
# File Explorer

## Ein Projekt für den Informaticup

**Autoren:**

Brandt Marco (Matrikel-Nr. 13779)

Lauritz Wiebusch (Matrikel-Nr. 13771)

Sören Panten(Matrikel-Nr. 13766)

Schatz Daniel (Matrikel-Nr. 13796)

**Projekt:** Informaticup

**Abgabedatum:** keine Ahnung

# Inhaltsverzeichnis

# 1 Intro

Software für den Eigengebrauch zu entwickeln – mit genau den Funktionen, die man sich über Jahre hinweg im Alltag gewünscht hat – stellt nicht nur eine kreative, sondern auch eine methodisch interessante Herangehensweise dar. Gerade im Bereich der Dateiverwaltung, der tagtäglich durch den Einsatz verschiedenster File Explorer geprägt ist, zeigt sich ein eklatantes Missverhältnis zwischen praktischer Nutzung und funktionaler Erfüllung. Während bestehende Systeme primär darauf ausgerichtet sind, Dateien anzuzeigen und ein minimales Interface zur Verfügung zu stellen, bleiben tiefere Anforderungen moderner Nutzerinnen und Nutzer häufig unberücksichtigt.

Die Möglichkeit, einen File Explorer von Grund auf neu zu denken, eröffnet daher nicht nur den Freiraum zur Integration lang ersehnter Features, sondern stellt auch ein erkenntnisreiches Experiment in der Human-Computer-Interaktion dar. Welche Strukturen, Navigationsprinzipien und Automatisierungsansätze fördern tatsächliche Effizienz? Wie kann ein Interface gestaltet sein, das nicht nur funktioniert, sondern den Nutzer aktiv unterstützt, informiert und entlastet? Eine solche Neuentwicklung ist nicht lediglich als technischer Fortschritt zu verstehen, sondern als reflektierter Beitrag zur Evolution alltäglicher Softwarewerkzeuge.

## 2 Daten vom Backend zum Frontend

Datenverarbeitung findet hauptsächlich im Backend statt. Dieses besteht im Framework Tauri [1] aus Rust Code. Das Frontend kann Methoden aufrufen und über die asynchrone runntime können Daten an das Frontend geschickt werden. Dies ist recht einfach über Json oder sonst auch normalen primitiven Datentypen. Wenn selbst erstellte Objekte durchgereicht werden sollen, müssen Felder usw. im Frontend den selben Namen tragen und auch gleich Strukturiert sein. Für manche Datentypen gibt es jedoch in diversen Javascript Frame Das Frontend kann Methoden aufrufen und über die asynchrone runntime können Daten an das Frontend geschickt werden. Dies ist recht einfach über Json oder sonst auch normalen primitiven Datentypen. Wenn selbst erstellte Objekte durchgereicht werden sollen, müssen Felder usw. im Frontend den selben Namen tragen und auch gleich Strukturiert sein. Für manche Datentypen gibt es jedoch in diversen Javascript Frameworks keine passenden Datentypen. Somit ist es für folgendes Modell zum Beispiel besser dies als Json zurück zu geben.

```
  pub struct MetaData {
    version: String,
    abs_file_path_buf: PathBuf,
    abs_file_path_for_settings_json: PathBuf,
    pub abs_folder_path_buf_for_templates: PathBuf,
    pub template_paths: Vec<PathBuf>,
    all_volumes_with_information: Vec<VolumeInformation>,
    current_running_os: String,
    current_cpu_architecture: String,
    user_home_dir: String
}
```

Das oben gezeigte *struct* besitzt Typen, welche aus einem *PathBuf* hervorgehen. Diese können in *Typescript* nicht ohne weiteres realisiert werden. Ein Ansatz hierfür wäre das folgende Konstrukt, welches aber auch nicht automatisch generiert werden kann.

```
export interface VolumeInformation {
  //Implementation for the VolumeInformation struct (Fields are not relevat for the example.)
}
export interface MetaData {
  version: string;
  abs_file_path_buf: string;
  abs_file_path_for_settings_json: string;
  abs_folder_path_buf_for_templates: string;
  template_paths: string[]; // PathBufs become strings
  all_volumes_with_information: VolumeInformation[];
  current_running_os: string;
  current_cpu_architecture: string;
  user_home_dir: string;
}
```

# Literaturverzeichnis

[1] Tauri Project. *Tauri - Build Smaller, Faster, and More Secure Desktop Applications with a Web Frontend*. Accessed: 2025-05-10. 2025. URL: https://v2.tauri.app/.

# Abbildungsverzeichnis

## Eidesstaatliche Erklärung

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Arbeit selbständig verfasst und dabei keine anderen als die angegebenen Hilfsmittel benutzt habe. Sämtliche Stellen der Arbeit, die im Wortlaut oder dem Sinn nach Publikationen oder Vorträgen anderer Autoren entnommen sind, habe ich als solche kenntlich gemacht. Die Arbeit wurde bisher weder gesamt noch in Teilen einer anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

| | | | |
|---|---|---|---|
| ..... | ..... | ..... | ..... |
| Marco Brandt | Daniel Schatz | Sören Panten | Lauritz Wiebusch |

Ort, den 10. Mai 2025