KB-CAPTCHA User Manual

Table of Contents

Table of Contents....................1

Using the KB-CAPTCHA Solution....................2

Before you start....................2

Installation....................3

Running the Solution....................3

Stopping the Containers....................3

Uninstalling....................4

Running the Non-containerized Solution....................5

Windows....................6

Linux....................7

Data Methodology Documentation....................8

Launching the Jupyter Notebook....................8

Extracting the Data (Windows)....................8

Extracting the Data (Linux)....................9

Installing and Running Jupyter (All platforms)....................9

Automated Tester....................10

Building the Docker Images....................11

Using the KB-CAPTCHA Solution

The KB-CAPTCHA solution provides two components that combine to form the demonstration. First is the Python machine learning backend. This actually performs the calculations. The other part is a sample frontend implementation that uses the service through a web interface. This is provided as a reference should you want to use the backend yourself as well as to provide a way to test the functionality of the product.

To run the backend and frontend demo on your local machine, follow the steps in the "**installation**" and **"running the solution**" sections below. After you have finished, you can stop the processes with the commands in the "**stopping the container**" section. To remove the solution files from your computer, run the commands in the "**uninstalling**" section.

For *advanced users*, you can choose to instead run the solution files directly without Docker. Instructions for that are in the "*Running the Non-containerized Solution*" portion of this manual.

Before you start

You will need Docker installed on your machine. Instructions for installing it vary by operating system, but you can get it for most x86_64 bit operating systems here. **For Windows, launch Docker Desktop before opening the command line.** This starts a background process that's needed for the command line to function.

If you are acquainted with Docker Desktop you can use that. However, the instructions below will assume you are using the command line. **On Unix-like systems, you will likely need to run the docker commands as root**. You can attempt to do it without root permissions, but it may break your configuration. On Windows, use an administrator command prompt. To open one:

- Go to your start menu
- Search "cmd"
- Right click "command prompt"
- Click "Run as Administrator"

The commands below are not platform specific and will work on any Docker install.

Installation

To install the front/backend, use the following commands.

```
❖ docker pull dowdyj/kb-captcha-frontend-demo

❖ docker pull dowdyj/kb-captcha-backend

❖ docker network create --driver=bridge kbcaptcha-network
```

Running the Solution

After installing, you can run the solution at any time with the following commands:

```
❖ docker run -d -p 5000:5000 --network=kbcaptcha-network
  --name=kbcaptcha-python-backend dowdyj/kb-captcha-backend

❖ docker run -d -p 8080:8080 --network=kbcaptcha-network
  --name=kbcaptcha-java-frontend dowdyj/kb-captcha-frontend-demo
```

Last, visit http://localhost:8080 in your browser to see the webpage. If it fails to load, wait a moment and refresh the page.

Stopping the Containers

To stop the solution, you can run these commands:

```
❖ docker stop kbcaptcha-java-frontend

❖ docker stop kbcaptcha-python-backend

❖ docker rm kbcaptcha-java-frontend

❖ docker rm kbcaptcha-python-backend
```

Uninstalling

After you've stopped the containers, you can remove the files from your system and clean up with these commands:

- **`docker network rm kbcaptcha-network`**
- **`docker image rm dowdyj/kb-captcha-frontend-demo`**
- **`docker image rm dowdyj/kb-captcha-backend`**

Running the Non-containerized Solution

For most users, the easier install is the Docker-based install described at the beginning of the manual. However, if you would like to run the solution without Docker for development purposes, you will need to have the following installed on your computer:

- Java SDK & Java JRE 17 or above
  - You can check with **`javac  --version`** and **`java --version`**

- Python 3.10 or above
  - You can check with **`python3 --version`**

- Lua 5.4 or above
  - You can check with **`lua -v`**

- Any modern version of Git
  - You can check if git is installed with **`git --version`**

Windows

1. Clone the repository with:
   `git clone https://github.com/DowdyJ/KB-CAPTCHA.git`

2. Download the dataset from here:
   https://userinterfaces.aalto.fi/136Mkeystrokes/data/Keystrokes.zip

3. Extract the zip contents into Data\raw in the project folder.
   a. Take care to extract the contents, not a nested folder

4. Move the `metadata_participants.txt` and `readme.txt` files out of Data/raw (We don't need them, they just need not to be with the data)

5. Go to the DataCleaner\ folder at the root of the project.

6. Run the command **`lua create_clean_csv.lua`**
   Wait for the command to finish. It will take a while.

7. Now, go into the KeyInputTester\ and open up a terminal in that folder. Run the following commands:

   a. pip install virtualenv
   b. python3 -m venv .venv
   c. .venv\Scripts\activate.bat
   d. pip install -r requirements.txt
   e. python3 main.py

   To run this in the future after initial install, you only need to run:

   a. .venv\Scripts\activate.bat
   b. python3 main.py

8. Open a new terminal inside of Frontend\kbcaptcha-demo. Run the following commands:

   a. .\mvnw.cmd install
   b. .\mvnw.cmd spring-boot:run

9. Now, you can navigate to http://localhost:8080 to see the demo page.

Linux

1. Clone the repository with:
   `git clone https://github.com/DowdyJ/KB-CAPTCHA.git`

2. Run the **`get_raw_data.sh`** script in the Data/ folder.
   a. You may need to run **`chmod +x get_raw_data.sh`**

3. Move into the DataCleaner/ folder at the root of the project.

4. Run the command **`lua create_clean_csv.lua`**
   Wait for the command to finish. It will take a while.

5. Now, go into the KeyInputTester/ and open up a terminal in that folder. Run the following commands:

   a. pip install virtualenv
   b. python3 -m venv .venv
   c. source .venv/bin/activate
   d. pip install -r requirements.txt
   e. python3 main.py

   To run this in the future after initial install, you only need to run:

   c. source .venv/bin/activate
   d. python3 main.py

6. Open a new terminal inside of Frontend/kbcaptcha-demo. Run the following commands:

   a. ./mvnw install
      i. You may need to run **`chmod +x mvnw`**
   b. ./mvnw spring-boot:run

7. Now, you can navigate to http://localhost:8080 to see the demo page.

Data Methodology Documentation

The repository for the solution contains a folder called “Docs”. Inside of this folder you can find a Jupyter Notebook that describes the data methodology and some analysis of the functionality of the solution. Additionally, there is an HTML render of the notebook. It is recommended that you launch the notebook with Jupyter for the best experience, but the HTML version is provided for convenience.

Launching the Jupyter Notebook

For the Jupyter Notebook version of the documentation, you need to have first downloaded, extracted and processed the data. If you have already done this as part of an earlier activity (i.e. running the solution without docker), you can skip to the “*Installing and Running Jupyter*” section.

Extracting the Data (Windows)

1. Clone the repository with:
   `git clone https://github.com/DowdyJ/KB-CAPTCHA.git`

2. Download the dataset from here:
   https://userinterfaces.aalto.fi/136Mkeystrokes/data/Keystrokes.zip

3. Extract the zip contents into Data\raw in the project folder.
   a. Take care to extract the contents, not a nested folder

4. Move the `metadata_participants.txt` and `readme.txt` files out of Data/raw (We don’t need them, they just need not to be with the data)

5. Go to the DataCleaner\ folder at the root of the project.

6. Run the command **`lua create_clean_csv.lua`**
   Wait for the command to finish. It will take a while.

Extracting the Data (Linux)

1. Clone the repository with:
   `git clone https://github.com/DowdyJ/KB-CAPTCHA.git`

2. Run the **`get_raw_data.sh`** script in the Data/ folder.
   a. You may need to run **`chmod +x get_raw_data.sh`**

3. Move into the DataCleaner/ folder at the root of the project.

4. Run the command **`lua create_clean_csv.lua`**
   Wait for the command to finish. It will take a while.

Installing and Running Jupyter (All platforms)

1. You need to have Python 3.10 or above
   a. You can check with **`python3 --version`**

2. Navigate to the Docs/ folder and open a terminal.

3. Run **`python -m venv .venv`**

4. Activate the virtual environment
   a. For Windows, use **`.venv\Scripts\activate.bat`**
   b. For Linux, use `source` **`.venv/bin/activate`**

5. Run **`pip install -r requirements.txt`**

6. Run **`jupyter notebook MachineLearningDesignDoc.ipynb`**

7. A new tab should open up in your browser that shows the document.

Automated Tester

The automated tester uses the browser automation software Puppeteer to simulate a variety of bot input styles. It serves as both a means to test the solution, as well as an interesting demonstration.

To run the solution, you must first have the backend and frontend demo page running. To do this, you can either use the Docker setup described in the "*Using the KB-CAPTCHA Solution*" section **or** the locally compiled version using the instructions in the "*Running the Non-containerized Solution*" section.

You can check by seeing if the solution is running by visiting http://localhost:8080/ and checking if it loads.

Lastly, you will need **Node.js version 18+** installed. You can check if it's installed with **`node --version`**. You can download it here.

After you have got the solution running and Node.js installed, do the following to use the automated testing software:

1. Navigate to the AutomatedInputTester folder.
2. Open a command prompt / shell in the current folder.
3. Run **`npm install`**
4. Run **`npx tsc`**
5. Run **`node index.js`**
6. The browser will now open and perform the tests. You just need to wait and watch it do so. It is done in headful mode so that you can see the test performed.

Building the Docker Images

If you would like to build the Docker images locally, it’s very simple. First, make sure Docker is installed. You can find instructions for this in the “*Using the KB-CAPTCHA Solution*” section.

After Docker has been installed, you can run the following commands to build the images. You should run the commands from inside the root folder of the solution. If you are on Windows, replace the slashes with backslashes in the commands below. Otherwise, the commands are the same.

As with other Docker commands, these may need to be run as **root / administrator**.

-
  ```
  docker build -t kbcaptcha-backend KeyInputTester/
  ```

-
  ```
  docker build -t kbcaptcha-frontend-demo Frontend/kbcaptcha-demo/
  ```