Московский государственный технический
университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»
Отчет по лабораторной работе №5-6
«Телеграмм-бот на Aiogram»

Выполнил:	Проверил:
Студент группы ИУ5-33Б	Преподаватель каф. ИУ5
Иванов Николай	Нардид А.Н.

Москва, 2023 г.

Описание работы

Создание телеграмм-бота для анализа рынка, позволяющего пользователю узнать курс обмена фиатных валют. API для получения данных о курсе криптовалют.

Пользователь может ввести базовую и котируемую валюты и указать сумму для перевода. Также пользователь может изменить базовую или котируемую валюты, а также сумму для перевода.

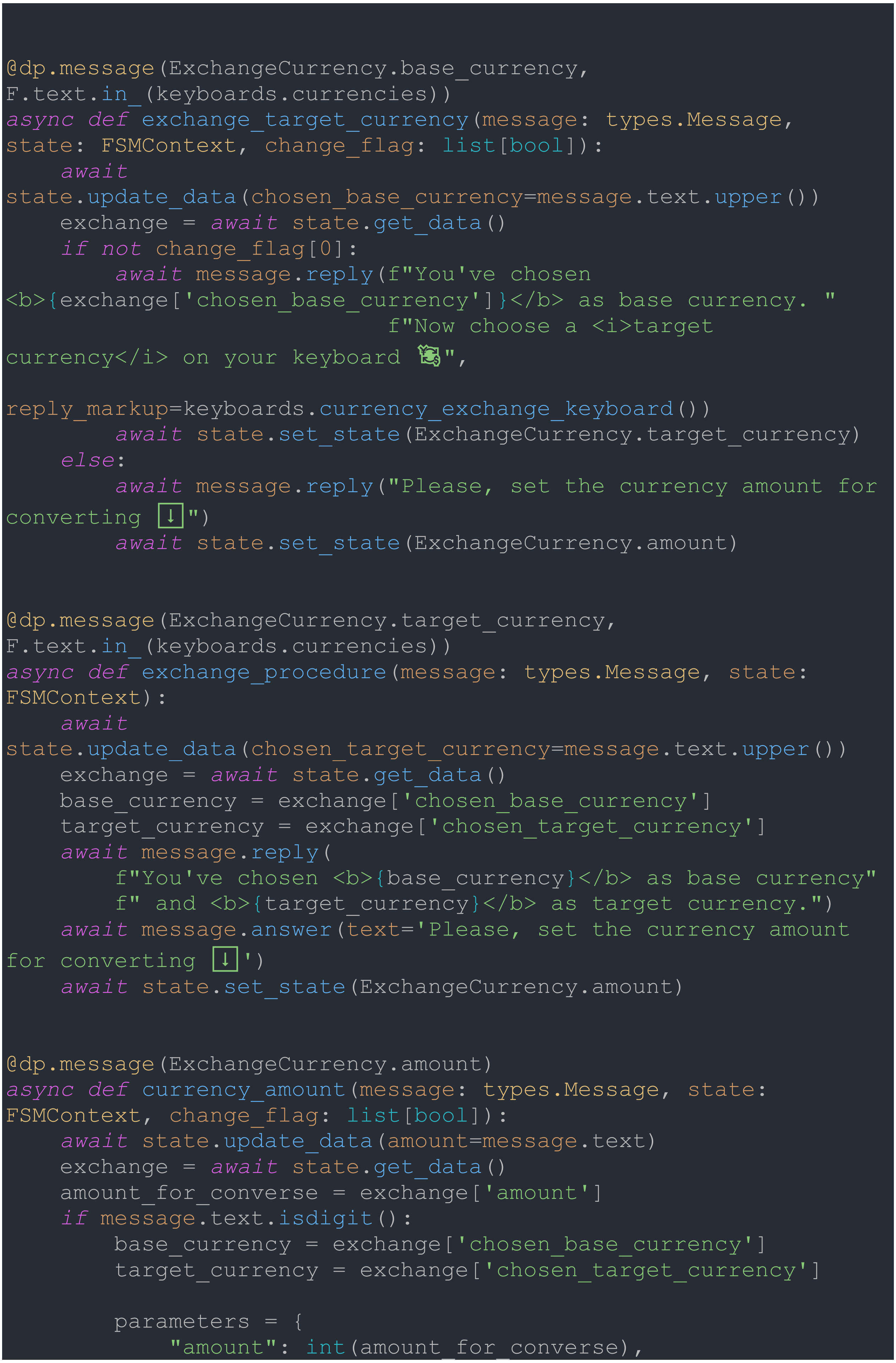
Для написания телеграмм-бота используется библиотека **Aiogram**. Хранение данных, вводимых пользователем, осуществляется с помощью механизма конечных автоматов (**FSM**).

Для надежного хранения конфиденциальной информации, такой как токена бота, API-ключи, используются dotenv файлы и библиотека **Pydantic**.

Файл bot.py

Текст программы

```
import asyncio
import logging

from aiogram import Bot, Dispatcher, types, F
from aiogram.filters import Command, StateFilter
from aiogram.fsm.context import FSMContext
from configs.config import configuration
from keyboards import keyboards
from FSM.StateMachine import ExchangeCurrency
from FSM.StateMachine import Menu
from api.api import api_crypto

logging.basicConfig(filename="../static/logger.txt",
level=logging.INFO)
dp = Dispatcher()
bot = Bot(token=configuration.BOT_TOKEN.get_secret_value(),
parse_mode="html")


@dp.message(Command("start"))
async def cmd_start(message: types.Message, state: FSMContext):
    await message.answer(f"Hello, {message.from_user.username}!
Welcome to NIVACryptoBot 📈\n\n"
                         f"Choose an option on your keyboard 📲",
                         reply_markup=keyboards.main_keyboard)
    await state.set_state(Menu.option)


@dp.message(F.text.lower().in_(['back']))
async def back(message: types.Message, state: FSMContext,
change_flag: list[bool]):
    await state.clear()
    if change_flag[0]:
        change_flag[0] = False
    await state.set_state(Menu.option)
    return await message.answer(text='Back to Menu',
reply_markup=keyboards.main_keyboard)


@dp.message(Menu.option, F.text.in_("Currency exchange prices"))
async def menu_option(message: types.Message, state:
FSMContext):
    await state.update_data(option=message.text)
    await message.answer("Choose a <i>base currency</i> on your
keyboard 💱",

reply_markup=keyboards.currency_exchange_keyboard())
    await state.set_state(ExchangeCurrency.base_currency)
```



```
@dp.message(ExchangeCurrency.base_currency,
F.text.in_(keyboards.currencies))
async def exchange_target_currency(message: types.Message,
state: FSMContext, change_flag: list[bool]):
    await
state.update_data(chosen_base_currency=message.text.upper())
    exchange = await state.get_data()
    if not change_flag[0]:
        await message.reply(f"You've chosen
<b>{exchange['chosen_base_currency']}</b> as base currency. "
                            f"Now choose a <i>target
currency</i> on your keyboard 💱",

reply_markup=keyboards.currency_exchange_keyboard())
        await state.set_state(ExchangeCurrency.target_currency)
    else:
        await message.reply("Please, set the currency amount for
converting ⬇️")
        await state.set_state(ExchangeCurrency.amount)


@dp.message(ExchangeCurrency.target_currency,
F.text.in_(keyboards.currencies))
async def exchange_procedure(message: types.Message, state:
FSMContext):
    await
state.update_data(chosen_target_currency=message.text.upper())
    exchange = await state.get_data()
    base_currency = exchange['chosen_base_currency']
    target_currency = exchange['chosen_target_currency']
    await message.reply(
        f"You've chosen <b>{base_currency}</b> as base currency"
        f" and <b>{target_currency}</b> as target currency.")
    await message.answer(text='Please, set the currency amount
for converting ⬇️')
    await state.set_state(ExchangeCurrency.amount)


@dp.message(ExchangeCurrency.amount)
async def currency_amount(message: types.Message, state:
FSMContext, change_flag: list[bool]):
    await state.update_data(amount=message.text)
    exchange = await state.get_data()
    amount_for_converse = exchange['amount']
    if message.text.isdigit():
        base_currency = exchange['chosen_base_currency']
        target_currency = exchange['chosen_target_currency']

        parameters = {
            "amount": int(amount_for_converse),
            "symbol": base_currency,
            "convert": target_currency
        }

        response = api_crypto(parameters)
        conversion =
response["data"][0]["quote"][target_currency]["price"]
        await message.answer(
            f'You are going to converse
<b>{float(amount_for_converse):,}</b> units of
<b>{base_currency}</b> into'
            f'
<b>{target_currency}</b>\n\n'f"{float(amount_for_converse):,}
<b>{base_currency}</b> equals {conversion:,.2f}
<b>{target_currency}</b>",

reply_markup=keyboards.currency_exchange_keyboard_expanded())
        change_flag[0] = False
        return await state.set_state(ExchangeCurrency.next_step)
    else:
        change_flag[0] = False
        await message.answer('Wrong data. Please, try again')
        return await state.set_state(ExchangeCurrency.amount)


@dp.message(ExchangeCurrency.next_step)
async def next_step(message: types.Message, state: FSMContext,
change_flag: list[bool]):
    await state.update_data(step=message.text)
    if message.text.lower() == "change base currency":
        change_flag[0] = True
        await state.set_state(ExchangeCurrency.base_currency)
        return await message.answer(text='Choose a new base
currency on your keyboard',

reply_markup=keyboards.currency_exchange_keyboard())
    if message.text.lower() == 'change target currency':
        change_flag[0] = True
        await state.set_state(ExchangeCurrency.target_currency)
        return await message.answer(text='Choose a new target
currency on your keyboard',

reply_markup=keyboards.currency_exchange_keyboard())


@dp.message()
async def wrong_input(message: types.Message):
    await message.answer('Wrong data. Please, try again')


async def main():
    await bot.delete_webhook(drop_pending_updates=True)
    await dp.start_polling(bot, change_flag=[False])
```

```
if __name__ == "__main__":
    asyncio.run(main())
```

Файл keyboards.py

Текст программы

```
from aiogram.types import (
    InlineKeyboardMarkup,
    InlineKeyboardButton,
    ReplyKeyboardMarkup,
    KeyboardButton
)
from aiogram.utils.keyboard import ReplyKeyboardBuilder

main_keyboard = ReplyKeyboardMarkup(
    keyboard=[
        [
            KeyboardButton(text="Currency exchange prices"),
            KeyboardButton(text="Cryptocurrency info"),
        ]
    ],
    resize_keyboard=True,
    selective=True,
    one_time_keyboard=True,
    input_field_placeholder="Choose an option from menu..."
)

currencies = [
    "USD", "EUR", "RUB",
    "CNY", "JPY", "QAR",
    "XAU", "XAG", "XPT"
]


def currency_exchange_keyboard():
    keyboard = ReplyKeyboardBuilder()
    [keyboard.button(text=fiat) for fiat in currencies]
    keyboard.button(text='Back')
    keyboard.adjust(*[3] * 3, 1)
    return keyboard.as_markup(resize_keyboard=True)


def currency_exchange_keyboard_expanded():
    keyboard = ReplyKeyboardBuilder()

    [keyboard.button(text=fiat) for fiat in currencies]
    keyboard.button(text='Change base currency')
    keyboard.button(text='Change target currency')
    keyboard.button(text='Back')
    keyboard.adjust(*[3] * 4)
    return keyboard.as_markup(resize_keyboard=True)
```

Файл StateMachine.py

Текст программы

```python
from aiogram.fsm.state import StatesGroup, State


class ExchangeCurrency(StatesGroup):
    base_currency: str = State()
    target_currency: str = State()
    amount: str = State()
    next_step : str = State()


class Menu(StatesGroup):
    option: str = State()
    menu = ["Currency exchange prices", "Cryptocurrency info"]
```

Файл api.py

Текст программы

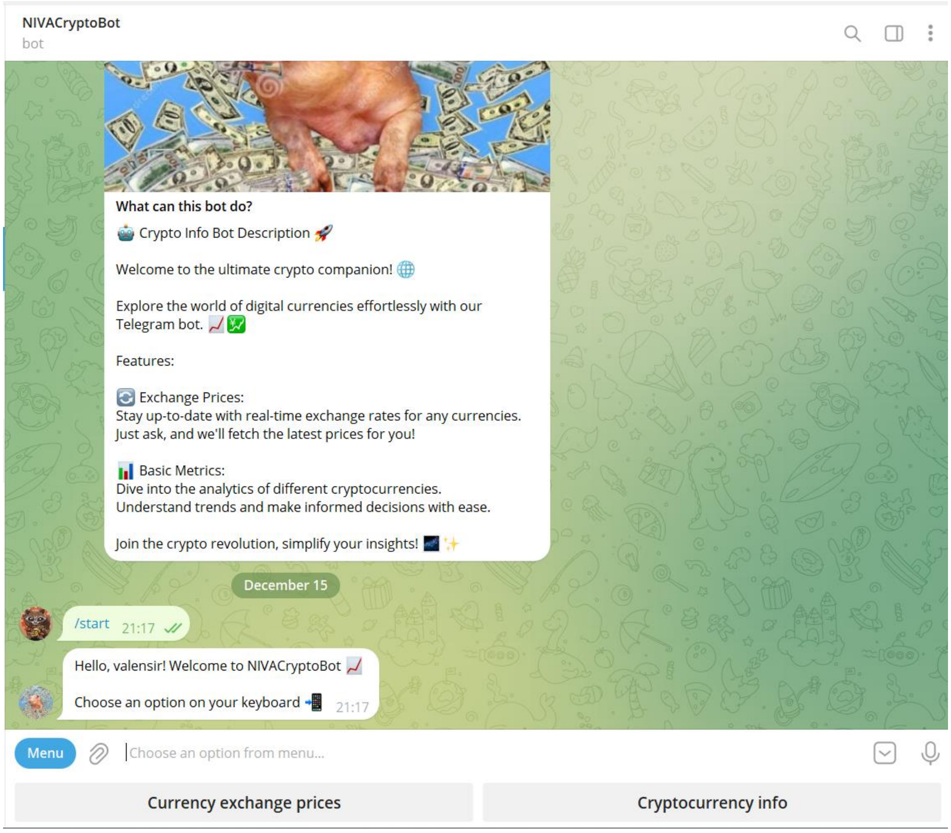
```python
import json
from typing import Dict

from requests import Session
from configs.config import configuration

api_key_coin = configuration.API_KEY_COIN.get_secret_value()
api_key_crypto = configuration.API_KEY_CRYPTO.get_secret_value()


def api_crypto(parameters: Dict):
    url = "https://pro-api.coinmarketcap.com/v2/tools/price-conversion"

    headers = {
        'Accepts': 'application/json',
        'X-CMC_PRO_API_KEY': api_key_crypto
    }

    session = Session()
    session.headers.update(headers)
    response = session.get(url, params=parameters)

    return json.loads(response.text)
```

Файл config.py

Текст программы

```
from pydantic_settings import BaseSettings, SettingsConfigDict
from pydantic import SecretStr


class Settings(BaseSettings):
    BOT_TOKEN: SecretStr
    API_KEY_COIN: SecretStr
    API_KEY_CRYPTO: SecretStr

    model_config = SettingsConfigDict(
        env_file="../static/.env", env_file_encoding="utf-8"
    )


configuration = Settings()
```

Результаты



What can this bot do?

🤖 Crypto Info Bot Description 🚀

Welcome to the ultimate crypto companion! 🌐

Explore the world of digital currencies effortlessly with our Telegram bot. 📈💹

Features:

🔄 Exchange Prices:
Stay up-to-date with real-time exchange rates for any currencies. Just ask, and we'll fetch the latest prices for you!

📊 Basic Metrics:
Dive into the analytics of different cryptocurrencies. Understand trends and make informed decisions with ease.

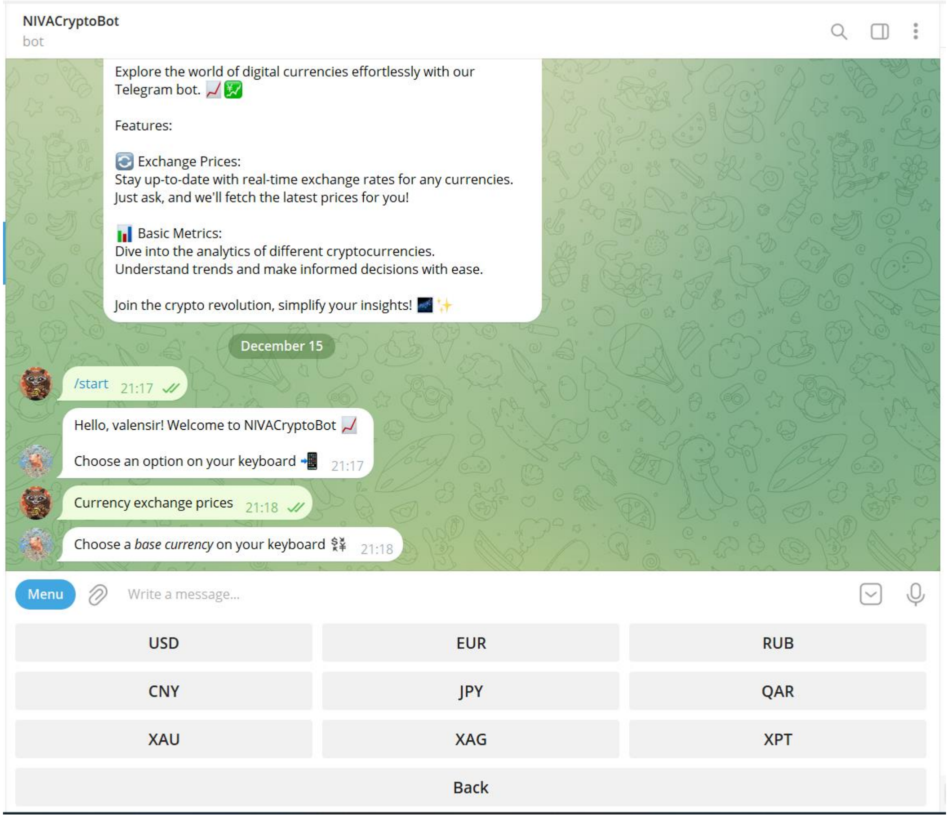
Join the crypto revolution, simplify your insights! 🌌✨

December 15

/start 21:17

Hello, valensir! Welcome to NIVACryptoBot 📈

Choose an option on your keyboard 📲 21:17



Explore the world of digital currencies effortlessly with our Telegram bot. 📈💹

Features:

🔄 Exchange Prices:
Stay up-to-date with real-time exchange rates for any currencies. Just ask, and we'll fetch the latest prices for you!

📊 Basic Metrics:
Dive into the analytics of different cryptocurrencies. Understand trends and make informed decisions with ease.

Join the crypto revolution, simplify your insights! 🌌✨

December 15

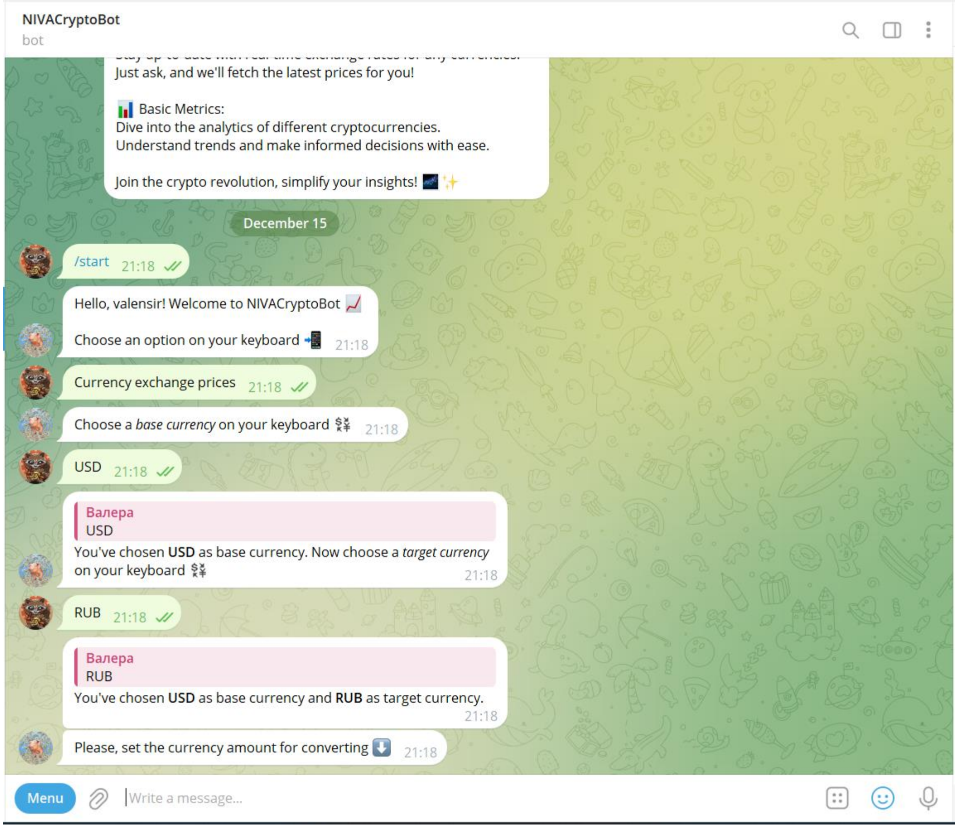
/start 21:17

Hello, valensir! Welcome to NIVACryptoBot 📈

Choose an option on your keyboard 📲 21:17

Currency exchange prices 21:18

Choose a *base currency* on your keyboard 💱 21:18



NIVACryptoBot
bot

Stay up-to-date with real-time exchange rates for any currencies.
Just ask, and we'll fetch the latest prices for you!

📊 Basic Metrics:
Dive into the analytics of different cryptocurrencies.
Understand trends and make informed decisions with ease.

Join the crypto revolution, simplify your insights! 🌌✨

December 15

/start 21:18 ✓✓

Hello, valensir! Welcome to NIVACryptoBot 📈

Choose an option on your keyboard 📲 21:18

Currency exchange prices 21:18 ✓✓

Choose a *base currency* on your keyboard 💱 21:18

USD 21:18 ✓✓

> Валера
> USD

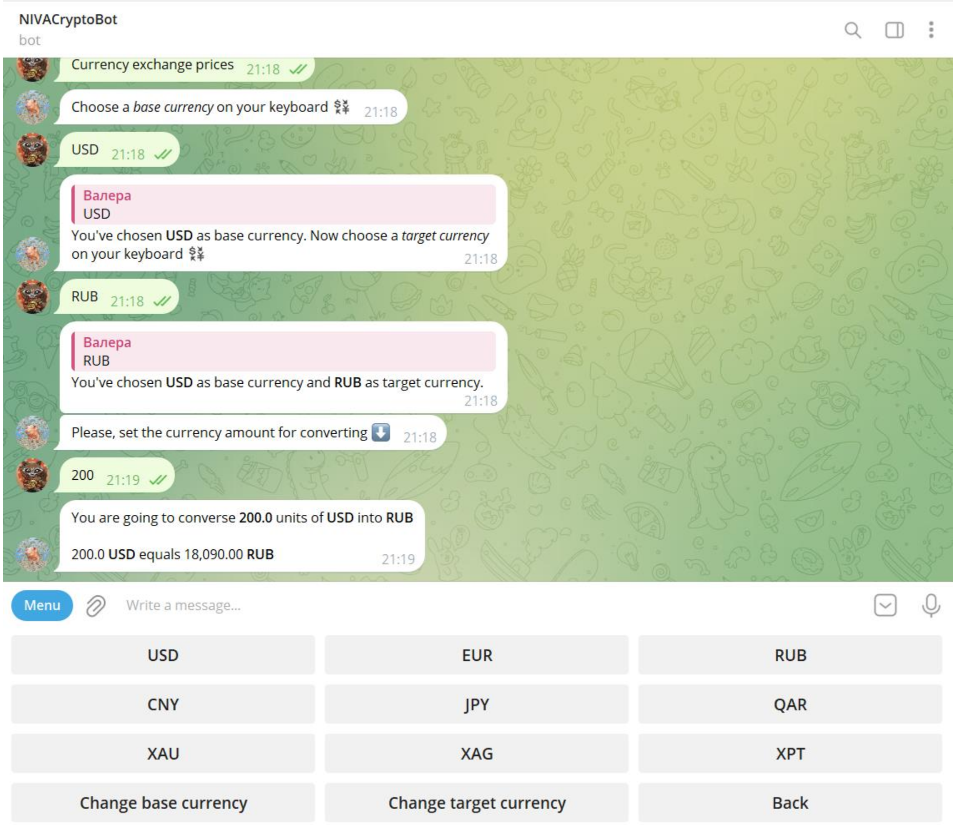
You've chosen **USD** as base currency. Now choose a *target currency* on your keyboard 💱 21:18

RUB 21:18 ✓✓

> Валера
> RUB

You've chosen **USD** as base currency and **RUB** as target currency. 21:18

Please, set the currency amount for converting ⬇️ 21:18



NIVACryptoBot
bot

Currency exchange prices 21:18 ✓✓

Choose a *base currency* on your keyboard 💱 21:18

USD 21:18 ✓✓

> Валера
> USD

You've chosen **USD** as base currency. Now choose a *target currency* on your keyboard 💱 21:18

RUB 21:18 ✓✓

> Валера
> RUB

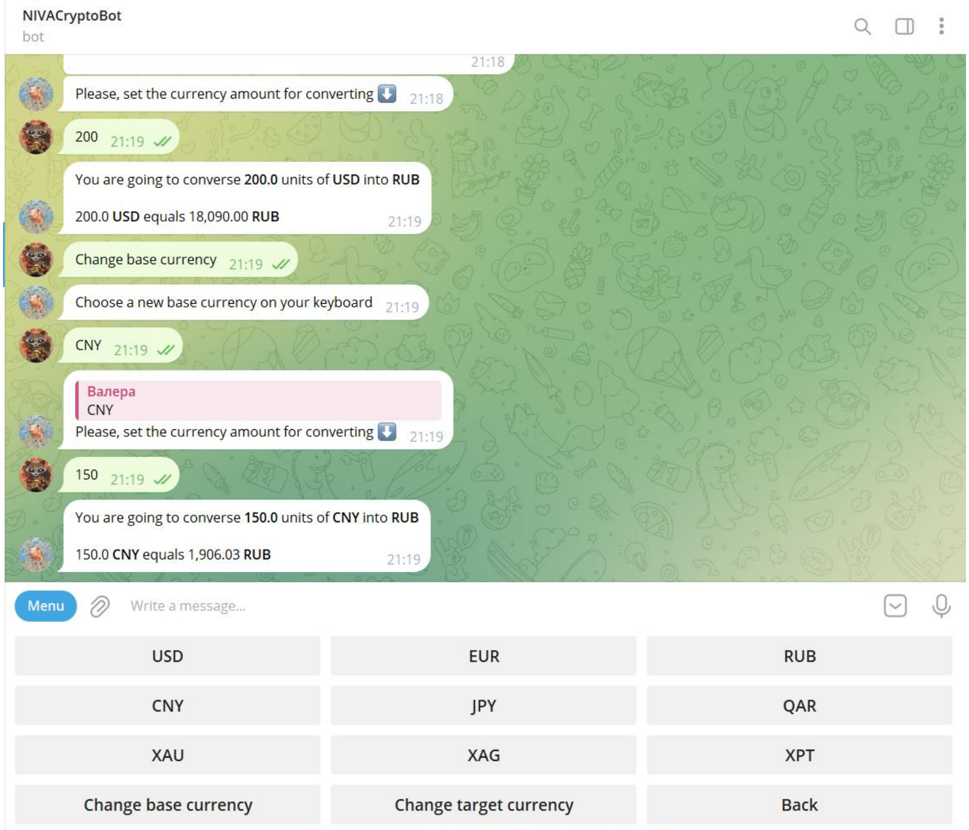
You've chosen **USD** as base currency and **RUB** as target currency. 21:18

Please, set the currency amount for converting ⬇️ 21:18

200 21:19 ✓✓

You are going to converse **200.0** units of **USD** into **RUB**

200.0 **USD** equals 18,090.00 **RUB** 21:19



NIVACryptoBot
bot

21:18

Please, set the currency amount for converting ⬇️ 21:18

200 21:19 ✓✓

You are going to converse **200.0** units of **USD** into **RUB**

200.0 **USD** equals 18,090.00 **RUB** 21:19

Change base currency 21:19 ✓✓

Choose a new base currency on your keyboard 21:19

CNY 21:19 ✓✓

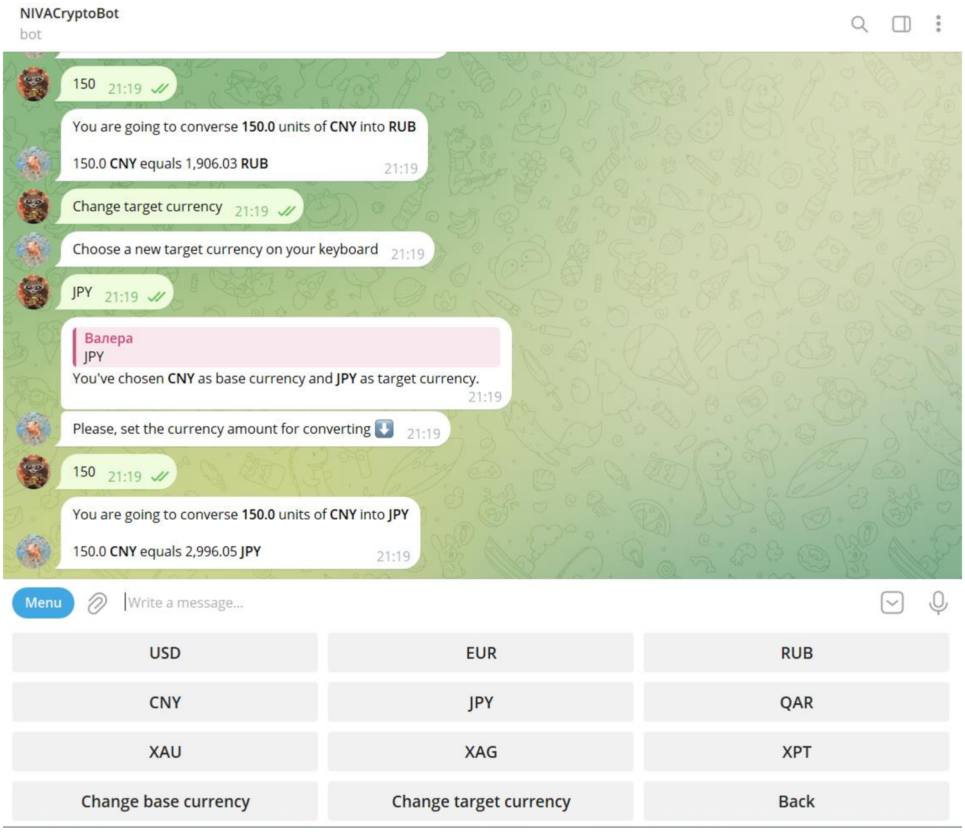
> Валера
> CNY

Please, set the currency amount for converting ⬇️ 21:19

150 21:19 ✓✓

You are going to converse **150.0** units of **CNY** into **RUB**

150.0 **CNY** equals 1,906.03 **RUB** 21:19



NIVACryptoBot
bot

150 21:19 ✓✓

You are going to converse **150.0** units of **CNY** into **RUB**

150.0 **CNY** equals 1,906.03 **RUB** 21:19

Change target currency 21:19 ✓✓

Choose a new target currency on your keyboard 21:19

JPY 21:19 ✓✓

> Валера
> JPY

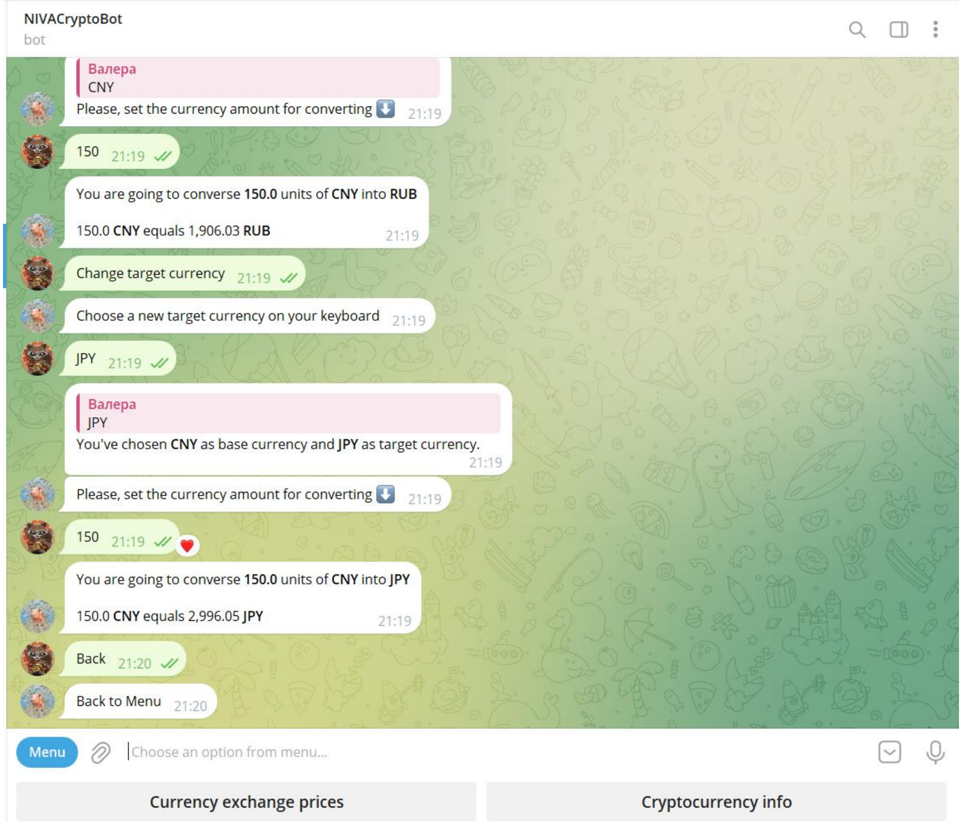
You've chosen **CNY** as base currency and **JPY** as target currency. 21:19

Please, set the currency amount for converting ⬇️ 21:19

150 21:19 ✓✓

You are going to converse **150.0** units of **CNY** into **JPY**

150.0 **CNY** equals 2,996.05 **JPY** 21:19



NIVACryptoBot
bot

> Валера
> CNY

Please, set the currency amount for converting ⬇️ 21:19

150 21:19 ✓✓

You are going to converse **150.0** units of **CNY** into **RUB**

150.0 **CNY** equals 1,906.03 **RUB** 21:19

Change target currency 21:19 ✓✓

Choose a new target currency on your keyboard 21:19

JPY 21:19 ✓✓

> Валера
> JPY

You've chosen **CNY** as base currency and **JPY** as target currency. 21:19

Please, set the currency amount for converting ⬇️ 21:19

150 21:19 ✓✓ ❤️

You are going to converse **150.0** units of **CNY** into **JPY**

150.0 **CNY** equals 2,996.05 **JPY** 21:19

Back 21:20 ✓✓

Back to Menu 21:20