, 10]
    },
    // Entre 2 y 10 caracteres
  },
  destino: {
    type: DataTypes.STRING(10),
    allowNull: false,
    validate: {
      isUppercase: true,
      len: [2, 10]
    },
  },
  valor: {
    type:
DataTypes.DOUBLE,
    allowNull: false,
    validate:
    {
      min: 0.000001,
      // Valor mínimo
      max: 1000000
      // Valor máximo
    }
  }
});

```

Este fragmento de código define la estructura utilizada para almacenar las monedas en la base de datos.

- **Origen:** representa el código de la moneda desde la cual se realiza la conversión (por ejemplo, 'USD').
- **Destino:** representa el código de la moneda a la cual se desea convertir (por ejemplo, 'MXN').
- **Valor:** indica la tasa de conversión aplicada entre las dos monedas.

Validaciones Importantes

```

const validarMoneda = (req, res, next) => {
const { origen, destino, valor } = req.body;
  if (!origen || typeof origen !== 'string' || origen.length > 10) {
return res.status(400).json({ error: 'Moneda de origen inválida' });
  }
  if (!destino || typeof destino !== 'string' || destino.length > 10)
{
return res.status(400).json({ error: 'Moneda de destino inválida' });
  }
  if (!valor || typeof valor !== 'number' || valor <= 0) {
return res.status(400).json({ error: 'Valor de conversión inválido' });
  }
  next();
};

```

Esta función se encarga de validar que los datos enviados sean correctos antes de procesar la solicitud.

- Verifica que los códigos de moneda estén en el formato adecuado (por ejemplo, tres letras en mayúsculas: 'USD', 'EUR').
- Asegura que los valores numéricos proporcionados (como la tasa de conversión) sean positivos.
- Comprueba que todos los campos requeridos estén presentes en la solicitud.

Resultados

Agregar Monedas

```
{  
  "origen": "USD",  
  "destino": "MXN",  
  "valor": 17.50  
}
```

Hacer Una Conversion

```
{  
  "origen": "USD",  
  "destino": "MXN",  
  "cantidad": 100  
}
```

Resultado

```
{  
  "origen": "USD",  
  "destino": "MXN",  
  "cantidad": 100,  
  "resultado": 1750.00,  
  "tasa": 17.50,  
  "timestamp": "2024-01-20T15:30:45.123Z"  
}
```