

## Semesterprojekt mit dem Thema: ***API-Entwicklung***

- Es soll eine Software entwickelt werden, die aus 2 Clients und einem Server besteht. Die beiden Clients sollen über ein ***REST-API*** mit dem Server kommunizieren.
- Das API soll zumindest die ***CRUD*** (Create, Read, Update, Delete) – Operationen abbilden und einen „Health“ – Status bereitstellen (der „UP“ zurückgibt, sobald der Server bereitsteht).
- Die Datensätze sollen serverseitig persistent gespeichert werden (die Technologie dafür wird aber nicht vorgegeben).
- Das Datenaustauschformat soll bevorzugt ***JSON*** (oder XML) sein.
- Die Clients sind von der zu verwendenden Technologie vorgeben. Eine Desktop-App muss in ***WPF und C#*** entwickelt werden, eine responsive Web-App in ***HTML, CSS und JavaScript*** (hier sind auch Frontend-Frameworks wie React möglich).
- Der Server ist nicht vorgegeben (kann z.B. Java mit Spring-Boot sein).
- Die Dokumentation der Software und des API muss in ***Markdown*** erfolgen.

Diese muss folgende Punkte enthalten:

- Softwaredesign (Architektur)
- Beschreibung der Software (was tut sie und wozu ist sie gut)
- API-Beschreibung
- Verwendung der API (ev. mit Code-Ausschnitten)
- Ev. Diagramme (Use-Cases, Übersichtsdiagramme)
- Diskussion der Ergebnisse (Zusammenfassung, Hintergründe, Ausblick, etc.)
- Quellenverzeichnis / Links
- Und wichtig: Die Grafiken müssen mittels ***Mermaid*** erstellt werden! (GitHub bietet eine Unterstützung für Mermaid-Diagramme)
- Als Abschluss des Projektes ist eine ***Ergebnispräsentation*** mit Darstellung des Software-Design und eines Code-Walkthrough vorgesehen.
- Alle Teile müssen in einem ***Git-Repository*** abgelegt werden (z.B. GitHub). Die README.MD ist der ideale Ort für die oben genannte Beschreibung in Markdown.
- Abgabe: Ein Link zum genannten Git-Repo und die Präsentation (PowerPoint)

## Hilfreiche Links:

- [Online FlowChart & Diagrams Editor - Mermaid Live Editor](#)
- [Swagger Editor](#)
- [\(72\) Sending JavaScript Http Requests with XMLHttpRequest - YouTube](#)
- [\(72\) Sending JavaScript Http Requests with the fetch\(\) API - YouTube](#)
- [Creating a simple JSON based API using Node.js - GeeksforGeeks](#)
- [HTML Cheat Sheet - Learn web development | MDN \(mozilla.org\)](#)
- [HTML Cheat Sheet !\[\]\(467d80e979964f7f8c752fb22248b5b7\_img.jpg\) - The best interactive cheat sheet](#)
- [Tutorial: Erstellen einer minimalen API mit ASP.NET Core | Microsoft Learn](#)

## Mögliche Themen:

- Eine einfache Chat-App (tiny WhatsApp)
- Eine einfache Notiz-App (bzw. ein Einkaufszettel)
- Ein Wettbüro für Schneckenrennen
- Eine einfache Terminfindungs-App (tiny Doodle)
- Eine App für Abstimmungen / Votings
- Eine Geburtstagsliste (Hochzeitsliste)
- Eine Linkssammlung mit Erreichbarkeits-Check
- Ein Textadventure (ev. mit Multiplayer-Modus)