

# GK771 Middleware Engineering "Marathon - REST" (MICT) - Protokoll

## 1.)

- Git-Repository download (<https://classroom.github.com/a/SxIBky6A>)
- In der Powershell in den heruntergeladenen Ordner mittels 'cd' wechseln
- 'gradle clean' und 'gradle bootRun' eingeben

```

Spring
:: Spring Boot ::
(v2.5.4)

2021-10-05 14:20:52.952 INFO 13848 --- [main] sectionpark.MarathonApplication : Starting MarathonApplication using Java 13.0.2 on LAPTOP-1Q86BSRF with
PID 13848 (C:\Users\ma240\Desktop\4_Klasse\Sem7\Syst_Labor\dezszy_gk_marathon_rest-mfrenslich-main\build\classes\java\main started by ma240 in C:\Users\ma240\Desktop\4_Klasse\Sem7\Syst_Labor\dezszy_gk_marathon_rest-mfrenslich-main)
2021-10-05 14:20:52.950 INFO 13848 --- [main] sectionpark.MarathonApplication : No active profile set, falling back to default profiles: default
2021-10-05 14:20:55.286 INFO 13848 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)
2021-10-05 14:20:55.311 INFO 13848 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2021-10-05 14:20:55.311 INFO 13848 --- [main] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.52]
2021-10-05 14:20:55.523 INFO 13848 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2021-10-05 14:20:55.523 INFO 13848 --- [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 80ms
2021-10-05 14:20:58.131 INFO 13848 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path ''
2021-10-05 14:20:58.153 INFO 13848 --- [main] sectionpark.MarathonApplication : Started MarathonApplication in 5.4s
<===== 80% EXECUTING [24m 35s]
<===== 80% EXECUTING [32s]
2021-10-05 14:21:28.281 INFO 13848 --- [nio-8080-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring DispatcherServlet 'dispatcherServlet'
2021-10-05 14:21:28.281 INFO 13848 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
2021-10-05 14:21:28.285 INFO 13848 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : Completed initialization in 3 ms
<===== 80% EXECUTING [24m 33s]
> :bootRun
```

- Nun ist der Simulator mit den Daten in 'JSON' über (<http://localhost:8080>) erreichbar

JSON	Rohdaten	Kopfzeilen
Speichern	Kopieren	Alle einklappen
Alle ausklappen	JSON durchsuchen	
timingstationID:	"001"	
timestamp:	"2021-10-05 14:47:45.162"	
distance:	1	
altitude:	200	

## 2.)

- Nun sollen die Daten auch in 'XML' angezeigt werden können
- Dafür (*implementation 'com.fasterxml.jackson.dataformat:jackson-dataformat-xml'*) in die 'build.gradle' unter 'dependencies' einfügen
- Und (*@RequestMapping(value="/timingstation/{timingstationID}/xml", produces = MediaType.APPLICATION\_XML\_VALUE)*) in die Klasse 'TimingstationController' einfügen

```
@RequestMapping(value="/timingstation/{timingstationID}/xml", produces = MediaType.APPLICATION_XML_VALUE)
public TimingstationData timingstationXml( @PathVariable String timingstationID ) {
    return service.getTimingstationData( timingstationID );
}
```

- Um zwischen 'JSON' und 'XML' wechseln zu können, müssen wir das gleiche auch für 'JSON' machen

```
@RequestMapping(value="/timingstation/{timingstationID}/data", produces = MediaType.APPLICATION_JSON_VALUE)
public TimingstationData timingstationData( @PathVariable String timingstationID ) {
    return service.getTimingstationData( timingstationID );
}
```

- Als nächstes wird 'mainPage' String um einen Link für 'XML' erweitert, damit wir zwischen 'XML' & 'JSON' wechseln können

```
"This is the timingstation application! (DEZSYS_MARATHON_REST) <br/><br/>" +
"<a href='http://localhost:8080/timingstation/001/data'>Link to timingstation/001/data</a><br/>" +
"<a href='http://localhost:8080/timingstation/001/xml'>Link to timingstation/001/xml</a><br/>" +
"<a href='http://localhost:8080/timingstation/001/transfer'>Link to timingstation/001/transfer</a><br/>";
```

- Damit das ganze funktioniert, muss noch ein import getätigt werden

```
import org.springframework.http.MediaType;
```

- Nun in der Powershell 'gradle build', 'gradle clean' & 'gradle runBoot' eingeben, damit es aktualisiert wird
- Jetzt kann man zwischen der Anzeige in 'XML' und 'JSON' wechseln

This is the timingstation application! (DEZSYS\_MARATHON\_REST)

[Link to timingstation/001/data](http://localhost:8080/timingstation/001/data)

[Link to timingstation/001/xml](http://localhost:8080/timingstation/001/xml)

[Link to timingstation/001/transfer](http://localhost:8080/timingstation/001/transfer)

```

-<TimingstationData>
  <timingstationID>001</timingstationID>
  <timestamp>2021-10-05 15:00:02.874</timestamp>
  <distance>1.0</distance>
  <altitude>200.0</altitude>
</TimingstationData>

```