



LEICHTGEWICHTIGE

MICROSERVICES

MIT JAYWIRE UND DAS SPARK FRAMEWORK

Róbert Bräutigam, MATHEMA Software GmbH.

A

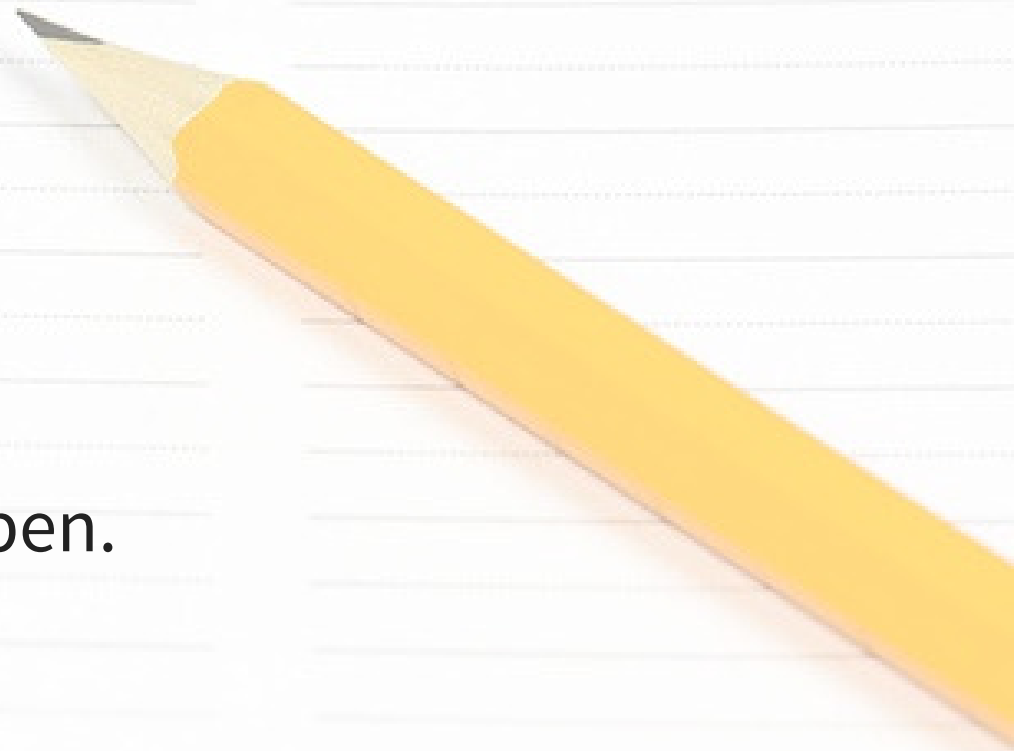
B

INHALT

Erst die Wörter definieren:

- Leichtgewichtig
- Microservice
- JayWire
- Spark Framework

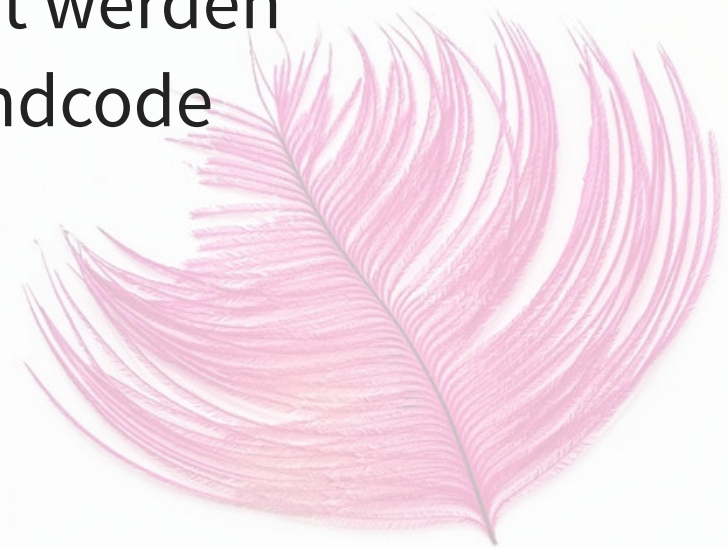
Dann einen Microservice schreiben.



LEICHTGEWICHTIG

Was ist eigentlich leichtgewichtig?

- Nur Abhängigkeiten die direkt genutzt werden
- Unabhängig von das Design von Fremddcode



LEICHTGEWICHTIG

Alternative versuche:

- Wenig Code. **Nein**
- Kleines Jar. **Nein**
- Kein "Boilerplate" Code. **Nein**



BEISPIEL

```
public class Employees {  
  
    public Employees(Database database) {  
        ...  
    }  
  
    ...  
    public void giveRaiseToAll(Percent amount) {  
        ...  
    }  
    ...  
}
```



GEGENBEISPIEL

```
@Stateless
public class Employees {

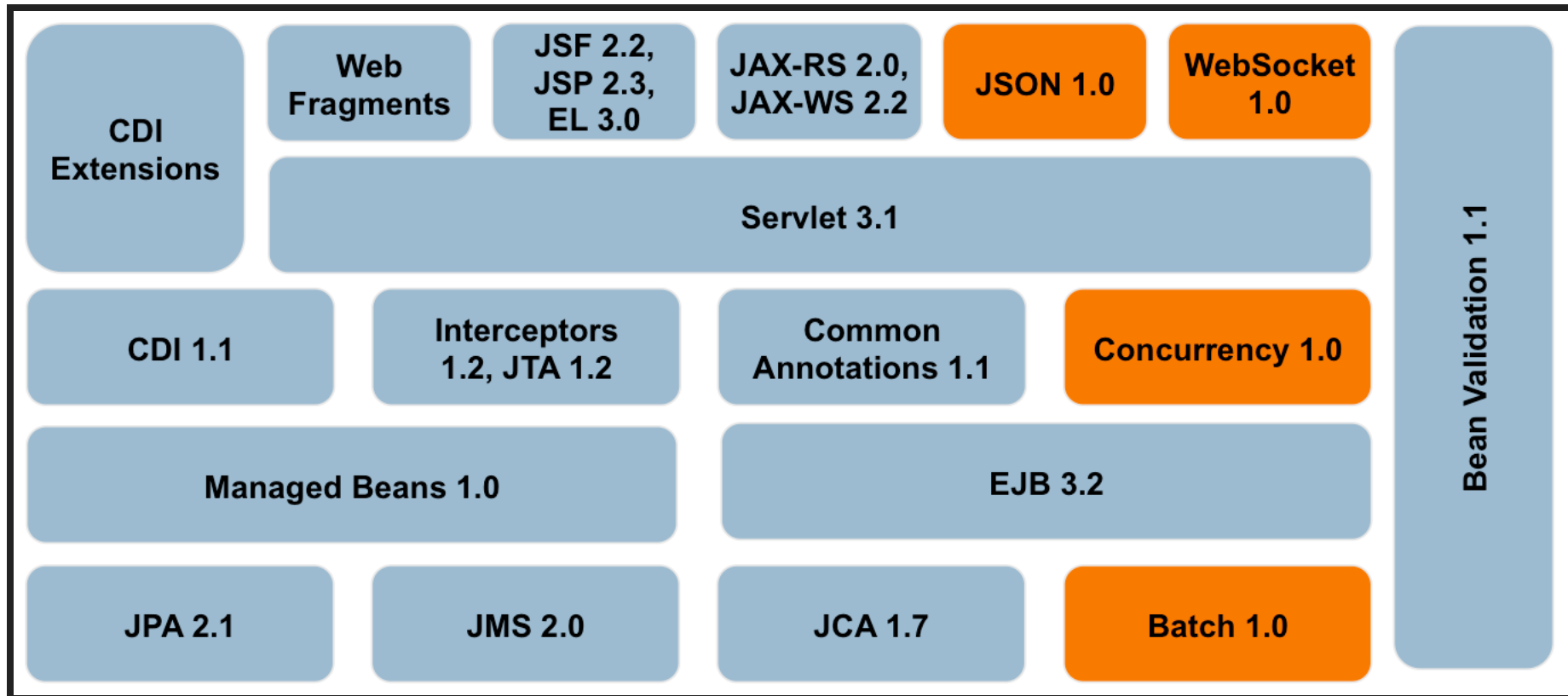
    @Inject
    public Employees(Database database) {
        ...
    }

    ...
    public void giveRaiseToAll(Percent amount) {
        ...
    }
    ...
}
```

Wieso ist das nicht leichtgewichtig?



GEGENBEISPIEL



Brauche ich wirklich das alles für Employees?

MICROSERVICE

Was ist also ein Microservice?

- Sollte "klein", und spezialisiert sein
- Läuft unabhängig, in ein oder mehrere Prozesse
- Kommuniziert über einheitliche Schnittstelle (über das Netzwerk)
- Ist entkoppelt, hat eine eigene Datenbank/Persistenz, oder auch GUI

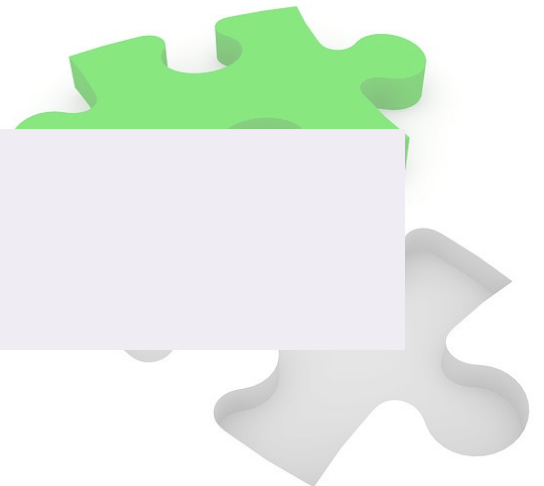


ENTKOPPELTE SERVICES

```
$ ls          # List files to stdout
a
b
c
$ wc          # Word-Count stdin to stdout
^C
$ ls | wc -l # List files and count lines
3
```

Im Gegensatz zu:

```
$ ls --count
3
```

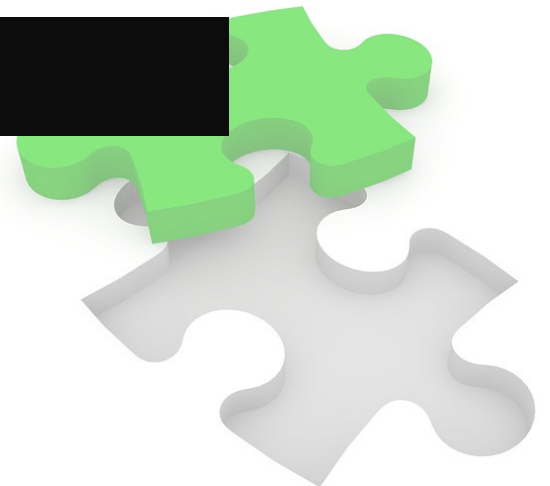


REST

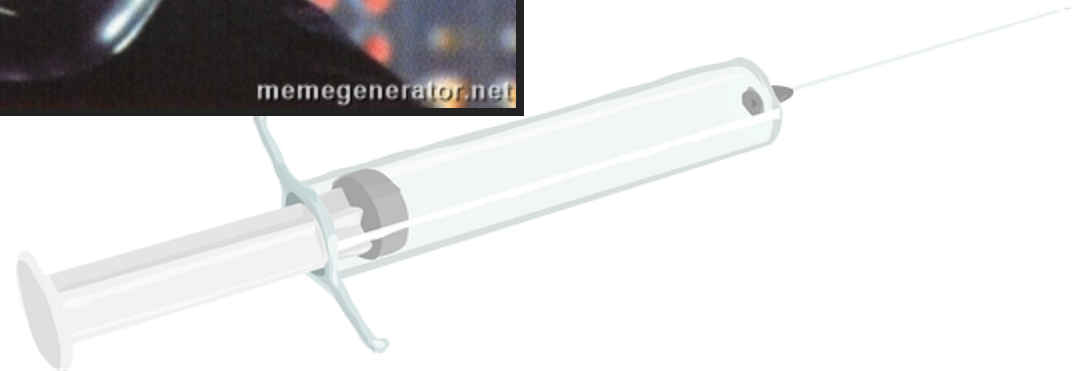
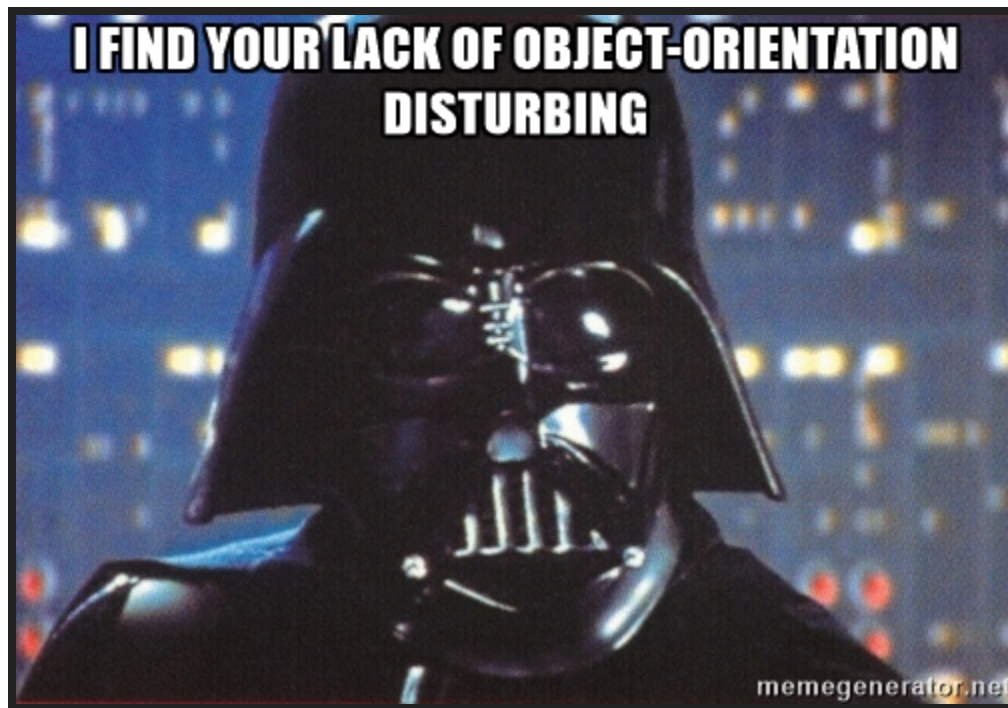
REST passt optimal für Microservices, weil:

- Schnittstelle HTTP bekannt ist
- Beinhaltet Kontrol-Informationen, z.B. Caching, Linking

"Smart endpoints and dumb pipes"

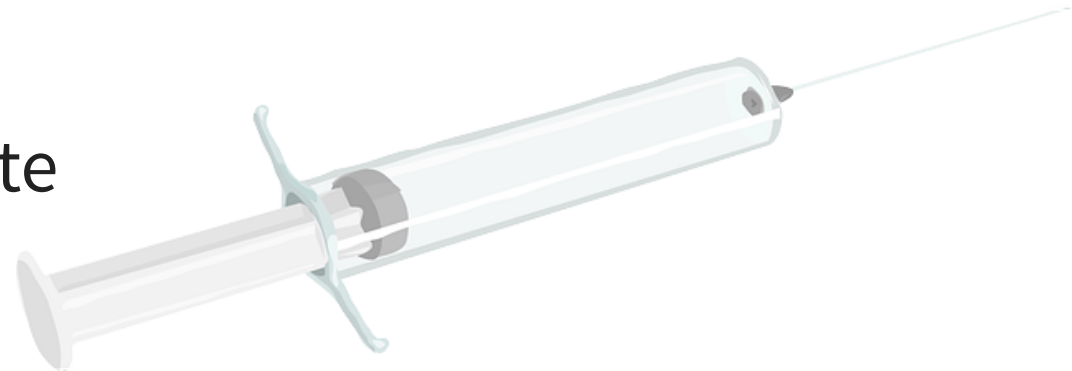


JAYWIRE



JAYWIRE

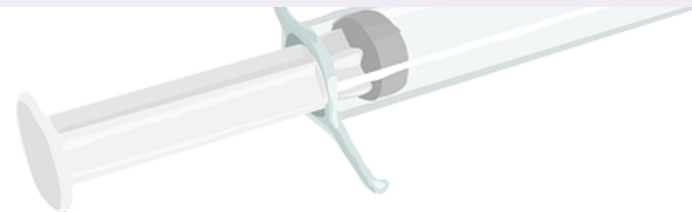
- Dependency Injection Framework für Java 8
- Statisch sicher
- Sehr klein, modular
- Übliche Features: scopes, singleton, serialisierbar, usw.
- 100% Java Code (kein Konfiguration)
- Ohne Classpath-Scanning
- Ohne Bytecode "Enhancement"
- Ohne Reflection
- Ohne Annotations
- Ohne implizite Proxy-Objekte
- Ohne Code-Generation



JAYWIRE

```
public class Application extends StandaloneModule {  
    public Database getDatabase() {  
        return singleton( () -> new JdbcDatabase(...) );  
    }  
  
    public ServiceA getServiceA() {  
        return new ServiceA(getDatabase()); // Injecting Database  
    }  
}
```

```
new Application().getServiceA().run();
```



SPARK

"A micro framework for creating web applications in Java 8"

- Nutzt Jetty Web Server
- "Nur" Web



SPARK

```
import static spark.Spark.*;

public class HelloWorld {
    public static void main(String[] args) {
        get("/hello", (req, res) -> "Hello World");
    }
}
```


FRAGEN?



DANKE!

JAYWIRE

GitHub: <https://github.com/vanillasource/jaywire/>

Maven: `com.vanillasource.jaywire:jaywire:1.1.0`

SPARK

GitHub: <https://github.com/perwendel/spark>

Maven: `com.sparkjava:spark-core:2.5`

