

# Web-Entwicklung: Übungsblatt 7

---

In dieser Übung soll schrittweise eine kleine, aber vollständige Web-Anwendung zur Anzeige einer Bildersammlung realisiert werden. Die Anwendung besteht aus einer clientseitigen Browser-Anwendung (Frontend) sowie einer serverseitigen Node.js-Anwendung (Backend), die REST-konform via HTTP kommunizieren (RESTful HTTP).

## Aufgabe 1: Frontend

Das in Aufgabe 2 zu realisierende Backend wird dem Frontend per HTTP eine Liste von URLs auf Bild-Dateien liefern. Da Sie das Backend erst später realisieren, müssen Sie zunächst wieder mit Mock-Daten arbeiten. Verwenden Sie dazu einen Dienst wie z.B. <https://placekitten.com> oder <https://picsum.photos>, um eine Liste von URLs auf Bild-Dateien zu erstellen.

Zum Beispiel liefert <https://placekitten.com/1920/1080> ein beliebiges Bild mit FullHD-Abmessung, <https://placekitten.com/1920/1080?image=0>, <https://placekitten.com/1920/1080?image=1>, <https://placekitten.com/1920/1080?image=2> usw. eine ganze Serie von Bildern mit entsprechenden Abmessungen.

Im Browser sollen die referenzierten Bilder beim Öffnen der Web-Seite mittels DOM-Manipulation in einer Art Galerie als verkleinerte sog. Thumbnails präsentiert werden. Klickt der Nutzer auf ein Thumbnail soll das Bild in der Originalgröße (höchstens aber fensterfüllend) angezeigt werden. Drückt der Nutzer erneut auf das Fenster schließt es sich wieder.

Während sich der Mauszeiger über einem Thumbnail in der Galerie befindet sollen in einem Overlay drei Schaltflächen (siehe zweites Bild in der zweiten Zeile im folgenden Screenshot der Galerieansicht) mit folgender Funktionalität angezeigt werden:

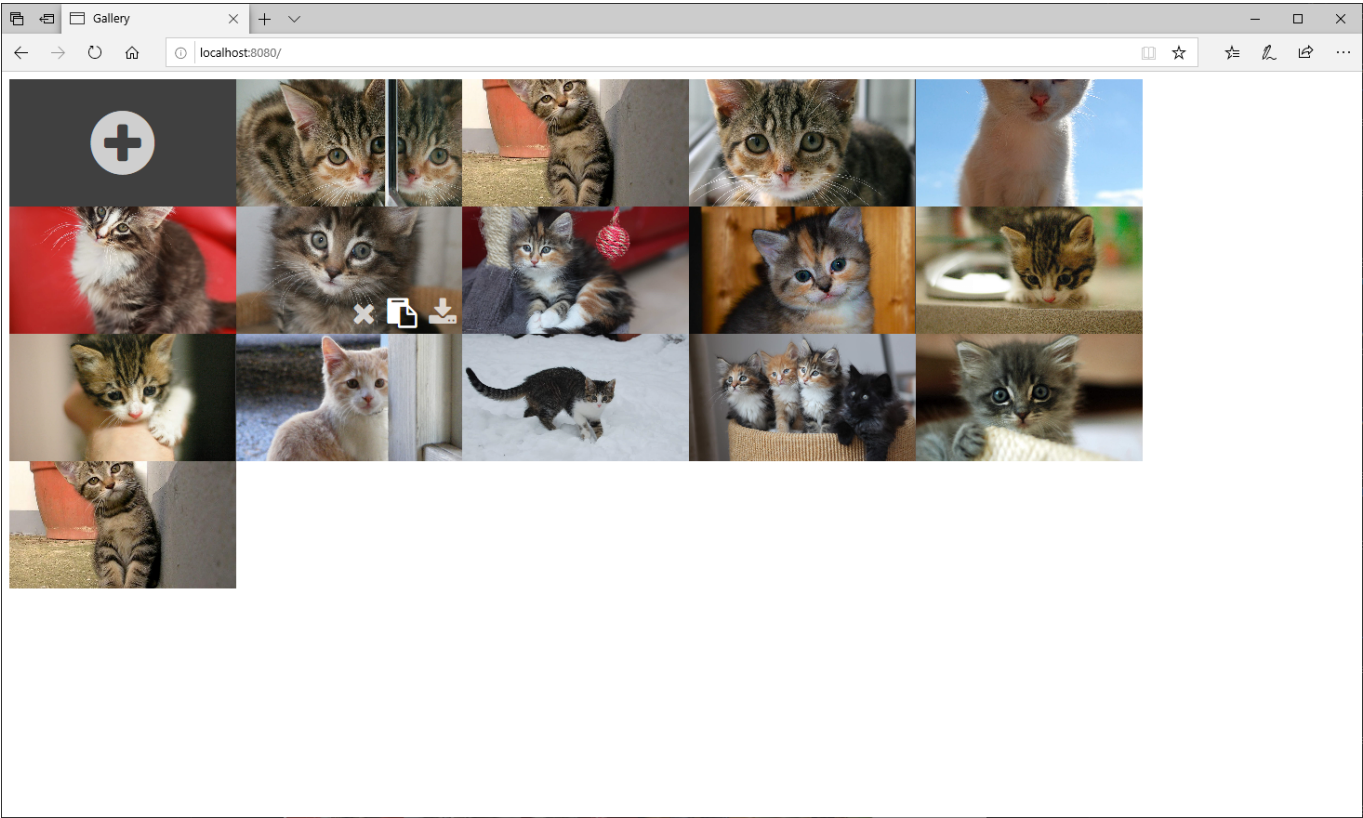
- Löschen des Bildes aus der Galerie
- URL des Bildes in die Zwischenablage kopieren
- Download der Bild-Datei

Zusätzlich dazu soll es eine Schaltfläche mit gleicher Größe wie die Thumbnails geben, um eine neue Bild-Referenz in die Galerie einfügen zu können. Zunächst ist hier keine manuelle Eingabemöglichkeit vorgesehen. Fügen Sie einfach ein weiteres zufälliges Bild vom o.g. Dienst in die Liste ein.

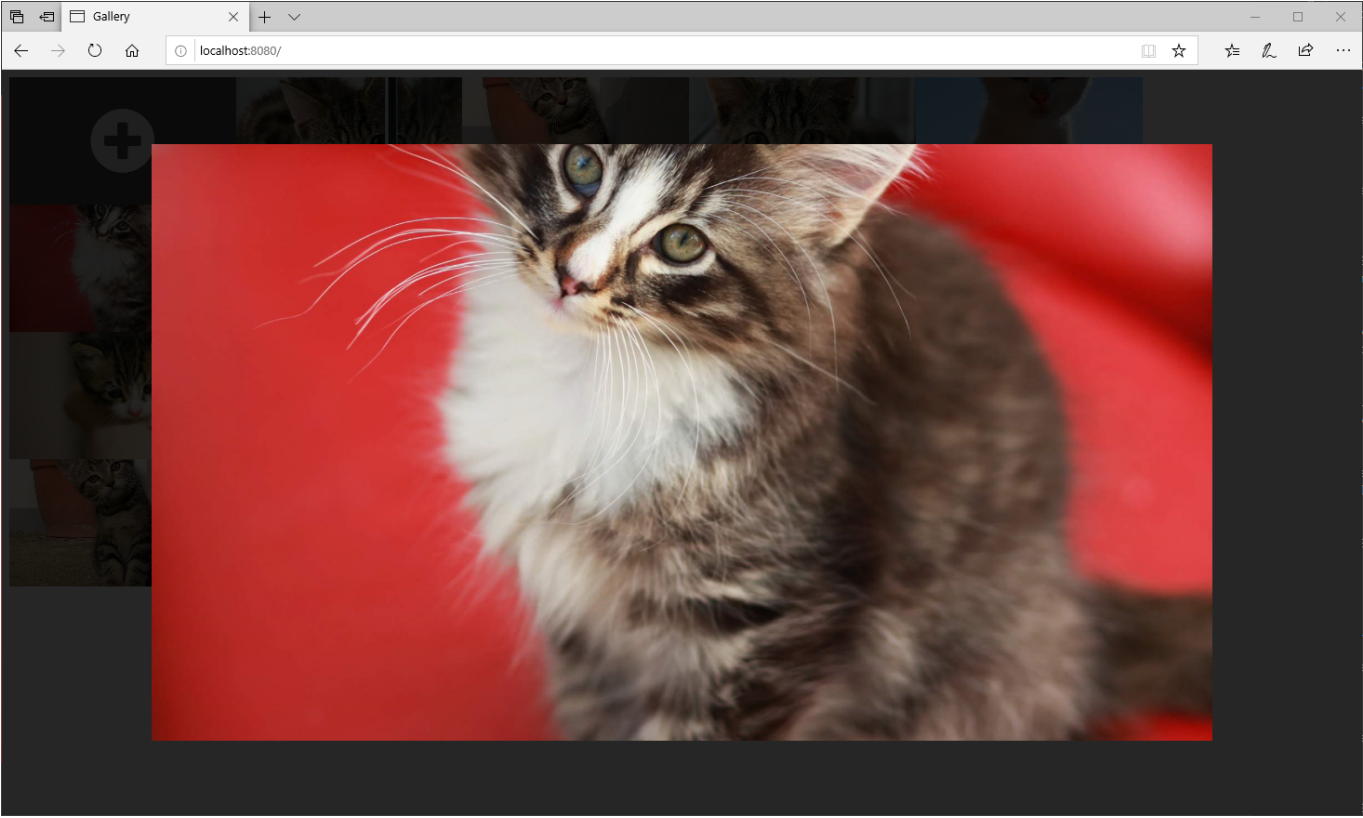
Achten Sie auf eine sinnvolle Strukturierung Ihrer Anwendung. Versuchen Sie, die Element Ihrer Oberfläche durch möglichst kompakte JavaScript-Komponenten abzubilden. Diese sollten möglichst lose gekoppelt sein und über Events miteinander kommunizieren.

Hinweis: Für Icons können Sie z.B. die Icon-Sets [Material Icons](#) oder [Font Awesome](#) verwenden.

Galerieansicht



Detailansicht



## Aufgabe 2: Backend

Ihr bisheriger Bild-Datensatz wird bei jedem Neuladen der Webseite neu erzeugt. Um Daten auch über die Laufzeit einer Browser-Anwendung hinweg speichern zu können, soll nun ein Backend realisiert werden, über welches Bildquellen (d.h. URLs auf Bilder) persistent zum Datensatz hinzugefügt oder aus dem Datensatz entfernt werden können.

Verwenden Sie hierfür eine der Datenbank-Lösungen, welche Sie in der Vorlesung kennengelernt haben (z.B. MongoDB oder SQLite).

Das Backend soll die von ihm verwalteten Bildquellen über eine HTTP-Schnittstelle zur Verfügung stellen. Falls Sie die Vorlesungseinheit zum Thema "RESTful HTTP" zum Zeitpunkt der Aufgabenbearbeitung bereits gehört haben, soll diese HTTP-Schnittstelle REST-konform umgesetzt werden. Führen Sie dazu einen Analyse- und Entwurfsprozess wie in der Vorlesung gezeigt durch und setzen Sie Ihren Entwurf mithilfe von Express.js um. Anderenfalls können Sie einen "einfachen" HTTP-Server realisieren und die REST-Grundprinzipien nachträglich umsetzen.

Passen Sie Ihre Frontend-Lösung aus Aufgabe 1 so an, dass diese die neue Backend-Schnittstelle anspricht.

## Aufgabe 3: Formular

Bis zu diesem Zeitpunkt fügen Sie über die entsprechende Schaltfläche zufällig generierte Bildquellen zur Datenbank hinzu. Nun soll im Frontend ein Formular erstellt werden, über welches man die Bild-URL manuell angeben kann. Die Schaltfläche zum Hinzufügen eines Bildes zeigt dieses Formular (z.B. in einem modalen Dialog) an. Bevor die URL an das Backend kommuniziert wird, soll eine Gültigkeitsprüfung stattfinden.

## Bonus-Aufgabe 4: Datei-Upload

Erweitern Sie Ihre bisherige Lösung, sodass Bilder auch aus Bild-Dateien im Dateisystems des Nutzers über den Browser hochgeladen und in der Datenbank gespeichert werden können. Im Frontend soll dazu das Formular um eine Schaltfläche ergänzt werden, die das Auswählen einer Datei im Dateisystem erlaubt. Im Backend muss nun auch der Handler zum Anlegen neuer Bild-Ressourcen angepasst werden.