

UI5con

# Migrating an existing freestyle UI5 project from JavaScript to TypeScript

Törehan Gören, TGW Systems Integration GmbH



# Agenda

## TypeScript

Ever wondered  
about the difference  
between JavaScript  
and TypeScript?

## Why is this Migration

Do we really need  
this?  
Motivation behind

## Migration Strategy

Preparation  
Progressive  
Migration  
Setting up

## Case Studies

The real world?  
Is it worth to-do?

# TypeScript



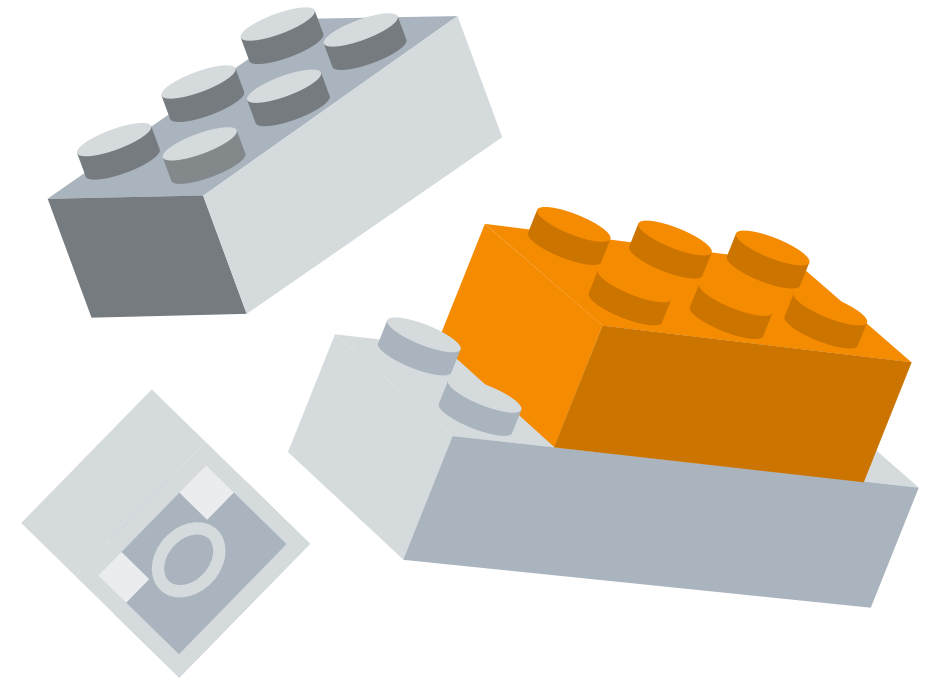
# TypeScript

## Basic Intro

- JavaScript plus types
- Static Typing
- Early Error Detection
- Improved Code Readability and Maintainability
- Scalability
- Integration with Existing JavaScript Code

```
let x:string;  
x = 1; // throws: Type 'number' is not  
assignable to type 'string'.ts(2322)
```

# Why we need this

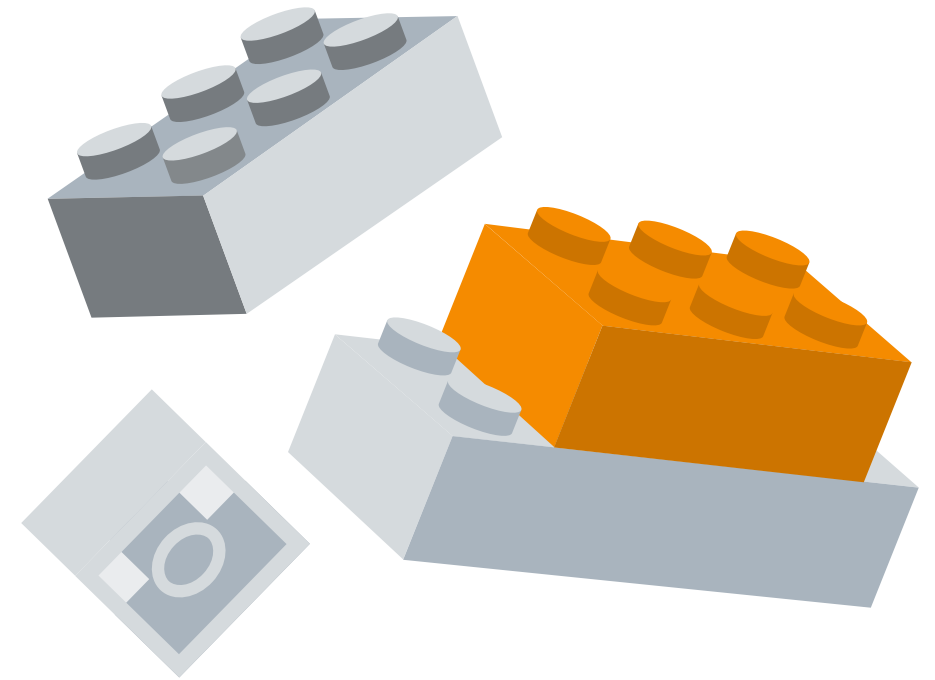


# Why we did this

- **Do we really need this changes?**
- **What was our motivation for making these major changes?**

# Migration Strategy

UI5con



# Migration Strategy

## Two directions

- **Analyse and develop Project in a better way**
  - **convert JS to TS of Project**
- 
- <https://sap.github.io/ui5-typescript/releasesnotes.html>  
<https://github.com/SAP/ui5-typescript>



# Migration Strategy - convert JS to TS of Project

## 1. Step: Enhanced Structural Code Modifications: Optimizing Module Loading and Class Definitions

- Before

```
sap.ui.define([
    "tgw/ui5/con/controller/BaseController"
], function (BaseController) {
    "use strict";

    return BaseController.extend("tgw.ui5.con.controller.App", {
        onInit: function () {

        }
    });
});
```

- After

```
import BaseController from "../BaseController";
/**
 * @namespace tgw.ui5.con.controller
 */
export default class App extends BaseController {
    public async onInit(): Promise<void> {

    }
}
```

# Migration Strategy - convert JS to TS of Project

## 2. Step: Standard TypeScript Code Adaptations

- Add type information to method parameters.
- Add private member class variables (with type information) to the beginning of the class definition.
- Convert conventional functions to arrow functions when `Function.bind(..)` is used
- Define further types and structures needed withing the code, if applicable.

# Migration Strategy - convert JS to TS of Project

## 2. Step: Standard TypeScript Code Adaptations

```
export default class Dialogs extends ManagedObject {  
  
    private mainController: Main  
  
    constructor(mainController: Main) {  
        super();  
        this.mainController = mainController;  
    }  
  
    public getDialog(dialogName: string): Dialog {  
        return this.dialogs.find(dialog => dialog.name === dialogName).dialog.getDialog();  
    }  
  
}
```

# Migration Strategy - convert JS to TS of Project

## 3. Step: Casts for Return Values of Generic Methods

- **Before**

```
onInit: function () {  
    this.getView().getModel("startUpModel").loadData("/sap/bc/ui2/start_up");  
}
```

- **After**

```
public async onInit(): Promise<void> {  
(this.getView().getModel("startUpModel") as JSONModel).loadData("/sap/bc/ui2/start_up");  
}
```

# Best practice

- The real world?
- Is it worth to-do?



# Useful Links

<https://github.com/SAP/ui5-typescript>

<https://sap.github.io/ui5-typescript/>

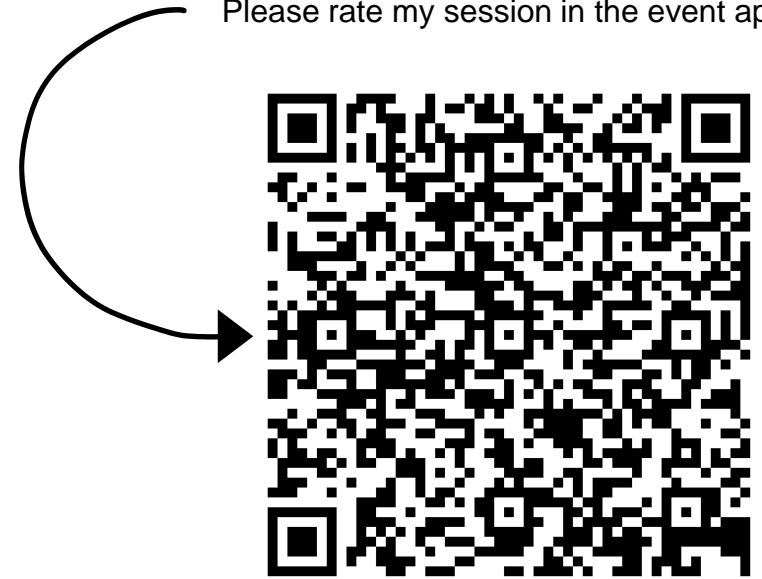
<https://odata2ts.github.io/>

UI5con

# Thank you!

Törehan Gören, TGW Systems Integration GmbH  
[goto@tgw-group.com](mailto:goto@tgw-group.com)

Like what you just saw?  
Please rate my session in the event app:





**Q & A**