

# 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);
  }
});
```

Lo que hace este endpoint es simplemente devolver todas las monedas que tenemos guardadas en la base de datos

### 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);
  }
});
```

El Endpoint que tenemos aquí, es para agregar nuevas monedas por ejemplo si se quisiera agregar la conversión de Dólar a Peso Mexicano

## 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);
  }
});
```

Aquí Lo Utilizamos Para Actualizar El valor De Una Conversion Existente

## Realizar Conversion (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 EdPoint Principal hace las conversiones

## 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 Código Define Como Se Guardan Las Monedas En La Base De Datos

**Origen:** el código de la moneda de origen

**Destino:** El Código De La Moneda Destino

**Valor:** Tasa De Conversión

## 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 Verifica Que Los Datos Enviados Sean Validos
- Códigos De Moneda En Formato Correcto
- Valores Numéricos Positivos
- Datos Requeridos Presentes

## 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"  
}
```