# How to Start Battykoda

This guide will walk you through setting up and running a Battykoda server. The setup ensures that Flask version 2.x is used within a virtual environment.

**## Step 1: Create and Activate a Virtual Environment**  
  
To keep dependencies isolated, it is best to use a virtual environment.  
  
### On macOS/Linux:  
```  
python3 -m venv venv  
source venv/bin/activate  
```  
  
### On Windows:  
```  
python -m venv venv  
venv\Scripts\activate  
```

**## Step 2: Install Flask (Version 2.x)**  
After activating the virtual environment, install Flask:  
```  
pip install 'Flask>=2.0,<3.0'  
```

**## Step 3: Verify Installation**  
To confirm that Flask 2.x is installed, run:  
```  
python -c "import flask; print(flask.\_\_version\_\_)"  
```  
Ensure the output shows a version between 2.0 and 2.9.

**## Step 4: Run the Flask Server**  
Navigate to the directory containing `main.py` and start the server:  
```  
python main.py  
```  
  
If `main.py` is correctly configured, the server should start, and you will see output similar to:  
```  
 \* Running on http://127.0.0.1:8060/ (Press CTRL+C to quit)  
```

**## Step 5: Access the Server**  
Open a web browser and visit:  
```  
http://127.0.0.1:8060/  
```  
  
If `main.py` defines routes properly, you should see the expected web page or API response.

**## Additional Notes**- If Flask is not found, ensure you have activated the virtual environment.  
- If you need to install additional dependencies, use `pip install <package>` within the virtual environment.

# Folder structure:

The server is replicating the home folder user structure. Additionally, there should be user called data. In the home folder of that user should be a folder called battykoda, with two subfolders, called static and tempdata. Each species that is to be used should have an ABC.jpg and ABC.txt file in the static folder. Once those files are created, subfolders of other users called ABC will be visible. An example for an ABC.txt and ABC.jpg files are provided in the git repository for Efuscus.

# Pickle generation and user interface

Once you have the Battycoda setup for your species (labels and template), then you can convert your files to be read by the software. The files need to be converted to pickle files. To do this you need the wave file and an xlsx file with acoustic parameters, most importantly start and end time of each call. The [file name] script is used to create a pickle file from the xlsx file. Once you have converted your files to pickle files, you will be able to use Battycoda to travel through your directory and open the files for analysis.

Below is the interface you will see once you have chosen your file in Battycoda. You will be prompted to enter your username. On the left you will see a spectrogram zoomed in to one specific call, which is the call you are labeling. The spectrogram on the right is a zoomed-out version of the same call, to better understand the context of the call.

Under the spectrogram number, you will see multiple options you can change. Increase the contrast if you are having trouble seeing calls clearly because they are too quiet and decrease if they are too loud. The channel observed is regarding multiple microphones in each recording. If you are running an experiment with multiple microphones, you will have multiple channels. You can see the spectrograms for the different channels when you scroll down. If one channel is clearer than the others, choose this as your main channel.

You see a call. Choose the correct label and hit either or next call. This will automatically save your label. If you change the settings, you can hit update. This also tends to change to the next call, so keep an eye on that. You can use the undo button to go back.

Underneath the left spectrogram, you will see a number labeled confidence. Once we have enough files analyzed, this will be how confident the classifier is at identifying this call. We will be able to set a level of confidence and only check the calls under the given threshold.