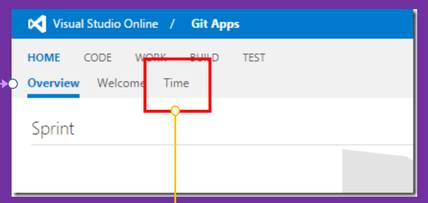
VS Team Services Extension Flow – Gordon's Time Ticker

Visual Studio ALM Rangers



3 – Contribution Point

4 – "Click" event

vss-extension.json

```
{
  "namespace": "VS-Team-Services-Time-Ticker",
  "version": "0.0.2",
  "name": "Time Ticker",
  "description": "Simple Time Ticker extension",
  "provider": { "name": "Gordon Beeming" },
  "icon": "images/icon.png",
  "files": [
    { "path": "index.html", "addressable": true },
    { "path": "images", "addressable": true },
    { "path": "scripts", "addressable": true },
    { "path": "css", "addressable": true }
    { "path": "sdk", "addressable": true },
  ],
  "contributions": {
    "vss.web#hubs": [
      {
        "id": "time",
        "name": "Time",
        "groupId": "home",
        "order": 22,
        "uri": "index.html",
        "usesSdk": true,
        "fullPage": false
      }
    ]
  }
}
```

2 - Static Hosting

5 – Load page

index.html

```
<!DOCTYPE html>

<html lang="en">
<head>
    <meta charset="utf-8" />
    <title>TypeScript HTML App</title>
    <link rel="stylesheet"
          href="app.css" type="text/css" />
    <script src="sdk/Scripts/VSS.SDK.js"></script>
</head>
<body>
    <script type="text/javascript">
        // Initialize the VSS sdk
        VSS.init({
            setupModuleLoader: true,
            moduleLoaderConfig: {
                paths: {
                    "Scripts": "scripts"
                }
            }
        });
        // Wait for the SDK to be initialized
        VSS.ready(function () {
            require(["Scripts/app"], function (app) { });
        });
    </script>
    <h1>TypeScript HTML App</h1>
    <div id="content"></div>
</body>
</html>
```

6, 7, 8 – Function calls

1 – Compile

app.ts

```
class Greeter {
    element: HTMLElement;
    span: HTMLElement;
    timerToken: number;

    constructor(element: HTMLElement) {
        this.element = element;
        this.element.innerHTML += "The time is: ";
        this.span = document.createElement('span');
        this.element.appendChild(this.span);
        this.span.innerText = new Date().toUTCString();
    }
    start() {
        this.timerToken = setInterval(() =>
              this.span.innerHTML = new Date().toUTCString(), 500);
    }
    stop() {
        clearTimeout(this.timerToken);
    }
}

var el = document.getElementById('content');
var greeter = new Greeter(el);
greeter.start();
```

10 – Update HTML element

9 – Asynchronously fetch and run script

app.js

```
var Greeter = (function () {
    function Greeter(element) {
        this.element = element;
        this.element.innerHTML += "The time is: ";
        this.span = document.createElement('span');
        this.element.appendChild(this.span);
        this.span.innerText = new Date().toUTCString();
    }
    Greeter.prototype.start = function () {
        var _this = this;
        this.timerToken = setInterval(function ()
         { return _this.span.innerHTML =
                new Date().toUTCString(); }, 500);
    };
    Greeter.prototype.stop = function () {
        clearTimeout(this.timerToken);
    };
    return Greeter;
})();

var el = document.getElementById('content');
var greeter = new Greeter(el);
greeter.start();
//# sourceMappingURL=app.js.map
```

app.css

```
body { font-family: 'Segoe UI', sans-serif; }

span { font-style: italic; }
```

sdk/scripts/vss.sdk.js

```
// Initiates the handshake with the host window.
VSS.init;
// Register a callback that gets called once initial setup/handshake has completed.
VSS.ready
// Ensures that the AMD loader from the host is configured and fetches a script
// (AMD) module and its dependencies.
VSS.require
```