

Datum: 10.01.2024	Art der Leistungsfeststellung: 1. Test		Wertung	Bewertung																				
Klasse: 4BHITS	Gegenstand: Medientechnik	1)	25	25																				
		2)	25	22																				
Zeit: 20 min + Vorbereitungszeit	Notenschlüssel: <table><tr><td>90</td><td>-</td><td>100</td><td>Sehr gut</td></tr><tr><td>76</td><td>-</td><td>89</td><td>Gut</td></tr><tr><td>63</td><td>-</td><td>75</td><td>Befriedigend</td></tr><tr><td>51</td><td>-</td><td>62</td><td>Genügend</td></tr><tr><td>0</td><td>-</td><td>50</td><td>Nicht Genügend</td></tr></table>	90	-	100	Sehr gut	76	-	89	Gut	63	-	75	Befriedigend	51	-	62	Genügend	0	-	50	Nicht Genügend	3)	25	22
		90	-	100	Sehr gut																			
		76	-	89	Gut																			
		63	-	75	Befriedigend																			
		51	-	62	Genügend																			
0	-	50	Nicht Genügend																					
4)	25	19																						
Lehrer: HOEF	Name: Hibek Matthias	Σ	100	88																				

1. Nodejs (25 Punkte)

- Erkläre das Feature vom node-Parameter `--watch` beziehungsweise dem Modul `nodemon`.

`nodemon` / `--watch` ist ein ~~feaut~~ Feature des Hot-Reloading.
Das bedeutet dass das Programm nicht ~~in~~ bei jeder Änderung neu gestartet werden muss. `Nodemon` und `--watch` lassen das Programm laufen und zeigen die Änderungen in Echtzeit an. 8P

- Was machen die Befehle `npm init` und `npm install`?

`init` - initialisiert im ordner eine node Grundstruktur (`package.json`)
`install` - ~~ist~~ installiert alle fürs Projekt nötige Dependencies 9P

- Woran erkennst du den Einsatz des ECMA-Script Modulsystems und wie kann es aktiviert werden?

Die `import` und `exports` sind anders als zu `CommonJS`.
Anweisung `"type": "module"` in `package.json` hinzufügen
`import fs from 'node:fs'` 8P

2. REST & API Design (25 Punkte)

- Was bedeutet die Abkürzung **REST** und was versteht man darunter?

Representational State Transfer

Zustandslose Kommunikation

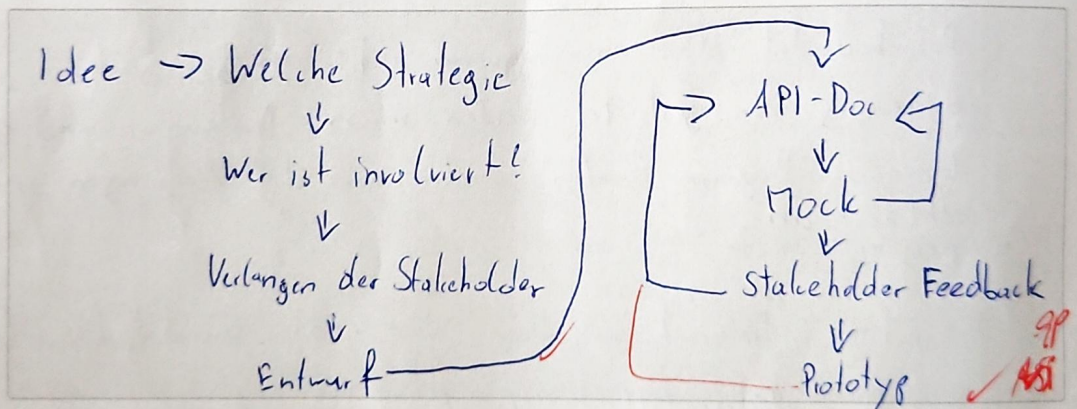
Rest ist ein Standard zur Dokumentation von APIs ✓ 70

- Welche API Typen kennst du und was zeichnet eine gute API aus?

public ✓
(private) internal
restricted ✓

Gute API: - Mehrwert fürs Unternehmen ✓
- Lohnt für immer ✓
- Entspricht den Erwartungen der Entwickler ✓ GP

- Beschreibe den Ablauf der Erstellung einer well designed API.



3. OpenAPI (25 Punkte)

- Was ist OpenAPI und warum wird es in der Webentwicklung eingesetzt?

OpenAPI ist eine Spezifikation zur Dokumentation von (RESTful) APIs ✓

- Es ist eine standardisierte Methode APIs zu dokumentieren
- Code generierung (Tests) ✓ 22P

- Analysiere die folgende OAS und trage die RESTful-Endpunkte in eine Tabelle (HTTP-Methode, URI) ein.
Korrigiere Fehler beim Eintragen in die Tabelle, sofern welche vorhanden sind

```

openapi: 3.0.0
info:
  title: Schülerverwaltungs-API
  version: 1.0.0
  description: Eine REST-API zur Verwaltung von Schülern
paths:
  /schueler:
    get:
      summary: Alle Schüler abrufen
      responses:
        '200':
          description: Erfolgreiche Abfrage
          content:
            application/json:
              example:
                schueler:
                  - id: 1
                    vorname: Max
                    nachname: Mustermann
                    klasse: 10A
                  - id: 2
                    vorname: Lisa
                    nachname: Beispiel
                    klasse: 10B
    post:
      summary: Neuen Schüler hinzufügen
      requestBody:
        required: true
        content:
          application/json:
            example:
              vorname: Max
              nachname: Mustermann
              klasse: 10A
      responses:
        '201':
          description: Erfolgreich erstellt
          content:
            application/json:
              example:
                id: 3
                vorname: Max
                nachname: Mustermann
                klasse: 10A
    delete:
      summary: Schüler löschen
      responses:
        '204':
          description: Erfolgreich gelöscht
  /schueler/{id}:
    parameters:
      - name: id
        in: path
        required: true
        description: Die ID des Schülers
        schema:
          type: integer
    get:
      summary: Schülerdetails abrufen
      responses:
        '200':
          description: Erfolgreiche Abfrage
          content:
            application/json:
              example:
                id: 1
                vorname: Max
                nachname: Mustermann
                klasse: 10A
    put:
      summary: Schülerdetails aktualisieren
      requestBody:
        required: true
        content:
          application/json:
            example:
              vorname: Max
              nachname: Mustermann
              klasse: 10B
      responses:
        '200':
          description: Erfolgreich aktualisiert
          content:
            application/json:
              example:
                id: 1
                vorname: Max
                nachname: Mustermann
                klasse: 10B

```

/schueler - get, post
 /schueler/{id} - get, put, delete

100

Alle Schüler Löschen ???

200

* content:
 application/json
 example
 vorname: deleted
 nachname: ...

4. Coding-Task (25 Punkte)

- Anbei ist eine Implementierung eines Express-Endpoints zu sehen.

```
const topics = [
  { title: 'javascript', description: 'programming stuff :)', teachingUnitsCount: 8 },
  { title: 'rest', description: 're...s...t...', teachingUnitsCount: 4 },
  { title: 'docker', description: 'containers everywhere :0', teachingUnitsCount: 6 }
];

// GET-Endpoint für alle Topics
app.get('/topics', (request, response) => {
  response.status(200).json(topics);
});
```

Schreibe den notwendigen Javascript-Code, der mittels fetch-API den Endpunkt (erreichbar unter <http://api.medien.tech/v1/topics>) abrufen, und speichere die Themen-Objekte mit den Titeln `javascript` und `docker` auf Konstanten, indem die Destrukturierung angewandt wird. Gib die Eigenschaften des Topics `docker` mittels String-Interpolation aus.

```
async getTopics() {
  const response = await fetch('http://api.medien.tech/v1/topics');
  const topics = response.json();
  if (x.title) console.log(dlc);
  const js = await response.json();
  const rs = await response.json();
  const dl = await response.json();
}
```

- Docker

- Lade das Image `python` in der Version `3.12.1` vom Dockerhub.

`docker pull python:3.12.1` ✓

- Überprüfe mit einem Command, ob das Image verfügbar ist.

`docker images` ✓

- Starte einen Container mit dem geladenen Image, sodass dein Terminal nicht blockiert. Stelle sicher, dass der Container im Hintergrund ausgeführt wird.

`docker run --name python -d python:3.12.1` ✓

- Docker teilt dir nach dem Absetzen des vorigen Befehls folgendes mit:

`2cf426345e35b4a15b5e9a576b3c892a1a6a0f8f7c15db33a7baff4e1e9fb1dc`

Überprüfe den aktuellen Status des Containers.

`docker ps python 2cf42` ✓

- Nenne alle Möglichkeiten die du kennst, um den Container zu beenden.

`docker stop / kill 2cf42` (Pc ausstecken) ✓
(Strom ausfall) ✓