

# SmartCoffee API



# Übersicht

- ⦿ Bibliothek zur Kommunikation mit der Kaffeemaschine
- ⦿ Java
- ⦿ TCP-Socket (Port 2081)
- ⦿ Kommunikation über lokales Netzwerk (KaaProject)

# Funktionen

- ⊙ Objekt „SmartCoffeeMachine“ mit Methoden
- ⊙ Verbindungsaufbau, Verbindungsabbau
- ⊙ Befehle an Kaffeemaschine senden:
  - Kaffee kochen
  - Kochvorgang unterbrechen
  - Brühstärke einstellen
  - Tassenanzahl einstellen
  - Warmhaltezeit der Warmhalteplatte einstellen

# Funktionen

## ⦿ Fehlerbehandlung der Befehle durch Exceptions:

- Ungültige Parameter
- Unbekannter Rückgabecode
- Keine Antwort der Kaffeemaschine
- Kaffee wird bereits gekocht
- Niedriger Wasserstand
- Kein Wasser
- Keine Karaffe

# Funktionen

- ⦿ Empfangen von Statusmeldungen der Kaffeemaschine:
  - Status: Bereit, Wasser brühen, Bohnen malen, Karaffe nicht vorhanden
  - Anzahl der eingestellten Tassen
  - Eingestellte Brühstärke
  - Wasserstand
  - Stärke des WiFi-Signals
- ⦿ Observer-Pattern zur Benachrichtigung bei Änderungen

# Funktionsweise der Kaffeemaschine

- Ansteuerung über TCP aus dem gleichen Netzwerk
- Befehle, Parameter, Statusmeldungen usw. sind hexadezimal codiert, werden jedoch in ASCII (UTF-8) übertragen
- Keine Dokumentation: Manuelle Entschlüsselung und Ausprobieren notwendig
- Kaffeemaschine sendet sekundlich Statuscodes („polling“)

# Funktionsweise der Kaffeemaschine

Befehlcodes:

- ⦿ **377e**: Kaffee kochen
- ⦿ **347e**: Kaffee kochen stoppen
- ⦿ **35XX7e**: Kaffeestärke ändern (XX: Stärke als Hexadezimalzahl)
- ⦿ **36XX7e**: Tassenanzahl ändern (XX: Anzahl als Hexadezimalzahl)
- ⦿ **3eXX7e**: Warmhaltezeit ändern (XX: Minuten als Hexadezimalzahl)

# Funktionsweise der Kaffeemaschine

Rückgabecodes:

- 03007e: OK
- 03017e: Maschine kocht bereits
- 03057e: Keine Karaffe
- 03077e: Wenig Wasser
- 03067e: Kein Wasser
- 03047e: Ungültige Parameter



# Funktionsweise der Kaffeemaschine

Statusmeldungen: 3206010002337e

- 32: Befehlskopf zur Unterscheidung von Rückgabecodes
- 06: Einer von vielen Statuscodes; hier: Bereit, Bohnen, keine Karaffe
- 01: Wasserstand; hier: wenig Wasser
- 00: WiFi-Stärke
- 02: Kaffeestärke; hier: „Strong“
- 33: Hintere Ziffer ist Tassenanzahl; hier: 3
- 7e: Befehlsende

# Beispiel - Observer

```
String host = „192.168.0.23“;  
int port = 2081;
```

```
SmartCoffeeMachine machine = new SmartCoffeeMachine(host, port);  
machine.connect();
```

```
machine.addPollingObserver(new Observer() {
```

```
    @Override
```

```
    public void update(Observable o, Object arg) {  
        CoffeeMachine model = (CoffeeMachine) arg;
```

```
        StatusCodes status = model.getStatus();  
        int cups          = model.getNumberOfCups();
```

```
    }
```

```
});
```

# Beispiel - Befehle

```
String host = „192.168.0.23“;  
int port = 2081;
```

```
SmartCoffeeMachine machine = new SmartCoffeeMachine(host, port);  
machine.connect();
```

```
try {  
    machine.setBrewingStrength(SmartCoffeeMachine.STRENGTH_WEAK);  
    machine.setCupsNumber(3);  
    machine.brewCoffee();  
  
} catch (IOException e) {  
    System.out.println(„Verbindungsfehler: " + e.getMessage());  
} catch (AlreadyBrewingException | NoCarafeException | NoWaterException |  
LowWaterException e) {  
    System.out.println(„Fehler: " + e.getMessage());  
}  
machine.disconnect();
```

- ▼ exceptions
  - > AlreadyBrewingException.java
  - > InvalidParameterException.java
  - > LowWaterException.java
  - > NoCarafeException.java
  - > NoReplyException.java
  - > NoWaterException.java
  - > UnknownCodeException.java
- ▼ gui
  - > Gui.java
- ▼ main
  - > HexConverter.java
  - > MessageReceiver.java
  - > ReplyCodeHandler.java
  - > SmartCoffeeMachine.java
- ▼ messages
  - > CommandCodes.java
  - > ReplyCodes.java
  - > StatusCodes.java
  - > StrengthCodes.java
  - > WaterLevelCodes.java
- ▼ model
  - > CoffeeMachine.java
- ▼ test
  - > Test.java
  - > UnitTestRemote.java
  - > UnitTestStatic.java

# Struktur