

## ACP Tutorial 1 (REST + docker)

This tutorial covers REST-programming for server and clients as well as some dockerization.

All code mentioned is available in github and can be cloned using:

```
git clone https://github.com/mglienecke/AcpTutorial
```

**Now we build (using intelliJ) and run it**

Calling Swagger: <http://localhost:8080/swagger-ui/index.html>

Health-Actuator: <http://localhost:8080/actuator/health>

**Now we build an image and run in docker (inside intelliJ)**

- See recording / presentation

**Now we build an image and run in docker (outside intelliJ)**

```
docker buildx build --platform linux/amd64,linux/arm64 -t  
acp-submission-image --load .
```

```
docker image save -o acp_submission_image.tar acp-  
submission-image
```

```
docker run -d -p HOST_PORT:CONTAINER_PORT IMAGE_NAME -e  
VAR_NAME=value
```

```
docker run -d -p 8080:8080 acp-submission-image -e  
"ACP_ILP_ENDPOINT=https://ilp-rest-2025-all-  
assigned.azurewebsites.net/"
```

**GET-calls (works with browser or curl)**

*GET HTML*

*GET isAlive*

```
curl -X GET http://localhost:8080/actuator/health
```

*GET JSON*

```
curl -X GET localhost:8080/restaurants
```

*GET file as static resource via URL*

```
curl -X GET localhost:8080/demoFile1.json
```

POST with JSON data (item 3 will be ignored on the server)

```
curl -X POST -H "Content-Type: application/json" -d '{
  "item1": "JSON1",
  "item2": "JSON2",
  "item3": "AAA"
}' localhost:8080/testPostBody
```

**This is transmitted over the wire (so it is just plain text with the content-body as JSON):**

```
POST /testPostBody HTTP/1.1
Content-Type: application/json; charset=utf-8
Host: localhost:8080
Connection: close
User-Agent: RapidAPI/4.2.0 (Macintosh; OS X/13.5.2) GCDHTTPRequest
Content-Length: 47
```

```
{"item1":"JSON1","item2":"JSON2","item3":"AAA"}
```

This works with the hosted instance as well:

```
curl -X POST -H "Content-Type: application/json" -d '{
  "item1": "JSON1",
  "item2": "JSON2"
}' https://ilptutorial.azurewebsites.net/testPostBody
```

*POST with request parameters (is quite unusual as POST mostly uses the body)*

```
curl -X POST localhost:8080/testPostPath -d item1=data1 -d item2=data2
```

**This is transmitted over the wire:**

```
POST /testPostPath?item1=AAAA&item2=BBBB HTTP/1.1
Host: localhost:8080
Connection: close
```

User-Agent: RapidAPI/4.2.0 (Macintosh; OS X/13.5.2) GCDHTTPRequest  
Content-Length: 0

### **Tools for REST:**

- Postman (market leader)
- curl (command line - simple, easy)
- ...