**Overview**

The labs in this repository use two coordinated environments: Python 3.11 for the reinforcement-learning notebooks and a Python 2.7 bridge that talks to the NAO robot through NAOqi. Follow the platform-specific steps below to obtain the NAOqi SDK, stand up both interpreters, and link them through the provided bridge code.

**Step 1 – Download the NAOqi SDK (all platforms)**

1. Sign in (or create an account) on SoftBank Robotics’ community/support portal.
2. Locate the NAOqi SDK that matches the NAO firmware on your robot (for example, NAOqi 2.1.x or 2.8.x). Download both:
   * The pynaoqi-python2.7-<version>-<platform> package for your workstation OS.
   * Any matching robot firmware/tools you need for the robot itself.
3. Extract the archive. Inside you’ll find a wheel or tarball such as pynaoqi-python2.7-<version>-win32.whl; note its path for installation.
4. Keep the extracted SDK handy. You’ll install the Python bindings on the workstation and copy any required firmware to the robot.

**Step 2 – Create the Python 2.7 NAO bridge environment**

The bridge depends on the NAOqi wheel plus Flask and simplejson in a Python 2.7 interpreter.

**Windows**

1. Install Miniconda or Anaconda (recommended because it still distributes Python 2.7).
2. Open “Anaconda Prompt” (or PowerShell) and create the environment:  
   conda create -n nao\_py27 python=2.7  
   conda activate nao\_py27
3. Install NAOqi:  
   pip install C:\path\to\pynaoqi-python2.7-<version>-win32.whl
4. Install the remaining bridge requirements:
5. cd <repo>\bridge\py27\_nao\_service
6. pip install -r requirements.txt
7. Verify with python -c "import naoqi" and keep this shell for running server.py.

**Linux**

1. Install Miniconda/Miniforge or use pyenv to provide Python 2.7. With Conda:  
   conda create -n nao\_py27 python=2.7  
   conda activate nao\_py27
2. Install the NAOqi SDK wheel:  
   pip install ~/Downloads/pynaoqi-python2.7-<version>-linux64.whl
3. Install bridge requirements:
4. cd /path/to/repo/bridge/py27\_nao\_service
5. pip install -r requirements.txt
6. Confirm python -c "import naoqi" succeeds.

**macOS**

1. Install Miniforge (ARM or Intel as appropriate) or Homebrew + pyenv. With Conda/Miniforge:  
   conda create -n nao\_py27 python=2.7  
   conda activate nao\_py27
2. Install the macOS wheel:  
   pip install ~/Downloads/pynaoqi-python2.7-<version>-mac64.whl
3. Install bridge dependencies:
4. cd /path/to/repo/bridge/py27\_nao\_service
5. pip install -r requirements.txt
6. Validate python -c "import naoqi".

**Step 3 – Create the Python 3.11 lab environment**

The repository provides scripts (and manual commands) to build a Python 3.11 virtual environment and install pinned requirements.

**Windows**

1. Open PowerShell in the repository root.
2. Run the helper script:  
   env\create\_env.ps1 -Name rl-labs-nao
3. (Optional) Register the Jupyter kernel once activated:  
   python -m ipykernel install --user --name rl-labs-nao --display-name "RL Labs (py311)"

**Linux**

1. In the repository root, execute:  
   bash env/create\_env.sh rl-labs-nao
2. Optionally register the kernel with the command above.

**macOS**

1. Use the same Bash script as Linux:  
   bash env/create\_env.sh rl-labs-nao
2. Register the IPython kernel if you plan to use Jupyter.

(If you prefer manual setup, follow the venv instructions in the README for your platform.)

**Step 4 – Wire up the bridge and notebooks**

1. **Python 2.7 service (robot or reachable host):**
2. conda activate nao\_py27
3. cd bridge/py27\_nao\_service
4. python server.py --host 0.0.0.0 --port 8000 --nao-ip <robot\_ip>

This exposes /say, /wave, /head, /posture, and /walk endpoints backed by NAOqi.

1. **Python 3.11 client (workstation):**
2. source .venv/rl-labs-nao/bin/activate # or .\.venv\rl-labs-nao\Scripts\activate
3. cd bridge/py311\_client
4. python send\_action.py --host http://<service\_ip>:8000 --action posture --payload StandInit

The helper handles rate-limiting and retries for you.

1. Inside the notebooks, select the “RL Labs (py311)” kernel and call the bridge helpers as shown in Lab 10 once both services respond.

**Step 5 – Final checks**

* Confirm the robot firmware matches the NAOqi SDK version before running motions.
* Keep the Python 2.7 environment active whenever you need the bridge server, and the Python 3.11 environment active for notebooks.
* Use the safety guidelines in the README (stable surface, rate limiting, supervised operation) whenever commanding the robot.