

# TypeScript in der Praxis

Livecoding

Johannes Dienst

Was nicht gezeigt wird!

**Endlose Downloads**

**Funktionale Magie**

**Schlaue Abkürzungen**

**Was gezeigt wird!**  
**Erkenntnisse aus der Praxis**  
**Aufsetzen eines Workflows**  
**Vorteile von TypeScript**  
**(Hoffentlich!) live**

# TypeScript

```
$> npm install -- save-dev typescript
```

# Deklarationsdateien

```
$> npm install --save-dev @types/jquery
```

# src/tsconfig.json

```
$> ../../node_modules/typescript/bin/tsc --init  
message TS6071: Successfully created a tsconfig.json file.
```

# tsconfig.json (2)

```
{
  "compilerOptions": {
    "module": "commonjs",
    "target": "es5",
    "noImplicitAny": false,
    "sourceMap": false,
    "rootDir": ".",
    "outDir": "../build",
    "removeComments": true
  },
  "compileOnSave": false,
  "moduleResolution": "node",
  "exclude": [
    "node_modules"
  ]
}
```

# package.json - TypeScript Tasks

```
"deploy:tsc" : "./node_modules/typescript/bin/tsc -p ./src" ,  
"watch:tsc" : "nodemon --verbose -w src/ --on-change-only -d 1 -e ts --exec \" npm run
```



# TypeScript Kompilieren

```
$> npm run deploy:tsc
```

```
> superherorelations@1.0.0 deploy:tsc /home/jdienst/git/SuperheroRelations
```

```
> ./node_modules/typescript/bin/tsc -p ./src
```

# TypeScript Watch Job

```
$> npm run watch:tsc

> superherorelations@1.0.0 watch:tsc /home/jdienst/git/ SuperheroRelations
> nodemon --verbose -w src/ --on-change-only -d 1 -e ts --exec "npm run deploy:tsc"

[nodemon] 1.11.0
[nodemon] to restart at any time, enter `rs`
[nodemon] ignoring: /home/jdienst/git/ SuperheroRelations /.git/**/* .nyc_output .sass-
/jdienst/git/ SuperheroRelations /node_modules /**/*
[nodemon] watching: /home/jdienst/git/ SuperheroRelations /src/**/*
[nodemon] watching extensions: ts
[nodemon] child pid: 10695
[nodemon] watching 8 files
```

# src/routes/base\_route.ts anlegen

```
export class BaseRoute {  
  constructor(){};  
}
```

# Datenbank- hero\_db.json

```
{
  "heros" : [
    {
      "id": 1,
      "name": "Arrow"
    },
    {
      "id": 2,
      "name": "Flash"
    },
    {
      "id": 3,
      "name": "Batman"
    },
    {
      "id": 4,
      "name": "Superman"
    }
  ]
}
```

# src/hero\_db.ts - Modell als Interfaces

```
interface Hero {  
  id: number;  
  name: string;  
}  
  
interface HeroDB {  
  heros: Array;  
}
```

# HeroRoute - src/routes/hero\_route.ts

```
/// <reference path="../../hero_db.ts" />  
const data: HerosDB = require("../../../../hero_db.json");
```

```
import {Request, Response, Router, NextFunction} from "express";  
import {BaseRoute} from "../base_route";
```

```
export class HeroRoute extends BaseRoute {  
  constructor() {  
    super();  
  }  
}
```



```
public static create(router: Router) {
    let routes = new HeroRoute();

    router.get("/heros",
        routes.list
    );
}

private list(req, res: Response, next) {
    res.status(200).json(data);
}
```

# server.ts

```
import {HeroRoute} from "../routes/hero_route" ;
```

# server.ts - routes()

```
HeroRoute.create (router) ;
```

# index.ts - Startpunkt

```
function getAsync() {
  console.log("getAsync");
}

async function getAsyncExp() {
  console.log("getAsyncExp");
}

$( document ).ready(function() {
  console.log("ready!");

  $("#get_async").on("click", getAsync);
  $("#async_await").on("click", getAsyncExp);
});
```

# Generisches getAsync

```
function getAsync<T>(): Promise<T>: {  
  return Promise.resolve(  
    $.getJSON( "http://localhost:8089/heros" )  
  );  
}
```

# Generisches getAsync - Verwendung

```
$("#get_async").on("click",  
  () => {  
    getAsync<HerosDB>(). then(appendHeros);  
  }  
);
```

# appendHeros - String Templating

```
function appendHeros(herosDB: HerosDB) {  
  let tableBody = $("table tbody");  
  tableBody.find("tr").remove();  
  herosDB.heros.forEach(  
    (hero) => {  
      tableBody.append(  
        `| ${hero.id}</td><td>${hero.name}</td></tr>`       )     }   ); } |

```

# webpack - Tasks

```
"deploy:webpack" : "./node_modules/webpack/bin/webpack.js build/index.js build/index_bu  
"watch:webpack" : "nodemon --verbose -w build/index.js --on-change-only -d 1 --exec \"
```



# async/await

```
async function getAsyncExp() {  
  let herosDB = await getAsync<HerosDB>();  
  appendHeros(herosDB);  
}
```



JohannesDienst



johannesdienst.net



info@johannesdienst.net