

Aufgabe 5: Vorbereitung

1. Was beschreibt das Interface `AssocStringString` und wozu kann es verwendet werden?

`AssocStringString` ist ein Interface, welches ein homogen assoziatives Array beinhaltet.

Über den Parameter `key:string` kann ein Element vom Typ `string` zugeordnet und darüber abgerufen werden. Zudem definiert `key:string` auch den Namen des Elements vom Typ `string`.

2. Was geschieht in Zeile 9 und was ist das Ergebnis?

Es wird ein `let` mit Namen `port` vom Typ `number` erstellt und es wird der Wert `process.env.PORT` zugewiesen. Ich habe das so verstanden, dass hier der Wert des `PORT`, des Environments auf dem die App „läuft“ dem `let` zugewiesen werden soll.

Wird hier nun kein gültiger Wert für den Port gefunden, wird er auf 8100 gesetzt.

3. Was geschieht in Zeile 25 von `ServerTest.ts`?

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

Ein `let` mit Namen `query` vom Typ `AssocStringString` wird erstellt. Der zugewiesene Wert ist das Ergebnis der Methode `Url.parse()`. Diese nimmt einen URL `string`, dieser wird geparkt, nach `boolean` auf `true` oder `false` geprüft und gibt bei einem Rückgabewert `true` ein URL Objekt zurück, welches dann in `query` eingefügt wird.

4. Wie arbeitet die `for-in`-Schleife in Zeile 29?

Die `for-in` schleife setzt hier die Bedingung für die „Ausgabe“ `_response`. Es wird geprüft, ob `key:string` des `AssocStringString` in `query` vorhanden ist (also das in Zeile 25 geklappt hat und der Rückgabewert des `Url.parse()` `true` ist)

Falls das der Fall ist wird der `CodeBlock` darunter ausgeführt, also die `_response` generiert.

5. Was bewirkt die Header-Information `Access-Control-Allow-Origin`?

Damit jede externe Ressource/Ursprung auf die Ressource zugreifen kann (Anfragen ohne Anmeldeinformationen), gibt man als `Access-Control-Allow-Origin` ein * als Platzhalter an. Um den Zugriff zu begrenzen kann man hier auch eine URL als `<origin>` angeben.