



ulm university universität  
**uulm**

**Fakultät für  
Ingenieurwissenschaften,  
Informatik und  
Psychologie**

Institut für Software-  
technik und Program-  
miersprachen

# Softwaregrundprojekt Meilenstein 6

Softwaregrundprojekt an der Universität Ulm

**Vorgelegt von:**

Gruppe 10

**Dozent:**

Florian Ege

**Betreuer:**

Stefanos Mytilineos

2019

---

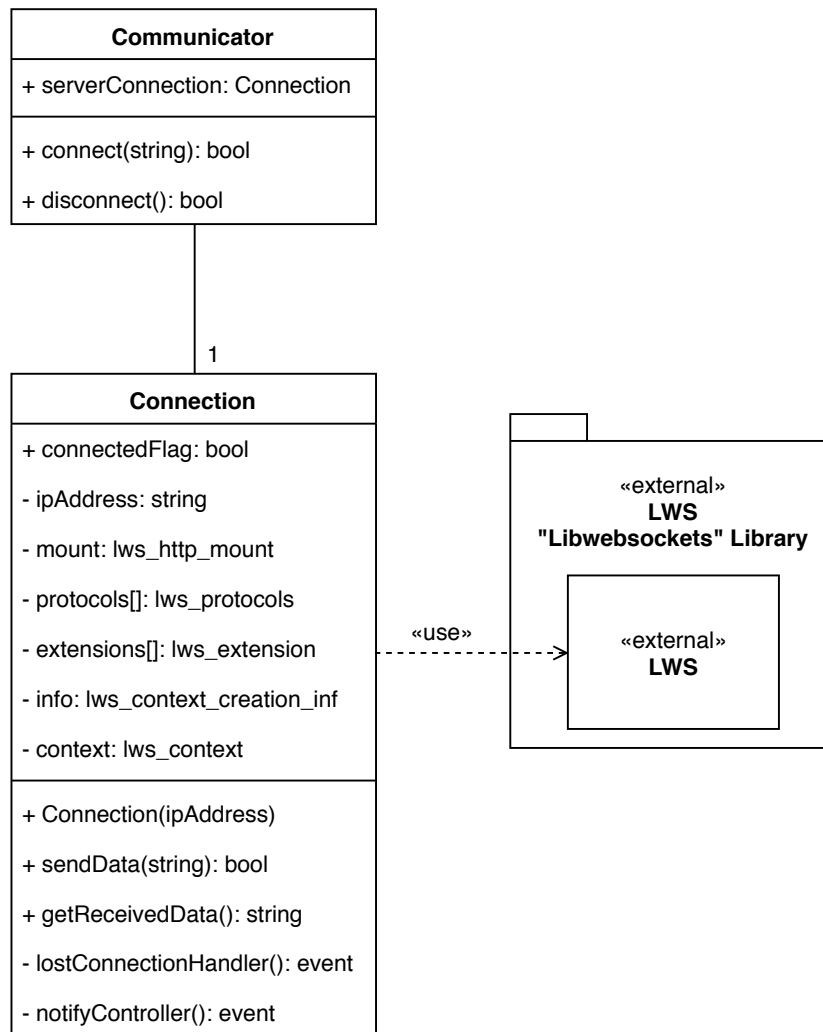
# Inhaltsverzeichnis

<b>1</b>	<b>Klassen Übersicht</b>	<b>3</b>
1.1	Klassen-Diagramm . . . . .	3
1.2	Beschreibung . . . . .	3
<b>2</b>	<b>JSON-Parser</b>	<b>4</b>
2.1	Klassen-Diagramm . . . . .	4
2.2	Beschreibung . . . . .	4
2.2.1	JSON-Parser (Klasse) . . . . .	4
2.2.2	nlohmann::json (Externe Bibliothek) . . . . .	5
2.3	Zuordnung der Funktionalen Anforderungen . . . . .	6

---

# 1 Kommunikator

## 1.1 Klassen-Diagramm



---

## 1.2 Beschreibung

### 1.2.1 Connector (Klasse)

Im Allgemeinen dient die Connector Klasse dazu, die Kommunikation zwischen der Client Anwendung und der Server Anwendung über das Netzwerk zu verwalten.

**connect (Methode)** Die connect Methode dient dazu, eine neue Verbindung mit einem Server aufzubauen. Die IP-Adresse des Servers wird dabei als Parameter übergeben. Ist die Aktion erfolgreich, so wird true zurück gegeben, anderenfalls false.

**disconnect (Methode)** Die disconnect Methode dient dazu, eine bestehende Verbindung mit einem Server zu trennen. Ist die Aktion erfolgreich, so wird true zurück gegeben, anderenfalls false.

### 1.2.2 Connection (Klasse)

**sendData (Methode)** Die öffentliche sendData Methode dient dazu, einen im JSON-Format formatierten String an den Server zu übertragen. Ist die Aktion erfolgreich, so wird true zurück gegeben, anderenfalls false. Zur Umsetzung dieser Methode werden Funktionen und Strukturen der Drittanbieter-Bibliothek LWS benötigt.

**getReceivedData (Methode)** Mit Hilfe der öffentlichen Methode getReceivedData kann der letzte vom Server gesendete JSON Datensatz ausgelesen werden. Zur Umsetzung dieser Methode werden ebenfalls Funktionen und Strukturen der Drittanbieter-Bibliothek LWS benötigt.

**lostConnectionHandler (Methode)** Die private Methode lostConnectionHandler kümmert sich im Falle eines außerplanmäßigen Verbindungsverlust darum, dass alle notwendigen Schritte eingeleitet werden, indem der Controller über den Verbindungsabbruch informiert wird. Auch hier werden zur Umsetzung Funktionen und Strukturen der Drittanbieter- Bibliothek LWS benötigt.

**notifyController (Methode)** Die private notifyController Methode ist dafür zuständig den Controller über das Eingehen von neuen Nachrichten des Servers zu informieren.

### 1.2.3 LWS (Externe Bibliothek)

Bei dieser Klasse Handelt es sich um eine externe Bibliothek, die verschiedenste Methoden und Strukturen zur Verfügung stellt um in einem C/C++ Projekt einfache Server Client Verbindungen mittels Web-Sockets zu implementieren. Genauere Informationen zu dieser Bibliothek sind unter folgendem Link zu finden:

<https://libwebsockets.org/>

---

## 1.3 Zuordnung der Funktionalen Anforderungen

Die Funktionalen Anforderungen werden den Methoden folgendermaßen zugeteilt:

<b>Funktionale Anforderungen</b>	<b>Methoden</b>
FA55	Connector::connect
FA55	Connector::disconnect
FA55	Connection::sendData
FA55	Connection::getReceivedData
FA55	Connection::lostConnectionHandler
FA55	Connection::notifyController