

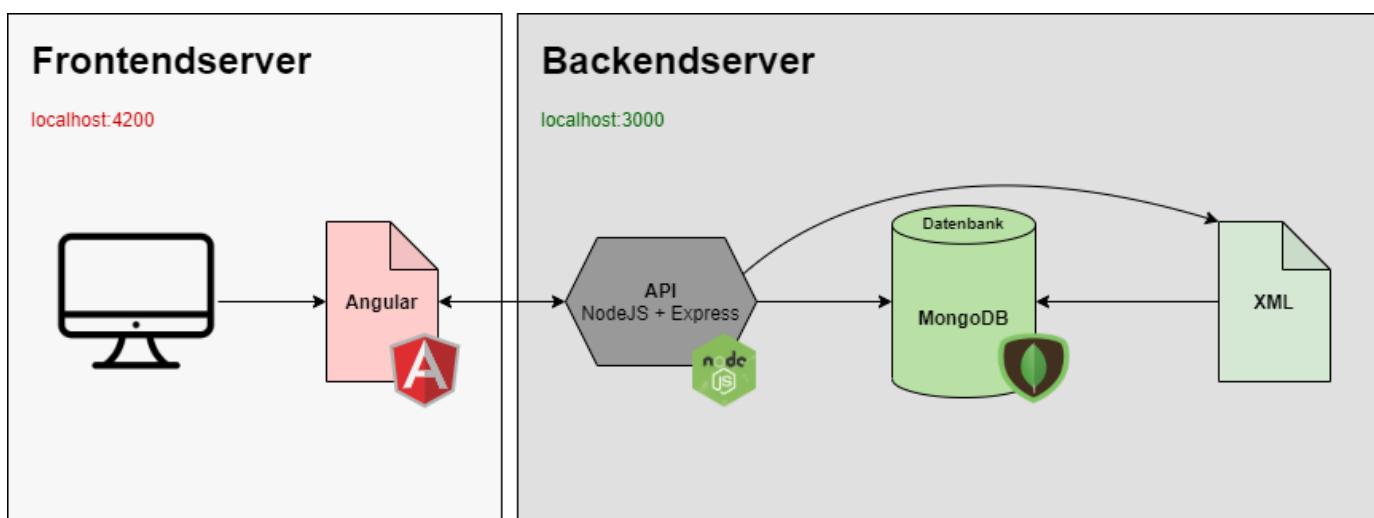
## Projektdokumentation Webprogrammierung – XML/JSON

Im Rahmen unseres Projektes ging es darum, einem Endnutzer eine Webapplikation mit einer einfach bedienbaren Oberfläche zur Verfügung zu stellen. Auf dieser soll es ihm möglich sein, Daten verschiedener Automobile abzufragen und diese bei Bedarf abzuändern, zu löschen oder einen neuen Datensatz zu erstellen. Um schnell ein gewünschtes Fahrzeug finden zu können, ist ein Suchfeld in unsere Applikation eingebaut. Genau diese Anforderungen und weitere Funktionen haben wir, das Team von AUDA, in unserer Anwendung umgesetzt und bieten Ihnen nun Zugriff auf über 1000 detaillierte Datensätze unterschiedlichster Fahrzeuge.

Um gemeinsam an dem Projekt arbeiten zu können, haben wir auf GitHub ein Repository erstellt. Dadurch hatten alle Teammitglieder Zugriff auf den aktuellsten Stand und Änderungen waren leicht nachvollziehbar. Im Nachhinein hätten jedoch zwei Repositories, jeweils eins für das Frontend und Backend, die Arbeit effizienter gemacht und Unklarheiten vermieden.

Für das Backend unserer Anwendung haben wir Node.js verwendet. Die Schnittstelle zum Frontend haben wir mittels Express-App realisiert. Eine ausführliche Dokumentation der API findet sich unter [localhost:3000/docs](http://localhost:3000/docs). Beim Frontend haben wir Angular genutzt und mit dem CSS-Framework Bootstrap die grafische Darstellung designed.

In der folgenden Grafik werden die verwendeten Tools übersichtlich dargestellt:

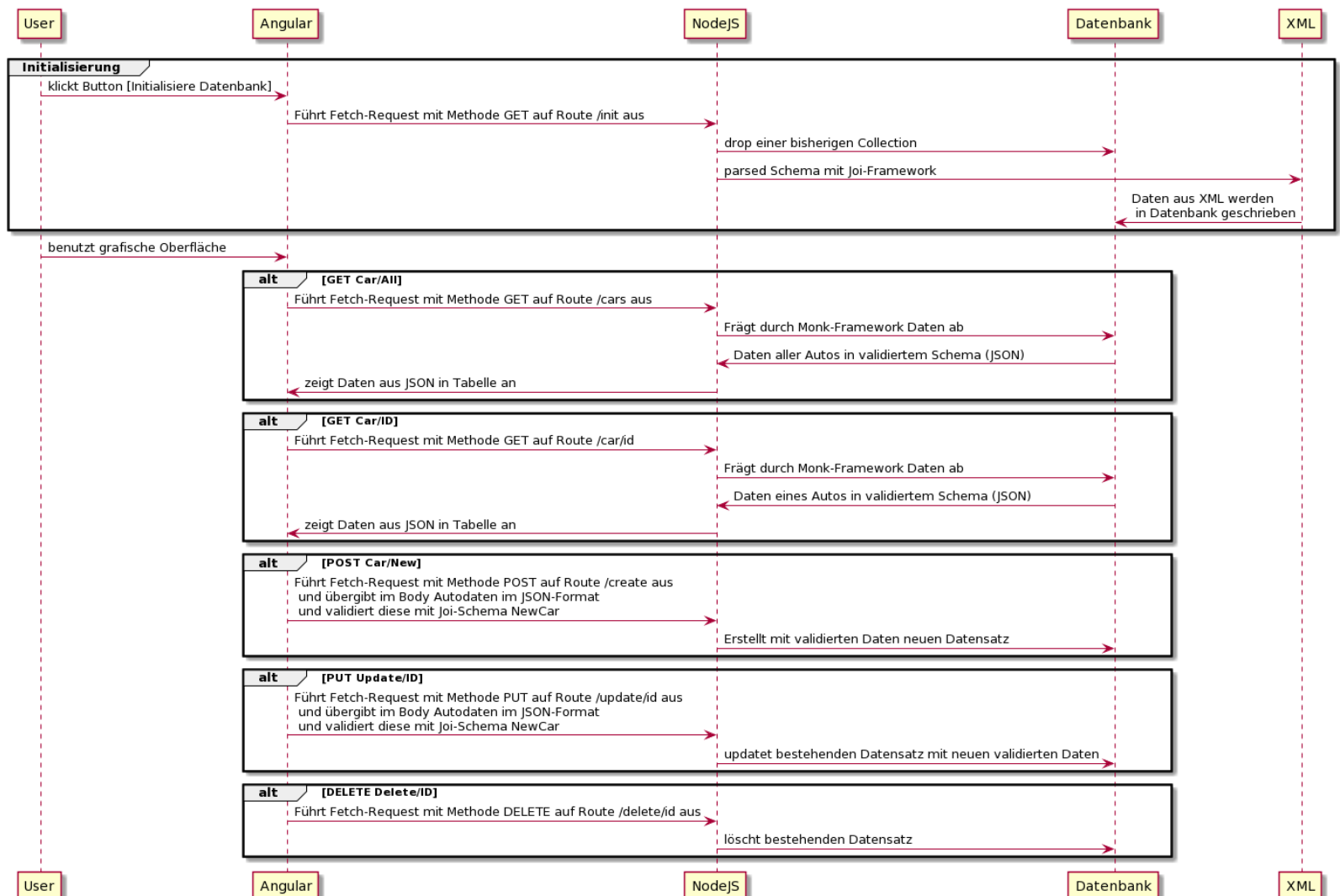


## Architektur

Wie bereits im vorangegangenen Text angeschnitten, griffen wir für das Backend auf Node.js zurück. Die darin erstellte Index-Datei ist sowohl für das Initialisieren der MongoDB verantwortlich, als auch für die Kommunikation zwischen dem Frontend und dem Backend in Form einer API. Die Initialisierungswerte der DB werden aus einer webgescrapten XML-Datei geladen, in der sich Daten der online verfügbaren ADAC-Auto-Datenbank befinden.

Auf diese kann der Endnutzer dann durch das in Angular erstellte Frontend zugreifen. Die Seiten, die der Nutzer sieht, bestehen aus einzelnen Angular-Komponenten. Mithilfe eines von uns definierten Services in Angular können wir per Fetch-Request auf die Daten der API zugreifen. Die aufrufbaren Endpunkte wurden dabei in der Express App definiert. Das zusätzlich verwendete Framework Monk kann auf die Datenbank MongoDB zugreifen und gewünschte Interaktionen durchführen. Handelt es sich bei der aufgerufenen Methode um eine POST oder PUT Methode, wird der mitgegebene Body mithilfe des Frameworks Joi validiert.

Im nachfolgenden Sequenzdiagramm sind alle möglichen Abläufe grafisch dargestellt:



## Funktionen und Aufbau

Um auf unsere umfangreiche Datenbank zugreifen zu können, muss man auf der Startseite, die unsere Anwendung kurz beschreibt, den „Los Geht's“ Button betätigen. Die Daten aus der XML-Datei werden mit Klick auf den „Datenbank initialisieren“-Button in die Datenbank geladen.

AUDA  
Die Autodatenbank deines Vertrauens  
Finde spannende Daten über fast jedes Auto

HERZLICH WILLKOMMEN BEI DER AUTODATENBANK


Wenn Autos Dein Ding sind, bist Du hier genau richtig.

Wir haben für dich eine Datenbank entwickelt,  
in der du sämtliche Daten von den unterschiedlichsten Autos finden kannst.  
Egal welche Daten dich interessieren, ob Kraftstoffverbrauch, PS-Zahl oder  
die Maximalgeschwindigkeit eines Fahrzeugs - hier wirst du fündig!


Um jetzt loszulegen, drück den "Start"-Button:

Los Gehts !

















Datenbank initialisieren

Man gelangt auf die tabellarische Übersicht der über tausend Fahrzeugdatensätze. Es besteht die Möglichkeit nach Fahrzeugen zu filtern oder zu suchen. Findet man sein gesuchtes Fahrzeug, gelangt man mit dem Button „“ zu der Fahrzeugübersicht mit den Detaildaten.

AUDA  
Die Autodatenbank deines Vertrauens  
Finde spannende Daten über fast jedes Auto

Suchen 

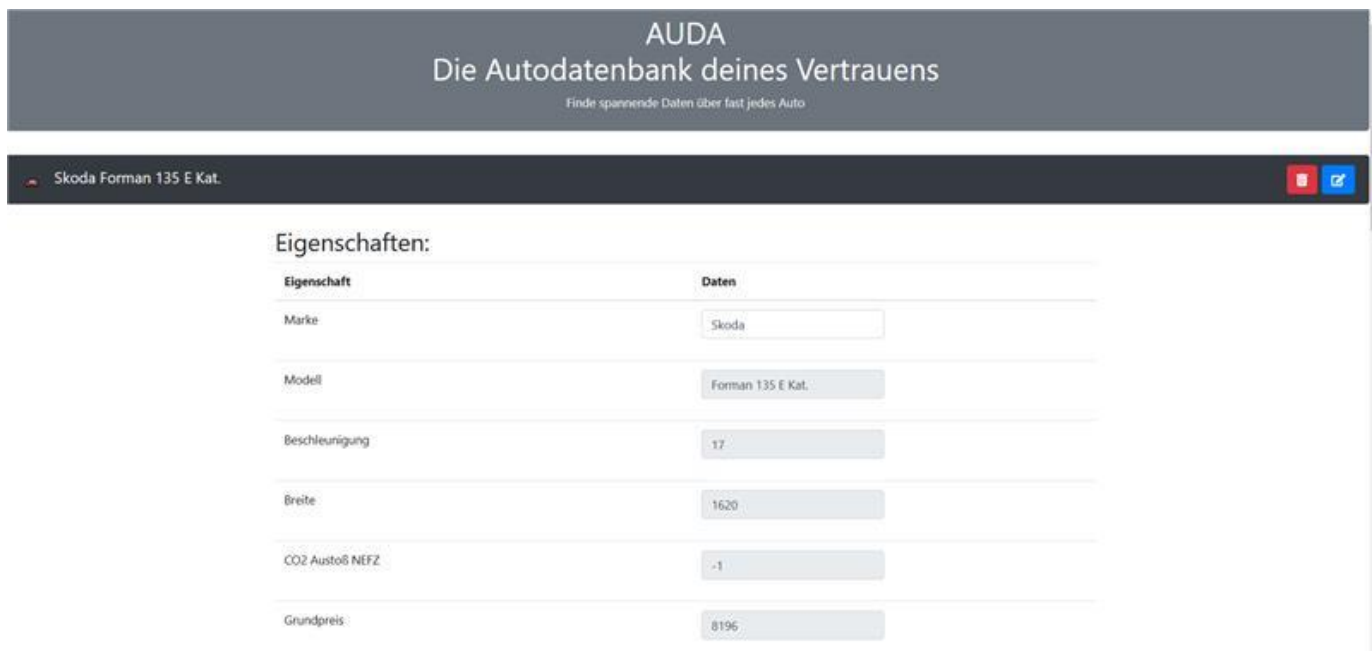
Übersicht aller Automobile:

Marke	Modell	ID	Details	Löschen
Skoda	Forman 1.35 E Kat.	602d59ee6e5a5c3810d51b73		
Skoda	Rapid Spaceback 1.0 TSI	602d59ee6e5a5c3810d51b74		
Skoda	Roomster 1.2 12V	602d59ee6e5a5c3810d51b75		
Skoda	Roomster 1.2	602d59ee6e5a5c3810d51b76		
Skoda	S105	602d59ee6e5a5c3810d51b77		
Skoda	Yeti 1.2 TSI	602d59ee6e5a5c3810d51b78		
Opel	Astra 1.2 DI Turbo	602d59ee6e5a5c3810d51b79		
Opel	Astra 1.4	602d59ee6e5a5c3810d51b7a		

## Funktionen und Aufbau (II)


Um auch in Zukunft eine umfangreiche Datenbank gewährleisten zu können, können neu auf den Markt gekommene Fahrzeuge mit dem „“ Button hinzugefügt werden.


Hier werden neben Marke, Modell und Baujahr auch spezifische Information wie Kraftstoffverbrauch, Leistung, Hubraum, etc. angezeigt.



The screenshot shows the AUDA web application interface. At the top, the header reads "AUDA Die Autodatenbank deines Vertrauens" with the tagline "Finde spannende Daten über fast jedes Auto". Below the header, a dark bar displays the selected car: "Skoda Forman 135 E Kat." with edit and delete icons. The main content area is titled "Eigenschaften:" and contains a table with two columns: "Eigenschaft" and "Daten".

Eigenschaft	Daten
Marke	Skoda
Modell	Forman 135 E Kat.
Beschleunigung	17
Breite	1620
CO2 Ausstoß NEFZ	-1
Grundpreis	8196

Trotz größter Sorgfalt, kann es vorkommen, dass manche Daten nicht korrekt sind. Um die Fehler zu beseitigen, kann man über den Button „“ das entsprechende Fahrzeug bearbeiten.

Für den Fall, dass es Redundanzen gibt und sich ein Fahrzeug zweimal in unserer Datenbank befindet, kann dieses über den Button „“ gelöscht werden.