CENTRE ESPECÍFIC
D´EDUCACIÓ A DISTÀNCIA DE
LA COMUNITAT VALENCIANA

ACTIVIDAD INTERMEDIA EVALUABLE

Desarrollo Web en entorno cliente
CFGS DAW

SEGUNDA EVALUACIÓN

Álvaro Maceda Arranz
alvaro.maceda@ceedcv.es
2022/2023

Versión:221214.1149

Licencia

regula la obra original.

ÍNDICE

ACTIVIDAD INTERMEDIA EVALUABLE - 2ª EVALUACIÓN

Revisa los criterios de evaluación para saber que es lo que debes tener en cuenta a la hora de realizar el ejercicio: no es suficiente que el programa cumpla la función, debe cumplir además otros criterios para ser considerado un buen código JavaScript.

1. ENUNCIADO

Crea un módulo **ES6** que exporte una función llamada `getMinRates(date, currency, weeks)`. No debe realizar la exportación por defecto, sino con nombre.

Dicha función recibirá tres parámetros:

1. Una fecha en formato ISO 8601 (YYYY-MM-DD)
2. Un texto con parte del nombre de una divisa
3. Un número de semanas

La función debe devolver un objeto con el nombre de la divisa y un array con el día y la información del cambio al euro de la divisa con mínimo cambio el día anterior (sigue leyendo y mira los ejemplos para comprenderlo mejor)

Utilizará el API de Frankfurter (https://www.frankfurter.app/) para obtener información sobre los tipos de cambio de esa divisa. La API es accesible a través de la dirección https://api.frankfurter.app

El módulo debe poder trabajar directamente en Node.js, por lo que necesitarás la librería `node-fecth`.

Su funcionamiento será el siguiente:

- Obtener la divisa a partir del texto
- Lanzar **en paralelo** las siguientes dos acciones para el día especificado y las `weeks-1` semanas anteriores:
  - Obtener la divisa con el mínimo cambio para ese día
  - Obtener el cambio al euro para el día anterior de la divisa con el mínimo cambio
- Una vez hayan terminado los procesos en paralelo, mostrar el resultado

La obtención del cambio al euro debe hacerse tan pronto como se sepa cual es la divisa de mínimo cambio para esa fecha: no es correcto esperar hasta tener los datos de mínimo cambio para todas las fechas.

Por ejemplo, esto no sería correcto:

- Obtener la divisa a partir del texto
- Obtener la divisa de mínimo cambio para todas las fechas
- Obtener el cambio al euro de la divisa de mínimo cambio para todas las fechas

1.1 Obtención de divisa

La función obtendrá todas las divisas disponibles utilizando el endpoint `/currencies`. Dicho endpoint devuelve datos en JSON:

```
{
  "AUD": "Australian Dollar",
  "BGN": "Bulgarian Lev",
  "BRL": "Brazilian Real",
  "CAD": "Canadian Dollar",
  ...
  "USD": "United States Dollar",
  "ZAR": "South African Rand"
}
```

La divisa con la que trabajará la función es la primera cuyo nombre contenga el texto pasado como parámetro. Por ejemplo, si se llama a la función con los parámetros:

```
getMinRates('2022-05-13', 'doll', 7)
```

la divisa con la que se trabajaría sería el dólar australiano (AUD)

Si no hay ninguna divisa que contenga el texto, la función fallará con el mensaje `'No se ha encontrado ninguna divisa con el nombre <nombre>'`.

1.2 Obtención de datos semanales

Una vez obtenida la divisa, la función debe realizar, en paralelo, un proceso para el día pasado como parámetro y el mismo día de las semanas anteriores hasta obtener un número total de semanas igual al número pasado como parámetro.

Por ejemplo, si la llamada a la función es:

```
getMinRates('2022-12-04', 'Dollar', 5)
```

Los días para los que se deberían obtener los datos en paralelo serían:

- 2022-12-04
- 2022-11-27
- 2022-11-20
- 2022-11-13
- 2022-11-06

Para cada uno de esos días se hará lo siguiente:

1.2.1 Obtener la divisa con el mínimo cambio

Para obtener el tipo de cambio de cada uno de esos días utilizará el endpoint `/YYYY-MM-DD?from=<DIVISA>`. Por ejemplo, para obtener el tipo de cambio para el dólar australiano el día 4/12/2022 la llamada sería:

```
/2022-12-04?from=AUD
```

Y un ejemplo de respuesta:

```
{
  "amount": 1.0,
  "base": "AUD",
  "date": "2022-12-02",
  "rates": {
    "GBP": 0.55544,
    "EUR": 0.64696,
    ...
    "TRY": 12.7057,
    "USD": 0.68176,
    "ZAR": 11.8231
  }
}
```

De esas divisas, hay que elegir la que tiene el mínimo cambio (GBP en el ejemplo)

1.2.2 Obtención del cambio del día anterior a EUR

Una vez calculada la divisa con el mínimo cambio para ese día, hay que lanzar una petición para obtener el cambio de esa divisa en euros para el día anterior mediante el endpoint `/YYYY-MM-DD?from=<DIVISA>&to=EUR`. La llamada para el ejemplo sería:

```
/2022-12-03?from=GBP&to=EUR
```

Y un ejemplo de respuesta:

```
{
"amount": 1.0,
"base": "GBP",
"date": "2022-12-02",
"rates": {
  "EUR": 1.1648
  }
}
```

Con lo cual, el campo `min` para ese día sería `min: { currency: 'GBP', EUR: 1.1648 } }` (GBP es la divisa con cambio mínimo y 1.1648 el cambio de esa divisa el día 2022-12-03)

Una vez se tengan esos datos, se generará un objeto con dos campos: `currency`, que contendrá el código de la divisa de la que se han obtenido los datos, y un array con los datos de mínimo cambio para ese día. Esto sería un ejemplo de respuesta; el campo `rates` contendrá los datos **en orden ascendente de fecha**:

```
{
  currency: 'AUD',
  rates: [
    { day: '2022-11-06', min: { currency: 'GBP', EUR: 1.1431 } },
    { day: '2022-11-13', min: { currency: 'GBP', EUR: 1.1424 } },
    { day: '2022-11-20', min: { currency: 'GBP', EUR: 1.1148 } },
    { day: '2022-11-27', min: { currency: 'GBP', EUR: 1.1643 } },
```

```
    { day: '2022-12-04', min: { currency: 'GBP', EUR: 1.1648 } }
  ]
}
```

1.3 Validación de tipos de cambio

Cuando se obtengan los datos de cambio de una divisa puede ser que no nos devuelva los datos de la fecha pedida. Por ejemplo, si pedimos los datos del domingo 4/12/2022 la aplicación nos devolverá los datos del viernes anterior (2/12/2022)

La función debe validar que haya como mucho dos días de diferencia entre la fecha que hemos pedido y la que nos ha devuelto. Por ejemplo, si pedimos los datos del día 1/3/2027 la aplicación nos devolverá los datos de la última fecha disponible. Como hay más de dos días de diferencia, esos datos no serían válidos.

En caso de que la petición al API nos devuelva una fecha que no esté en el rango la función fallará con el mensaje `“No se ha podido obtener el cambio para la divisa <DIVISA> el día <DIA>: El cambio no pertenece a la fecha solicitada”`

1.4 Gestión de errores

Cuando se produzca un error en la llamada para obtener las divisas la aplicación fallará con el error: `“No se han podido obtener las divisas”`

Cuando se produzca algún error en la llamada al API para la obtención del cambio la aplicación fallará con el error: `“No se ha podido obtener el cambio para la divisa <DIVISA> el día <DIA>: <ERROR>”`

1.5 Ejemplos

Ejemplo 1

```
getMinRates('2022-12-04', 'Dollar', 5)

{
  currency: 'AUD',
  rates: [
    { day: '2022-11-06', min: { currency: 'GBP', EUR: 1.1431 } },
    { day: '2022-11-13', min: { currency: 'GBP', EUR: 1.1424 } },
    { day: '2022-11-20', min: { currency: 'GBP', EUR: 1.1486 } },
    { day: '2022-11-27', min: { currency: 'GBP', EUR: 1.1643 } },
    { day: '2022-12-04', min: { currency: 'GBP', EUR: 1.1648 } }
  ]
}
```

Ejemplo 2

```
getMinRates('2022-05-13', 'Dollar', 0)

{ currency: 'AUD', rates: [] }
```

Ejemplo 3

```
getMinRates('2022-05-13', 'banana', 5)

No se ha encontrado ninguna divisa con el nombre banana
```

Ejemplo 4

```
getMinRates('2023-12-12', 'Dollar', 5)

No se ha podido obtener el cambio para la divisa AUD el día 2023-12-12: El
cambio no pertenece a la fecha solicitada
```

Ejemplo 5

```
getMinRates('2022-12-04', 'pou', 5)

{
  currency: 'GBP',
  rates: [
    { day: '2022-11-06', min: { currency: 'CHF', EUR: 1.0139 } },
    { day: '2022-11-13', min: { currency: 'CHF', EUR: 1.0158 } },
    { day: '2022-11-20', min: { currency: 'CHF', EUR: 1.012 } },
    { day: '2022-11-27', min: { currency: 'CHF', EUR: 1.0167 } },
    { day: '2022-12-04', min: { currency: 'CHF', EUR: 1.0169 } }
  ]
}
```

Ejemplo 6

```
getMinRates('2022-05-13', 'banana', 5)

(error de Internet en la llamada a /currencies)

No se han podido obtener las divisas
```

No puedes utilizar librerías externas, a excepción de `node-fetch` para realizar las peticiones al API.

2. Linter

Para que el ejercicio se corrija, el linter no debe devolver ningún error. En caso de que el linter devuelva un error, la máxima nota del trabajo será de 1.

Las reglas del linter son las que están especificadas en el fichero `.eslintrc.json` de la plantilla de linter (https://github.com/CEED-2022/linter-template) No puede modificarse este fichero. Asimismo no se admitirá deshabilitar ninguna de las reglas de eslint por ningún medio: en ese caso se procederá como si el programa hubiese fallado el linter.

3. Entrega

Debes entregar el ejercicio como un ficheros de texto plano con el nombre `get_rates.js`. Los ficheros se entregará en la tarea del curso habilitada a tal efecto.

No se admitirán entregas pasada la fecha límite. No se admitirán entregas con formato incorrecto.

4. Criterios de evaluación

- El programa es correcto, realiza la función que se solicita en el enunciado
- Las operaciones se paralelizan correctamente, de modo que la operación se realice en el menor tiempo posible.
- Se utilizan correctamente las estructuras de programación asíncronas.
- Se han utilizado estructuras del lenguaje adecuadas: bucles, condicionales, operadores, etc.
- Se han utilizado variables y constantes de forma adecuada
- Se utilizan correctamente y cuando corresponda los tipos de datos y objetos predefinidos del lenguaje (Arrays, objetos planos, Map, Set, etc.)
- Se han utilizado funciones para estructurar el código, definiendo y utilizando parámetros y valores de respuesta de forma adecuada
- El programa es lo más sencillo posible para realizar su función.
- No existe código repetido: se han extraído los comportamientos comunes a funciones y se ha intentado hacer el código genérico.
- El programa cumple todas las reglas definidas para el linter.

5. Recursos

Para trabajar con fechas puedes usar estas funciones:

```
function addDays(date, days) {
  const newDate = new Date(date)
  newDate.setDate(date.getDate() + days)
  return newDate
}

function YYYYMMDD(date) {
  return date.toISOString().slice(0, 10)
}
```