

Vorbereitung

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

Mit `AssocStringString` wird ein homogenes assoziatives Array beschrieben. Diesem werden Daten von Typ `string` zugeordnet, welche dann mit dem „key“ als Schlüssel gespeichert und anschließend auch aufgerufen werden können.

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

Zeile 9: `let port: number = process.env.PORT`

Mit „`port: number`“ wird die Variable „`port`“ dem Typ `number` zugeordnet. Übersetzt man `port` aus dem Englischen so bedeutet dies „Öffnung“ oder „Hafen“. Ein Port ist also ein Hafen, in dem gewisse Informationen ankommen können. Dieser „Hafen“ bekommt nun „`process.env.PORT`“ zugeordnet. Dies repräsentiert den Status der Systemumgebung (`process.env`) einer Applikation wenn sie startet. Die im folgenden Verlauf des Codes auftauchende `if`-Abfrage prüft dann den Port. Ist dieser „`undefined`“, so wird er auf `8100` festgelegt. Mit „`localhost`“ kann man dann über den Browser auf ihn zugreifen.

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

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

Hier wird die Variable „`query`“ vom Typ „`AssocStringString`“ erstellt. Dieser Variable wird „`Url.parse(_request.url, true).query`“ zugewiesen. „`Url.parse`“ wandelt eine URL-String in ein URL-Objekt um. Ist die „`URL.parse`“ wahr, also „`true`“, so wird aus der zuvor definierten Variablen „`query`“ ein neues Objekt vom Typ „`AssocStringString`“. Durch „`.query`“ greift man dann auf den Queryteil der URL zu. Ist „`query`“ jedoch falsch (`false`) so wird anstelle des Objekts nur ein String erstellt.

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

Zeile 29: `for (let key in query)`

Bei dieser `for-in`-Schleife werden im Array nur die Stellen, die auch den Schlüssel beinhalten beachtet. Anders als bei der normalen `for`-Schleife werden die Stellen ohne den Schlüssel einfach ignoriert. Auch wird bei hier nicht die Reihenfolge des Arrays beachtet, d.h. man kann sich also nicht darauf verlassen, in welcher Reihenfolge die Array-Elemente durchlaufen werden.

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

„`Access-Control-Allow-Origin`“ ist ein „Cross-Origin Resource Sharing“ (CORS) header. Dies ist ein Mechanismus, der eine domänenübergreifende Kommunikation des Browsers erlaubt, d.h. es bestimmt die Freigabeeinstellungen der Ressource.