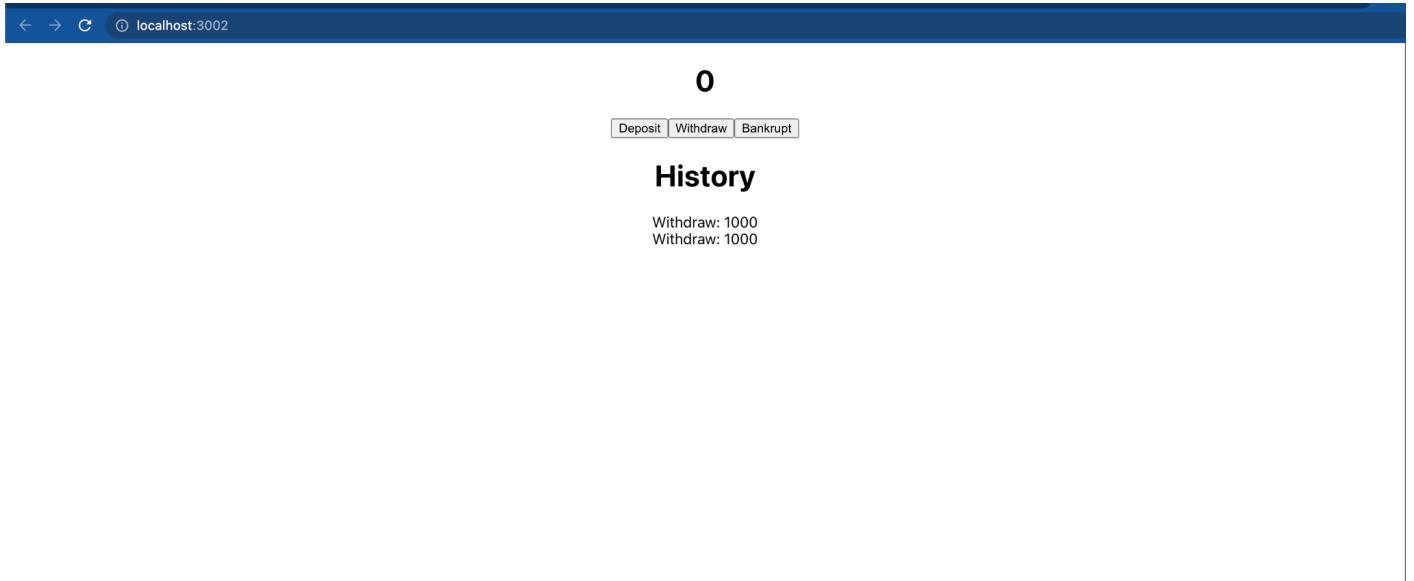


# Code Challenge Retalapp

En el archivo adjunto vas a encontrar un archivo zip que contiene el código fuente del proyecto.

Una vez lo descomprimes puedes correr el comando `yarn start` o `npm start`  
Y vas a poder ver en el navegador la siguiente pantalla:



Este código aun no está conectado con datos del servidor y no utiliza ninguna llamada a el API aun, las tareas consisten en conectar este proyecto con un API REST local.

Para correr el API REST local debes correr:  
`yarn server` o `npm run server`

\*Nota: Es posible que encuentres un error al correr este comando, debes entender porque es este error y encontrar cómo solucionarlo

\*Nota: Debes correr el comando de instalar paquetes

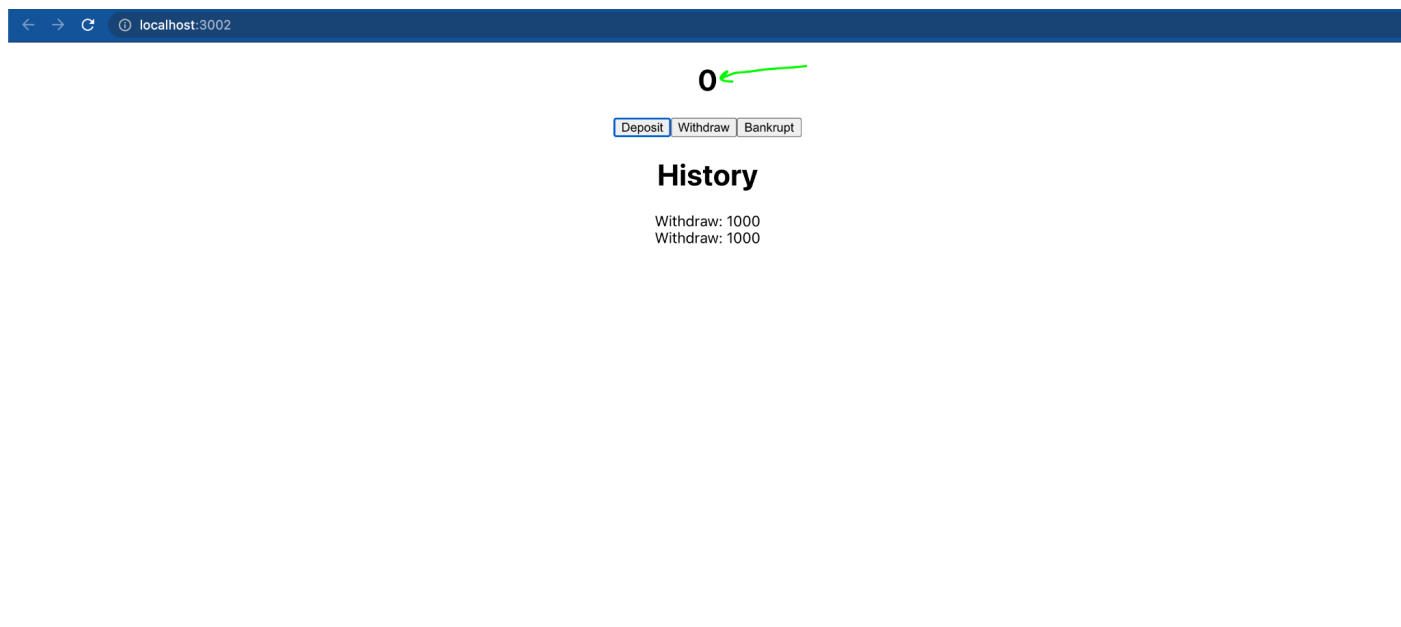
Luego de sobrepasar ese error deberías poder acceder al API localmente desde  
<http://localhost:3041/history>

```
localhost:3041/history

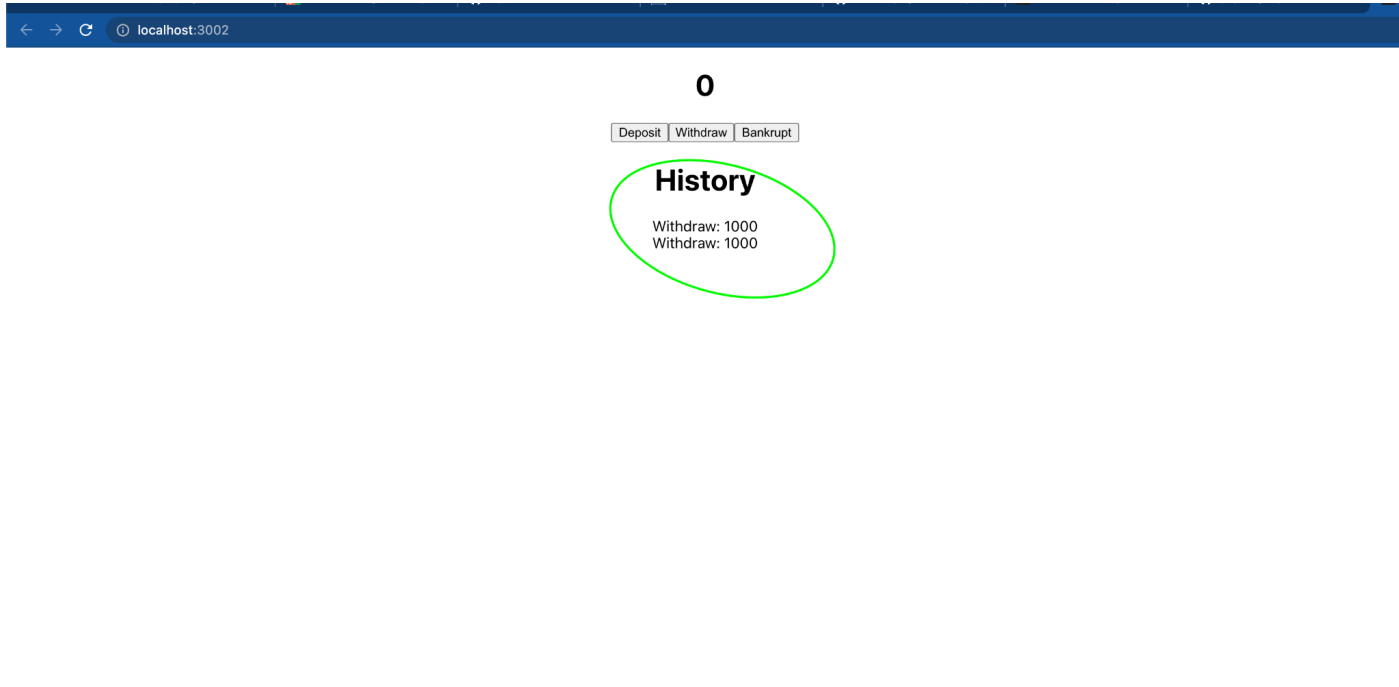
[
  {
    "type": "withdraw",
    "amount": 1000
  },
  {
    "type": "deposit",
    "amount": 1000
  },
  {
    "type": "deposit",
    "amount": 1000
  }
]
```

### Desafio:

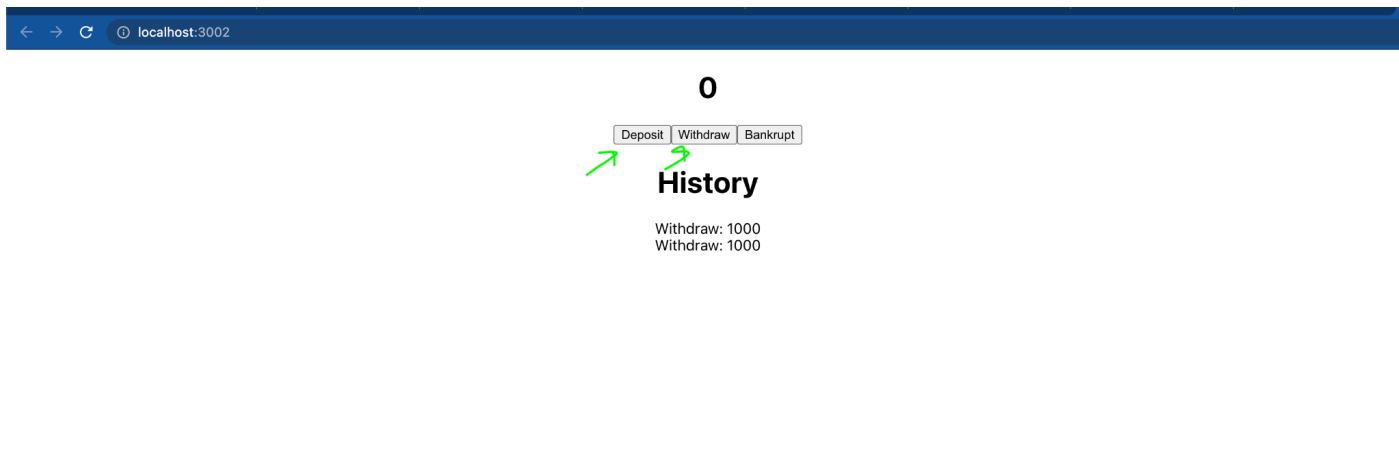
1. (Requerido) Leer los datos del siguiente endpoint <http://localhost:3041/history> para mostrar el valor en la cuenta tan pronto la pagina se carga: (Este valor es la suma de todos los valores en el endpoint the GET /history)



2. (Requerido) Leer los datos del siguiente endpoint <http://localhost:3041/history> para mostrarlos en el listado de historia usando Redux



3. (Requerido) Agregar nuevos registros en el historial cada vez que el usuario da click en “Deposit” o “Withdraw”



4. (Opcional) Si el usuario da click en “Bankrupt” debes hacer un request como si fuera “Withdraw” con el total de dinero que la cuenta tenga en ese momento para que la cuenta quede en cero.

\*\*Ten en cuenta que vas a necesitar instalar nuevos paquetes para lograr esto

\*\*Ten en cuenta que solo debes usar Hooks, no componentes de clases ni estados de clases

\*\*Vas a necesitar refactorizar el reducer y las acciones para usar datos del API

**\*\*Si utilizas selectores y useSelector tendrás puntuación adicional comparado con otros candidatos**

**\*\*Si usas los types de typescript tienes puntos adicionales comparado con otros candidatos**

**\*\*Si aplicas estilos para hacerlo que se vea mas atractivo tienes puntos adicionales comparado con otros candidatos**