1. Was ist REST und wie könnte es in einer Microservice Umgebung zum Einsatz kommen?

REST (kurz für Representational State Transfer) ist eine Schnittstelle über die Rechner miteinander kommunizieren können. Ein Client fragt den Server nach bestimmten Daten an, die ihm dann durch ein HTTP Protokoll geliefert werden.

Meistens sind diese Anfragen:

- GET: Daten aus einer bestimmten Ressource holen
- DELETE: Eine bestimmte Ressource löschen
- PATCH: Updatet einen Teil der Ressourcen
- PUT: Eine bestimmte Ressource updaten
  [Source: https://www.youtube.com/watch?v=7YcW25PHnAA]

Microservices stellen eine Programmarchitektur dar bei der die Gesamt-Anwendung aus kleineren unabhängigen Prozessen besteht. Dadurch kann man sein Team besser auf die kleine Anwendung anpassen und diese besser Warten.

Möchte man die Funktionen eines Microservices in einem anderen Verwenden muss man diese miteinander kommunizieren lassen damit es einen Datenfluss gibt.

REST-Services ermöglichen diese Kommunikation. Sie stellen sicher, dass sich die Microservices untereinander austauschen können und damit ihre Funktionen für die Gesamt-Anwendung bereitstellen.

2. Erklären Sie was UnitTests sind und an einem Beispiel wie sie eingesetzt werden.

Unit-Tests sind dazu da um Teile des Codes (Units) auf ihre Funktionalität zu überprüfen.
Sie sollen die Einzelteile des Codes getrennt voneinander überprüfen, um ihre unabhängige Funktionalität und Stabilität sicherzustellen.

In einem Unit-Test initialisiert man anfangs u.U. Variablen, die im Test genutzt werden, dann ruft man die zu testende Funktion auf und führt sie mit den initialisierten Variablen auf. Am Ende steht das Ergebnis der getesteten Funktion.

Dieses Ergebnis wird dann überprüft indem man den entstandenen Ist-Wert mit einem Soll-Wert vergleicht.

Kontinuierliche oder automatisierte Tests sind sehr wichtig für die Qualität der Software, da sie ständig die Funktionalität der Code-Einheiten überprüfen und damit Erkenntnisse über den Zustand der Software geben.

In meinem Beispiel lasse ich den Nutzer angeben wie viele Stunden er in welchem Bereich gearbeitet hat. Gibt er eine Stunden Zahl für Zeit ein (*setZeit*) die unter 8 liegt gibt es eine IllegalArgumentException. Im Konstruktor ist festgelegt, dass der Name eines Bereichs min. 3 Buchstaben haben muss.

Das Beispiel finden sie auf der nächsten Seite.

Der Code:

Arbeit.java

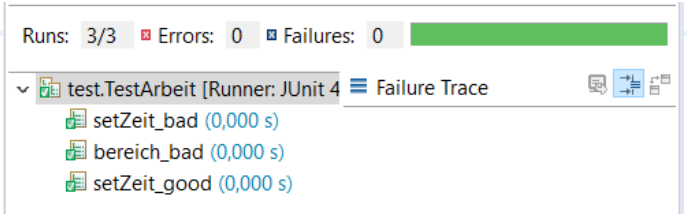
```java
package code;

public class Arbeit {

    private int zeit;
    private String bereich;

    public Arbeit(int zeit, String bereich) {
        this.setZeit(zeit);
        if (bereich.length() <= 3 || bereich.length() > 20) {
            throw new IllegalArgumentException("Invalider Arbeitsbereich");
        } else {
            this.bereich = bereich;
        }
    }

    public void setZeit(int zeit) {

        if (zeit >= 8) {
            this.zeit = zeit;
        } else {
            throw new IllegalArgumentException("Zu wenig Arbeitsstunden.");
        }
    }

    public int getZeit(Arbeit arb) {
        return this.zeit;
    }
}
```

Die Tests:

Ich teste ob zu wenig angegebene Zeit zu einer IllegalArgumentException führen, ob ein Arbeitsbereich mit unter 3 Buchstaben zu einer IllegalArgumentException führen und ob eine korrekte Eingabe in der Methode setZeit korrekt übernommen wird.

TestArbeit.java

```java
package test;

import static org.junit.Assert.assertEquals;

import org.junit.Test;

import code.Arbeit;

public class TestArbeit {

    Arbeit arb = new Arbeit(12, "IT-Bereich");

    @Test(expected = IllegalArgumentException.class)
    public void setZeit_bad() {
        arb.setZeit(3); // es werden < 8 Stunden eingetragen
    }

    @Test(expected = IllegalArgumentException.class)
    public void bereich_bad() {
        new Arbeit(8, "IT"); // der Arbeitsbereich hat weniger als 3 Buchstaben
    }

    @Test
    public void setZeit_good() {
        arb.setZeit(8); //die Arbeitszeit wird auf eine akzeptable Menge gesetzt
        assertEquals(8,arb.getZeit(arb));
    }
}
```



3. Schreiben Sie ein Programm, das leere Seiten aus einem PDF löscht.

Siehe GitHub sowie Anhang:
https://github.com/hani-j-asf/pdf-blank-page-remover