Setting up the Environment

Welcome to the Rasa Certification Workshop - we're glad to have you joining us! Before we begin the workshop, we have a few things to set up to make sure you're ready to go on day 1. In this session, we'll set up the environment, by installing all required tools and dependencies and downloading the starter project.

Use this handout to follow along with the video walkthrough.

Reminder: Please fill out this form and provide your Google email address. We need this to give you access to our Google Platform account later in the week, when we deploy our assistants.

I. Pre-workshop checklist:

Before you begin, you'll need:

- ❏ GitHub account
- ❏ IDE or Text editor
- ❏ Python 3.6 or 3.7

Setting up the project:

- ❏ Virtual environment
- ❏ Financial Demo bot
    - ❏ Fork the repo
    - ❏ Clone a local copy
    - ❏ Install dependencies (including Rasa Open Source)
- ❏ Run the bot!
- ❏ Telegram account

II. Setting up the Virtual Environment

Why set up a virtual environment? Virtual environments let you scope Python packages to a project directory, instead of installing the package on the system (global) level. This means you can use different versions of a package for different projects and helps to prevent conflicts.

We're using venv to create the virtual environment, because it's built into the Python standard library. If you prefer, you can use Anaconda or virtualenv.

Steps:

A. Create a new directory for the workshop project
    a. mkdir rasa-workshop

B. Navigate into the directory
    a. cd rasa-workshop
C. Create the virtual environment
    a. python3 -m venv ./venv
D. Activate the virtual environment
    a. (Mac/Linux) source ./venv/bin/activate
    b. (Windows) .\venv\Scripts\activate

III. Fork and clone the project

For this workshop, we're using the Financial Services demo bot, an open source banking assistant.

Steps:

A. Fork the repository: https://github.com/RasaHQ/financial-demo
B. Clone the repo to create a local copy on your computer:
    a. git clone <url to the repository forked to your GitHub account>

IV. Git Cheatsheet

We'll be using GitHub to manage different versions of the assistant, and to port the assistant's code into Rasa X later in the workshop. Here are a few common commands we'll be using:

git remote -v	Show remote repositories
git remote add upstream https://github.com/RasaHQ/financial-demo.git	Set upstream repo so we can fetch changes from the repo we forked from
git fetch <remote name>	Download new changes from remote repo
git branch -a	List all branches
git checkout <existing branch name>	Switch to a different branch
git status	Track uncommitted changes
git stash	Temporarily shelve uncommitted changes (so you can switch to a different branch)

V. Install Dependencies

To install Rasa Open Source (along with the other packages required by the Financial Demo assistant), we'll use the requirements.txt included in the repository.

Steps:

A. Navigate into the Financial Services demo assistant folder
    a. cd financial-demo
B. Install dependencies
    a. pip install -r requirements.txt

VI. Download the SpaCy Language Model

Steps:

A. Download the model:
    a. python3 -m spacy download en_core_web_md
B. Link the model:
    a. python -m spacy link en_core_web_md en

VII. Optional - Download Docker

The Financial Services demo bot uses Docker to run Duckling locally. Duckling is an entity extractor for dates (more on that later in the workshop).

If you want to follow along, you can download Docker using the instructions here.

If you run into trouble, don't worry - we'll be providing a live Duckling server later on that you can use instead of running Docker.

VIII. Run the Assistant

Before we can start talking to the assistant, we need to train the model and start the servers.

Steps:

A. Train the model
    a. rasa train
B. Open a new terminal window and start the Duckling server
    a. docker run -p 8000:8000 rasa/duckling

C. Once the model has finished training, go back to the first terminal window and start the action server and Rasa shell session
    a. rasa run actions & rasa shell
D. Talk to the bot! Check the README to see the questions the assistant can recognize.

IX. Create a Telegram account

Steps:

A. Download Telegram for desktop or mobile
B. Create your account
C. Log in

X. Office Hours

Have questions? Join our Slack workspace (look out for an invite soon), where we'll be hosting an Office Hours session to answer your setup questions. We'll have presenters from the workshop available for Office Hours on **May 7, 16:00-18:00 CEST.**