

# Feuerwerk

ENDABGABE EIA2 WS22/23

ANN-KATHRIN PFEFFER, VIVIEN PESCHKE, CARA BRÜGGENDIECK

## Funktionale Analyse

Sobald der Nutzer die Webseite öffnet sieht er einen leeren Canvas. Unter dem Canvas befinden sich Einstellungen. Hier kann man das Aussehen von Feuerwerkskörpern verändern. Diese Einstellungen beinhalten die Explosionsgröße, die Farbe und die Form der einzelnen Partikel. Unter den Einstellungen sieht man einen Speicherbutton und vier Speicherstände in Form von Raketen.

Drückt man auf den Canvas erfolgt eine Feuerwerksexplosion, die die Einstellungen aus dem Formular übernimmt und ausführt. Betätigt man den Speicherbutton, wird der momentane Speicherstand des Formulars für die ausgewählte Rakete überschrieben und somit gespeichert.

Klickt der Nutzer nun auf eine der anderen Raketen, wird dessen Speicherstand in dem Formular angezeigt und kann mit Klick auf den Canvas auch wieder ausgeführt werden.

Die Anwendung kann beispielsweise so aussehen.



## Installation

Will man die Anwendung vielleicht selbst haben oder auch bearbeiten, muss man einfach die ZIP-Datei auswählen, welche in diesem Repository zu finden ist, und auf herunterladen klicken.

Sobald sie heruntergeladen wurde, kann man sie an die gewünschte Stelle in seiner eigenen Datenstruktur ziehen und dort extrahieren. Es entfaltet sich die von dem Konzepterer angelegte Datenstruktur.

Aber hier hört der Spaß noch nicht auf.

Für den Datenserver muss zunächst ein VPN aktiviert werden. Falls dieser noch nicht installiert wurde, findet man eine Anleitung auf der Website der HFU. Dieser kann von Studentis der Hochschule genutzt werden.

Der Nutzer muss nun die Anwendung „Filezilla“ herunterladen. Um Filezilla nutzen zu können braucht man einen Zugang. Ist man Student der Hochschule Furtwangen, kann man ganz einfach auf der Hochschulwebsite nach den Zugangsdaten suchen. Hier wird der Server und der Port benötigt. Nutzernamen und Passwort, sind die altbekannten Daten des Studentis selbst, welche man für jede Hochschulwebsite braucht. Hat man diese nicht, sollte man sich dringlichst im Sekretariat melden. Wahlweise kann man natürlich auch ein Issue posten. Jirka hilft sicherlich gerne.

Hat man sich erfolgreich mit Filezilla verbunden sollte ein neuer Ordner erstellt werden. Dieser wird Database genannt. In diesem muss eine index.php liegen. (Pro Tipp: unter [MingiDB/index.php at master · JirkaDellOro/MingiDB \(github.com\)](https://github.com/JirkaDellOro/MingiDB) kann man sich diese einfach kopieren, natürlich nur mit Erlaubnis)

Im selben Ordner muss eine Rockets.json Datei angelegt werden.

Jetzt sollte die Anwendung funktionieren.

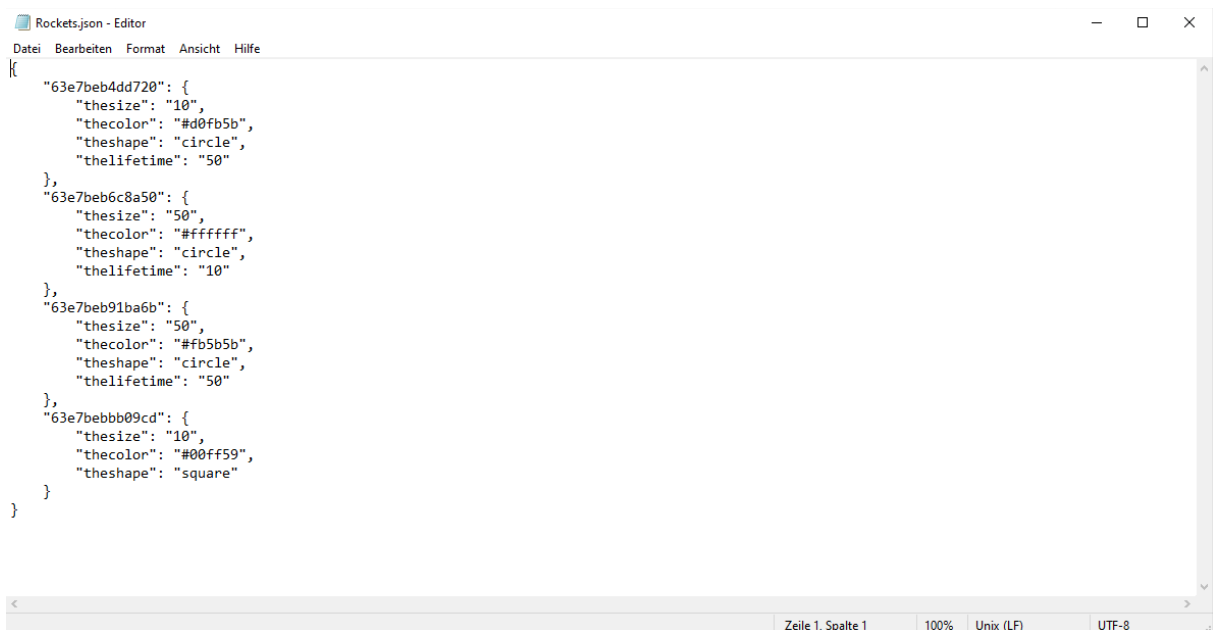
Wenn man in Typescript nun die HTML auf Live schaltet, kann die Anwendung gesehen und in den TS-Dateien verändert werden.

Allerdings ist die Serverdatei darauf angelegt, vorhandene Speicherstände nur zu überschreiben. Man müsste nun also in die Server.ts gehen und update durch „insert“ ersetzen.

```
//erzeugt URL query Befehl für Server
let query: URLSearchParams = new URLSearchParams();
query.set("command", "update");
query.set("collection", "Rockets");
```

Jetzt geht man auf die Website der HTML und drückt 4-mal speichern um 4 Speicherstände zu erstellen.

Danach geht man zurück auf Filezilla und klickt mit Rechtsklick auf die Rockets.ts und geht auf „Ansehen und Bearbeiten“. Hier sieht man vier Collections. Diese sehen ungefähr so aus.



```

Rockets.json - Editor
Datei Bearbeiten Format Ansicht Hilfe
{
  "63e7beb4dd720": {
    "thesize": "10",
    "thecolor": "#d0fb5b",
    "theshape": "circle",
    "thelifetime": "50"
  },
  "63e7beb6c8a50": {
    "thesize": "50",
    "thecolor": "#ffffff",
    "theshape": "circle",
    "thelifetime": "10"
  },
  "63e7beb91ba6b": {
    "thesize": "50",
    "thecolor": "#fb5b5b",
    "theshape": "circle",
    "thelifetime": "50"
  },
  "63e7bebbb09cd": {
    "thesize": "10",
    "thecolor": "#00ff59",
    "theshape": "square"
  }
}
Zeile 1, Spalte 1 100% Unix (LF) UTF-8
```

Um seine Speicherstände zu referenzieren, kopiert man die IDs und setzt sie an folgende Stelle in der Server.ts.

```
//je nach ausgewählter Raktete wird auf eine andere ID zugegriffen und diese geupdatet
if (rs1 == true) {
  query.set("id", "63e7beb4dd720");
} else if (rs2 == true) {
  query.set("id", "63e7beb6c8a50");
} else if (rs3 == true) {
  query.set("id", "63e7beb91ba6b");
} else if (rs4 == true) {
  query.set("id", "63e7bebbb09cd");
}
query.set("data", JSON.stringify(json));
```

Jede ID darf nur einmal vorkommen.

Zuletzt sollte man das insert, was man vorher etwas weiter oben eingefügt hat, wieder durch ein “update“ ersetzen. Jetzt hat man eigene Speicherstände erstellt die beim speichern der Einstellungen immer wieder überschrieben werden.