

# Setup RobotSwarm Guide

📅 Letzte Bearbeitung	@July 2, 2021
🏷️ Tags	Studium Vector
👇 noch relevant	Relevant

## Setup of the Programming Environment

Download and install Anaconda3

```
https://www.anaconda.com/products/individual
```

Run Conda-Propt and install Python 3.9 environment by using

```
conda create -n USER_CHOISE_LABEL python=3.9
```

Download and install Python IDE (PyCharm Pro Used, Community version and alternate IDE might work as well)

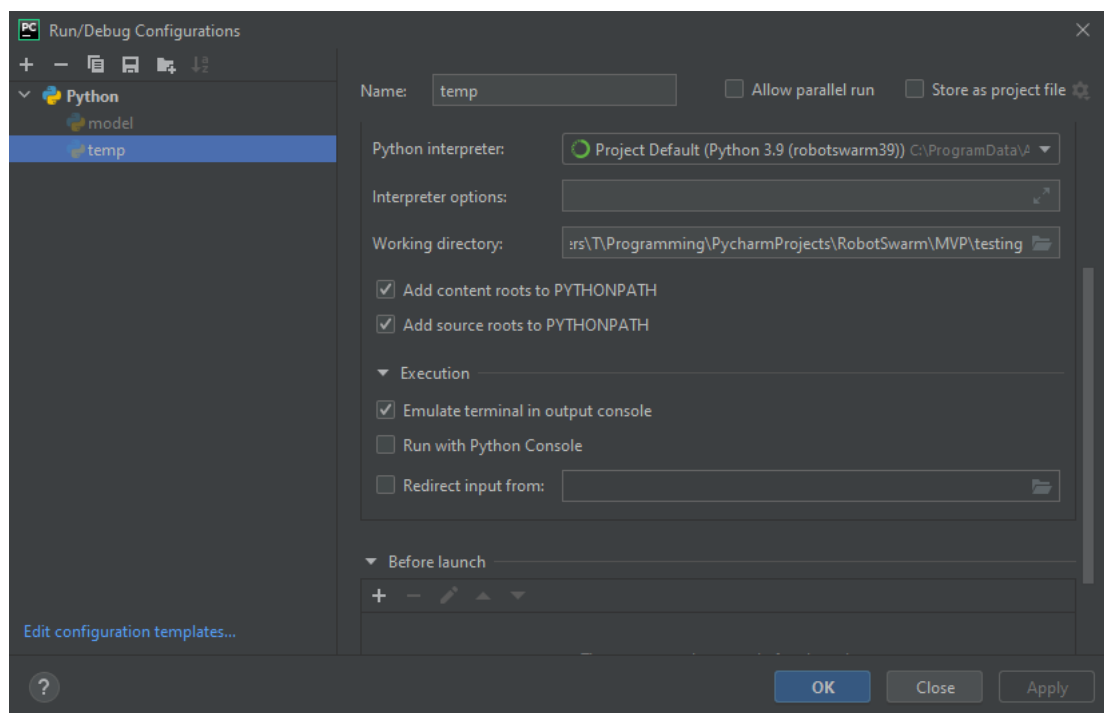
```
https://www.jetbrains.com/de-de/pycharm/download/#section=windows
```

Checkout Project from VCS (You need access to it, alternative you may want to ask for a copy to work on it)

```
https://dev.isst.fraunhofer.de/stash/scm/riotanav2/digitaltwin.git
```

Set Python Interpreter to Existing environment and use the one you created beforehand

Recommaded: Pycharm does not allow for Program interrupts by default, but you might want to allow a clean close of the Program while implementing/debugging (not the stop button - that acts like a kill). In the Run/debug Configuration you can set the terminal to interpret interrupt



## Setup of Openfire

Prepare the Multiagent-System by installing the Openfire-XMPP Server to your machine

<https://www.igniterealtime.org/downloads/#openfire>

After installation the setup will automatically show up in the Browser. In the Setup you want to keep most of the recommended options but you need to choose the "integrated Database" as for now we won't need the performance of an external service and we want to keep it simple. As Email u may use whatever Mail and password you want - My Choise is shown below:

### Administrator-Konto

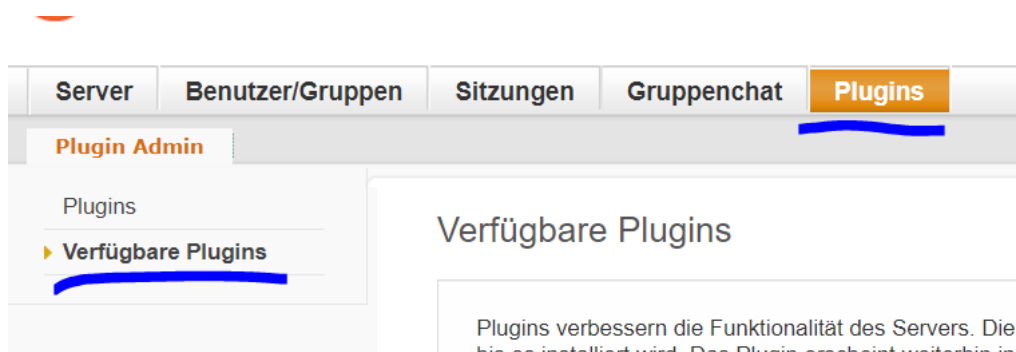
Einstellungen für das Konto des Systemadministrators eingeben (Benutzername von "admin"). Es ist wichtig, ein Passwort erraten werden kann -- zum Beispiel sollte es mindestens sechs Zeichen lang sein und einen Mix aus Zeichen und Zahl übersprungen werden, wenn bereits ein Konto für den Administrator angelegt wurde (nicht bei erstmaligem Setup).

<b>E-Mail Adresse des Administrators:</b>	<input type="text" value="roman.sliwinski@fh-bielefeld.de"/>
	<small>Eine gültige E-Mail Adresse für das Administrator-Konto.</small>
<b>Neues Passwort:</b>	<input type="text" value="Vector4win"/>
<b>Passwort bestätigen:</b>	<input type="text"/>
<div>Fortfahren</div>	

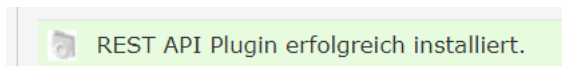
After setup you need to Login - Username is "admin" and password is the one you chose in the step before

Now we need to build the administration connection between the Server and our Program. At first we need to install the REST-API Plugin to the server - go to the given adress and search for the API. Click on the install-Button next to it.

<http://localhost:9090/available-plugins.jsp>



If everything worked you will see this message:



Next we need to process some settings in the API to ensure that it is configured in the right way and to copy shared secret Key - we will need this to authorise against the server later:

**REST API Properties**

Use the form below to enable or disable the REST API and configure the authentication.

**REST API**

The REST API can be secured with a shared secret key defined below or a with HTTP basic authentication. Moreover, for extra security you can specify the list of IP addresses that are allowed to use this service. An empty list means that the service can be accessed from any location. Addresses are delimited by commas.

☒ **Enabled** - REST API requests will be processed. **4**  
☐ **Disabled** - REST API requests will be ignored.

☐ HTTP basic auth - REST API authentication with Openfire admin account.  
☒ **Secret key auth** - REST API authentication over specified secret key. **5**  
 Secret key:

☐ Custom authentication filter classname - REST API authentication delegates to a custom filter implemented in some other plugin.

**Note: changing back and forth from custom authentication filter forces the REST API plugin reloading**

Filter classname:

Allowed IP Addresses:

Additional Logging  
☐ **Enabled** - Logging Enabled  
☒ **Disabled** - Logging disabled

## Setup of the Program itself

now we will do the setup at the program side: first we want so install the API-Library by

```
pip install openfire-restapi
```

then we need to update the information in the settings.py

```
# constants and settings
XMPP_SERVER_IP_ADDRESS = "YOUR_SERVER_IP" # The URL of where the Openfire-XMPP Server is running - usually Localhost if it runs on th
XMPP_SERVER_SHARED_SECRET = "YOUR_SECRET_KEY" # The shared Secret Key of the Openfire-REST API
XMPP_ALL_AGENTS_PASSWORD = "mvp_password" # in the MVP all agents share the same password
XMPP_SERVER_REST_API_USERS_ENDPOINT = "/plugins/restapi/v1/users" # see https://github.com/seamus-45/openfire-restapi/blob/master/doc
XMPP_SERVER_REST_API_PORT = ":9090" # Port is 9090 by default and 9091 for SSL
```

💡 instead of using the REST-API you can also add and manage users in the openfire-Admin-Panel:  
 To create a new user go to "User/Groups"- "Users"- "Add new User" and enter whatever you need. Mail-Adress per User will not be checked so you can leave this empty.  
 Each Agent need his own account. Add them depending on the System you want to create and store passwords and usernames.

Next you want to Install the robot framework SPADE to your environment. There are multiple ways to accive this - I usualy use the Terminal in Pycharm and enter

```
pip install spade
```

💡 Now we are at a Point where at least all implemented Soft-Agents have everything to work. So the Next step is to bring everything together and connect the realworld / the demonstration platform. Or in other words: we need to install install the Anki-Robot SDK in our environment.

install vector-Robot SDK by Terminal input

```
pip install anki_vector
```

Anki expects a certificate for each robot at a specific path - you need to copy / create the certificate-Folder and copy the Robot Certificate-Information there.

```
C:\Users\YOUR_USER_NAME\.anki_vector
```

Replace YOUR\_USER\_NAME with the current Windows user.

Maybe change the "C" directory if your windows is running from a different directory.

Start the Robots (and pair them to the same Network). Lookup the IP-Address of each Robot in your Router and modify the sdk-ini-File

```
C:\Users\YOUR_USER_NAME\.anki_vector\sdk_config.ini

....
[00404205]
cert = C:\Users\T\.anki_vector\Vector-X4V4-00404205.cert
ip = 192.168.0.178 //MODIFY THESE
name = Vector-X4V4
guid = H3GPVX0ARZStKtAnkY7kPg==

[00701643]
....
```



Notice that the design of the Robots does not allow Code running on the robots themselves (other than the given firmware). This is a security-action by the developers to ensure the robots work no matter what you do. This means that you need to run a Server (where the robot-Code is executed) in the same network as the robots permanently to ensure that the demonstration platform can be showcased!

That's It ! If Code works, the Robots are in the same Network as the Openfire-Server and the development-computer and you did not forget to update the .ini file everything implemented should now work.