# mlrMBO as a Webservice

This project is split into two parts: the server and the client.

The server is the actual project. It contains `mlrMBO`, the "highly configurable R toolbox for model-based / Bayesian optimization of black-box functions."

The client contains a REST client and some abstractions to make your life easier. But you can access the server with any REST client in any language you like, since the data exchange is done in JSON.

## How it works

Our server serves a REST-API which can be used to interact with mlrMBO. You upload your data and our machine learning algoriths will calculate an interesting point for you. You can interact with the server by using either our R client or any other client that can communicate with an REST-API.

# Our reference client

This client provides a high-level interface to the mlrMBO-Server provided in this repo. Without installing machine learning packages or other software (except of this package), you can benefit from mlrMBO!

You can install the R package with devtools. We also provide a server.

```
devtools::install_github("mboecker/mlrmbo-webservice/client")
mboServiceConnect("https://rombie.de:5001")
```

## Example usage

Although you can use the server with any programming language on the client side, we currently only supply an R client.

```
library("ParamHelpers")

# A simple test function with its optimum at (4, 4)
test_func = function(x) {
  sum((x - c(4,4))^2)
}

# Create a parameter set using ParamHelpers.
# It contains two variables from -5 to 5
size_of_initial_design = 10
par.set = makeNumericParamSet(len = 2,
                              lower = c(-5,-5),
                              upper = c(5,5))
```

```
# Generate a random initial design
data = generateDesign(n = size_of_initial_design, par.set)

# Run our test function on the initial design
Y = apply(data, 1, test_func)
data = data.frame(data, y = Y)

# Connect to service
# This is our open, free-to-use server
obj = mboServiceConnect("https://rombie.de:5001")

# Upload parameter set
# Use "expected improvement" as infill criterium
mboServiceSetConfigKey(obj, par.set = par.set, crit = "ei")

# Upload data
mboServiceUpload(obj, data)

# Request point proposal
point = mboServicePropose(obj)
print(point)

mboServiceDisconnect(obj)
```

## mlrMBO-Webservice Server

We provide mlrMBO as a Webservice. This server wraps the R package `mlrMBO` along with some software management software written in Python in order to handle multiple users simultanously. We expose a REST-API, which is documented in the file server/API_DOCS.md. You can use any REST client to access the API, but we also implemented an easy to use R package to go along with it.

### Do I have to run my own server?

Since we want you to have the best user experience, we are currently running an open, free-for-all server at `https://rombie.de:5001`. You may use that as much as you wish. If you're having performance problems, consider running your own server. How to do that is documented in the next section.

### How do I run my own server?

It's simple! The `image` folder contains a Docker image with everything you need. You can just run the `rebuild-server.sh` script and it will create a docker image, which you can upload to a server somewhere. It exposes port 5000, which will serve our `mlrMBO` API.

## May I use this? (License)

We provide both the server application as well as the reference client library as Open Source under the MIT license. This basically means you can use both the server and the client in open source and commercial projects.