



## Bachelorarbeit - Sean

Aktueller Stand vom 21.08.2018

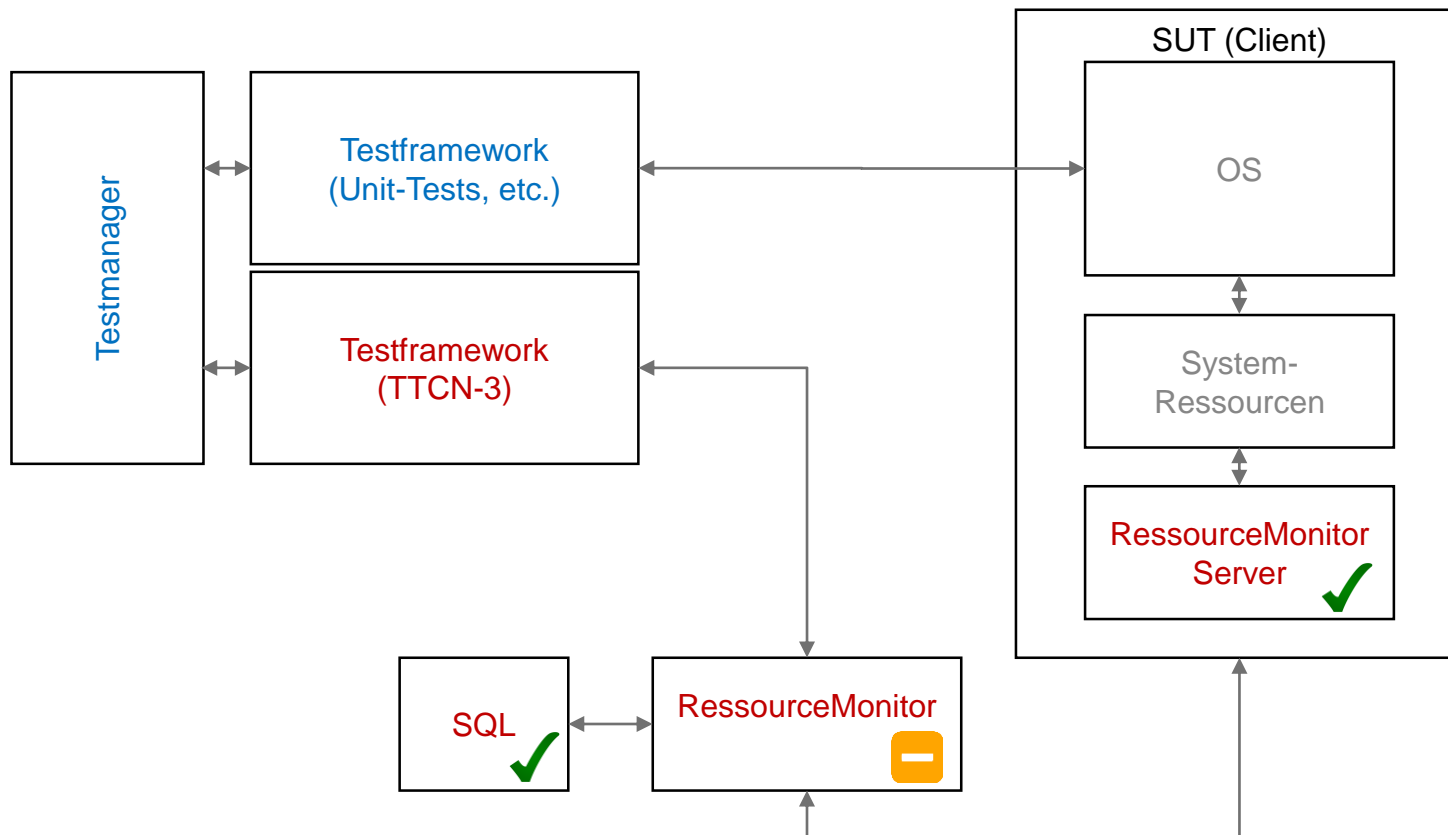
---

# Gliederung

1. Projekt-Status
2. Eingesetzte Technologien
  1. Swagger
3. Organisatorisches

# 1. Projekt-Status

## RECAP: Status vom 07.08.2018



## RECAP: Spring Boot Data

- Automatische Übersetzung von Klassen zu Tabellen

processorId	processorName	...
-------------	---------------	-----

- OneToMany, OneToOne, ManyToOne, ManyToMany

```
@Entity
public class Processor implements Serializable {

    /**
     *
     */
    private static final long serialVersionUID = 1L;

    @Id
    String processorId;
    String processorName;
    String processorVendor;
    String processorFamily;
    String processorModel;
    String processorStepping;
    int processorPhysicalPackageCount;
    int processorPhysicalProcessorCount;
    int processorLogicalProcessorCount;
    ...

    @OneToMany(mappedBy="processorValueProcessor")
    Set<ProcessorValue> processorValues;
```

## Recap: JpaRepository (Java Persistence API)

- Abstrakte Schicht zwischen Datenbank und Applikation
- Keine differenzierte Implementierung für diverse Datenbanken
- Methodennamen werden in SQL-Anfragen übersetzt

```

count() : long - CrudRepository
count(Example<S> arg0) : long - QueryByExampleExecutor
delete(Memory arg0) : void - CrudRepository
deleteAll() : void - CrudRepository
deleteAll(Iterable<? extends Memory> arg0) : void - CrudRepository
deleteAllInBatch() : void - JpaRepository
deleteById(Long arg0) : void - CrudRepository
deleteInBatch(Iterable<Memory> arg0) : void - JpaRepository
equals(Object arg0) : boolean - Object
exists(Example<S> arg0) : boolean - QueryByExampleExecutor
existsById(Long arg0) : boolean - CrudRepository
findAll() : List<Memory> - JpaRepository
findAll(Example<S> arg0) : List<S> - JpaRepository
findAll(Pageable arg0) : Page<Memory> - PagingAndSortingRepository
findAll(Sort arg0) : List<Memory> - JpaRepository
findAll(Example<S> arg0, Pageable arg1) : Page<S> - QueryByExampleExecutor
findAll(Example<S> arg0, Sort arg1) : List<S> - JpaRepository
findAllById(Iterable<Long> arg0) : List<Memory> - JpaRepository
findById(Long arg0) : Optional<Memory> - CrudRepository
findOne(Example<S> arg0) : Optional<S> - QueryByExampleExecutor
flush() : void - JpaRepository
getClass() : Class<?> - Object
getOne(Long arg0) : Memory - JpaRepository
hashCode() : int - Object
notify() : void - Object
notifyAll() : void - Object
save(S arg0) : S - CrudRepository
saveAll(Iterable<S> arg0) : List<S> - JpaRepository
saveAndFlush(S arg0) : S - JpaRepository

```

```

public interface MemoryRepo extends JpaRepository<Memory, Long> {
}

```






## Recap: Scheduler

- Feste Scheduler für diverse Messungen
- `isRunning = true` → Messung wird geupdated
- `isRunning = false` → iteriere zur nächsten Messung

```
@Component  
public class ScheduledTask {  
  
    @Scheduled(fixedRate = 1000)  
    public void execute() {
```

## RECAP: Geplante Ziele bis Heute

### Bis zum 21.08.2018:

- Vollständige REST-Beschreibung (GET, PUT, POST, DELETE) ✓   
- Vollständige JPA-Repositories ✓ 
- Vollständige, automatisierte Testdurchläufe mit Localhost und bekanntem Remote-PC ✓
- Daten visualisiert, nicht nur JSON 

### Eingeschobene Tasks

- Automatisierte REST-Dokumentation inkl. Beispiele, Parameter und Test-Queries ✓

### Anschließende Aufgaben (Restzeit: 5 Wochen | ~200h , Vollzeit!):

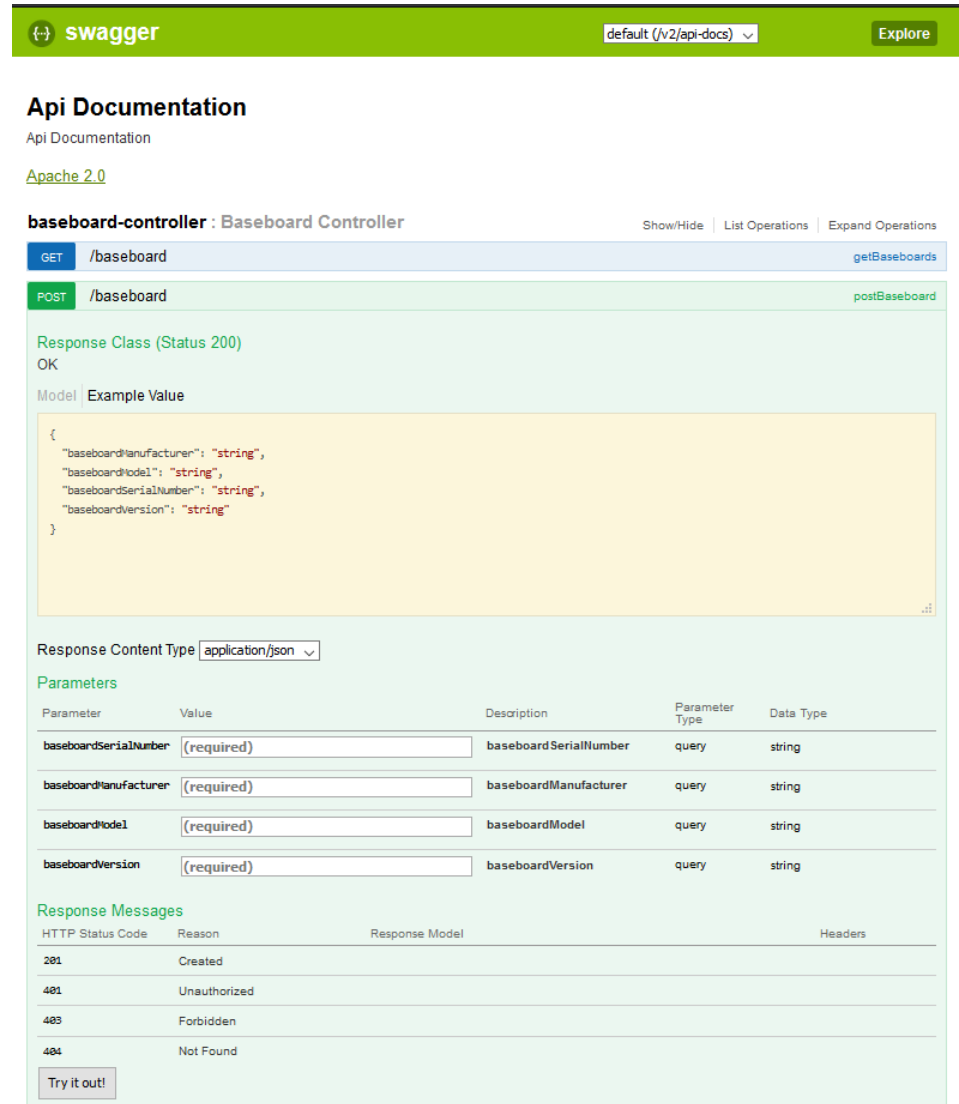
- Netzwerkaufdeckung
- TTCN-3 Testfälle
- Bachelorarbeit nieder schreiben



## 2. Eingsetzte Technologien

## 2. Swagger UI

- Automatisierte REST-Beschreibung
- einer der Standards
- Beispiel-Value
- Live-Demo mittels Eingabe von Parametern
- Mögliche Response-Messages inkl. Status
- Beschreibung der API:
  - mittels UI (siehe Bild)
  - als JSON (maschinen-lesbar)



The screenshot shows the Swagger UI for the **baseboard-controller** API. The interface is in German and displays the **POST /baseboard** endpoint. The response class is **Status 200 OK**. An example JSON response is shown:

```
{
  "baseboardManufacturer": "string",
  "baseboardModel": "string",
  "baseboardSerialNumber": "string",
  "baseboardVersion": "string"
}
```

Below the example, the **Parameters** section lists four query parameters, all required and of type string:

Parameter	Value	Description	Parameter Type	Data Type
baseboardSerialNumber	(required)	baseboardSerialNumber	query	string
baseboardManufacturer	(required)	baseboardManufacturer	query	string
baseboardModel	(required)	baseboardModel	query	string
baseboardVersion	(required)	baseboardVersion	query	string

The **Response Messages** section lists the following HTTP status codes and reasons:

HTTP Status Code	Reason	Response Model	Headers
201	Created		
401	Unauthorized		
403	Forbidden		
404	Not Found		

A **Try it out!** button is located at the bottom left of the response messages section.

# 3. Organisatorisches

### 3. RECAP: GitHub

- Projekt ist auf GitHub hochgeladen (privates Repository, MIT-Lizenz)
- Einladung per E-Mail