Monaco, TypeScript, Angular Sample

Demo Script

# Introduction

This sample script can be used to demonstrate various topics concerning Visual Studio Online Monaco, TypeScript, Azure, and web development in general (e.g. grunt, jasmine).

# Setup Environment

1. Create a new Azure Website
2. Enable Monaco for the new Website
3. Do a first short tour through Monaco

# Connect to Git

1. Create a new VSOnline Project
2. Connect Azure Website with VSOnline using Monaco

# Monaco Hello World

1. Add *default.html* with *html* snippet 🡪 Ctrl + Blank
2. Add *default.css* with *css* snippet
3. Show some editor features
   1. Ctrl+E $ 🡪 Editor features
   2. CSS Intellisense
   3. Side-by-side editing
4. Show site in browser
5. Commit to git
6. Push back to VSOnline
   1. git remote –v
   2. git push origin master
7. Show some editor features
   1. Changes view
   2. Undo

# TypeScript

1. Install TypeScript
   1. npm install typescript –g
2. Add *default.ts* with *sayHello* function with *alert()*, call it in *onload*
3. Compile *default.ts* with *tsc*
4. Show site in browser

# Prepare for Larger Sample

1. Speak about .d.ts files
   1. Show *DefinitelyTyped* project
2. Install type definitions
   1. npm install tsd –g
   2. tsd query jquery –a install
   3. tsd query angular –a install
   4. tsd query angular-route –a install
   5. tsd query AzureMobileServicesClient –a install
3. Change *default.ts* to use TypeScript
   1. */// <reference path="./typings/jquery/jquery.d.ts" />  
      function sayHello() { $("#target").html("<h1>Hello World</h1>"); }*
4. Show IntelliSense based on *.d.ts*
5. Remove *default.\** demo files

# Prepare Azure Mobile Service

1. Create Mobile Service in Azure
   1. Make sure to update service URI in controllers
2. Install type definitions
   1. tsd query AzureMobileServicesClient –a install

# Implement Solution

1. Upload *01 Sourcecode.zip* via drag-and-drop
2. Unzip file in the browser
3. Compile TS files in console
4. Show some editor functions
   1. Peek Definition
   2. Goto Definition

# Automate Build Using Grunt

1. Upload *02 Grunt Typescript.zip*
2. Unzip file in browser
   1. npm install
3. Show Build via Grunt
   1. Single Grunt
   2. Grunt Watch

# Add Unit Testing With Jasmine

1. Install type definitions
   1. tsd query Jasmine –a install
2. Upload *03 Grunt With Tests.zip*
3. Unzip file in browser
   1. npm install
4. Show Build via Grunt
   1. Single Grunt
   2. Grunt Watch
   3. Change unit test to fail and correct it again