

1. Vorbereitung

1. **Das Interface AssocStringString beschreibt ein homogenes (=gleichmäßig aufgebaut) assoziatives Array. Was bedeutet dies und was ist der Unterschied zu den heterogenen(uneinheitlich, intern gegliedert), die Sie bereits kennen?**

Quelle: <https://matthewhorne.me/javascript-array-types/>

Ein homogener Array ist ein Array der nur mit einem Datentyp gleichzeitig arbeitet, dh. mit string, number oder boolean, z.B.

```
var array = ["Matthew", "Simon", "Luke"];
```

```
var array = [27, 24, 30];
```

```
var array = [true, false, true].
```

Ein heterogener Array ist ein Array der mit mehreren Datentyp gleichzeitig arbeitet, dh. mit string, number und/oder boolean, z.B. var array = ["Matthew", 27, true] .

2. **Was geschieht in Zeile 27 von NodeTest.ts?**

```
let query: AssocStringString = Url.parse(_request.url, true).query;
```

Quelle: https://nodejs.org/api/http.html#http_message_url

Es wird eine Variable namens „query“ deklariert, sie bildet einen homogenen Stringarray der auf die query, dh. auf den Part hinterm Fragezeichen einer URL, sofern vorhanden ist, zu greift. Durch das parsen der URL werden aus dem string die Daten herausgelesen und true wirkt als zweites Argument unterstützend dafür.

Beispiel:

<https://www.youtube.com/watch?v=BoG7c4SCGEA&index=8&list=PLiWZD0929jPfqtVj06uxZUT14SAgVbw>

Query extrahiert:

v=BoG7c4SCGEA &

index=8 &

list=PLiWZD0929jPfqtVj06uxZUT14SAgVbw

Dabei ist das „&“ Zeichen lediglich das Verbindungsglied zwischen den Parametern.

3. **Die for-in-Schleife in Zeile 30 ist ein Ihnen wahrscheinlich unbekanntes Konstrukt, wie arbeitet sie?**

Quelle: <https://www.teialehrbuch.de/Kostenlose-Kurse/JavaScript/12750-for-in-Schleife.html>

Mit ihr kann über alle Eigenschaften eines Objekts iteriert/wiederholt laufend werden.

Codebeispiel NodeTest.ts:

```
let query: AssocStringString = Url.parse(_request.url, true).query;
```

```
let key: string;
```

```
for (key in query)
```

```
    console.log(key + ":" + query[key]);
```

```
for (indexvariable in array)
```

```
    //führe eine Aktion mit array[indexvariable] aus
```

Die Indexvariable durchläuft den Index jedes belegten Array-Elements genau einmal durch, wobei die Reihenfolge der durchgenommenen Elemente nicht konstant ist. Die Schleife braucht erst geschweifte Klammern sobald man mehr als nur eine for-Schleifenbedingung einbauen möchte.

4. Was bewirkt die Header-Information "Access-Control-Allow-Origin"?

```
_response.setHeader("Access-Control-Allow-Origin", "*");
```

Quelle: https://de.wikipedia.org/wiki/Cross-Origin_Resource_Sharing

Sendet der Server den Header, so erlaubt er den Abruf Webbrowsern oder auch anderen Webclients von Seiten, die auf dem im Header genannten Server gespeichert sind, somit wird Cross-Origin-Requests ermöglicht.

4. Ajax

1. Untersuchen Sie den Code in `SendData.ts` und beschreiben Sie die einzelnen Funktionen.

Init: öffnet ein Consolenlog mit „Init“ zur Kontrolle ob die Funktion läuft, danach startet sie die Funktion `setupColorDivs`

setuopColorDivs: deklariert die Variable `colors`, welche ein StringArray enthält und die Variable `divs`, welche auf die Divs im html.Dokument zu greift, danach springt eine Vorschleife an, die über alle Divs aus `divs` läuft. Die divs bekommen einen Style zugeschrieben, eine Hintergrundfarbe rot, grün oder blau, welche aus dem Array `colors` herausgelesen wird. Und alle Divs bekommen ein click-Event namens `handleClickOnDiv`.

handleClickOnDiv: ruft die Funktion `handleEvent` auf, in der Konsole wird die `BackgroundColor` des divs ausgegeben und die Farbe wird an die Funktion `sendRequest` weitergeleitet

sendRequest: es wird eine Variable deklairert `xhr`, welche den Request an den Server weiterleitet, dann wird ein Port 8100 geöffnet und ein Eventlistener mit Funktionsaufruf `handlestateChange` hinzugefügt

handleStateChange: nimmt XMLHttpRequest von `sendRequest` entgegen und wirft sobald der Prozess endet eine Konsolenausgabe aus der dies anzeigt, ebenso den Datentypen und Anfrage erfolgreich war und ein Alert der Response.

2. Was macht die Klasse XMLHttpRequest bzw. deren Objekte?

Diese Objekte sorgen für einen Datenaustausch zwischen Browser und Server. Dabei ist es möglich Daten auf einer Webseite zu aktualisieren, ohne diese komplett neu zu laden.

3. Was bedeutet Ajax?

Ajax sorgt für asynchrone Daten zwischen Server und Browser. Hierbei wird ein XMLHttpRequest Objekt erstellt und an den Server weitergesendet. Dort wird dann eine Antwort erstellt und zurück an den Browser gesendet, die dort ausgelesen wird und die Seite kann aktualisiert werden.