

# מסלול DEVOPS

שם המרצה: אלכס קומנוב





# PYTHON COURSE

# Python Challenge



# Challenge

You are going to build a **Telegram bot**.





# Challenge

First of all, create a new Python project. Also, create a new GitHub repository and connect your Python project to the repository.



git



Python

# Challenge

Make sure you have or install the required packages (use **pip install** or **pip3 install**):

- aiogram
- python-dotenv



**.ENV**



**AsyncIOgram**  
Telegram

# Challenge

In the project root directory create a **.gitignore** file and fill it with the following:

```
.gitignore x
.idea/

**/__pycache__

.env
```

# Challenge

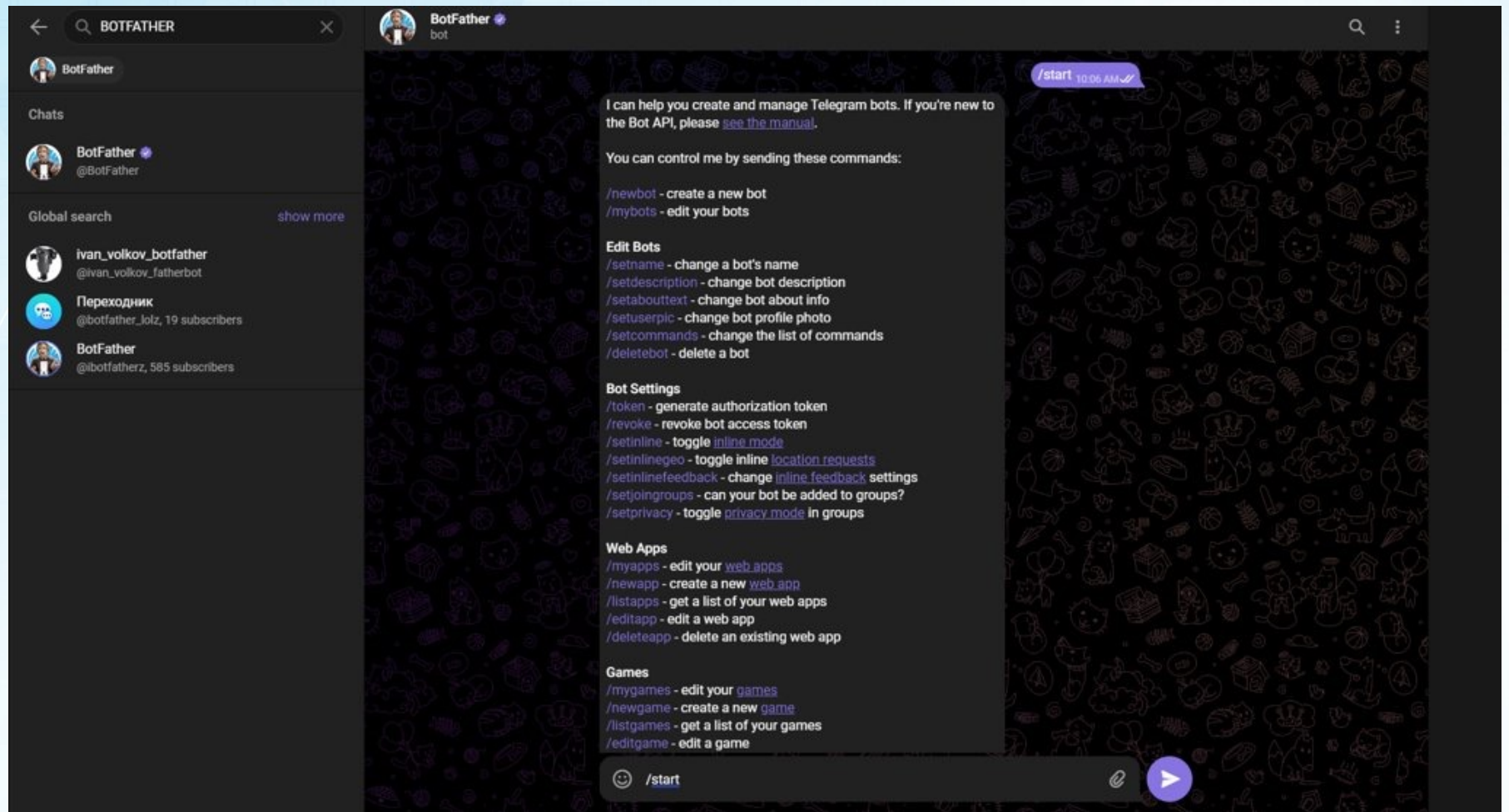
Search for **@BotFather** inside a telegram and type **/start** to get a menu.





# Challenge

Choose a **/newbot** option



# Challenge

- Choose the following name for the bot: **INT Python Challenge**
- Choose the following name for the bot: **int\_python\_challenge\_bot**
- Make sure you're getting a message about a successful creation:

Done! Congratulations on your new bot. You will find it at [t.me/SeatAlexBot](https://t.me/SeatAlexBot). You can now add a description, about section and profile picture for your bot, see [/help](#) for a list of commands. By the way, when you've finished creating your cool bot, ping our Bot Support if you want a better username for it. Just make sure the bot is fully operational before you do this.

Use this token to access the HTTP API:

**6964513723:AAG\_W1dUtPgro1dFZfWWcF6jrWZJJVIs1fo**

Keep your token secure and store it safely, it can be used by anyone to control your bot.

For a description of the Bot API, see this page:

<https://core.telegram.org/bots/api>

# Challenge

In the project root directory create a **.env** file. Then, copy a token from the Telegram and paste a token into the **.env** file. Store it as a **BOT\_TOKEN** variable like the following:

```
.env x
1 BOT_TOKEN=6373447736:AAGpF_s0T0L_7DFn2MWBvFE1Q-5_TutwC60
2
3
```



# Challenge

Create a file (**main.py**) inside the project and make sure to add the following:

- Command **/start** handler
- **Any** command handler that will just send back the same message of any type. Don't forget to include the errors handler inside a function.
- Function **main()** that will include the **INFO** logger and bot polling.
- A function **asyncio.run()** that will execute the **main()** function.
- You can take the [following python file](#) as an example.

If anything is correct, executing a file will show the following output:

```
INFO:aiogram.dispatcher:Start polling
INFO:aiogram.dispatcher:Run polling for bot @alex_komanov_bot id=6373447736 - 'INT DEMO ROBOT'
```

# Python Challenge

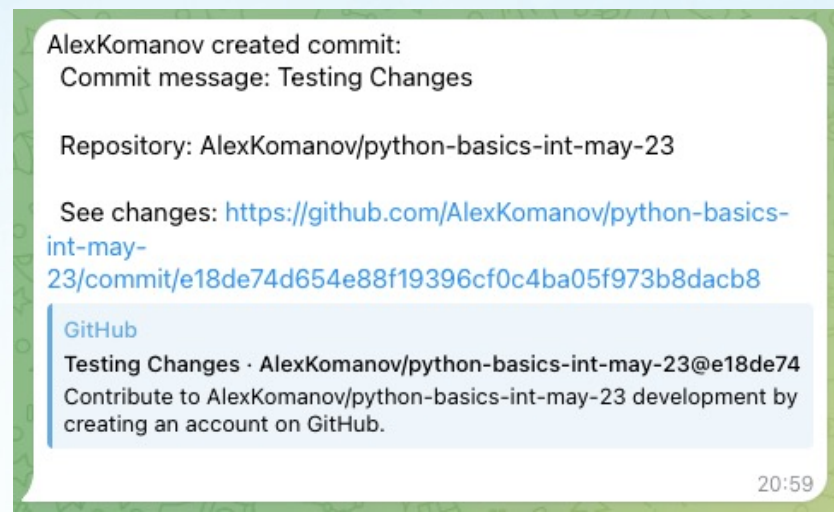


# Challenge

As a bonus part, you can add an integration with the **GitHub Actions Workflow**:

- See the relevant part of the lesson where it was presented.
- **Don't forget** to create relevant **secret** variables inside a repository.
- You can use the [following YAML file](#) as an example.

If anything is correct, every push to the repository will trigger the following message:





# Challenge

- You'll need to perform at least the main part. The bonus part is not a mandatory, but it is a good experience, so try to do it also.
- Add several screenshots that present how the bot is working. Create a folder inside the project and store the screenshots there.
- Commit and push the project (including the screenshots).
- Add the GitHub link (make sure that a repository is **public**) to the task inside the LMS platform and submit.

