

```
import requests

from diskcache import Cache

from kivy.app import App

from kivy.uix.label import Label

from kivy.uix.textinput import TextInput

from kivy.uix.spinner import Spinner

from kivy.uix.button import Button

from kivy.uix.boxlayout import BoxLayout

from kivy.uix.screenmanager import ScreenManager, Screen


cache = Cache('currency_cache')


API_URL = "https://open.er-api.com/v6/latest/"


conversion_history_file = 'conversion_history.txt'
conversion_history = []


def save_history():
    with open(conversion_history_file, 'w') as file:
        for conversion in conversion_history:
            file.write(conversion + '\n')


def load_history():
    global conversion_history

    try:
        with open(conversion_history_file, 'r') as file:
```

```

        conversion_history = [line.strip() for line in file.readlines()]
except FileNotFoundError:
    pass

def get_exchange_rate(base_currency):
    if base_currency in cache:
        return cache[base_currency]

    response = requests.get(API_URL + base_currency)
    if response.status_code == 200:
        data = response.json()
        if data['result'] == 'success':
            cache[base_currency] = data['rates']
            return data['rates']
    return None

def get_available_currencies():
    response = requests.get(API_URL + 'USD')
    if response.status_code == 200:
        data = response.json()
        if data['result'] == 'success':
            return list(data['rates'].keys())
    return []

def convert_currency(amount, from_currency, to_currency, rates):

```

```

if from_currency == to_currency:
    return amount
return amount * rates[to_currency]

class MainScreen(Screen):
    def __init__(self, **kwargs):
        super(MainScreen, self).__init__(**kwargs)
        self.build()

    def build(self):
        layout = BoxLayout(orientation='vertical', padding=10, spacing=10)

        self.amount_input = TextInput(hint_text='Amount', input_filter='float',
        multiline=False)
        layout.add_widget(self.amount_input)

        available_currencies = get_available_currencies()

        self.from_currency = Spinner(
            text='USD',
            values=available_currencies,
            size_hint=(1, 0.2),
            background_color=(0.1, 0.5, 0.8, 1),
            color=(1, 1, 1, 1)
        )
        layout.add_widget(self.from_currency)

        self.to_currency = Spinner(

```

```

        text='EUR',
        values=available_currencies,
        size_hint=(1, 0.2),
        background_color=(0.1, 0.5, 0.8, 1),
        color=(1, 1, 1, 1)
    )
    layout.add_widget(self.to_currency)

    self.convert_button = Button(text='Convert', size_hint=(1, 0.2))
    self.convert_button.bind(on_press=self.convert)
    layout.add_widget(self.convert_button)

    self.result_label = Label(text='Result will be shown here', size_hint=(1, 0.2))
    layout.add_widget(self.result_label)

    self.history_button = Button(text='Show History', size_hint=(1, 0.2))
    self.history_button.bind(on_press=self.show_history)
    layout.add_widget(self.history_button)

    self.add_widget(layout)

def convert(self, instance):
    try:
        amount = float(self.amount_input.text)
        from_currency = self.from_currency.text
        to_currency = self.to_currency.text

        rates = get_exchange_rate(from_currency)

```

```

    if rates:
        result = convert_currency(amount, from_currency, to_currency, rates)
        result_text = f'{amount} {from_currency} = {result:.2f} {to_currency}'
        self.result_label.text = result_text
        conversion_history.append(result_text)
        save_history()
    else:
        self.result_label.text = 'Error fetching exchange rates'
except ValueError:
    self.result_label.text = 'Invalid amount'

def show_history(self, instance):
    self.manager.current = 'history'

class HistoryScreen(Screen):
    def __init__(self, **kwargs):
        super(HistoryScreen, self).__init__(**kwargs)
        self.build()

    def build(self):
        layout = BoxLayout(orientation='vertical', padding=10, spacing=10)

        self.history_label = Label(text='Conversion History', size_hint=(1, 0.1))
        layout.add_widget(self.history_label)

        self.history_list = BoxLayout(orientation='vertical', size_hint=(1, 0.8))
        layout.add_widget(self.history_list)

```

```

self.update_history()

self.back_button = Button(text='Back', size_hint=(1, 0.1))
self.back_button.bind(on_press=self.go_back)
layout.add_widget(self.back_button)

self.add_widget(layout)

def on_pre_enter(self, *args):
    self.update_history()

def update_history(self):
    self.history_list.clear_widgets()
    for conversion in conversion_history:
        self.history_list.add_widget(Label(text=conversion, size_hint=(1, None),
height=30))

def go_back(self, instance):
    self.manager.current = 'main'

class CurrencyConverterApp(App):
    def build(self):
        load_history()

        sm = ScreenManager()
        main_screen = MainScreen(name='main')
        sm.add_widget(main_screen)

```

```
history_screen = HistoryScreen(name='history')
sm.add_widget(history_screen)

return sm

def on_stop(self):
    save_history()

if __name__ == '__main__':
    CurrencyConverterApp().run()
```