Lernatelier: Projektdokumentation

[**Gruppe Deluxe** / Herrmann/Dal Corso/Hartmann]

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Version** | **Änderung** | **Autor** |
| 31.05.2023 | 0.0.1 | Informiert, Anforderungen gemacht, Entscheidungen gefällt und geplant, Vorlage für Projekt programmiert | Dal Corso |
| 07.06.2023 | 0.0.2 | programmiert: Frontend mit Backend funktionsfähig,  Get, Delete, Post Operationen,  Fertig programmieren | Hartmann |
| 14.06.2023 | 1.0.0 | Projekt kontrollieren mit Testprotokoll und Testfazit schreiben | Hartmann |

# Informieren

## Ihr Projekt

Wir entwickeln ein altes Projekt weiter, das es, wenn es fertig ist, ein funktionierendes Backend hat.

Das alte Projekt funktionierte das Frontend nicht zusammen mit dem Backend. In diesem Projekt entwickeln wir das Projekt weiter, damit das Backend mit dem Frontend zusammen funktionieren.

## Quellen

<https://www.freecodecamp.org/news/javascript-post-request-how-to-send-an-http-post-request-in-js/>

<https://stackoverflow.com/questions/5311124/how-to-empty-a-list-in-c#5311135>

<https://stackoverflow.com/questions/5378429/check-if-a-url-is-reachable-help-in-optimizing-a-class>

<https://stackoverflow.com/questions/70585335/access-control-allow-origin-in-asp-net-core-6>

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math/random>

## Anforderungen

|  |  |  |  |
| --- | --- | --- | --- |
| **Nummer** | **Muss / Kann?** | **Funktional? Qualität? Rand?** | **Beschreibung** |
| 1 | Muss | Funktional | Die Daten werden für einen User in einem Backend abgespeichert. |
| 2 | Muss | Funktional | Get: die verschiedenen Links können angezeigt werden. |
| 3 | Muss | Funktional | Delete: die verschiedenen Links können gelöscht werden. |
| 4 | Muss | Funktional | Post: ein neuer Link kann erstellt und in Backend gespeichert werden. |

## 1.4 Diagramme

---

## 1.5 Testfälle

|  |  |  |  |
| --- | --- | --- | --- |
| **Nummer** | **Voraussetzung** | **Eingabe** | **Erwartete Ausgabe** |
| 2.1 | Programm gestartet | Klick auf ‘get’ Button | Gibt alle Links |
| 3.1 | Programm gestartet | Klick auf ‘delete’ Button | Löscht Link |
| 4.1 | Programm gestartet | Gibt den Namen ein  Gibt die URL ein  Klickt auf ‘post’ Button | Link ist im Backend gespeichert |

1. Planen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nummer** | **Frist** | **Beschreibung** | **Zeit (geplant)** | **Person** |
| 1.1 | 7.6.2023 | Backend erstellen | 4 | Alle |
| 2.1 | 7.6.2023 | Get Funktion | 2 | Michael |
| 3.1 | 7.6.2023 | Delete Funktion | 2 | Dominik |
| 4.1 | 7.6.2023 | Post Funktion | 2 | Luca |

1. Entscheiden

Wir haben uns für API.net entschieden als API für das Backend.

1. Realisieren

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nummer** | **Frist** | **Beschreibung** | **Zeit (geplant)** | **Zeit (effektiv)** |
| 1.1 | 7.6.2023 | Backend erstellen | 4 | 2 |
| 2.1 | 7.6.2023 | Get Funktion | 2 | 1 |
| 3.1 | 7.6.2023 | Delete Funktion | 2 | 1 |
| 4.1 | 7.6.2023 | Post Funktion | 2 | 1 |

1. Kontrollieren

## Testprotokoll

|  |  |  |  |
| --- | --- | --- | --- |
| **Nummer** | **Datum** | **Resultat** | **Durchgeführt** |
| 2.1 | **14.6.2023** | **OK** | Herrmann |
| 3.1 | **14.6.2023** | **OK** | Herrmann |
| 4.1 | **14.6.2023** | **OK** | Herrmann |

Fazit:

Wir haben unser Programm erfolgreich weiterentwickelt. Wir konnten unser Backend mit dem Frontend zusammen verbinden und so entstand ein lauffähiges Programm. Die unterschiedlichen CRUD-Operationen konnten wir gut erstellen.

1. Auswerten

Siehe Portfolio